

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA STROJNÍ

# DISERTAČNÍ PRÁCE

2023

Ing. Miroslav Malaga

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
**FAKULTA STROJNÍ**

Studijní program: P0715D270029 Průmyslové inženýrství a management

**DISERTAČNÍ PRÁCE**

Architektura HW Learning factory s digitálním dvojčtem v konceptu  
Industry 4.0

Autor:

**Ing. Miroslav Malaga**

Školitel:

**doc. Ing. Zdeněk Ulrych, Ph.D.**

Akademický rok 2022/2023

## PROHLÁŠENÍ O AUTORSTVÍ

Předkládám tímto k posouzení a obhajobě disertační práci (DISP), zpracovanou na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem předloženou DISP na téma:

***Architektura HW Learning factory s digitálním dvojčetem v konceptu Industry 4.0***

vypracoval samostatně, dle Studijního a zkušebního řádu Západočeské univerzity v Plzni, pod odborným dohledem školitele a s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této práce.

V Plzni dne:

---

Podpis autora

## **PODĚKOVÁNÍ**

V první řadě bych rád poděkoval celé své rodině za všestrannou pomoc, podporu a vytvoření vhodných podmínek k práci a svému školiteli doc. Ing. Zdeňkovi Ulrychovi, Ph.D., za podporu, všestrannou pomoc, cenné a inspirativní rady, za množství podnětů, doporučení a připomínek a hlavně trpělivost, kterou se mnou měl po celou dobu doktorského studia.

Ing. Věrece Markové bych rád poděkoval za odborné rady týkající se statistického vyhodnocení, Mgr. Ondřeji Jandovi za pomoc s jazykovou korekturou a v neposlední řadě bych rád poděkoval celému kolektivu katedry průmyslového inženýrství a managementu Fakulty strojní Západočeské univerzity v Plzni za dlouholetou příjemnou spolupráci a pomoc při řešení úkolů a problémů.



# ANOTACE DISERTAČNÍ PRÁCE

Fakulta strojní Západočeské univerzity v Plzni

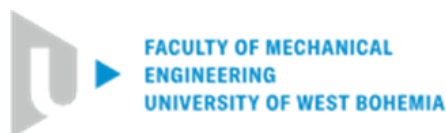
ak. rok odevzdání: 2023

<b>AUTOR</b>	<i>Příjmení (včetně titulů)</i> Ing. Malaga	<i>Jméno</i> Miroslav
<b>STUDIJNÍ PROGRAM</b>	P0715D270029 – Průmyslové inženýrství a management	
<b>VEDOUCÍ PRÁCE</b>	<i>Příjmení (včetně titulů)</i> doc. Ing. Ulrych, Ph.D.	<i>Jméno</i> Zdeněk
<b>PRACOVISŤE VEDOUCÍHO</b>	Západočeská univerzita v Plzni, Fakulta strojní, Katedra průmyslového inženýrství a managementu	
<b>NÁZEV PRÁCE</b>	Architektura HW Learning factory s digitálním dvojčtem v konceptu Industry 4.0	

## Počet stránek (A4 a ekvivalentů A4)

<b>Celkem</b>	146		<b>Textová část</b>	143		<b>Grafická část (přílohy)</b>	3
---------------	-----	--	---------------------	-----	--	--------------------------------	---

<b>STRUČNÝ POPIS ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY</b>	Disertační práce se zabývá návrhem a ověřením architektury HW learning factory s digitálním dvojčtem využívajícím diskretní simulaci v konceptu Industry 4.0 se zaměřením na průmyslové inženýry. Cílem je návrh architektury využitelné a realizovatelné průmyslovými inženýry strojírenského typu, kteří nejsou programátory, elektrotechniky ani např. kybernetiky. V rámci výzkumu byla ověřena realizovatelnost a vhodnost navržené architektury pro průmyslové inženýry. Hlavními výstupy je navržená architektura, HW model s digitálním stínem realizovaný podle navržené architektury a výstupy z provedeného výzkumu.
<b>KLÍČOVÁ SLOVA</b>	Learning factory, training factory, digitální dvojče, kyber-fyzikální systém, Industry 4.0, simulace



## SUMMARY OF DISSERTATION WORK

FAKULTY OF MECHANICAL ENGINEERING

UNIVERSITY OF WEST BOHEMIA

Submitted: 2023

<b>Author</b>	<i>Surname (including of Degrees)</i> Ing. Malaga	<i>Name</i> Miroslav
<b>Study Programmes</b>	P0715D270028 – Industrial Engineering and Management	
<b>Supervisor</b>	<i>Surname (including of Degrees)</i> doc. Ing. Ulrych, Ph.D.	<i>Name</i> Zdeněk
<b>Instution/ Department</b>	University of West Bohemia in Pilsen, Faculty of Mechanical Engineering, Department of Industrial Engineering and Management	
<b>Title of the Work</b>	HW Learning factory architecture with digital twin in Industry 4.0 concept	

### Number of pages

<b>Total</b>	146	<b>Part of Text</b>	143	<b>Attachment</b>	3
--------------	-----	---------------------	-----	-------------------	---

<b>Brief Description of the Theme, Findings and Benefits</b>	The dissertation deals with the design and validation of a HW learning factory architecture with a digital twin using discrete simulation in the Industry 4.0 concept with a focus on industrial engineers. The aim is to design an architecture usable and implementable by industrial engineers who are not computer programmers, electrical engineers, or cyberneticists for example. The research verified the realizability and suitability of the proposed architecture for industrial engineers. The main outputs are the proposed architecture, the HW model with digital shadow implemented according to the proposed architecture and the outcomes of the research conducted.
<b>Key Words</b>	Learning factory, training factory, digital twin, cyber-physical system, Industry 4.0, simulation

## Obsah

Seznam obrázků.....	4
Seznam tabulek.....	7
Seznam zkratk.....	9
Glosář.....	11
Úvod.....	13
1 Cíle disertační práce .....	14
2 Průmyslové inženýrství a průmyslový inženýr .....	15
3 Koncept Industry 4.0 .....	16
3.1 Adaptace Industry 4.0 v podnicích .....	18
3.2 Nástupce konceptu Industry 4.0 .....	18
4 Simulace v konceptu Industry 4.0 .....	20
4.1 Systém .....	20
4.2 Model .....	20
4.3 Nástroje pro diskrétní simulace.....	21
4.4 Digitální dvojče .....	23
4.5 Cyber-Physical System a DT.....	25
5 Koncept STEM a prototypovací hardware.....	28
5.1 Komplexní STEM modely a Industry 4.0.....	30
5.2 Legislativní omezení pro průmyslového inženýra .....	33
6 Rešerše odborných článků vztahujících se k dané problematice .....	35
6.1 Plánování výroby pomocí simulačního modelu v kontextu Industry 4.0 .....	35
6.2 Přístup k tvorbě a využití modelů a simulace v kontextu Industry 4.0.....	37
6.3 HW modely a jejich využívání v rámci konceptu Industry 4.0.....	39
6.4 Shrnutí .....	44
7 Shrnutí poznatků a teoretických východisek práce.....	46
8 Výzkumný záměr .....	48
8.1 Cíle práce .....	48
8.2 Hypotézy.....	48
9 Architektura pro training factory s digitálním dvojčetem .....	50
9.1 Hardware/Hardware model .....	52
9.2 Řídící systémy hardwaru.....	52
9.3 Interface pro komunikaci s HW modelem a jeho řídicími systémy .....	55
9.4 Zpracování a zpřístupnění dat .....	58
9.5 Digitální dvojče/Digitální stín .....	60

9.6	Cloud.....	61
9.7	Možnost napojení dalšího SW a HW .....	61
9.8	Kompetence pro využití navržené architektury vs architektury Fischertechnik .....	62
9.8.1	Kompetence pro využití Fischertechnik Training Factory Industry 4.0 9V .....	62
9.8.2	Potřebné kompetence pro využití navržené architektury.....	64
10	Vývoj modelu komplexní výrobní linky s digitálním dvojčetem dle navržené architektury..	66
10.1	Modul sklad .....	67
10.2	Modul kontroly a řezání .....	69
10.3	Modul robot .....	70
10.4	Modul obrábění.....	71
10.5	Modul řízení pracovníků.....	72
10.6	Modul přehřevu a svařování .....	73
10.7	Modul sensorové stanice .....	74
10.8	Uplatnění training factory realizované podle navržené architektury .....	75
11	Způsob testování a sběru dat pro ověření hypotéz.....	79
11.1	Způsob testování a dotazování.....	83
11.2	Průběh bodového hodnocení odpovědí pro H1 a H2.....	85
11.3	Výzkumný vzorek.....	86
12	Vyhodnocení a výsledky výzkumu .....	88
12.1	Metodika statistického vyhodnocení .....	88
12.1.1	Vyhodnocení normality .....	88
12.1.2	Testování tří a více skupin a mnohonásobné porovnávání .....	88
12.1.3	Testování dvou skupin.....	88
12.1.4	Grafické srovnání skupin .....	89
12.2	Příklady na programování HW – test.....	89
12.2.1	Okruh otázek k příkladu č. 1 .....	89
12.2.2	Okruh otázek k příkladu č. 2 .....	93
12.2.3	Okruh otázek k příkladu č. 3 .....	96
12.2.4	Okruh otázek k příkladu č. 4 .....	99
12.2.5	Otázka 5 - hodnocení náročnosti testu .....	102
12.2.6	Shrnutí .....	103
12.3	Hodnocení znalostí – dotazník .....	106
12.3.1	Zkušenosti s programovacími jazyky .....	106
12.3.2	Zkušenosti s vývojovými prostředími .....	109
12.3.3	Ostatní programátorské zkušenosti .....	110



---

12.3.4	Zkušenosti s uživatelskou prací v operačním systému .....	112
12.3.5	Zkušenosti se softwarovými nástroji .....	113
12.3.6	Zkušenosti s administrací SW nástrojů .....	114
12.3.7	Ostatní uživatelské zkušenosti .....	115
12.3.8	Shrnutí hypotézy 3 .....	116
13	Použité vědecké metody .....	118
13.1	Empirické vědecké metody .....	118
13.2	Logické vědecké metody .....	118
13.3	Specifické vědecké metody .....	119
14	Přínosy disertační práce .....	120
14.1	Teoretický přínos práce .....	120
14.2	Praktický přínos práce .....	120
14.3	Doporučení pro další výzkum v dané problematice .....	121
15	Diskuse .....	122
16	Závěr .....	124
	Seznam literatury .....	127
	Publikační činnost studenta doktorské formy studia .....	140

## Seznam obrázků

Obrázek 3-1 – Model Industrie 4.0 (RAMI4.0) [14] .....	17
Obrázek 4-1 – Teoretický vliv míry abstrakce na přesnost modelu, zpracováno dle textu [28] .....	21
Obrázek 4-2 – Spuštěná simulace v nástroji Tecnomatix Plant Simulation .....	23
Obrázek 4-3 – Konceptuální model dvojčete dle [39], vlastní úprava .....	23
Obrázek 4-4 – Digitální model, stín, dvojče podle [40], vlastní úprava.....	24
Obrázek 4-5 – vztah mezi CPS, DT a IoT [53], vlastní úprava .....	26
Obrázek 4-6 – Kyber-fyzikální systém využitý pro baterie elektromobilů [54], vlastní úprava.....	26
Obrázek 4-7 – 5C architektura kyber-fyzikálního systému [50], vlastní úprava.....	27
Obrázek 5-1 – Řídící jednotky (mikrokontrolery a mikropočítače) .....	29
Obrázek 5-2 – Ukázka modelu s využitím BBC Micro:bit a řídicího kódu v BBC Micro:bit MakeCode, JavaScript a MicroPython .....	30
Obrázek 5-3 – FESTO CP Factory [86] .....	31
Obrázek 5-4 – CP Lab 404-1 [87] .....	32
Obrázek 5-5 – Fischertechnik Training Factory Industry 4.0 .....	32
Obrázek 5-6 – Fischertechnik Factory Simulation 24V [90] .....	33
Obrázek 5-7 – Merkur programovatelná minilinka s MG [91] .....	33
Obrázek 5-8 – Odborné znalosti PLC programátora podle národní soustavy povolání [94].....	34
Obrázek 6-1 – Stavebnice robotické ruky použitá S. Langem a kol. [111] .....	38
Obrázek 6-2 – Model postavený na platformě Fischertechnik pro využití ve výuce Pechmannem a kol. [121] .....	40
Obrázek 6-3 – HW model a uživatelské rozhraní emulace modelu dle Hofmanna a kol. [122].....	40
Obrázek 6-4 – Architektura learning factory, návrh layoutu v MS Excel a vygenerovaný 3D model dle Langa a kol. [111] .....	41
Obrázek 6-5 – Pyramida automatizace L0 – L3 dle Phillipse a kol. [124] .....	41
Obrázek 6-6 – Ukázka stromu projektu v ABB Freelance [124] .....	42
Obrázek 6-7 – Schéma zapojení, verifikační model a ukázka z venkovního testu Padillou a kol. [125] .....	42
Obrázek 6-8 – Ukázka laboratoře Industry 4.0 a logické architektury demonstrátoru Industry 4.0 představená Zartem a kol. [126] .....	43
Obrázek 6-9 – Dvouvrstvá architektura pro CPS a fyzický model dle Lyu a kol. [127] .....	43
Obrázek 6-10 – Architektura FTsim a ukázka modeluv FTsim [128] .....	44
Obrázek 9-1 – HW model s digitálním dvojčetem v nástroji Tecnomatix Plant Simulation .....	50
Obrázek 9-2 – Navržená architektura pro Training factory s digitálním dvojčetem .....	51
Obrázek 9-3 – Model třídičky .....	52
Obrázek 9-4 – Ukázka řídicího kódu jednotky Fischertechnik Robotics TXT .....	53
Obrázek 9-5 – Ukázka řídicího kódu jednotky Fischertechnik TXT 4.0.....	53

Obrázek 9-6 – Ukázka řídicího kódu jednotky BBC Micro:bit.....	54
Obrázek 9-7 –Ukázka řídicích jednotek Fischertechnik TXT 4.0 a Fischertechnik Robotics TXT .....	54
Obrázek 9-8 – Ukázka řídicí jednotky BBC Micro:bit.....	55
Obrázek 9-9 – Interface pro komunikaci s HW modelem využívající RaspberryPi Pico .....	56
Obrázek 9-10 – Sketch pro řízení překladače (Interface pro komunikaci s HW modelem) .....	57
Obrázek 9-11 – Ukázka Arduina Uno zapojeného a využívaného jako překladače v prvotní realizaci .	58
Obrázek 9-12 – Aplikace ControllModSim s náhledem na převodní tabulku HW modelu třídičky .....	59
Obrázek 9-13– Aplikace ControllModSim s náhledem na data z HW modelu třídičky .....	59
Obrázek 9-14 – Ukázka metody pro synchronizaci digitálního dvojčete s HW modelem v jazyce SimTalk .....	60
Obrázek 9-15 – Ukázka 3D digitálního dvojčete vytvořeného v nástroji Tecnomatix Plant Simulation	61
Obrázek 9-16 – Architektura Fischertechnik Training Factory Industry 4.0 9V [140] .....	64
Obrázek 10-1 – Komplexní model HW training factory.....	67
Obrázek 10-2 – Modul sklad.....	67
Obrázek 10-3 – Ukázka HW propojení překladače a řídicí jednotky Fischertechnik TXT 4.0 .....	68
Obrázek 10-4 – Modul kontroly a řezání.....	69
Obrázek 10-5 – Modul robot.....	70
Obrázek 10-6 – Modul obrábění .....	71
Obrázek 10-7 – Modul řízení pracovníků .....	72
Obrázek 10-8 – Modul přehřevu a svařování.....	73
Obrázek 10-9 – Modul sensorové stanice .....	75
Obrázek 10-10 – Digitální stín training factory ve virtuální realitě [141].....	76
Obrázek 10-11 – Mobilní aplikace pro zobrazení stavu a vizualizaci dat HW modelu [142] .....	77
Obrázek 10-12 – Ukázka hlavní části navrženého a ověřeného SW modelu [143].....	78
Obrázek 10-13 – Ukázka realizovaného prototypu AGV pro realizovanou training factory [144].....	78
Obrázek 11-1 – Ukázka modelu pro příklad 1 .....	79
Obrázek 11-2 – Ukázka modelu pro příklad 2 .....	80
Obrázek 11-3 – Ukázka modelu pro příklad 3 .....	80
Obrázek 11-4 – Ukázka modelu pro příklad 4 .....	81
Obrázek 11-5 – Ukázka otázky č. 1.1., tedy 1. otázka základního příkladu pro vizuální programovací jazyk.....	82
Obrázek 11-6 – Ukázka aplikace Questionnaire v režimu testování pro programovací jazyk Python ..	84
Obrázek 11-7 – Ukázka poučení při přihlášení do testu v aplikaci Questionnaire.....	85
Obrázek 11-8 – Ukázka náhledu části dotazníku v Google Forms.....	85
Obrázek 12-1 – Popis boxplotu – krabicového grafu [148].....	89
Obrázek 12-2 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 1 .....	91

Obrázek 12-3 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 1 .....	92
Obrázek 12-4 – Boxplot zobrazující časovou efektivitu k příkladu č. 1 .....	93
Obrázek 12-5 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 2 .....	94
Obrázek 12-6 – Boxplot zobrazující čas strávený řešením úkolů k příkladu č. 2 .....	95
Obrázek 12-7 – Boxplot zobrazující časovou efektivitu k příkladu č. 2 .....	96
Obrázek 12-8 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 3 .....	97
Obrázek 12-9 – Boxplot zobrazující čas strávený řešením úkolů k příkladu č. 3 .....	98
Obrázek 12-10 – Boxplot zobrazující časovou efektivitu k příkladu č. 3 .....	99
Obrázek 12-11 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 4 .....	100
Obrázek 12-12 – Boxplot zobrazující čas strávený řešením úkolů k příkladu č. 4 .....	101
Obrázek 12-13 – Boxplot zobrazující časovou efektivitu k příkladu č. 4 .....	102
Obrázek 12-14 – Boxplot zobrazující hodnocení náročnosti testů .....	103
Obrázek 12-15 – Boxplot zobrazující zkušenosti respondentů s programovacími jazyky .....	107
Obrázek 12-16 – Boxplot zobrazující zkušenosti respondentů s vývojovými prostředími .....	110
Obrázek 12-17 – Boxplot zobrazující ostatní programátorské zkušenosti .....	111
Obrázek 12-18 – Boxplot zobrazující zkušenosti s uživatelskou prací v operačním systému .....	112
Obrázek 12-19 – Boxplot zobrazující zkušenosti se softwarovými nástroji .....	114
Obrázek 12-20 – Boxplot zobrazující zkušenosti s administrací softwarových nástrojů .....	115
Obrázek 12-21 – Boxplot zobrazující ostatní uživatelské zkušenosti .....	116
Obrázek 13-1 – Kolbův experimentální cyklus [150] .....	119

## Seznam tabulek

Tabulka 4-1 – Generace simulačního software dle [36].....	22
Tabulka 4-2 – Pětiúrovňové dělení Digitálních dvojčat (úroveň zralosti digitálního dvojčete) podle Yong-Woona [41].....	25
Tabulka 9-1– Potřebné kompetence pro využití Fischertechnik Training Factory Industry 4.0 9V .....	63
Tabulka 9-2 – Potřebné kompetence pro využití navržené architektury.....	65
Tabulka 10-1 – Informace předávané z modulu sklad do aplikace ControllModSim.....	68
Tabulka 10-2 – Informace předávané z modulu kontroly a řezání do aplikace ControllModSim .....	70
Tabulka 10-3 – Informace předávané z modulu robot do aplikace ControllModSim .....	71
Tabulka 10-4 – Informace předávané z modulu obrábění do aplikace ControllModSim.....	72
Tabulka 10-5 – Informace předávané z modulu předeřevu a svařování do aplikace ControllModSim .....	74
Tabulka 11-1 – Velikost populace, na kterou se zaměřuje výzkum .....	87
Tabulka 12-1 – Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 1.....	90
Tabulka 12-2 – P-hodnoty kombinací způsobů programování pro hodnocení řešení úkolů k příkladu č. 1.....	90
Tabulka 12-3 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 1 ....	91
Tabulka 12-4 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 1 .....	92
Tabulka 12-5 – P-hodnoty kombinací způsobů programování pro hodnocení časovou efektivitu k příkladu č. 1 .....	92
Tabulka 12-6 – Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 2.....	93
Tabulka 12-7 – P-hodnoty kombinací způsobů programování pro hodnocení řešení úkolů k příkladu č. 2.....	93
Tabulka 12-8 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 2 ....	94
Tabulka 12-9 – P-hodnoty kombinací způsobů programování čas strávený řešením úkolů k příkladu č. 2.....	94
Tabulka 12-10 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 2 .....	95
Tabulka 12-11 – P-hodnoty kombinací způsobů programování pro hodnocení časovou efektivitu k příkladu č. 2 .....	95
Tabulka 12-12 – Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 3.....	96
Tabulka 12-13 – P-hodnoty kombinací způsobů programování pro hodnocení řešení úkolů k příkladu č. 3.....	97
Tabulka 12-14 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 3 ..	97
Tabulka 12-15 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 3 .....	98
Tabulka 12-16 – P-hodnoty kombinací způsobů programování pro hodnocení časovou efektivitu k příkladu č. 3 .....	98
Tabulka 12-17– Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 4.....	99
Tabulka 12-18 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 4	100

Tabulka 12-19 – P-hodnoty kombinací způsobů programování čas strávený řešením úkolů k příkladu č. 4.....	100
Tabulka 12-20 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 4 .....	101
Tabulka 12-21 – Základní statistické charakteristiky pro hodnocení náročnosti testu.....	102
Tabulka 12-22 – P-hodnoty kombinací způsobů programování pro hodnocení náročnosti testu.....	102
Tabulka 12-23 – Shrnutí výsledků pro H1.....	104
Tabulka 12-24 – Shrnutí výsledků pro H2.....	105
Tabulka 12-25 – Stupnice pro hodnocení zkušeností s programovacími jazyky .....	106
Tabulka 12-26 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s programovacími jazyky“.....	107
Tabulka 12-27 – P-hodnoty dvojic se statisticky významným rozdílem pro okruh „Zkušenosti s programovacími jazyky“.....	108
Tabulka 12-28 – Stupnice pro hodnocení zkušeností s vývojovými prostředími .....	109
Tabulka 12-29 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s vývojovými prostředími“.....	109
Tabulka 12-30 – Stupnice pro hodnocení ostatních programátorských zkušeností .....	110
Tabulka 12-31 – Tabulka četnosti odpovědí pro okruh „Ostatní programátorské zkušenosti“ .....	111
Tabulka 12-32 – Stupnice pro hodnocení zkušeností s uživatelskou prací v operačním systému .....	112
Tabulka 12-33 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s uživatelskou prací v operačním systému“.....	112
Tabulka 12-34 – Stupnice pro hodnocení zkušeností se softwarovými nástroji .....	113
Tabulka 12-35 – Tabulka četnosti odpovědí pro okruh „Zkušenosti se softwarovými nástroji“ .....	113
Tabulka 12-36 – Stupnice pro hodnocení zkušeností s administrací SW nástrojů.....	114
Tabulka 12-37 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s administrací SW nástrojů“ .....	114
Tabulka 12-38 – Stupnice pro hodnocení ostatních uživatelských zkušeností .....	115
Tabulka 12-39 – Tabulka četnosti odpovědí pro okruh „Ostatní uživatelské zkušenosti“ .....	116
Tabulka 12-40 – Shrnutí výsledků pro H3.....	117

## Seznam zkratk

AGV	Automated Guided Vehicle (Automaticky řízený vozík)
AR	Augmented reality (rozšířená realita)
BP	Bakalářská práce
CPS	Cyber-physical System (kyber-fyzikální systém)
CRM	Customer Relationship Management (řízení vztahů se zákazníky)
DM	Digital Model (digitální model)
DP	Diplomová práce
DS	Digital Shadow (digitální stín)
DT	Digital Twin (digitální dvojče)
E-PPS	Energy in Production Planning and Scheduling (plánování výroby s ohledem na spotřebu energií)
ERP	Enterprise Resource Planning (plánování podnikových zdrojů, většinou chápáno jako druh IS)
EV	Electric vehicle (elektromobil)
FS	Fakulta strojní
FST	Fakulta strojní (ve smyslu fakulty Západočeské univerzity v Plzni)
HW	Hardware
H1	hypotéza č. 1 definovaná v rámci této disertační práce
H2	hypotéza č. 2 definovaná v rámci této disertační práce
H3	hypotéza č. 3 definovaná v rámci této disertační práce
I2C/I <sup>2</sup> C	Internal Integrated-Circuit Bus
IoT	Internet of Things (internet věcí)
IS	Information System (informační systém)
IT	Informační technologie
JSON	JavaScript Object Notation (JavaScriptový objektový zápis)
KPV	Katedra průmyslového inženýrství a managementu
LED	Light-Emitting Diode (elektroluminiscenční dioda)
MES	Manufacturing Execution Systems (výrobní informační systém)
ML	Machine Learning (strojové učení)
MQTT	Message Queuing Telemetry Transport
NASA	National Aeronautics and Space Administration (Národní úřad pro letectví a kosmonautiku)
OA	Odborný asistent
ORM	Object-Relational Mapping (Objektově relační mapování)
OS	Operační systém
PC	Personal Computer (osobní počítač)

PLC	Programmable Logic Controller (programovatelný logický automat)
PWM	Pulse Width Modulation (pulsně šířková modulace)
QS	Quality management system (systém řízení kvality)
RFID	Radio Frequency Identification (identifikace na rádiové frekvenci)
SME	Small and Medium Enterprise (malé a střední podniky)
SoC	State of Charge (stav nabití baterie)
SPC	Statistical Process Control (statistická regulace procesu)
SRM	Supplier Relationship Management (řízení vztahů s dodavateli)
SVS	Předmět Simulace výrobních systémů a procesů na KPV FST ZČU
SW	Software
TCP	Transmission Control Protocol
TOC	Theory of Constraints (teorie úzkých míst)
USB	Universal Serial Bus (univerzální sériová sběrnice)
VBA	Visual Basic for Applications
VPJ	Vizuální programovací jazyk
VR	Virtual Reality (virtuální realita)
WPF	Windows Presentation Foundation
ZČU	Západočeská univerzita v Plzni



## Glosář

Slovo	význam
<b>AI (UI, umělá inteligence)</b>	Obor informatiky zabývající se tvorbou strojů řešících komplexní úlohy např. z oblastí logistiky, robotiky, zpracování přirozeného jazyka, či zpracování velkých objemů dat.
<b>AR</b>	Technologie používaná pro rozšíření reálného obrazu světa, který je díky spolupráci fotoaparátu chytrého telefonu a chytrého telefonu doplněný o počítačem vytvořený objekt/objekty.
<b>C#</b>	Vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET
<b>CNC</b>	Zkratka anglického „Computer Numerical Control“, která se používá ve spojení s obráběcím strojem. Je to tedy počítačem řízený obráběcí stroj.
<b>COM Port</b>	COM port je jednoduše I/O rozhraní, které umožňuje připojení sériového zařízení k počítači. Dnes často virtualizovaný přes USB.
<b>Entity Framework</b>	Framework pro objektově relační mapování v ADO.NET.
<b>I2C/I<sup>2</sup>C</b>	Multi-masterová počítačová sériová sběrnice vyvinutá firmou Philips, která je používána k připojování nízkorychlostních periférií např. k základní desce
<b>IoT</b>	Označení pro síť fyzických zařízení (vozidel, domácích spotřebičů, strojů a dalších zařízení) vybavených síťovou konektivitou, která umožňuje těmto zařízením se propojit a vyměňovat si data.
<b>Kobot</b>	Kolaborativní robot, tedy robot spolupracující s člověkem.
<b>Learning factory, Teaching factory</b>	Reprezentace realistického výrobního prostředí pro vzdělávání, školení a výzkum. Nejčastěji jde o vzdělávací HW model nebo laboratoř.
<b>Master/Follower</b>	V informatice model počítačové komunikace, kdy jedno zařízení nebo jeden proces přebírá řízení nad jedním nebo více dalšími zařízeními.
<b>ML</b>	Podoblast umělé inteligence, zabývající se algoritmy a technikami, které umožňují počítačovému systému 'učit se'.
<b>MQTT</b>	Odlehčený síťový protokol pro publikování a odebírání pro službu fronty zpráv.

<b>ORM</b>	Programovací technika nebo SW nástroj v softwarovém inženýrství pro automatickou konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem.
<b>SAP</b>	SAP je německá softwarová firma se sídlem ve Walldorfu (Bádensko), vyvíjející systémy z oblasti ERP. Název firmy vznikl ze zkratky „ <b>S</b> ysteme, <b>A</b> nwendungen, <b>P</b> rodukte in der <b>D</b> atenverarbeitung“.
<b>Sériová (Seriálová) komunikace</b>	V telekomunikacích a informatice přenos dat komunikačním kanálem nebo po sběrnici postupně po jednotlivých bitech (sekvenčně).
<b>SimTalk</b>	Skriptovací a programovací jazyk pro simulační nástroj Tecnomatix Plant Simulation
<b>Sketch</b>	Skript, nebo program pro řízení mikrokontrolérů rodiny Arduino, nebo všeobecně skript/program napsaný v jazyce Wiring
<b>TCP</b>	Protokol transportní vrstvy v sadě protokolů TCP/IP používaných v síti Internet
<b>Testbed</b>	Experimentální zařízení, ve kterém je určitá množina strojů, transportních zařízení, skladů atd. Lze z něj velmi jednoduše sestavit malou experimentální výrobní linku.
<b>USB</b>	Univerzální sériová sběrnice, moderní způsob připojení periférií k počítači.
<b>VR</b>	Technologie umožňující uživateli ocitnout se v simulovaném prostředí a často s možností s ním i interagovat.
<b>Wiring</b>	Programovací jazyk využívaný pro mikrokontroléry rodiny Arduino, podporuje ale i jiné mikrokontroléry, např. Raspberry Pi Pico
<b>WPF</b>	Windows Presentation Foundation je v informatice knihovna tříd pro tvorbu grafického rozhraní, která je součástí .NET

## Úvod

Technické vzdělání není a ani nemůže být souborem samostatných znalostí a dovedností bez jakékoli provázanosti. Více než v jakémkoliv jiném typu vzdělání je u technického vzdělání, a zejména u multioborových zaměření typu průmyslových inženýrů, potřeba klást důraz na provázanost znalostí a dovedností a dosáhnout tak synergie. Dá se říct, že toto platilo již dávno v historii, kdy největší objevitelé a vynálezci měli znalosti nejen v rámci jedné vědní kategorie, ale průřezově i napříč vědními disciplínami (přírodní vědy, technické vědy, filozofie, umění). V době čtvrté průmyslové revoluce (Průmysl 4.0) toto začíná být opět aktuální, protože nejen takto vzdělaný člověk bude mít pro společnost vysokou hodnotu, ale také bude mít ty nejlepší možné předpoklady pro to, aby v konkurenčním světě obstál. Tento tlak je i na společnosti, které nejenže musí neustále implementovat a přivádět k životu nové technologie, postupy práce a trendy, ale souběžně s tím musí neustále vzdělávat své zaměstnance, aby tyto technologie a postupy práce mohli využívat a aplikovat.

V době čtvrté průmyslové revoluce je mnoho možností, jak průmysloví inženýři mohou modelovat a optimalizovat výrobní procesy ve svých organizacích, ale dnes se to oproti dřívějším dobám musí často dělat ve spolupráci s dodavateli i odběrateli. Moderní technologie stírají ostré hranice výrobního podniku díky technologiím, jako je například internet věcí, který umožňuje zavádění komunikace mezi stroji nejen v rámci jedné organizace, ale dnes už i napříč organizacemi, nebo online sdílení dat mezi ERP systémy podniku, jeho dodavateli a jeho odběrateli. To vše může vést – a je snaha, aby to vedlo – k vyšší racionalizaci práce, vyšší produktivitě a ke snížení zmetkovitosti (a všeobecně plýtvání jakýchkoli zdrojů), ale zároveň to klade vyšší požadavky na všechny pracovníky organizace bez ohledu na jejich pozici. Je potřeba tyto lidi nejen zaškolit, jak s technologiemi pracovat, ale zároveň je i seznámit s jejich možnostmi a hlavními myšlenkami, protože jen tak lze do budoucna očekávat, že budou přicházet návrhy na optimalizace a zlepšení procesů i zevnitř organizace.

Disertační práce se tedy zabývá rešeršním výzkumem této problematiky včetně představení vědeckých metod, představuje navrženou architekturu training factory s digitálním dvojčetem a v rámci stanovených tezí ověřuje, že danou architekturu jsou schopni průmysloví inženýři nejen využívat, ale jsou schopni podle architektury i realizovat nové a popř. upravovat stávající training factories podle svých potřeb, potřeb svých průmyslových organizací a potřeb vzdělávacích institucí zaměřených na průmyslové inženýry nebo je využívat i pro představování průmyslového inženýrství a s tím spojených aktuálních trendů a konceptů, jako je např. koncept Industry 4.0 široké veřejnosti např. v rámci popularizace vědy nebo oboru.

## 1 Cíle disertační práce

Rámcovým tématem disertační práce je téma „Modelování a optimalizace podnikových procesů“. Toto rámcové téma je v době rozmachu konceptu Industry 4.0 velmi aktuální a velmi rozsáhlé. Při modelování a optimalizaci se jako jeden z hlavních pilířů konceptu Industry 4.0 využívají digitální dvojčata, která mohou být vytvořena jako HW model, digitální model/simulace, nebo kombinací HW modelu a digitálního modelu/simulace. Důvodů pro tvorbu modelů a následné provádění simulací je mnoho, od vzdělávání, přes snahu pochopit daný systém, až např. po optimalizaci systému (nebo dnes už i provázaných systémů), nebo návrh zcela nového systému „na zelené louce“ se snahou minimalizovat různé chyby vzniklé např. na základě nesprávného rozhodnutí v rámci návrhu a realizace, které by se mohly projevit až po rozběhnutí tohoto nového systému. Koncept Industry 4.0 s myšlenkou propojování, neustále dostupných dat a komunikace „všeho se vším“ rozšiřuje modelování a simulace o další rozměr, a to napojení modelu, simulace nebo např. jakéhokoliv informačního systému na fyzický, skutečný systém, popř. jeho HW model – tímto se dostáváme k digitálním dvojčatům různých úrovní a kyber-fyzikálnímu systému (CPS), který je aktuálně jedním z nejdiskutovanějších prvků konceptu Industry 4.0.

Na základě teoretických poznatků získaných při práci na tezi disertační práce byl stanoveným hlavním cílem disertační práce:

### **Návrh architektury HW learning factory s digitálním dvojčetem, využívajícím diskrétní simulaci, pro vzdělávání v konceptu Industry 4.0.**

Stanovený hlavní cíl vycházející ze zpracovaných tezí spadá do problematiky rámcového tématu „Modelování a optimalizace podnikových procesů“. Název práce je adekvátně upraven na základě zpracovaných tezí k disertační práci a podle stanoveného hlavního cíle disertační práce:

### **Architektura HW learning factory s digitálním dvojčetem v konceptu Industry 4.0**

Navržená komplexní architektura HW learning factory s digitálním dvojčetem je cílena pro využití ve vzdělávacích organizacích vyučujících problematiku Industry 4.0, výrobních podnicích typu SME a společnostech zaměřených na průmyslové inženýrství a samotné průmyslové inženýry. Tato architektura umožňuje jak tvorbu vzdělávacích HW modelů, softwarových modelů, tak jejich propojení např. s cíli učit pracovníky nové pracovní postupy, umožnit průmyslovým inženýrům pochopit a implementovat digitální dvojče/dvojčata, sdílet a vytěžovat data z výroby, zavádět a využívat kyber-fyzikální systémy ve svých organizacích, nebo identifikovat důležité procesy a modelovat/simulovat/optimalizovat je za jakýmkoliv jiným účelem dle vlastní potřeby.

## 2 Průmyslové inženýrství a průmyslový inženýr

U výstupů disertační práce se předpokládá využití hlavně průmyslovými inženýry a v průmyslovém inženýrství. Proto je potřeba nejprve tyto pojmy definovat.

Průmyslové inženýrství (dále jen „PI“) je podle jedné z definic vedoucí vědní obor zabývající se návrhem, zaváděním a zlepšováním integrovaných systémů majících socio-technickou povahu a integruje lidi, informace, stroje, energie, materiál a procesy v rámci celého životního cyklu výrobku, služby nebo programu.[1]

Jiná definice označuje PI jako multidisciplinární obor řešící aktuální potřeby podniků v oblasti moderního průmyslového managementu, kombinuje technické znalosti inženýrských oborů s poznatky z podnikového řízení a na jejich základě kombinuje, racionalizuje, optimalizuje a zefektivňuje výrobní i nevýrobní procesy. Systematicky se zabývá projektováním, plánováním, zaváděním a zlepšováním průmyslových procesů a implementacemi.[2]

Kosky a kol. [3] říká, že se průmyslové inženýrství týká výroby jakéhokoliv ekonomického zboží v rámci ekonomiky. Průmyslové inženýrství pak nedefinují jako celek, ale člení jej na kategorie:

- Výroba a kontrola kvality
- Metody inženýrství
- Simulační analýza a operační výzkum
- Ergonomie
- Manipulace s materiálem

Jestliže se podíváme na průmyslového inženýra a požadavky, které jsou na něj kladeny, pak národní soustava povolání definuje průmyslového inženýra jako pracovníka, který plánuje, projektuje, řídí a implementuje komplexní integrované výrobní systémy a systémy pro poskytování služeb, zabezpečuje výkonnost a spolehlivost systémů, řídí náklady, zlepšuje procesy, zvyšuje produktivitu práce a efektivitu výroby. Mezi výčtem činností průmyslového inženýra v kategorii „Nutné“ nechybí ani vzdělávání a trénink pracovníků např. v oblasti zlepšování procesů a moderace workshopů. Z obecných dovedností je jako nejdůležitější uváděna počítačová způsobilost [4].

Ivan Mašín a Milan Vytlačil pak definují průmyslového inženýra jako osobu upozorňující ostatní inženýrské profese, že existuje něco jako obchodní realita, pomáhá překonávat mezeru mezi liniovými pracovníky a manažery. Průmyslový inženýr ví, že nákup drahého stroje nemusí znamenat podstatné zvýšení produktivity, musí mít nadhled a bere v potaz vždy celkové řešení, tedy je si vždy vědom souvislostí. [1]

### 3 Koncept Industry 4.0

V dnešní době už snad nemůže být průmyslový inženýr, který se ještě nesetkal s pojmem Industry 4.0. Ovšem problémem pro průmyslového inženýra může být definování tohoto pojmu. Existuje velké množství definic, např. Culot a kol. se pokusili analyzovat téměř 100 různých definic pojmu Industry 4.0. [5]

Jedna ze stručnějších definic Industry 4.0 např. říká, že je definován jako nová úroveň organizace a kontroly nad celým hodnotovým řetězcem životního cyklu produktů a je zaměřen na stále více individualizované požadavky zákazníků. Jedná se o realistický koncept zahrnující internet věcí, průmyslový internet, inteligentní výrobu a využívání cloudu. Jeho součástí je i integrace člověka do výrobního procesu takovým způsobem, aby se proces neustále zlepšoval a je zde zaměření se na činnosti s přidanou hodnotou a předcházení plýtvání. [6]

Autoři Oztemel a Gursev [7] na základě vlastního výzkumu definic pojmu Industry 4.0 nakonec definují tento pojem ve stručné verzi jako kombinaci šesti principů, a to interoperabilita, virtualizace, reálný čas, orientace na služby a modularita.

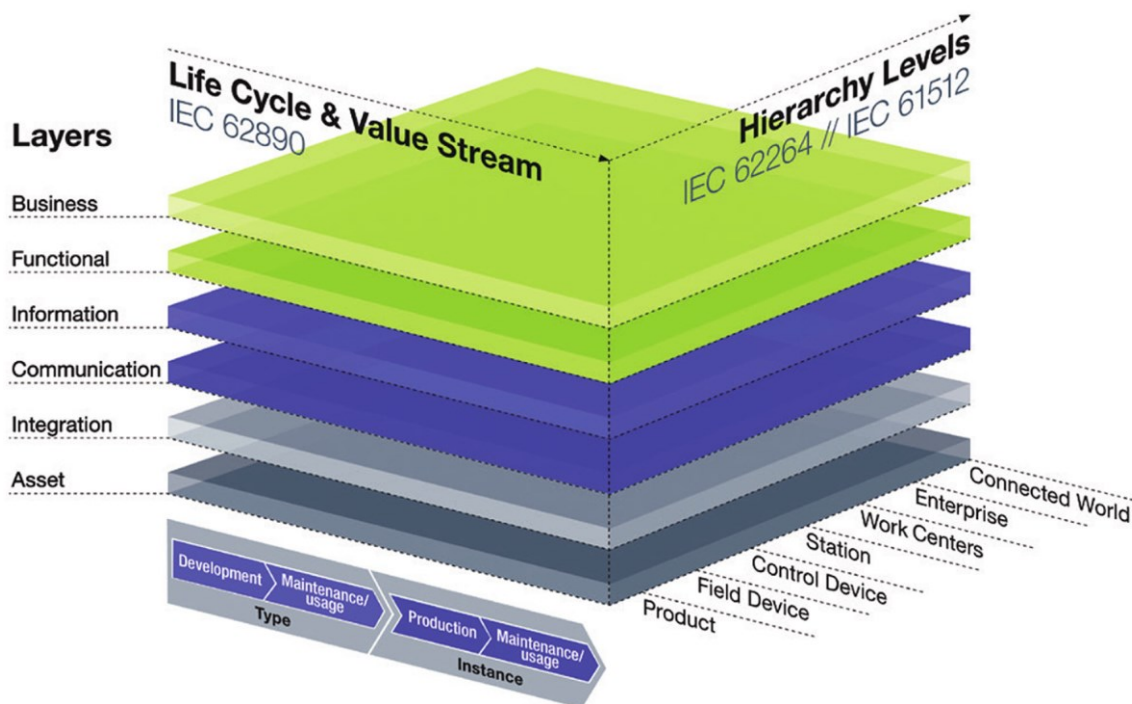
Rozsáhlejší definice pojmu Industry 4.0 se pak už často shodují v tom, že pojem Industry 4.0 je postavený na několika pilířích [8] [6] [9] [7]:

- Big Data a jejich analyzování
- Autonomní robotika
- Simulace
- Vertikální i horizontální systémové integrace
- Průmyslového internetu věcí (IIoT)
- Kyber-fyzikální systémy (CPS)
- Kyberbezpečnost
- Cloud
- Aditivní výroba
- Rozšířená reality (AR)

Kromě výše uvedených technologií a přístupů se dnes bere jako standardní a důležitá část konceptu Industry 4.0 i Machine learning (strojové učení) a Artificial intelligence (umělá inteligence). [10] [11]

S implementováním technologií kyber-fyzikálních systémů (CPS, detailně v kapitole 4.5) a IIoT, které jsou podle téměř všech definic součástí konceptu Industry 4.0, do podniků přichází i nový pojem označující tyto podniky jako Smart factory. [12]

Jiným způsobem, jak se dostat k určité definici pojmu Industry 4.0 a představě co tento pojem označuje, může být např. přes referenční model podle IEC PAS 63088:2017(E) snažící se o určitou standardizaci. Ve Specifikaci je popsána referenční architektura nazvaná „Model Industrie 4.0“, nebo také „RAMI4.0“, viz Obrázek 3-1, na kterou se při definování pojmu Industry 4.0 odvolává např. Han Soonhung [13]. Jedná se o kubický model, kde technické objekty jsou ve formě vrstev a je možné je popsat a sledovat po celou dobu jejich životnosti včetně přiřazení k technickým, nebo organizačním hierarchiím [14]. Toto vytváří představu, kam až sahá vliv pojmu Industry 4.0, a že se dotýká všech struktur a úrovní nejen v rámci podniku, ale i jeho širokého okolí.



Obrázek 3-1 – Model Industrie 4.0 (RAMI4.0) [14]

Situace s definováním pojmu Industry 4.0 se může komplikovat kromě svého rozsahu i tím, že se pojem Industry 4.0 (popř. česky Průmysl 4.0) často zaměňuje s pojmem 4. průmyslová revoluce, kdy velká část autorů je považuje za synonyma a pouze malá část autorů v těchto pojmech vidí rozdíl [15]. Tato práce se zabývá aktuálním stavem konceptu Industry 4.0 a využíváním jeho principů, není tedy potřeba používat i pojem 4. průmyslová revoluce, který může evokovat např. časově ohraničený a ukončený historický milník a zanášet tak do práce nejasnosti, které by mohla záměna těchto pojmů způsobit.

Mnohdy se lze setkat i s označením Smart factory (první zmínka v USA [16]), které je často chápáno jako synonymum k pojmu Industry 4.0, popř. jeho překlady dle země, jako např. Fabbrica Intelligente (Itálie), Factory of the future, nebo Digital factory [17]. Většina autorů ovšem chápe pojem „chytrá továrna“ jako důležitou část nebo klíčový prvek konceptu Industry 4.0 [18] [19]. Pojem Smart factory je tedy prvkem konceptu Industry 4.0, který často představuje kyber-fyzikální systém.

4 klíče k porozumění konceptu Industry 4.0 jsou dle Yáñez [16]:

- Německo jako první země v Evropě zavedla high-tech strategii nazvanou Industry 4.0 pro zvýšení konkurenceschopnosti německého průmyslového odvětví. Paralelně se v Americe objevil koncept Smart factory.
- Koncept je založený na kyber-fyzikálních systémech, kombinujících fyzickou infrastrukturu se softwarem, senzory, nanotechnologiemi a komunikačními technologiemi.
- Pokrok v těchto nových technologiích a vznik adekvátních obchodních modelů způsobí hospodářskou a sociální nerovnováhu, kterou budou muset zvládnout vlády.
- Industry 4.0 změní svět.

Pokud pak přijmeme Smart factory jako prvek konceptu Industry 4.0, musí splňovat 6 kritérií [16]:

- **Je automatizovaná** – Zrychluje a automatizuje neustále se opakující procesy a procesy s nižší přidanou hodnotou pomocí kobotů.
- **Je digitalizovaná** – Začleňuje elektroniku pro hromadný sběr dat a řízení výrobních procesů v reálném čase.

- **Je inteligentní** – Interpretuje procesní data a umožňuje prediktivní přístup.
- **Je flexibilní** – V návrhu, výrobě, nebo logistice se okamžitě přizpůsobuje změnám v poptávce.
- **Je udržitelná** – Racionálně a zodpovědně využívá zdroje a energii
- **Je lidská** – Továrna může být Smart pouze za předpokladu, že řádně školí své zaměstnance a využívá jejich talent. Lidé v ní i nadále zůstávají centrem činnosti a mají v něm vliv.

### 3.1 Adaptace Industry 4.0 v podnicích

V České republice existuje podle Vrchoty a kol. [20] závislost mezi velikostí podniku a využíváním prvků konceptu Industry 4.0. Schopnost využívat a adaptovat principy Industry 4.0 rostou s velikostí podniku. Dle průzkumů nejhůře vychází SME. Výsledky ukazují, že v rámci Německa je situace stejná [21].

Aby se situace SME zlepšila, je potřeba zavést určité podpory [21]:

- Investice do Industry 4.0 v rámci SME musí být podporovány veřejným financováním.
- Je potřeba požadovat a zavádět interní zaměstnanecké kvalifikační programy a vzdělávací programy pro školy a univerzity.
- SME musí být podporovány jako samostatná skupina, protože mají nižší schopnost vyrovnat se s finančními, technologickými a personálními výzvami v porovnání s velkými podniky.

Paradoxně z pohledu vlastnické struktury podniků podle Šoltése [22] vychází SME jako nejlepší adepti na zavádění prvků Industry 4.0. Podle tohoto kritéria je možné rozdělit podniky do tří skupin:

- České pobočky nadnárodních korporací, které v zásadě nejsou schopné určit, kdy a jak implementovat nové procesy.
- České společnosti vlastněné velkou finanční skupinou, a proto jsou více nakloněny krátkodobé vizi vedení společnosti, a tedy je méně pravděpodobné, že se v přechodu na Industry 4.0 stanou lídry na českém trhu.
- SME, které mají možnost volně rozhodovat o své budoucnosti. U nich však může existovat nedostatek finančních prostředků a chybějící informace.

### 3.2 Nástupce konceptu Industry 4.0

Pojem Industry 4.0 označuje aktuální trendy digitalizace a automatizace výroby včetně jejich dopadů na trh práce. V rámci snahy vymezit pojem Industry 4.0 byly nalezeny i publikace, jež zmiňují možného nástupce, nebo jsou přímo zaměřené na pokračovatele označovaného jako Industry 5.0 nebo 5. průmyslová revoluce. V rámci představ nebo při definování konceptu Industry 5.0 se autoři v tuto chvíli často rozcházejí, přesto může být důležité vědět, jakým směrem by se mohl vývoj ubírat. I když se většinou nejedná o „revoluci“, ale spíše evoluci stávajícího pojmu Industry 4.0. Ve skutečnosti pak může být další průmyslová revoluce odstartována něčím úplně jiným, např. celosvětovou pandemií, jakou prožíváme nyní, nebo přechodem na kvantovou fyziku ve světě počítačů.

Evropský hospodářský a sociální výbor (EHSV, v originále EESC – European Economic and Social Committee) popisuje Průmysl 5.0 jako zaměřený na kombinaci kreativity a řemeslného umění lidí s rychlostí a produktivitou robotizované výroby [23].

Podle Petera Sachsenmeiera a kol. [24] bude spouštěčem 5. průmyslové revoluce Bionika (napodobování nebo abstrakce „vynálezů“ přírody) a v ještě větší míře syntetická biologie. Dle autorů to bude rozvoj v tomto odvětví pro průmysl a strojírenství stejně zásadní jako křemíkový čip za posledních 50 let. Jiní autoři [25] zase vidí smysl Industry 5.0 v odhlédnutí od masové produkce a zaměření se na udržitelnost a životní prostředí.



Jiný pohled na vývoj představuje koncept Industry 5.0 jako zaměření se na personalizaci výrobků a autonomii výroby, přičemž jako nejdůležitější pilíře by měly být [26]:

- Síťové sdílení dat ze senzorů
- Víceúrovňové dynamické modelování a simulace s využitím DT
- Shopfloor trackers
- Virtuální trénink
- Skutečně inteligentní autonomní systémy
- Pokročilé snímací technologie, vnímání okolí stroji a napojení strojů na lidský mozek

Podle Özdemira a kol. [27] je Industry 4.0 zaměřený na extrémní automatizaci a z toho plynoucí extrémní propojování dle autorů přímo „*všeho se vším*“. Podle nich tedy bude koncept Industry 5.0 zaměřený na zabezpečení a zvýšení robustnosti systémů vytvořených na bázi Industry 4.0. Mělo by se jednat o ochranu jak před systémovými riziky, jako např. kolaps sítě, nebo jednotlivých prvků/částí systému, tak napadením hackery, viry nebo možným ohrožením způsobeným sociálními nebo politickými změnami a tlaky.

## 4 Simulace v konceptu Industry 4.0

Definice simulace říká, že simulace je napodobování průběhu procesu z reálného světa nebo chování systému v průběhu doby [28]. Simulace je vědeckou metodou [29]. Abychom mohli simulace provádět, je potřeba mít simulační model, nad kterým se simulace provádí – simulace je tedy jinými slovy experimentování s modelem za účelem poznání jeho chování.

Simulace jako vědecká metoda má několik výhod, podle [30] to mohou být např.:

- Ochrana reálného systému a objevení chyb před ostrým nasazením nových politik, operačních procedur, rozhodovacích pravidel, při změně toku informací v rámci systému apod.
- Testování nového designu, rozvržení layoutů, transportních systémů apod. předtím, než se provedou investice a případné nákupy nových zařízení.
- Testování hypotéz, proč v rámci reálného systému dochází k určitým stavům, situacím.
- Analýza úzkých míst a testování případných návrhů jejich řešení.
- Simulační studie je cesta, jak skutečně poznat chování vlastního systému včetně reakcí, rozhodování a chování jedinců.
- Simulace odpovídá na otázky typu „Co když...“.

Pelánek [31] vidí využití simulací a modelů v návrhu a řízení systémů, předpovídání chování, porozumění, učení, tréninku a zábavě.

Kromě množství výhod se můžeme setkat i s nevýhodami. Některé z nich podle [28], [32] a [30] jsou:

- Pro tvorbu modelů, simulaci i interpretaci výsledků simulace je potřeba zkušeného odborníka.
- Výsledky simulačních modelů může být složité interpretovat.
- Tvorba modelu, simulace a analýzy mohou být časově náročné.
- Rychlost simulací. Včera byly simulace pomalejší než dnes, zítra budou naopak rychlejší. Tento bod odkazuje na neustálý nárůst výkonu počítačů.

V případě počítačových simulací můžeme mluvit o základním rozdělení simulací na diskrétní simulace a spojitě simulace [33]. Diskrétní simulace je charakteristická změnou stavu modelu nespojitě – skokově. Tedy stav modelu se změní až ve chvíli, kdy nastane nějaká změna. V případě spojitě simulace se zjišťuje stav modelu v pravidelném časovém intervalu.[34]

Jako typický příklad diskrétní simulace lze uvést např. strojírenskou výrobu, kdy nás zajímají změny stavu ve výrobě – ty nebudou v pravidelných časových intervalech. Jako spojitou simulaci lze uvést např. míšení dvou tekutin, kdy se zjišťuje stav míšení v periodicky se opakující časové konstantě, např. každou sekundu.

### 4.1 Systém

Systém je definovaný jako soubor entit, např. lidí, strojů, které provádí nějakou činnost a interagují spolu [34]. Tyto entity mají své vlastnosti (atributy), jejichž hodnoty se mohou v čase měnit. V rámci systému nestačí pouze znát jednotlivé entity, ale je potřeba vědět, jak jsou spolu propojené a jak na sebe reagují, jinými slovy je potřeba vědět, jak se systém chová jako celek.

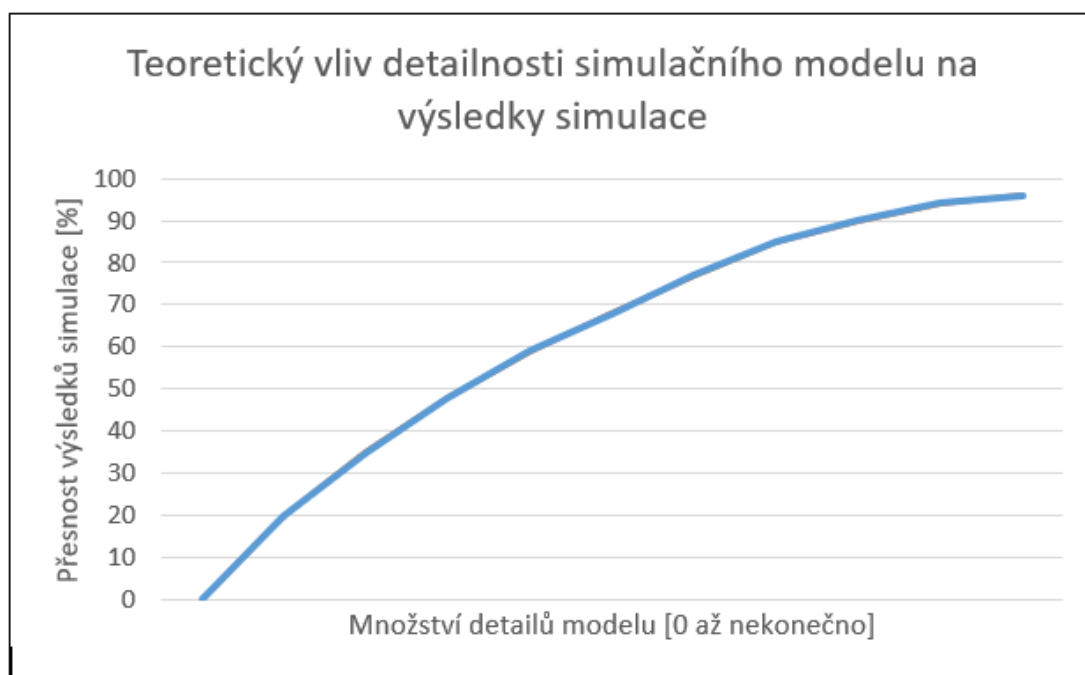
### 4.2 Model

Model je napodobenina systému jiným systémem vytvořeným např. v digitální formě, tedy jedná se o abstraktní reprezentaci systému [32]. Tvorbu modelu označujeme jako modelování.

Model může být v principu matematický nebo fyzický. Matematický model využívá symbolické notace a matematických rovnic k reprezentaci systému, je tedy abstraktní. Digitální simulační model je konkrétně matematickým typem [32].

Fyzický (fyzikální) model vychází z fyzikální nebo geometrické podobnosti mezi modelovaným systémem a modelem a je hmatatelný (fyzický). Příkladem může být model automobilu pro aerodynamické zkoušky.

Dle Banksyho [28] je při tvorbě modelu důkladně zanalyzovat prostředí/systém na základě kterého se vytváří model. To je časově velmi náročné jak pro odborníka, jenž má model vytvořit, tak pro stranu uživatele/vlastníka systému, pro který se model vytváří. Důvodem je potřeba důkladně vysvětlit, jak daný systém funguje a často to i zjistit, protože ne vždy uživatel/vlastník systému rozumí do potřebných detailů a úrovně (jde především o vyloučení velkého zkreslení, nebo zjednodušování, které by mělo fatální dopady na důvěryhodnost a reálnost modelu). Při tvorbě modelu je pak potřeba jej udělat dostatečně komplexní, aby simulace poskytovala rozumné a použitelné výsledky, ale zároveň se komplexnost modelu musí udržet v rozumných mezích, protože se vzrůstající komplexností rapidně narůstá pracnost modelu, náklady na tvorbu modelu i doba běhu simulace. Vztah přesnosti výsledků simulace a detailnosti modelu je znázorněn grafem na následujícím obrázku, viz Obrázek 4-1.



Obrázek 4-1 – Teoretický vliv míry abstrakce na přesnost modelu, zpracováno dle textu [28]

Tvorba simulačního modelu se v základu skládá ze šesti základních fází, přičemž tyto fáze se dále rozpadají na základní kroky [31]:

1. Formulace problému
2. Základní návrh modelu
3. Implementace
4. Verifikace a validace
5. Simulace a analýza
6. Sumarizace výsledků

### 4.3 Nástroje pro diskrétní simulace

V principu lze využít tří typů simulačního softwaru pro tvorbu simulací:

- Využití univerzálního programovacího jazyka, dříve např. FORTRAN, ALGOL, dnes by to mohla být navíc např. JAVA, C/C++, C# apod.
- Využití simulačních programovacích jazyků, jako příklad lze uvést SIMAN, SIMULA, SIMULA 67 apod. Tyto programovací jazyky jsou přímo navrženy a vytvořeny pro simulace. Uspadňují tedy jak jejich tvorbu a běh, tak i řízení, i když se stále jedná o programovací jazyky.
- Využití komplexních simulačních prostředí jako např. Tecnomatix Plant Simulation, ARENA, WITNESS apod. Tato simulační prostředí často využívají některý ze standardních simulačních jazyků nebo mají svůj vlastní. Např. simulační prostředí Tecnomatix Plant Simulation obsahuje objektově orientovaný simulační jazyk SimTalk [35].

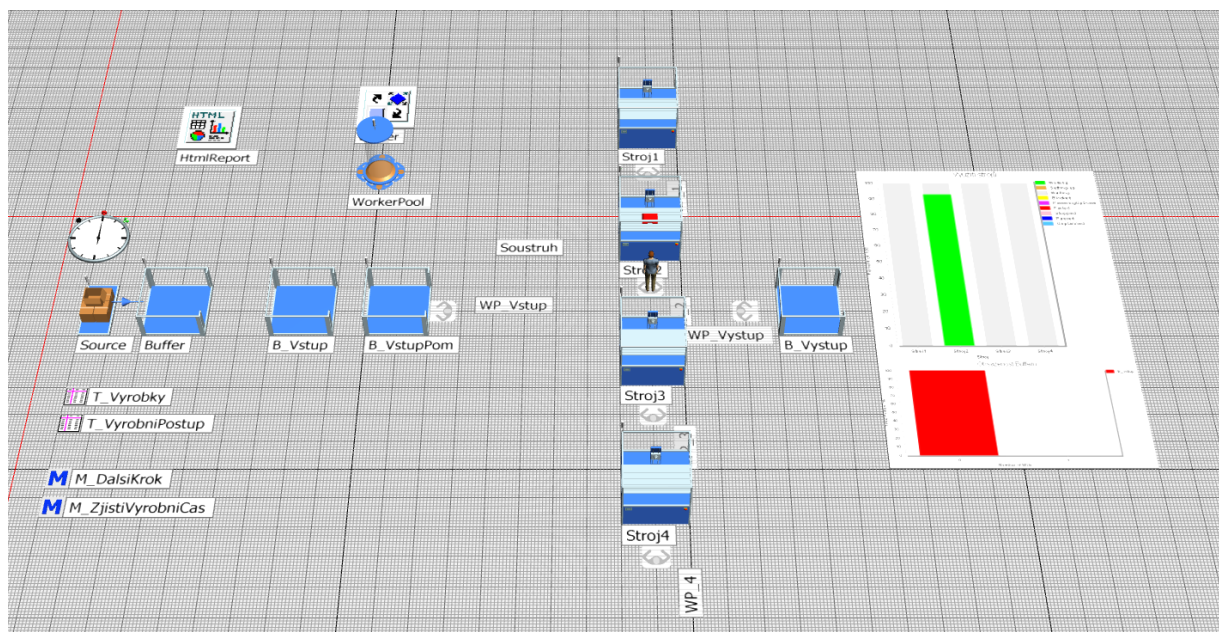
Autoři Debnár, Košturiak a Kuric [36] dělí simulační software do generací, viz Tabulka 4-1. Dnes by se už mohlo mluvit o 7. generaci využívající ML, AR nebo např. VR.

Tabulka 4-1 – Generace simulačního software dle [36]

Vývoj	Doba	Charakteristika	Příklad SW
1. generace	50'	Obecné programovací jazyky, žádná speciální podpora pro simulaci	FORTTRAN, ALGOL ASSEMBLER
2. generace	60'	Programovací jazyky, podpora pro tvorbu náhodných čísel, zpracování statistiky	GPSS, SIMSCRIPT GASP II, SIMULA
3. generace	70'	Diskrétní události, spojitá a kombinovaná simulace	GASP IV, ACSL
4. generace	80'	Speciálně problémově orientované simulátory, animace, simulace jako projekt	SIMAN/CINEMA, SIMFACTORY, SEEWHEY
5. generace	85'	Umělá inteligence a expertní systémy v simulaci	SIMKIT, SIMULATION CRAFT
6. generace	90'	Objektově-orientované simulační systémy	SIMPLE++, ARENA, WITNESS

V rámci KPV/ZČU se využívá jak pro výuku, tak pro projekty komplexní simulační prostředí Tecnomatix Plant Simulation od společnosti Siemens PLM Software. Podle Debnára a kol. [36] se tento nástroj řadí do 6. generace simulačního softwaru. Ukázka spuštěné simulace v tomto nástroji je na následujícím obrázku, viz Obrázek 4-2.

Existují i open-source řešení, jako příklad plnohodnotného open-source simulačního prostředí pro diskretní simulace lze uvést JaamSim, který Sebastian Lang a kol. [37] hodnotí ve svém výzkumu jako nejlepší z porovnávaných open-source simulačních nástrojů.

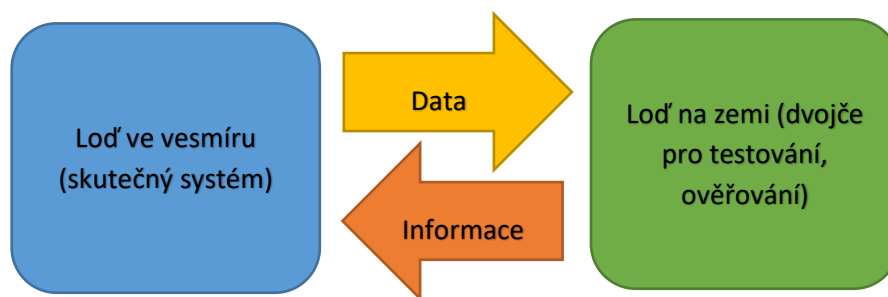


Obrázek 4-2 – Spuštěná simulace v nástroji Tecnomatix Plant Simulation

#### 4.4 Digitální dvojče

Prakticky první skutečné využití dvojčete si připsala americká vesmírná agentura NASA během Apollo programu, kdy zavedla systém dvojčat. Princip spočíval v tom, že byly dvě stejné lodě. Jedna, která letěla do vesmíru, a pak dvojče, na němž před odletem probíhaly tréninky posádky a během vesmírné mise pak sloužilo dvojče na zemi k ověřování postupů a řešení případných problémů před jejich provedením na sourozenci ve vesmíru. Toto ověřování postupů probíhalo podle skutečných dat z lodě ve vesmíru. [38]

Digitální dvojče pak lze názorně vysvětlit obrázkem vycházejícím z výše popsaného principu, viz Obrázek 4-3.

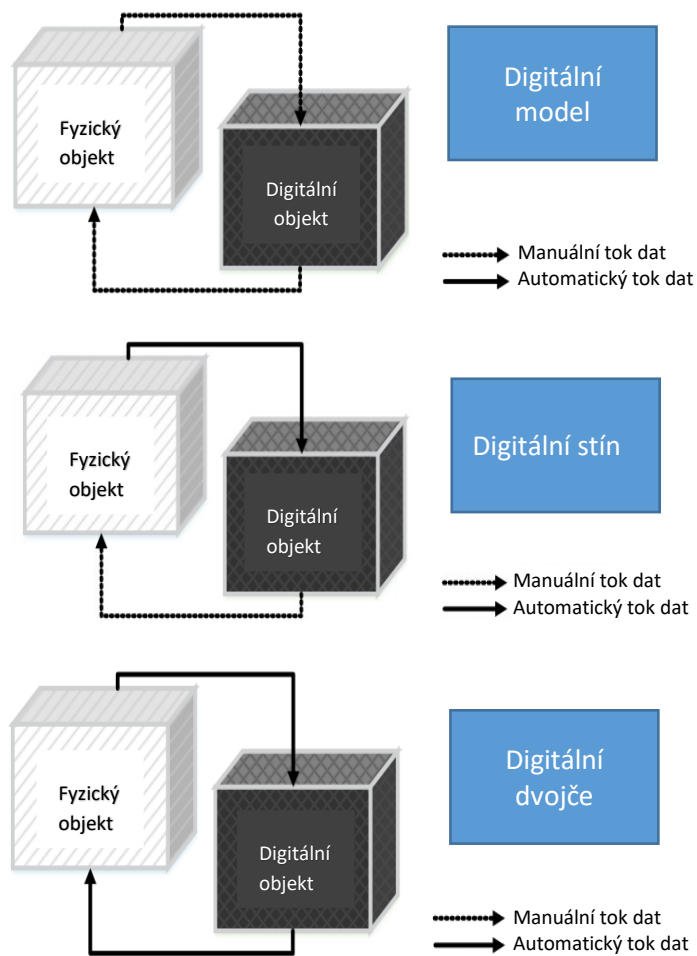


Obrázek 4-3 – Konceptuální model dvojčete dle [39], vlastní úprava

Podle [39] označuje pojem Digitální dvojče procesy a metody pro popis a modelování vlastností, chování, procesu formování a výkonu fyzických objektů pomocí digitální technologie. Digitální model dvojčat odkazuje na virtuální model, který odpovídá fyzickým entitám v reálném světě a je s nimi konzistentní, může tak simulovat jeho chování a výkon v reálném čase. Lze říct, že digitální dvojčata jsou techniky, procesy a metody a že modelem digitálního dvojčete jsou objekty, modely a data. Technologie digitálních dvojčat nevyužívá pouze lidské teorie a znalosti k vytváření virtuálních modelů, ale také může pomocí technologie simulace virtuálních modelů zkoumat a předpovídat neznámý svět, hledat lepší způsoby a prostředky a podporovat inovativní myšlení člověka. Snaha o optimalizaci a pokrok proto v technologii digitálních dvojčat poskytuje nové nápady a nástroje pro současné inovace a vývoj výroby.

Podle způsobu předávání dat a informací mezi fyzickým systémem a digitálním dvojčetem lze ještě rozlišovat DM (Digital Model/digitální model), DS (Digital Shadow/digitální stín), a DT (Digital Twin/digitální dvojče). Rozdíl podle [40] je ve způsobu předávání dat a informací mezi nimi, tedy zda předávání probíhá automatizovaně, nebo manuálně, viz Obrázek 4-4. Při této interpretaci lze Kritzingerovo digitální dvojče považovat za kyber-fyzikální systém.

Z tohoto obrázku vyplývá, že o digitálním dvojčeti mluvíme ve chvíli, kdy se data i informace předávají automatizovaně. V případě digitálního modelu se data i informace předávají manuálně a v případě digitálního stínu se data předávají automaticky, ale informace manuálně.



Obrázek 4-4 – Digitální model, stín, dvojče podle [40], vlastní úprava

Je potřeba říct, že někteří autoři zatím nerozlišují mezi DM, DS a DT a vše označují jako DT. Pak v jejich studiích nebo odborných článcích nelze vždy zcela určit míru propojenosti a je potřeba odhadovat, zda se jedná o DM, DS nebo DT z kontextu jejich článku/publikace. Existuje i jiný přístup, dle [41], který rozřazuje digitální dvojčata do pěti úrovní zralosti, přičemž autor vůbec nebere v potaz dělení na DM/DS a DT, přestože také zvolil jako klíč schopnost výměny informací a dat mezi fyzickým systémem a digitálním dvojčetem. Popis těchto pěti úrovní je na následující tabulce, viz Tabulka 4-2.

Tabulka 4-2 – Pětiúrovňové dělení Digitálních dvojčat (úroveň zralosti digitálního dvojčete) podle Yong-Woona [41]

Level (úroveň)	Název	Funkční požadavky na zpracování	Příklady
Level 5	Autonomní digitální dvojčata	<ul style="list-style-type: none"> <li>Autonomní operace pomocí on-line synchronizace a orchestrace bez jakéhokoliv lidského zásahu</li> </ul>	N/A
Level 4	Interaktivní digitální dvojčata	<ul style="list-style-type: none"> <li>Federovaný, synchronizovaný a interaktivní operace s digitálními dvojčaty se zásahy člověka</li> <li>Synchronizace pomocí rozhraní (např. dig. vlákno) během životního cyklu fyzického dvojčete</li> </ul>	N/A
Level 3	Dynamická digitální dvojčata	<ul style="list-style-type: none"> <li>Chování a dynamika modelování pro provoz a simulaci</li> <li>Simulace poskytuje odpovědi na otázku „Co když...“</li> <li>Analýzy příčin reprodukční simulací</li> <li>Synchronizace pomocí datové linky (např. MTConnect) během provozní doby</li> </ul>	CAE, Digital factory, Virtual Singapore, HILS, CPS, etc.
Level 2	Statická digitální dvojčata	<ul style="list-style-type: none"> <li>Inicializační trvalé a statické datové připojení</li> <li>Použita logika řízení procesu, ne model chování a dynamiky systému</li> <li>Monitorování v reálném čase</li> <li>Z části automatizovaná kontrola, ale z většiny využívány lidské zásahy</li> </ul>	SCADA, DCS, CAM, etc.
Level 1	Tvářící se jako digitální dvojčata	<ul style="list-style-type: none"> <li>Fyzická entita modelována takovým způsobem, aby model měl podobnou vizuální podobu, vykreslená ve 2D nebo 3D</li> </ul>	CAD, etc.

Digitální dvojčata jsou jedním z nejdůležitějších trendů v konceptu Industry 4.0 a průmyslovém inženýrství [42], někteří autoři přímo označují dnešní dobu Industry 4.0 jako éru digitálních dvojčat [43]. Koncept Industry 4.0 rozšiřuje možnosti a využití digitálních modelů, digitálních stínů a digitálních dvojčat jak např. pro podporu rozhodování a plánování výroby [44], řešení nečekaných nastalých situací/problémů nebo predikování takových situací [45, 46], tak i vzdělávání a přenos znalostí vedení, managementu a výkonných pracovníků [47]. DM, DS, nebo DT může být jak čistě digitální, tak teoreticky i kombinace digitálního a hardwarového modelu. Mohl by být vytvořen i čistě hardwarový model řízený i napřímo online daty ze skutečné výroby – takové řešení bychom neoznačili jako digitální dvojče, ale jako hardwarové dvojče, popř. pouze dvojče.

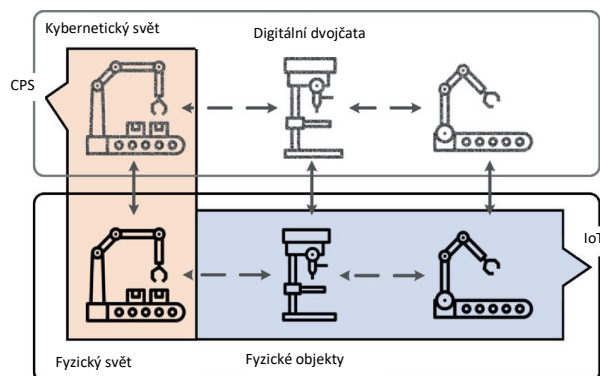
#### 4.5 Cyber-Physical System a DT

V souvislosti s konceptem Industry 4.0 se čím dál častěji zmiňuje i využívání kyber-fyzikálních systémů (někdy česky označované i jako kyberfyzické, nebo kyber-fyzikální systémy). CPS je systém skládající se z fyzických entit řízených počítačovými algoritmy, které těmto entitám umožňují fungovat zcela samostatně včetně autonomního rozhodování, tedy např. mohou řídit svěřený technologický celek, nebo mohou být samostatným členem komplexních výrobních celků. Stejně jako je průmyslové inženýrství multioborová specializace, platí toto označení i pro samotné CPS systémy, které vyžadují znalosti kybernetiky, mechatroniky, informatiky, konstrukční a výrobní vědy apod. [48]. Často jsou postaveny na umělé inteligenci a strojovém učení [49], využívají simulací pro rozhodování [50] a další oblasti informatiky, které se rozvíjejí v rámci konceptu Industry 4.0.

CPS i DT jsou zaměřeny na dosažení kyber-fyzikální integrace, která je potřeba pro inteligentní výrobu, ale každý z přístupů klade důraz na něco jiného. Zatímco CPS se zaměřuje na senzory a akční členy, DT se zaměřuje na data a modely.[51]

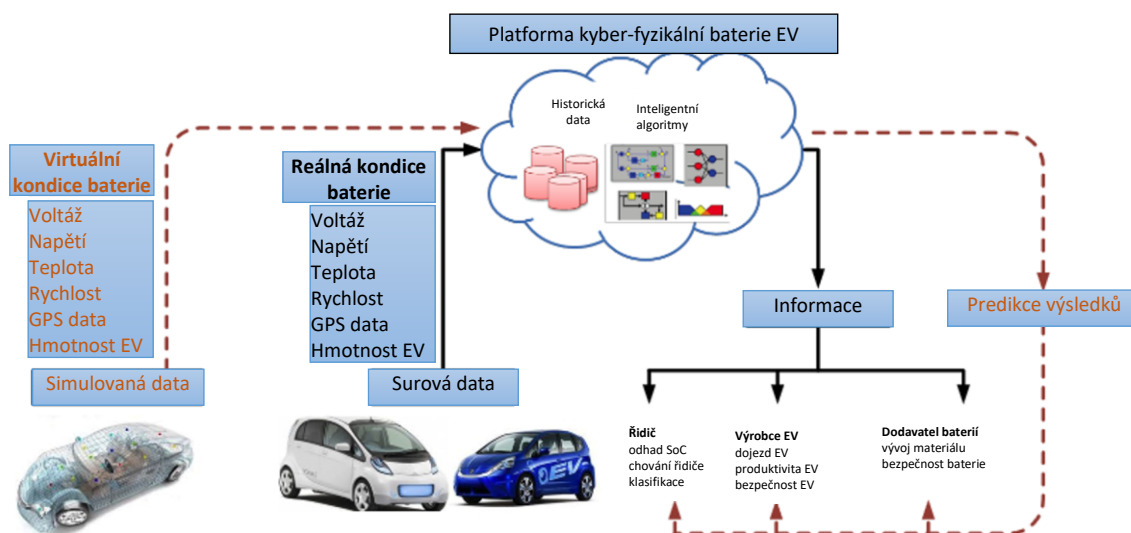
Vztah mezi CPS a DT lze popsat tedy takovým způsobem, že CPS využívá DT, a existující DT je nezbytný předpoklad pro CPS[52]. Ukázka vztahu mezi CPS, DT a IoT, viz Obrázek 4-5. Lze tedy říct, že propojením

simulačního modelu s reálnou výrobou vzniká kyber-fyzikální systém a můžeme jej označit jako Smart factory.



Obrázek 4-5 – vztah mezi CPS, DT a IoT [53], vlastní úprava

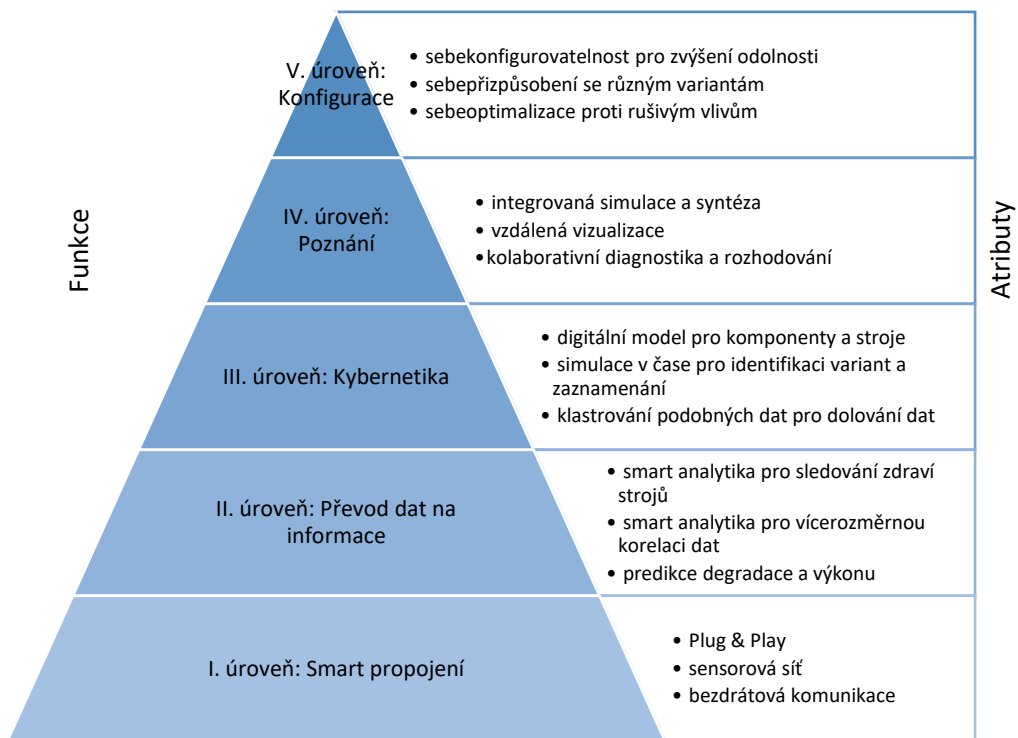
Kyber-fyzikální systémy produkují velké množství dat, proto se u nich předpokládá zpracování a analýzy Big Data, která pak mohou být využívána pro simulace pro potřeby prediktivity, plánování a rozhodování [54]. Ukázka možné platformy je na následujícím obrázku, viz Obrázek 4-6.



Obrázek 4-6 – Kyber-fyzikální systém využitý pro baterie elektromobilů [54], vlastní úprava

Potřeba DT pro CPS je znázorněna i ve standardně používané 5C architektuře ve III. úrovni „Kybernetika“. Integrovaná simulace je pak ve IV. úrovni „Poznání“. Ukázka 5C architektury je na následujícím obrázku, viz Obrázek 4-7.





Obrázek 4-7 – 5C architektura kyber-fyzikálního systému [50], vlastní úprava

## 5 Koncept STEM a prototypovací hardware

V rámci technického vzdělání se stále ve větší míře prosazuje koncept STEM, který vznikl v USA v 90. letech minulého století pro označení integrovaného vzdělávání v oborech přírodní vědy (Science), techniky (Technology), technologie (Engineering) a matematiky (Mathematics) [55–59], kdy vzdělávání v těchto oborech neprobíhá separátně, ale v rámci projektů kombinujících potřebné znalosti z více oborů. Přestože tento koncept není žádnou novinkou, je v dnešní době dále rozvíjen a rozšiřován. Jako příklady lze uvést:

- STEAM (A – Arts, schopnost tvořit, formulovat, prezentovat) [60–62]
- STREAM (R – Riting, zvládnutí jazyka vědy) [63]
- STEMIE (IE – Include everyone, tedy každý může být vzděláván) [64]

Kromě využití v technickém vzdělávání se nástroje vytvořené na konceptu STEM čím dál tím častěji a stále ve větší míře využívají v profesním životě průmyslových inženýrů a technicky zaměřených pracovníků, včetně akademiků, kdy slouží např. pro řízení, sběr dat, vyhodnocování dat nebo pro levné a jednoduše nahraditelné lokální servery (např. postavené na počítačích z rodiny Raspberry Pi) či pro tvorbu digitálních dvojčat, prototypů apod., viz část této práce zabývající se rešerší zaměřenou na tuto problematiku. Tyto produkty jsou zároveň využívány technicky zaměřenými uživateli pro jejich „bastlení“, tedy např. pro vlastní řešení chytrých domácností [65] apod.

Pojem „bastlení“ se využívá jako český výraz pro v anglickém jazyce používané označení „tinkering“, vychází z německého „basteln“ a česky je mu blízké slovo „kutit“ [66]. Bastlíři bývají velmi technicky zdatní a často jsou schopni získat na svá řešení i patenty, ale nejsou profesionály a bastlení je pro ně hlavně koníčkem [67].

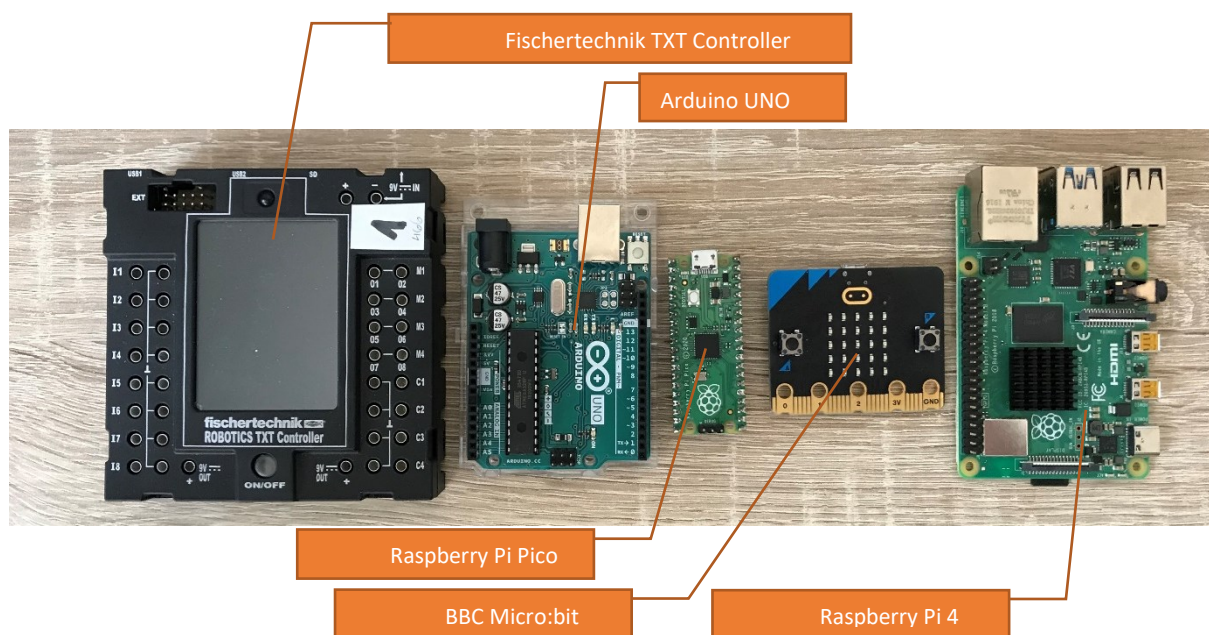
V době Průmyslu 4.0 lze určitě tento koncept rozšířit o velké množství dalších oblastí, jako jsou např.:

- Komunikace (jak ve smyslu způsobu komunikace a toku informací mezi technologicky moderními výrobními stroji, tak i komunikace mezi lidmi různých odborností (ekonomové, právníci, strojní inženýři, elektroinženýři, matematici, programátoři, technickohospodářští pracovníci, provozní zaměstnanci a také i uživatelé).
- Vyhledávání a zpracování informací s aplikací kritického myšlení.
- Samostatnost – ve smyslu obsáhnutí úplných základů většího rozsahu oborů, jejich implementace a využití.
- Úroveň znalostí – ve smyslu vysoké odbornosti, specializace v některé ze zvolených vědních disciplín.
- Simulace – ve smyslu schopnosti simulovat složité technologické výrobky zjednodušeným vlastním fyzickým modelem postaveným a řízeným vlastním programem.

Vzdělávacích řešení postavených na konceptu STEM [68] a jeho derivátech je na trhu velké množství. Lze se setkat s kompletními stavebnicemi, které pokrývají vše od návrhu a konstrukce, přes elektrotechniku až po programování, nebo samostatné vývojové počítače, kde je pouze na uživateli, jaké nářadí a materiál zvolí pro sestavení modelu, a začíná např. pouze se zvoleným typem programovatelné řídicí jednotky. Jako příklad komplexních stavebnic lze uvést Lego MINDSTORMS [69, 70], stavebnici Fischertechnik [71], českou stavebnici MERKUR [72] nebo pro malé děti např. SmartLab Toys [73].

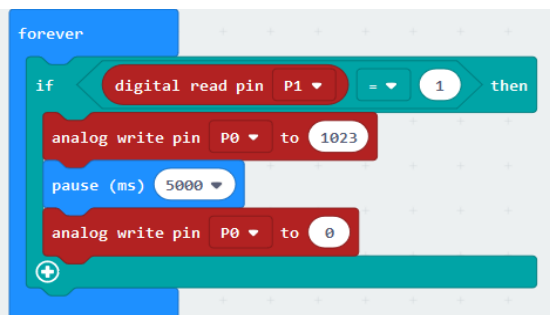
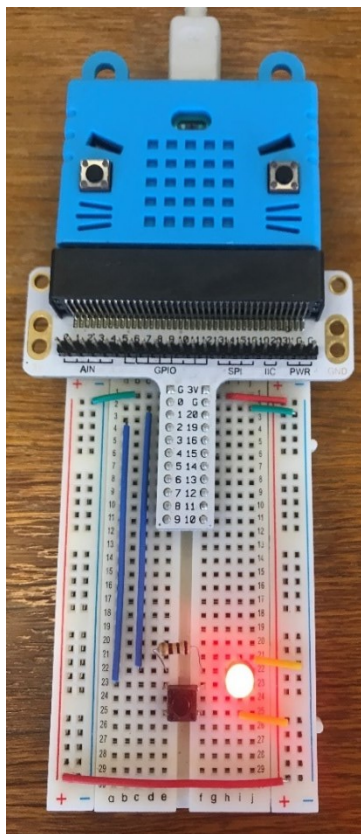
Příkladem samostatných mikrokontrolerů je velmi rozšířený vzdělávací jednodeskový počítač BBC Micro:Bit [74], počítače z rodiny Raspberry Pi [75], Arduino [76] a jeho klony nebo Edison robot [77]. Lze je pořídit jak samostatně, tak i v různě bohatých setech od různých dodavatelů, často doplněných postupy pro vytvoření základních projektů, jež využívají doplňky z daného setu. Většinu těchto stavebnic a počítačů lze programovat pomocí vizuálních programovacích jazyků/prostředí na bázi jazyka/prostředí SCRATCH (<https://scratch.mit.edu>) - v některých případech i přímo ve SCRATCH a pak

je téměř vždy dostupné programování v C/C++, Pythonu, a např. u BBC Micro:Bit i JavaScriptu. Je tedy možné zvolit si způsob programování vlastních modelů podle aktuálních schopností a potřeb a postupně se propracovat hlouběji k „dospělým“ programovacím jazykům jako je např. Python nebo C/C++. Zde je třeba ještě zmínit, že v případě Raspberry Pi se jedná o plnohodnotný armový počítač, na kterém lze provozovat OS včetně grafického prostředí s výkonem dostačujícím pro základní kancelářskou práci, vývoj aplikací nebo hraní her (v případě Raspberry Pi 4 se 4 nebo 8GB RAM) a lze v něm programovat v jakémkoliv jazyce dostupném pro linuxové distribuce [78]. Ukázka některých mikrokontrolerů a mikropočítačů je na následujícím obrázku, viz Obrázek 5-1.



Obrázek 5-1 – Řídící jednotky (mikrokontrolery a mikropočítače)

Jako triviální ukázkou vytvořeného a naprogramovaného modelu lze uvést např. model využívající počítač BBC Micro:bit pro ovládání LED tlačítkem. K mikropočítači je přes odpor a tlačítko připojena LED pomocí nepájivého pole. Stisknutím tlačítka dojde k rozsvícení LED na dobu 5 sekund. Program je zacyklený, aby jej bylo možné opakovaně spouštět. Pro ukázkou bylo využito nastavení analogové hodnoty kvůli možnosti řízení intenzity světla pomocí PWM, proto jsou hodnoty v rozsahu [0,1023] a ne 0/1. Ukázka modelu včetně řídicího kódu v programovacím grafickém nástroji BBC Micro:bit MakeCode a programovacích jazycích JavaScript Python jsou na následujícím obrázku, viz Obrázek 5-2.



Program v JavaScriptu:

```
basic.forever(function () {  
  if (pins.digitalReadPin(DigitalPin.P1) == 1) {  
    pins.analogWritePin(AnalogPin.P0, 1023)  
    basic.pause(5000)  
    pins.analogWritePin(AnalogPin.P0, 0)  
  }  
})
```

Program v pythonu (přesněji řečeno MicroPythonu):

```
from microbit import *  
  
while True:  
    if pin1.read_digital() == 1:  
        pin0.write_analog(1023)  
        sleep(5000)  
        pin0.write_analog(0)
```

Obrázek 5-2 – Ukázka modelu s využitím BBC Micro:bit a řídicího kódu v BBC Micro:bit MakeCode, JavaScript a MicroPython

## 5.1 Komplexní STEM modely a Industry 4.0

Pro praktické seznámení s konceptem Industry 4.0 je vhodné využívat fyzických modelů, které podpoří zájem, představivost, kreativitu i zvýšení úrovně pochopení dané problematiky [79, 80]. Tím se dá docílit i zvýšení náročnosti studia, aniž by se to negativně projevilo na neúspěšnosti, nebo nepochopení problematiky. Výhodou takových modelů je i možnost ukázky procesního myšlení a vybraných metod jako jsou Lean, Six Sigma, TOC a další ve výrobní praxi [81]. Pro vzdělávání v problematice Industry 4.0 jsou tedy v podstatě 4 možnosti, jak můžeme postupovat, popř. jejich kombinace:

1. Nevyužívat fyzických modelů (vzdělávacích pomůcek)
2. Využívat vlastních modelů (vlastní stavební prvky, např. využití 3D tisku)
3. Využívat stavebnic modelů
4. Využívat koupených od výrobce sestavených modelů

Jak uvádí Votrubec a Koblasa [82], lze postavit poměrně složitý model ilustrující koncept Industry 4.0 na vývojové platformě Arduino s komunikací pomocí wifi sítě a samostatným AGV, kde pak lze řešit i takové problémy, jako je např. přesnost navádění a plynulost jízdy AGV v kombinaci fyzické úpravy a optimalizace algoritmu pro vyhodnocení sledování čáry (popř. návržení nového algoritmu).

U mikropočítače BBC Micro:bit se nejčastěji jedná o tvorbu samostatných zařízení (robotická ruka, sklad, dopravník, výtah apod.) [83]. V případě potřeby tvorby spolu komunikujících zařízení je možné využít Bluetooth rozhraní (programování v grafickém prostředí Microsoft MakeCode) nebo rádio signál (Python, nebo programování v grafickém prostředí Microsoft MakeCode) [74]. V případě Raspberry Pi se už pak často jedná o řídicí jednotky nasazované do provozu, jako Industry 4.0 Embedded Systems Network [84] (kde je kombinace Raspberry Pi jako serveru a Arduina), nebo např. řídicí jednotky pro robotickou ruku [85].

V případě řešení formou hotových modelů postavených na principu Industry 4.0 nebo modelů, které by po lehkých úpravách mohly prezentovat principy Industry 4.0, není na trhu velký výběr, ale příklady lze najít:

### FESTO CP Factory

CP Factory je univerzální modulární vývojová a vzdělávací platforma v oblasti Industry 4.0. Skládá se z modulů, které lze postupně přikupovat a libovolně propojovat, ukázka viz Obrázek 5-3. CP Factory využívá standardních PLC systémů, takže je možné napojení i jiného systému včetně komunikace mezi nimi. Všechny moduly mají standardizovanou velikost a rozměry většiny modulů jsou 1200 x 800 x 900 mm.[86]



Obrázek 5-3 – FESTO CP Factory [86]

### FESTO CP LAB

Jedná se o předem připravené sestavy modelů z CP Factory a samostatné moduly vhodné pro výrobu, jako např. moduly pro vrtání, ohýbání, lisování apod. Jako příklad lze uvést CP Lab 404-1 [87], která se skládá ze čtyř aplikačních modulů o rozměru 1800 x 1800 x 1800 mm, ukázka viz Obrázek 5-4. Obsažené moduly jsou v tomto případě:

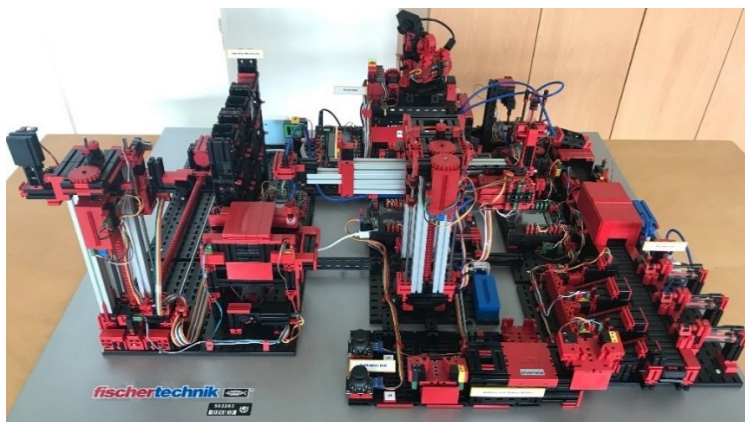
- Magazine application module, který umožňuje využívat např. RFID, slouží pro spouštění výrobního procesu
- Analog measuring application module, který umožňuje např. QS, SPC, analýzy
- Drilling application module, který umožňuje CPS, nastavení výrobních parametrů a variantnost
- Output application module, který umožňuje např. zpracování a vytěžení dat, flexibilní manipulaci, řešení logistiky



Obrázek 5-4 – CP Lab 404-1 [87]

### Fischertechnik Training Factory Industry 4.0 - 9V

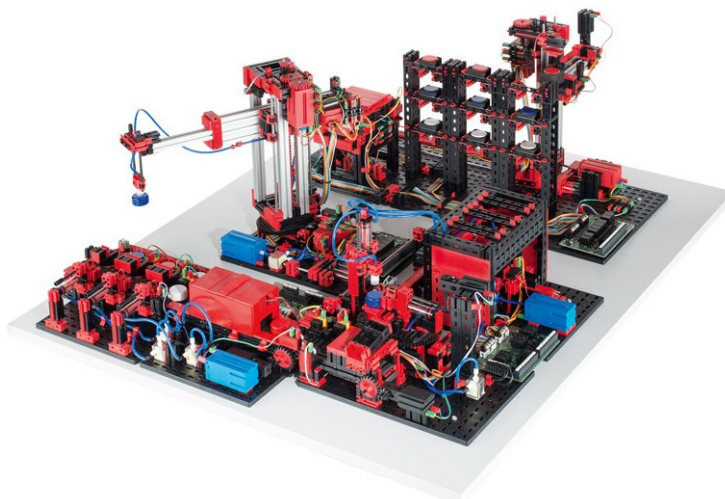
Model využívá napájení 9V, ovládání modelu je přes cloud společnosti Fischertechnik, který je možné customizovat. Model obsahuje 6 řídicích jednotek Fischertechnik TXT Controller, WI-FI router zajišťující vnitřní WI-FI síť a komunikaci s Cloudem přes WAN. Skládá se z robotického otočného ramene s vakuovým systémem pro uchopení/upuštění materiálu a NFC čtečkou, automatizovaného skladu, obráběcího zařízení, dopravníku, sortovacího zařízení pracujícího na principu detekce barvy a stanice senzorů prostředí s dálkově ovládatelnou kamerou, která zajišťuje online přenos obrazu z modelu do cloudu. Rozměry modelu jsou přibližně 9720 x 7720 x 4020 mm [88]. Ukázka viz Obrázek 5-5.



Obrázek 5-5 – Fischertechnik Training Factory Industry 4.0

### Fischertechnik Factory Simulation – 9V, nebo 24V

Model je z velké části podobný modelu Fischertechnik Training Factory Industry 4.0 a je ve variantě běžící na 9V [89] nebo 24V [90]. Model využívající 9V je potřeba osadit řídicími jednotkami, které jsou schopné řídit výkonové prvky běžící na 9V, např. řídicí jednotky Fischertechnik TXT Controller, kterými je osazený i model Fischertechnik Training Factory Industry 4.0. Model využívající 24V je potřeba osadit PLC systémem. Další podstatný rozdíl oproti verzi Fischertechnik Training Factory Industry 4.0 je chybějící příjem a výdej materiálu se čtečkou NFC, chybějící napojení na FischerTechnik Cloud a chybějící senzorová stanice včetně kamery. V případě varianty běžící na 24V je výhoda jednoduchého rozšíření jak o jakékoliv komponenty z PLC techniky, tak o nasazení a provozování vlastních PLC systémů a zapojení na jiné modely, nebo zařízení např. pro sběr nebo předávání informací. Ukázka viz Obrázek 5-6.



Obrázek 5-6 – Fischertechnik Factory Simulation 24V [90]

### **Merkur Toys – Programovatelná minilinka s MG**

V případě Merkur Toys se nejedná o zcela hotové, kompletní řešení. Jako vhodné se jeví využití modelů řady Education, popř. Robotics, a jako základ použít programovatelný model nazvaný „Programovatelná minilinka s MG“, jehož součástí je robotické rameno se šesti stupni volnosti a dopravník se senzory [91], viz Obrázek 5-7. Model se pak dá rozšířit dalšími modely z řad Merkur Toys, ovšem na jejich propojení a komunikaci a řízení musí uživatel nasadit nějaké vlastní, vhodné řešení.



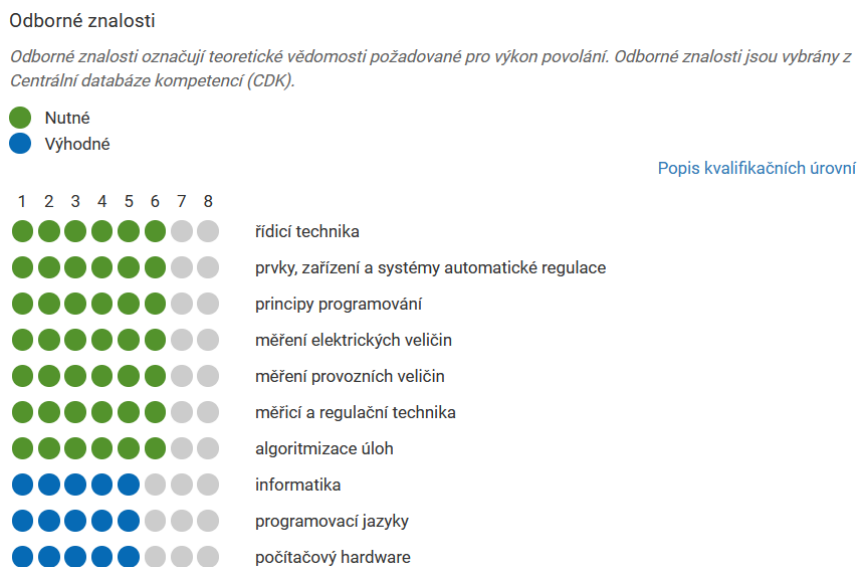
Obrázek 5-7 – Merkur programovatelná minilinka s MG [91]

## **5.2 Legislativní omezení pro průmyslového inženýra**

Při využívání modelů nebo stavebnic postavených na PLC naráží klasický průmyslový inženýr na legislativní omezení. Sice je možné, aby PLC programoval, ale pokud nedisponuje odbornou způsobilostí, neměl by PLC obvody zapojovat nebo upravovat. Zde je potřeba si uvědomit, že i když PLC obvody jsou často pouze na napětí 24 V, často se napojují na systémy s podstatně vyšším napětím, které řídí, regulují nebo s nimi jinak spolupracují. Toto omezovala do 14. 7. 2022 vyhláška č. 50/1978 Sb., o odborné způsobilosti v elektrotechnice § 5 – Pracovníci znalí nebo § 6 – Pracovníci pro samostatnou činnost [92]. Stávající osvědčení, která byla vydána podle vyhlášky 50/1978 Sb., budou podle přechodného ustanovení zákona platit po celou dobu, na kterou byla vydána. Od 1. 7. 2022 je to pak regulováno nařízením vlády 194/2022 Sb. [93], které slučuje vybrané paragrafy a upravuje

názvosloví. Pro získání této odborné způsobilosti je dle platné legislativy potřebné elektrotechnické vzdělání, minimálně vyučení (v případě § 6 a vyšších pak v kombinaci s praxí v délce podle typu el. vzdělání) [93]. Toto nesplňuje pracovník např. se strojírenským vzděláním ani průmyslový inženýr technického (strojírenského) typu.

I národní soustava povolání spravovaná Ministerstvem práce a sociálních věcí ČR ([www.nsp.cz](http://www.nsp.cz)) považuje povolání Programátor PLC za elektrotechnicky zaměřené než za informaticky zaměřené. Vzdělání v elektrotechnice označuje za nutné, zatímco v informatice, programovacích jazycích a počítačovém hardware pouze za výhodné [94]. Přehled odborných znalostí programátora PLC je uveden na následujícím obrázku, viz Obrázek 5-8.



Obrázek 5-8 – Odborné znalosti PLC programátora podle národní soustavy povolání [94]

Toto legislativní omezení tak brání klasickému průmyslovému inženýrovi vzdělávat ostatní v zapojování hardwarových modelů (nebo stavět vlastní HW modely např. pro ověřování nových procesů, nebo algoritmů), postavených i na 24 V využívajících PLC a omezuje jej pouze na programování hotových modelů. Tím se snižuje rozsah předávaných praktických znalostí potřebných pro pochopení principů konceptu Industry 4.0, který je na sběru dat ve výrobě, digitální komunikaci ve výrobě a robotizaci z velké části postavený.



## 6 Rešerše odborných článků vztahujících se k dané problematice

Pro rešerše byly využity primárně databáze Web of Science a Scopus. Vzhledem k tomu, že práce je zaměřena na koncept Industry 4.0, průmyslové inženýrství a využívání hardwarových i softwarových simulačních modelů a training factories v rámci průmyslového inženýrství, lze rešeršní část rozdělit do tří kategorií, které jsou následně popsány v samostatných kapitolách, a to:

- Plánování výroby pomocí simulačního modelu v kontextu Industry 4.0
- Přístupy k tvorbě a využití modelů a simulace v kontextu Industry 4.0
- HW modely a jejich využívání v rámci konceptu Industry 4.0

### 6.1 Plánování výroby pomocí simulačního modelu v kontextu Industry 4.0

Rozumně navržený a vytvořený model a nad ním následně běžící simulace s vhodnými parametry je využitelný jako nástroj pro výběr vhodného toku výrobků výrobním procesem. Jinými slovy lze využít model a nad ním běžící simulaci pro plánování výroby. S tím, jak se vyvíjí přístup k výrobě a přichází stále nové trendy ve výrobě, je potřeba adekvátně přizpůsobovat způsoby plánování. Aktuálně takovým novým přístupem, resp. paradigmatem, je koncept Industry 4.0, který se snaží přesunout jednoduché nebo stále se opakující činnosti od člověka ke stroji a zaměřuje se i na komunikaci jak mezi lidmi, lidmi a stroji, tak samotnými stroji. Jako vhodné využití simulací se ukazuje plánování výroby v podniku, který staví na konceptu Industry 4.0 a věnuje se převážně malosériové výrobě, kdy je potřeba pro nový produkt nebo novou sérii řešit přenastavení výrobních strojů, vhodnou zaměnitelnost (jak strojů, tak výrobků nebo vyr. procesů), řešit stavy, kdy dojde k souběhu výroby několika druhů výrobků, popř. jeden druh ve výrobě dobíhá a souběžně s tím se rozbíhá výroba nového. Vize je pak taková, že materiál může při výrobě procházet několika výrobními cestami, zaměnitelné stroje hlásí svůj stav (nastavení, využití atd.) a pak lze plánovat výrobu jak pro nový výrobek, tak teoreticky i pro každý jeden kus při vstupu do systému podle stavu na celé lince. S plánováním výroby má spojitost i prediktivní údržba, která do plánování může zanášet možné odstávky strojů apod.

Stálým trendem pro sběr dat (pro simulace) nebo jejich vizualizaci je využívání digitálních dvojčat, která se doplňují o moderní možnosti jako např. strojové učení. Tato digitální dvojčata jsou využívána pro vývoj inteligentních algoritmů pro autonomní řízení a další případy zahrnující optimalizaci, mentální modely a systémy podpory rozhodování [95]. Digitální dvojče umožňuje pro výrobní proces propojení produkčního systému s jeho digitálním ekvivalentem. Aby byla zajištěna maximální shoda kybernetického procesu s jeho skutečným modelem, musí být dle autorů provedeno multimodální získávání a hodnocení dat [52]. Jaensch a kol. [96] představují možnost rozšíření digitálních dvojčat o možnost strojového učení modelů založených na datech až po učení logiky řízení složitých systémů. Ukazují, jak používat strojové učení ve spojení s těmito modely, aby se dosáhlo kratší doby vývoje výrobních systémů.

V rámci konceptu Industry 4.0 se začalo rozvíjet i téma prediktivní údržby, které má velký dopad na plánování a umožňuje plánovat výrobu s ohledem na prostoje, jež budou způsobeny údržbou, poruchami apod. Tento přístup sleduje cíl eliminovat selhání systému dříve, než k němu dojde. Tím se zabrání stavům, kdy zařízení vstoupí do neočekávaných prostojů, a ušetří se vysoké náklady. Jedna z možností je využití generické prediktivní simulace údržby např. v nástroji AnyLogic. Koncept obsahuje popis funkce a cíle simulace, rozhraní pro přenos dat do/ze simulace, obecnou strukturu simulace a ukazatele používané k vyhodnocení simulovaných plánů výroby. První případová studie ukazuje, jak lze simulační model použít k simulaci vzorových dat z testbedu pro zlepšení výrobních plánů [97]. Dle Teufela a kol. [98] dnes není neobvyklé mít prediktivní model továrny, i když tyto modely jsou často zjednodušené povahy. Takové modely však jen zřídka odrážejí aktuální provozní výkonnost systému, používají jednoduché a oddělené datové proudy a nemají rozlišení stroje/pracovní stanice. Mohou podporovat plánování výroby, ale obvykle mají omezené využití pro optimalizaci výkonu továrny v

reakci na měnící se vnější podněty. Proto výzkumný projekt OPTIMIZED vyvíjí holistickou platformu pro řízení závodu, který reaguje na neočekávané poruchy v továrně. Tím představují aplikaci optimalizace založené na simulaci na podporu plánování výroby výrobního procesu jednoho z průmyslových demonstrantů. Simulační model zachycuje příslušná výrobní omezení továrny až na každý stroj a pracovní stanici. Simulační model čte data z podnikových informačních systémů, živá data ze strojů a data z dovybavených senzorů ve výrobním závodě. Výsledkem je, že dochází k optimalizaci úrovně služeb i výrobních nákladů.

Podle Fantiniho a spol. [99] s ohledem na rozrůstající se cyber-physical systémy existuje velká shoda v tom, že se práce změní a budou nutné různé dovednosti. Otázkou však zůstává, zda se organizace práce bude vyvíjet směrem k vyšší odpovědnosti a rozhodování zaměstnanců nebo k vyšší technologické kontrole. Výzvou je tento vývoj řídit a cíleně vést proces integrace lidí do CPS. Autoři navrhují metodologii na podporu navrhování a hodnocení různých pracovních konfigurací a zvažují jedinečnost lidské práce a charakteristiky kybernetické výroby v komplexním rámci. Metoda zahrnuje běžnou výrobu i nepravidelné scénáře, jako je detekce poruch nebo zásah údržby, zvláště zajímavé pro lidskou práci. Použitelnost této metody je ilustrována dvěma průmyslovými případy, které vedou k návrhům na školení personálu a ke zlepšení celého kyberneticko-fyzikálně-sociálního systému. Z kontextu tedy vyplývá, že se na simulaci lze dívat jako na podpůrný nástroj pro rozhodování z pohledu osoby mající vliv na to, co bude dělat stroj, a co člověk.

Simulace, popř. digitální dvojče se dnes běžně využívá jako podklad pro rozhodování. Autoři B. Denkena a F. Winter [100] využívají simulaci výroby rotorových listů v odvětví větrných turbín jako podklad pro rozhodování s cílem snížení doby plánování a nákladů a aby se odhadl vzájemný dopad na výrobní kapacitu podle produktu, technologie a továrny v rámci plánovací fáze. Autoři W. Yang a S. Takakuwa [101] jdou ještě dále a zaměřují se na plánování výroby s ohledem na možnou probíhající komunikaci mezi výrobcem a zákazníkem. Výrobce je díky implementaci prvků Industry 4.0 schopen rychleji reagovat na požadavky zákazníků, což znamená, že výrobci musí nyní pevněji kontrolovat plánování výroby. Do simulačního modelu se importuje objednávka zákazníka a tabulka zpracování produktů. Experimenty jsou prováděny pro případy, kdy systém narazí na neočekávané podmínky. Navrhovaný přístup představuje pro výrobce potenciální nástroj pro rozhodování v reálném čase. C. L. Constantinescu a kol. [102] považují za jednu z hlavních výzev Industry 4.0 zajistit, aby správní lidé měli správné informace ve správný čas, aby mohli učinit správná rozhodnutí. Proto přichází s víceúrovňovým simulačním nástrojem just-in-time pro podporu rozhodování v plánování digitálního závodu. Ten využívá vhodné prostředky pro zachycení a reprezentaci znalostí výrobního systému v několika typech průmyslových odvětví a typů společností (velké, malé a střední podniky). Pro podporu modelovací činnosti, rozhraní člověk-systém simulačního nástroje využívá vyhledávání informací v reálném čase (JITIR) pro proaktivní poskytování požadované informace ve správný čas na základě kontextu během modelování a simulace.

Možností, jak využít simulaci a vizualizaci pro rychlé ověření plánů výrobních toků, je metoda generování poloautomatických modelů řízených daty [103], která je ilustrována na příkladu implementace pomocí simulačního softwaru FlexSim. A. Caggiano a R. Teti [104] využívají nástrojů digitálního modelování a simulace pro krátkodobou analýzu a validaci strategií řízení výroby, popř. pro střednědobé plánování výroby nebo návrh/redesign výrobního systému. Navrhují metodiku Digital Factory na podporu rozšíření stávající výrobní buňky pro výrobu lopatek turbín leteckých motorů pomocí robotické automatizace její odhrotovací stanice. K ověření upgradované výrobní buňky s cílem zvýšit její výkonnost z hlediska využití zdrojů a doby průchodu je využito 3D modelování a diskrétní simulace. Validace simulačního modelu se provádí sbíráním skutečných dat z fyzicky rekonfigurované výrobní buňky a jejich porovnáním s prognózou modelu.

V současné době chybí dle G. Avventurosa a kolektivu [105] znalosti a strukturované přístupy týkající se přechodu výrobních systémů z rychlého prototypování na velkovýrobu. Představují tedy postup pro simulační analýzu FMS podporující plánování, návrh a hodnocení výkonu. Vyvinuli a použili

diskrétní simulační model pro provozní vyhodnocení průmyslového případu, který zahrnuje 3D tisk, automatizovaný transport a skladovací systémy.

Dle autorů P. Goodalla, R. Sharpa, a A. Westa [106] mohou ze simulací těžit obzvláště repasovací operace jako podpora posuzování různých strategií pro scénáře v reálném čase kvůli nejisté povaze návratnosti produktu. Navrhují simulaci založenou na datech, která se skládá ze tří prvků:

1. adaptivní algoritmus remanufacturing simulace pro modelování komplexního materiálového toku nalezeného v procesu repase obecným a opakovaně použitelným způsobem,
2. informační model pro strukturování a zvýraznění požadavků na simulační data,
3. vrstvu informačních služeb pro shromažďování a analýzu dat senzorů pro použití v simulaci.

Simulace je implementována za účelem demonstrace toho, jak lze automaticky překonfigurovat a přizpůsobit se změnám v datových vstupech (procesní a tovární modely). Případová studie je provedena u „výrobce“ elektroniky, který je zaměřený na repasování.

## 6.2 Přístup k tvorbě a využití modelů a simulace v kontextu Industry 4.0

Standardní přístup ve vytváření modelů a následných simulací je vytvoření modelu bez vrstvy, která by zajišťovala komunikaci mezi jednotlivými stroji, popř. stroji a pracovníky na různých pracovních pozicích i různých hierarchických stupních. V tuto chvíli se v případě potřeby pokrytí rozhodování využívá standardních „if“ podmínek na daném vstupu/výstupu, popř. u přepravního systému (dopravník, vysokozdvizný vozík, roznos apod.). Otázkou tedy například může být, zda tok materiálu/výrobku by mohl být vyhodnocovaný ne pouze podle následujícího pracoviště/úkonu, ale podle komplexního pohledu na stav výrobního úseku, haly, popř. celé továrny, včetně odloučených pracovišť nebo závodů. V článku „Industry 4.0 and the New Simulation Modelling Paradigm“ [107] představují autoři vliv konceptu Industry 4.0 na vývoj nového paradigmatu simulace, ztělesněného konceptem Digital Twin. Hlavní poučení je, že zatímco nové paradigma simulačního modelování je přijímáno velkými společnostmi a malými a středními podniky, existují značné rozdíly v závislosti na velikosti společnosti v problémech, jimž čelí, a používaných metodologiích a technologiích. Zatímco zkoumané případy naznačují přijetí nového modelu simulačního modelování v průmyslových a vědeckých komunitách, jeho přijetí v akademickém prostředí vyžaduje úzkou spolupráci s průmyslovými partnery a diverzifikaci znalostí výzkumných pracovníků, aby bylo možné vybudovat integrované, víceúrovňové modely kybernetických systémů. Jak ukazují předložené případy, nedostatek nástrojů není problém, protože současná generace nástrojů pro modelování pomocí diskretních simulací pro všeobecné použití nabízí adekvátní možnosti integrace.

Častým požadavkem při tvorbě simulačních modelů je možnost využívat je jako vzdělávací platformu, a to nejčastěji pro vzdělávání v simulacích, komunikaci, konceptu Industry 4.0 apod. Dle Grubeho a kol. [108] je simulace a digitální dvojče cestou, jak v malých a středních podnicích využít jejich potenciál. Přičemž autoři kladou důraz na propojení virtuálních a fyzických artefaktů. To následně inspiruje podniky ke změnám. Na nedostatek znalostí konceptu Industry 4.0 a možností simulací schopných automatizovaného sběru dat upozorňují i Uhlemann a kol. [109], kteří na digitálním dvojčeti s využitím sběru dat v reálném čase demonstrují výhody těchto řešení pro SME a považují jej za vhodnou vzdělávací platformu pro tuto velikost organizací. Na neznalost podniků upozorňují i D. Mourtiz a kol. [110], kteří se zaměřují na snahu propojit akademickou sféru ovládající simulace se soukromým sektorem, pro které jsou simulace užitečným, ale poměrně neznámým nástrojem. Autoři teaching factory považují za chybějící článek, jehož cílem je překlenout propast mezi akademickou obcí a průmyslem. Proto ukazují návrh a vyhodnocení skutečného výrobního systému pomocí diskretní simulace a na základě reálných dat získaných z metalurgického průmyslu prezentovaného v kontextu paradigmatu teaching factory. Naopak S. Lang a kol. [111] představují koncept a současný stav vývoje modulární, decentralizované a digitální továrny na učení pro využití studenty a výzkumnými pracovníky. Kromě prezentace průmyslových modulů a jejich nastavení představují řídicí a

komunikační strategii, která umožňuje vytvoření a provoz libovolného preferovaného rozvržení v podniku. Dále stručně popisují model diskrétní simulace a model emulace pro virtuální uvedení do provozu. Teaching factory je postavena na kombinaci stavebnice FischerTechnik (vytvoření otočného stojanu pod robotickou ruku a dopravník), Arduino a stavebnice samostatné robotické ruky, ukázané na následujícím obrázku, viz Obrázek 6-1. Tato stavebnice robotické ruky pro Arduino se dá k 6. 6. 2023 pořídit např. na [www.rpishop.cz](http://www.rpishop.cz) za 7 089Kč pod názvem TinkerKit Braccio robot.



Obrázek 6-1 – Stavebnice robotické ruky použitá S. Langem a kol. [111]

Dalším zajímavým trendem se ukazuje využití simulace a popř. digitálních dvojčat pro nastavení spolupráce výrobních podniků a využívání kooperací. B. Kádár a kol. [112] navrhuje využívat simulaci s cílem řídit síť výrobců, kteří mohou dynamicky překonfigurovat a sdílet své zdroje v rámci předregistrované komunity a kteří mohou analyzovat, jak kardinalita takového „společenství“ ovlivňuje jak spolupráci, tak globální a místní produkční KPI. Naopak T. Kaihara a kol. [113] představují Crowdsourced production, nový typ výroby, ve kterém společnosti sdílejí své výrobní zdroje v závislosti na jejich poptávce a kapacitě. Podle toho společnosti sdílejí své informace o zdrojích a mohou najít outsourcingovou společnost, když potřebují konkrétní/specifické zdroje. Autoři vyvinuli simulační model výroby s crowdsourcetem s modelem zdroje a vyjednávacím algoritmem založeným na agentech pro vyhodnocení efektivity výroby na základě dodávky a využití stroje. Podle kombinace a vyváženosti obchodního stylu se míra doručování mění s bodem nasycení, který závisí na shodnosti zdrojů. A. K. Türker a kol. [114] pak využívají sběr dat v reálném čase a simulace k rozhodování, zda při výrobě v budoucnu dojde, nebo nedojde ke zpoždění, a navržený systém na základě toho rozhoduje o outsourcingu výroby zakázky.

Využití simulace je i možnost nejen pro predikce potřebných oprav, popř. prostožů, ale i reorganizaci a realizaci jiného plánu výroby než standardního, např. při nenadálé poruše [115]. V takovém případě je problém předvídat možné poruchy, a hlavně mít připravené scénáře pro chování systému. Proto autoři článku navrhuje metodiku pro získání znalostí o údržbě pomocí metody počítačové simulace. Předložený příklad vytvořili pomocí softwaru Tecnomatix Plant Simulation. Obdobnému tématu se věnuje i H.-J. Shin a kol. [116], kteří studují aktivní ochranu pro umožnění funkce „sebeopravy“. Autoři navrhli modelování dílny a dynamické rekonfigurovatelné schéma CPS, které může předpovídat výskyt anomálií v modelu a implementuje sebeochrany. Za tímto účelem byla využita technologie strojového učení umožňující omezit přetížení ve výrobním procesu. Výsledkem je navržený simulační model autonomně detekující abnormální situace a je dynamicky rekonfigurovatelný pomocí sebeoprav.

V člancích zabývajících se simulací se pak ukazuje i další směr, kterým se simulace vydávají – propojování. Nejde jen o propojování s hardwarem, který např. umožňuje sběr dat v reálném čase, ale

i o propojení softwaru pro diskretní simulace se softwarem pro simulaci dynamiky [117], kde autoři představují integrační mechanismus pro online komunikaci mezi těmito nástroji, a navíc ji využívají pro ověření logiky naprogramovaného PLC. Jiným příkladem je propojení diskretní simulace a virtuální reality (VR DES) [118]. V tomto případě je jako VR DES označována nová platforma pro testování scénářů a rozhodování. V práci se řeší potřebné komunikační protokoly, aspekty návrhu systému, validace modelu a aplikace VR a DES. Dle autorů je potřeba další výzkum v oblastech zpracování obrazu s nižší latencí, dodávky DES jako služby, rozpoznávání gest pro VR Interakce DES a propojení DES s datovými toky v reálném čase a soubory velkých dat. Delbrügger a kol. [119] se pak zaměřují na propojování simulačních systémů. Podle autorů nelze simulační modely snadno kombinovat, a proto často není možné dosáhnout interdisciplinárních globálních optimalizací. Navíc aspekt lidské interakce s tak komplexními ko-simulačními systémy je často opomíjen. Proto představují koncept pro kombinaci různých simulačních modelů s interdisciplinárními víceúrovňovými simulacemi produkčních systémů. Koncept zahrnuje tři hierarchické úrovně produkčních systémů (Process Simulation, Factory Simulation a Human Interaction) a umožňuje interakci člověka se simulačním systémem. Koncept lze ale snadno rozšířit, aby podporoval další úrovně. V rámci víceúrovňové struktury provádí každý simulační systém vícecílou optimalizaci. Paretooptimální řešení jsou předávána simulacím na vyšších hierarchických úrovních, aby je mohla kombinovat a splnit flexibilně přizpůsobitelné cíle celého výrobního systému. Koncept je testován pomocí zjednodušeného výrobního systému, aby byl optimalizován z hlediska doby výkonu a spotřeby elektrické energie. Výsledky ukazují na potenciál takto optimalizovat produktivitu a efektivitu výrobních systémů.

Y. Lu a kolektiv [53] se zabývají nedávným vývojem technologií Digital Twin ve výrobních systémech a procesech, analyzují konotace, scénáře aplikací a výzkumné otázky inteligentní výroby digitálním dvojčtem řízené výroby v kontextu Industry 4.0. Ve své práci identifikují nevyřešené výzkumné problémy vývoje digitálních dvojčat pro inteligentní výrobu. Jsou to:

1. Standardizovat vzor architektury pro digitální dvojče ideální kombinací dvou aktuálně používaných architektur.
2. Zaměřit se na snížení latence komunikace pro digitální dvojčata.
3. Standardizovat mechanismus sběru dat (definovat best practices).
4. Definovat standardy pro digitální dvojčata.
5. Standardizovat/navrhnout výchozí funkce digitálního dvojčete.
6. Zaměřit se na správu verzí modelu dvojčete.
7. Zaměřit se na „simulování“ lidí v aplikacích Digital Twin.

S. Kern a J. Scholz [120] se zabývají přístupem k simulaci založeným na modelování agentů pro vnitřní výrobní prostory. Simulační metody používají k optimalizaci prostorové dimenze s ohledem na simulaci výrobních procesů. Model založený na agentech vyvinutý v tomto článku je využíván k simulaci výrobního prostředí pomocí umělého vnitřního prostoru a sady testovacích dat. Výsledky ukazují, že různé úrovně dostupných prostorových informací mají vliv na výsledky simulace vnitřních výrobních prostředí a procesů. Vzdálenosti překonávané pracovními agenty mohou být významně sníženy stejně jako jejich neproduktivní pohyby (bez výrobních aktiv).

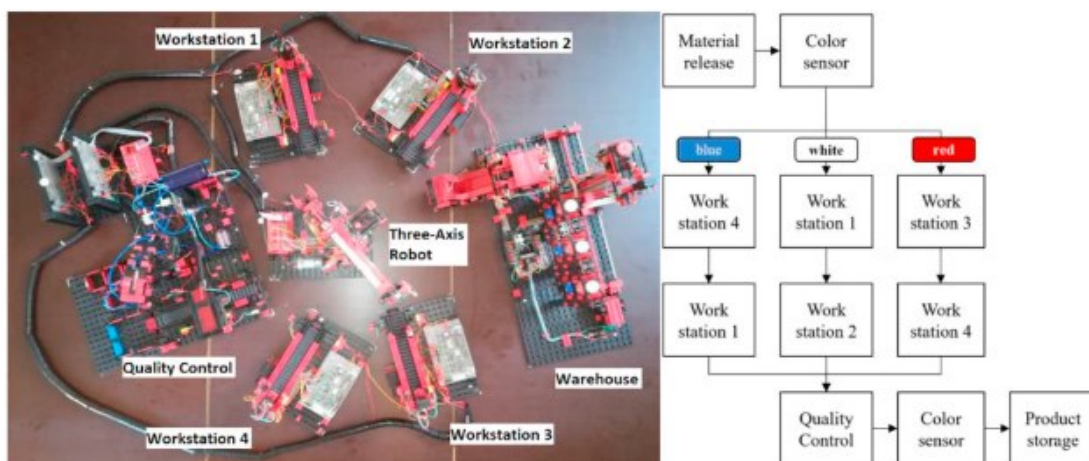
### 6.3 HW modely a jejich využívání v rámci konceptu Industry 4.0

Pechmann a kol. na University of Applied Sciences Embedn/Leer<sup>1</sup>, fakultě technologie, a Institutu for Industrial Informatics, Automation and Robotics vytvořili s využitím stavebnice Fischertechnik kyber-fyzikální systém (nejedná se o zakoupený kompletní model) skládající se ze 4 pracovních stanic, skladu, tříosého robota, dopravníků a stanice pro kontrolu kvality s vizí využití ve vzdělávání, ukázka je

---

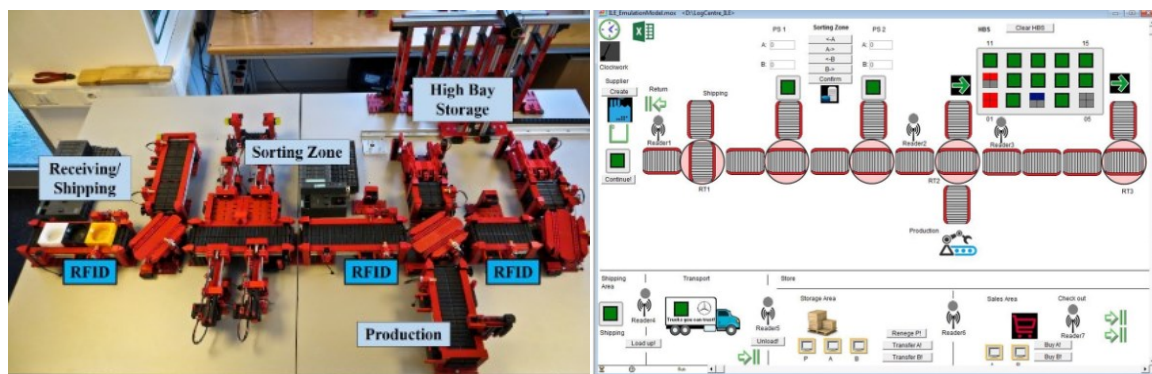
<sup>1</sup> Univerzita i fakulta zaměřena na informatiku, elektrotechniku. Tento kontext je důležitý pro tyto teze k disertační práci.

na následujícím obrázku, viz Obrázek 6-2. Tento systém „vyrábí“ 3 výrobky, které jsou rozlišeny barvou. Postup využití je takový, že model je představený jako klasická výroba bez jakýchkoliv Industry 4.0 prvků, rozdělí studenty do skupin po max. třech a každá skupinka má za úkol implementovat prvky Industry 4.0 do jim svěřené části modelu. Studenti napojují model na ERP systém, pracují s různými databázemi (MySQL, Oracle) a způsoby komunikace (např. MQTT). Zároveň je model využíván při bakalářských a diplomových pracích, kdy student zpracoval napojení modelu na software AnyLogic pro diskrétní simulace. Není řečeno, jakým způsobem je zpracováno řízení modelu a práce s ním, ale z kontextu vyplývá, že se využívají buď napřímou pracovní stanice, nebo PLC. Autoři vyzdvihují jako přínos využití modelu ve výuce v předání ucelených znalostí a jejich mezioborovým propojení, kdy se kromě rozšíření znalostí z oblasti informatiky seznámí i s principem různých el. prvků, jejich využitím, programováním apod.



Obrázek 6-2 – Model postavený na platformě Fischertechnik pro využití ve výuce Pechmannem a kol. [121]

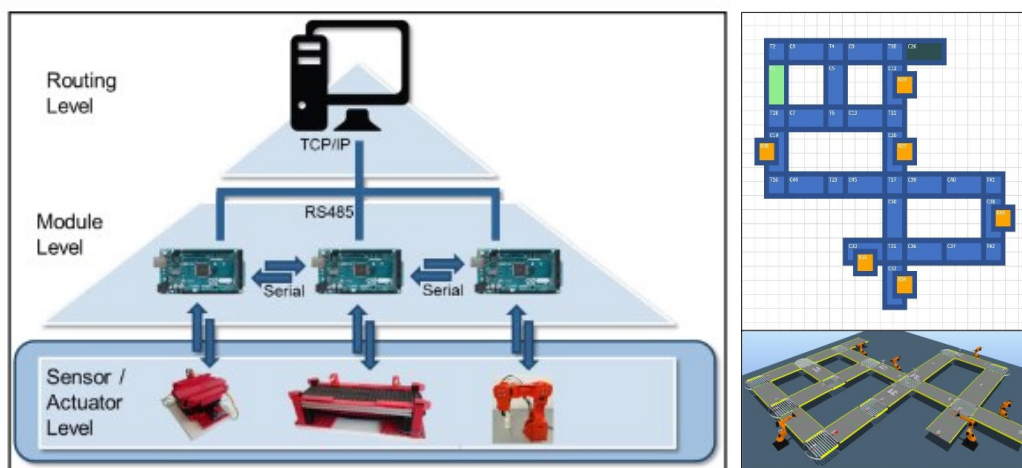
Hofmann a kol. [122] využívají vlastní learning factory, kde HW model je postavený na platformě Fischertechnik a k němu je vytvořena vlastní emulace sloužící k rozhodování. Tuto learning factory využívají autoři ve výuce logistiky. Výuka s využitím learning factory je pak zaměřena na řešení problémů. Studenti jsou konfrontováni s existujícím problematickým procesem nebo procesem, který je potřeba nově nasadit, a jejich úkolem je se s daným problémem vypořádat. Zadáání může např. znít: Navrhněte a implementujte proces inventury pro High Bay Storage. Autoři odkazují na Freemana a kol., kteří říkají, že praktická práce podstatně zvyšuje šance vysokoškoláků na pochopení problematiky a úspěšné absolvování předmětu [123]. Zároveň také říkají, že nejlepším učitelem je neúspěšný experiment.



Obrázek 6-3 – HW model a uživatelské rozhraní emulace modelu dle Hofmanna a kol. [122]

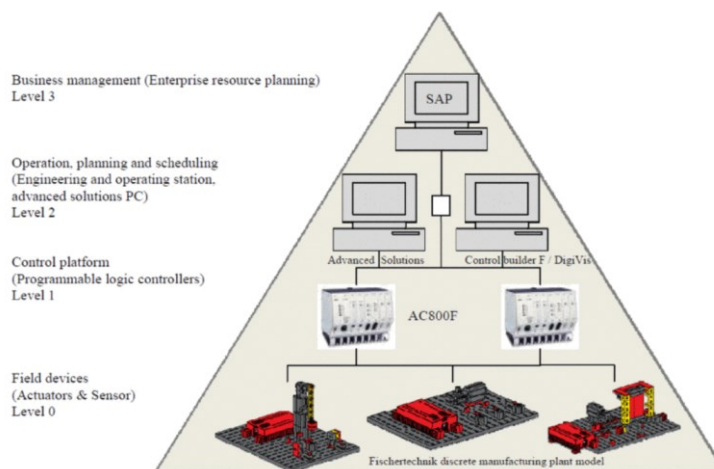
Lang a kol. [111] pracují na modulární a decentralizované Industry 4.0 learning factory pro vzdělávání studentů v rámci konceptu Industry 4.0 a na vývoji a evaluaci nových konceptů a technologií

pro využití v Industry 4.0. V rámci modelu je řešeno i jeho řízení a komunikace jak s modelem, tak v rámci modelu. Aktuálně jej využívají např. pro ověřování routovací strategie/algorithmu, na kterém aktuálně pracují. Samotný model je postavený na stavebnici Fischertechnik (10 dopravníků, 5 otočných stolů, 8 vytlačovačů a dalších „modulů“), 3 robotické ruce TinerKit Braccio robot a arduino MEGA 2560 pro každý výrobní modul jako řídicí jednotky. Arduina byla zvolena z důvodu malých rozměrů řídicích jednotek, rychlého vývoje a poměrně jednoduché integrace, zároveň dle autorů zajišťují do budoucna jednoduchou rozšiřitelnost learning factory takovým způsobem, aby bylo možné integrovat budoucí požadavky a nápady a rozšiřovat ji. Pro návrh a grafickou tvorbu byla vyvinuta VBA aplikace pro MS Excel, ve kterém se navrhne linka a pomocí VBA aplikace se podle návrhu v Excelu vytvoří 3D model v nástroji Emulate3D. Learning factory ještě není hotova, posledním přidaným prvkem je napojení na nástroj pro diskretní simulace, v tomto případě Siemens Plant Simulation 14, všeobecně pro jakýkoliv přenos dat autoři využívají souborový formát AutomationML.



Obrázek 6-4 – Architektura learning factory, návrh layoutu v MS Excel a vygenerovaný 3D model dle Langa a kol. [111]

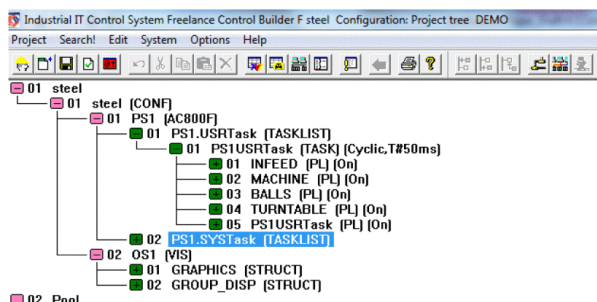
Phillips a kol. [124] pro výzkumné práce a edukaci propojili hardwarový model diskretní simulace (model postavený na platformě Fischertechnik, dle autorů úroveň 0) řízený PLC ABB AC800F (dle autorů úroveň 1), přičemž cílem je možnost řídit výrobu ze systému na vyšší úrovni (např. SAP) nebo pro tuto úroveň alespoň sbírat data. Sami autoři rozlišují celkem 4 úrovně, viz Obrázek 6-5, přičemž práce řeší první 3 úrovně.



Obrázek 6-5 – Pyramida automatizace L0 – L3 dle Phillipse a kol. [124]

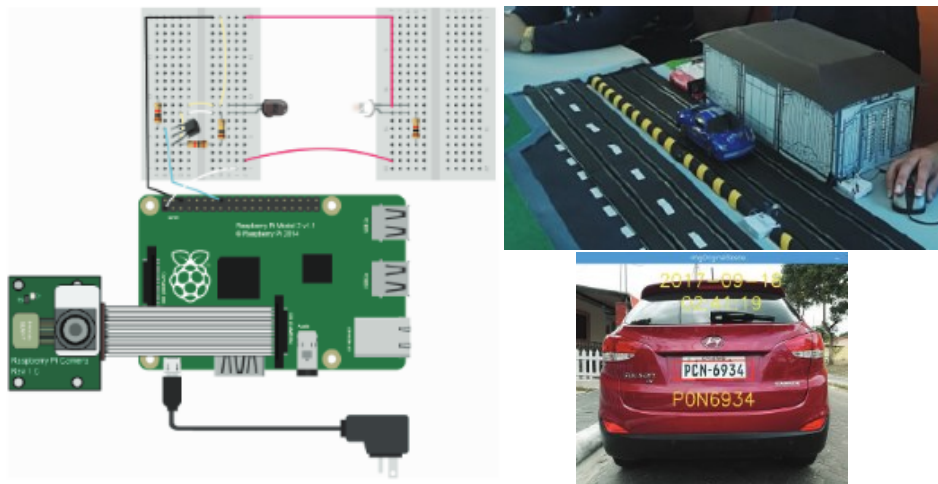
Pro úroveň 3 byl zvolen nástroj ABB Freelance 800F jako integrované prostředí pro správu a řízení hw modelu včetně sběru dat z něj. Vytvořené kyber-fyzikální prostředí slouží k seznámení studentů

s automatizací a u odborníků k rozšíření jejich znalostí v bezpečném laboratorním prostředí. Ukázka stromu projektu v nástroji ABB Freelance je na obrázku, viz Obrázek 6-6. Do budoucna autoři uvažují o rozšíření o napojení na databázi, napojení na 3. úroveň (na ERP systém, např. SAP) a implementaci plánovacího a optimalizačního řešení. Při využívání navrženého prostředí ve vzdělávání se ukázalo, že je pro studenty velmi obtížné sekvenování, tedy stanovení pořadí událostí v modelové továrně a rozšiřování modelu o novou stanici/stanice, které pro ně zesložituje nejen celý řídicí kód, ale zasahuje i do stanovených sekvencí, jež dělá komplikovanějšími a zasahuje do jejich stávajícího pořadí. Learning factory je a nadále bude využívána studenty pro návrh a evaluaci různých layoutů výrobních podniků. Pro výzkumníky slouží jako prostředí pro vývoj a vyhodnocování nových konceptů a používání nových technologií vztažených ke konceptu Industry 4.0.



Obrázek 6-6 – Ukázka stromu projektu v ABB Freelance [124]

Padilla a kol. [125] představují případovou studii, kdy pro zlepšení průjezdnosti v rámci provozu využívají neurálních sítí a rozeznávání obrázků. Prototyp postavili na jazyce Python a mikropočítači RaspberryPi 3 vybavený kamerou pro sběr obrázků a branou spouštějící kameru využívající infra LED. Pro verifikaci využívali vlastní model. Při testování rozeznávání značek skutečných aut pak dosáhli přesnosti rozeznávaných aut dle značek ze 78,04 %. Pro vyšší přesnost by dle autorů byla potřeba už lepší kamera a normalizace obrázků (výsledek je hodně ovlivněn např. úhlem, pod kterým je značka snímána). Pro skutečné nasazení by dále bylo potřeba řešit implementaci GSM modulů. Ukázka je na následujícím obrázku, viz Obrázek 6-7.



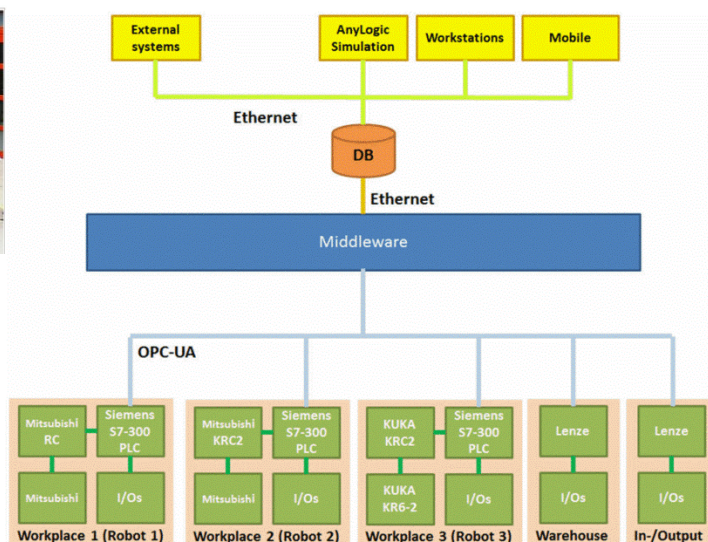
Obrázek 6-7 – Schéma zapojení, verifikační model a ukázka z venkovního testu Padillou a kol. [125]

Zarte a kol. [126] ve své studii představují vybudovanou Industry 4.0 laboratoř pro demonstrování komunikace mezi dílnou a různými úrovněmi IT v rámci organizace. Laboratoř je využívána v rámci bakalářského a magisterského studia oborů elektrotechnika, informatika, počítačové vědy a průmyslové informatika a skládá se z několika částí:



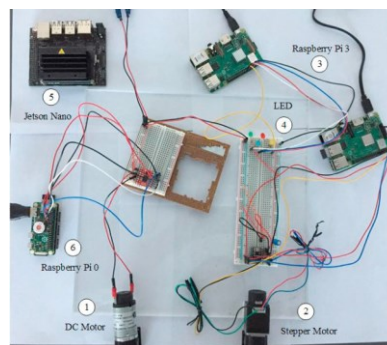
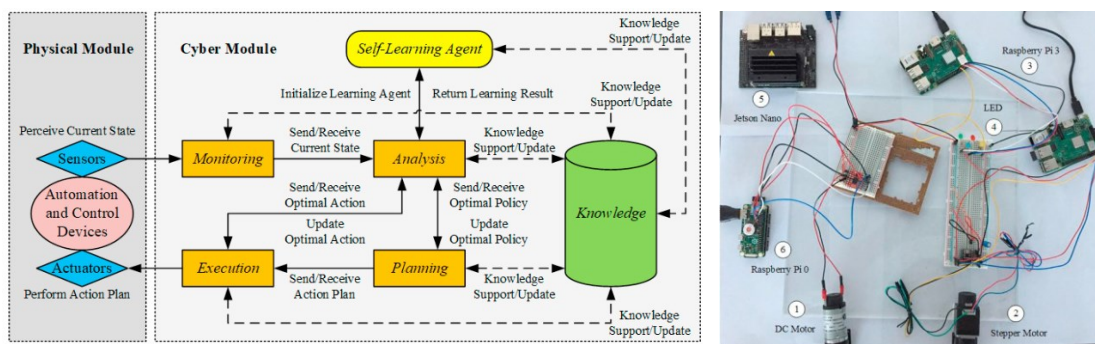
- HW model (dílnu) skládající se ze skladu, 3 pracovních stanic, dopravníku a vstupně/výstupního modulu. Vše je řízeno PLC jednotkami SIEMENS S300 PLC, pracovní stanice jsou vybaveny roboty KUKA nebo Mitsubishi. Produkt představují sestavy z LEGO Duplo obohacenými o RFID.
- ERP systém, ze kterého využívají moduly CRM, SRM, e-PPS a Warehouse (pro správu skladu).
- Databáze Oracle, která je součástí ERP systému
- Nástroj AnyLogic pro simulace a prediktivní údržbu

V pojetí autorů představuje samotná dílna, resp. HW model, úroveň 1, na úrovni 2 jsou PLC jednotky, úroveň 3 je simulační software pro plánování a optimalizaci a úroveň 4 je ERP systém. Od této learning factory je očekáván přímý dopad na studenty a způsob jejich vzdělávání. Studenti byli zapojeni přímo do tvorby laboratoře a stejně tak se plánuje jejich využití i při jejím rozšiřování. Laboratoř se využívá jak k demonstracím, tak závěrečným pracím studentů a ověřování nových metodik, technik, postupů, optimalizací apod. Ukázka je na následujícím obrázku, viz Obrázek 6-8.



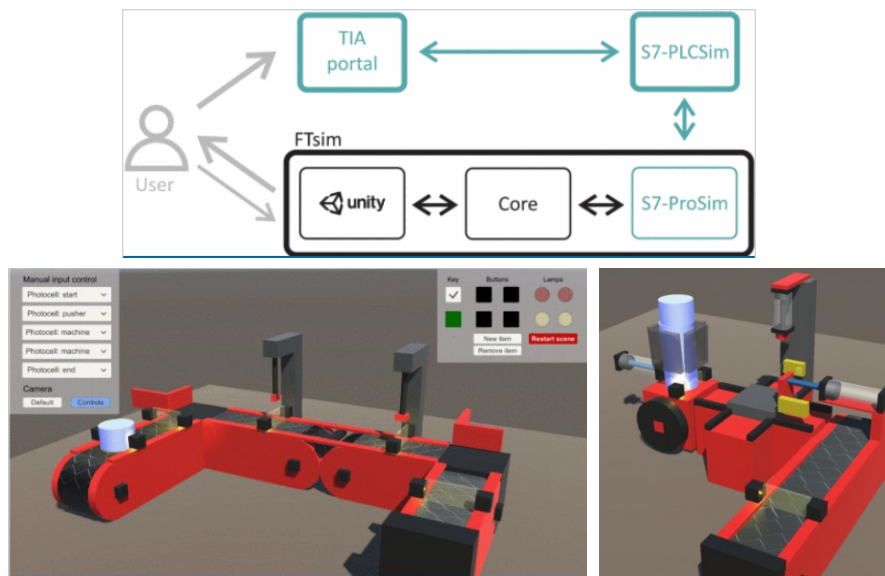
Obrázek 6-8 – Ukázka laboratoře Industry 4.0 a logické architektury demonstrátoru Industry 4.0 představená Zartem a kol. [126]

Lyu a kol. [127] vyvíjejí dvourvrstvou architekturu pro modelování kyber-fyzikálních systémů, kterou cílí na integraci umělé inteligence a real-time adaptace systému. Využívají k tomu počítač Jetson Nano, který zajišťuje analyzování, plánování, znalostní agenty a samoučení a počítače RaspberryPi 3 a RaspberryPi Zero, jež zajišťují monitorování a výkonné agenty (v tomto případě řízení motorů). Ukázka architektury a modelu je na následujícím obrázku, viz Obrázek 6-9.



Obrázek 6-9 – Dvourvrstvá architektura pro CPS a fyzický model dle Lyu a kol. [127]

Ilc a Lotrič [128] vytvořili simulátor tréninkového modelu nazvaný FTsim, který je 3D virtualizací tréninkového modelu využívajícího platformu Fischertechnik řízenou PLC SIEMENS S7, využívající Fischertechnik modely (model výrobní stanice s dopravníkem, model pneumatického procesního centra a model 3D uchopovacího robota). Model v FTsim se pak může chovat jako digitální dvojče ke skutečné výrobě představené fyzickým modelem postaveným na platformě Fischertechnik, limitujícím faktorem je pouze omezení na řízení modelu pomocí výše zmíněného PLC. FTsim je využíván pro výuku studentů oboru Computer Science a slouží k seznámení studentů s programováním PLC a s jejich interakcí se systémy na vyšší úrovni. Díky využití unity autoři plánují rozšíření o napojení na VR. Architektura vytvořeného nástroje FTsim a ukázka modelu je na následujícím obrázku, viz Obrázek 6-10.



Obrázek 6-10 – Architektura FTsim a ukázka modeluv FTsim [128]

Faller a Feldmüller [129] vytvořili learning factory se zaměřením na využití pro regionální SME podniky v okolí své univerzity, pro které je dle autorů velmi složité implementovat prvky Industry 4.0 hlavně kvůli nedostatku znalostí a strachu z tohoto nového konceptu. Learning factory propojuje ERP systém (SAP) s MES a systémem pro monitorování spotřeby energie. Tyto systémy čerpají data z robota Mitsubishi (přes PLC) a CNC stroje SIEMENS 840D (napřímo, bez PLC).

Navržená learning factory cílí na základní (!) ukázkou a vzdělání ve 3 směrech, jsou to 3 samostatná, na sobě nezávislá (a nepodmiňující) školení:

- Technická a organizační integrace managementu a výroby
- Automatizace skrz moderní komunikační technologie (komunikace s PLC, Robot Control, CNC)
- Zvýšení energetické účinnosti ve výrobních procesech

Kromě těchto školení je využívána learning factory pro výuku studentů a pro vyhodnocování výzkumných projektů. Autoři zmiňují, že u takového řešení je potřeba vyvinout velké úsilí k dosažení rozumně spolehlivé integrace a že do budoucna bude potřeba vytvořenou learning factory rozšířit.

## 6.4 Shrnutí

Jak ve výzkumných ústavech, na univerzitách i v podnicích se využívají learning factories (někde označované jako training factories, jedná se o synonymum). Aktuálně je trendem rozšiřovat tyto fungující modely o digitální dvojčata a využívat je k ověřování nových algoritmů např. z problematiky plánování, k prezentaci konceptu Industry 4.0, vzdělávání studentů a pracovníků v konceptu Industry

4.0 a v technologiích, které tento koncept představuje, a současně i k ověřování těchto technologií. Z konkrétních technologií se často jedná právě o digitální dvojče, využívání simulací, nové typy aktivních prvků pro stroje a linky, implementaci ML v prediktivní údržbě, AI apod.

Často je zmiňovaný problém neznalosti simulací a digitálních dvojčat ve SME podnicích, kde by se často velmi rychle ukázaly výhody implementace těchto technologií. Zkušenosti ukazují, že simulace a digitální dvojče jsou cestou, jak v malých a středních podnicích využít jejich potenciál. Aby se však využilo těchto technologií, je potřeba nějakým způsobem dané možnosti prezentovat a mít vhodný nástroj, který by nesloužil pouze k prezentaci těchto technologií a jejich výhod, ale i trénování – pro obojí je označována learning factory jako vhodná cesta.

Autoři teoretických výzkumných článků i článků zaměřených prakticky mnohdy zdůrazňují nárůst potřebných znalostí pracovníků všech úrovní v podnicích implementujících Industry 4.0 a learning factories v tomto směru mohou pomoci. Často se to pak řeší různými vzdělávacími akcemi na výzkumných pracovištích a univerzitách pořádanými pro okolní podniky těchto organizací zaměřených na vzdělávání s využitím learning factories. Rozšiřuje se trend testbedů, tedy zkušebních pracovišť sloužících k transparentnímu a opakovatelnému testování vědeckých teorií, výpočetních nástrojů, nových technologií nebo k prezentaci moderních postupů, znalostí, schopností, technologií podnikům v okolí vytvořeného testbedu. Z důvodu velké složitosti testbedů jsou pak často realizovány spojením vědeckých pracovišť s podniky, které jsou ve své oblasti těmi, kdo trendy udává, nebo je má implementované, a má s nimi zkušenosti, jež může pomoci předat dál.

K tvorbě hardwarových modelů je evidentně s oblibou používána stavebnice Fischertechnik. Druhou variantou pak bývá vlastní tvorba HW modelu např. pomocí 3D tisku, využitím dřeva, kartonu a podobně. V čem se různé výzkumné týmy více liší, je způsob řízení těchto modelů, kde se používají PLC, kontroléry Arduina, mikropočítače RaspberryPi nebo napřímo standardní počítače. Lze se setkat i s learning factory postavenou na skutečných velkých strojích, robotech, dopravnících apod. Dalo by se říct, že opakem těchto learning factories jsou pak čistě softwarové learning factories. Autoři se opakovaně shodují, že pokud learning factory není tvořena individuálně pro řešení nebo ověření jednoho daného problému, pak tvorba learning factory je nekonečný proces a je potřeba jej neustále rozšiřovat o nové poznatky, nové algoritmy, nové technologie a o neustále hlubší integraci jednotlivých prvků.

## 7 Shrnutí poznatků a teoretických východisek práce

Na základě provedené rešerše lze říct, že koncept Industry 4.0 klade vysoké nároky na znalosti a schopnosti průmyslových inženýrů. I když je průmyslové inženýrství multioborovým vzděláním, je potřeba si uvědomit, že průmyslový inženýr není elektrotechnik, programátor, datový specialista, síťář, matematik specializovaný na optimalizace nebo např. kybernetik. Přesto všechno koncept Industry 4.0 tyto odbornosti propojuje a vyžaduje v těchto jednotlivých oborech specializaci. Aby průmyslový inženýr získal alespoň základní znalosti v takto široké škále oborů, využívají organizace i univerzity learning factories zaměřené jak na vzdělávání vlastních studentů, tak i pracovníků z okolních výrobních podniků. Akademičtí pracovníci je pak využívají např. na ověřování hypotéz, nových algoritmů vycházejících z jejich výzkumných činností apod. V případě vzdělávání studentů a pracovníků z výrobních podniků se využívají learning factories jak pro komplexní vzdělávání, tak pro vzdělávání v dílčích problematikách z konceptu Industry 4.0. V takovém případě vždy záleží na možnostech a komplexitě využívaných modelů. Jedná se tedy o modely od řídicí jednotky se třemi LED až po skutečné výrobní linky (nebo stroje) v laboratoři schopné skutečné výroby.

Jako znevýhodněné typy organizací jsou z pohledu konceptu Industry 4.0 často identifikované organizace typu SME. Důvodem je jejich neznalost a rigidita. Např. vědí, že simulační nástroje jsou jednorázově poměrně drahé, ale nevědí, co jim simulační nástroje mohou přinést za výhody, a jaký mohou mít dopad na jejich profitabilitu a výrobu jako takovou. Paradoxně jsou tyto organizace uváděny jako ty, pro které by měly prvky konceptu Industry 4.0 např. simulace, CPS, DT, IoT apod. vysoký přínos a realizace je u nich podstatně jednodušší než u větších společností, protože často nemají vedení v zahraničí a „rozhodují sami o sobě“. U těchto organizací je pak často potřeba vzdělávat jejich pracovníky zastávající nejen pozice průmyslových inženýrů v základech těchto moderních technologiích obdobně jako studenty na univerzitách.

Přístupy k learning factories jsou 4:

1. Nevyužívat fyzických modelů (vzdělávacích pomůcek)
2. Využívat vlastních modelů (vlastní stavební prvky, např. využití 3D tisku)
3. Využívat stavebnic modelů
4. Využívat koupených od výrobce sestavených modelů

Výzkumníci a vzdělávací organizace pak často volí kombinaci nákupu hotového modelu a stavby vlastního. Zakoupený hotový model se využívá k prezentacím, ale zasahuje se do něj minimálně např. z důvodu jeho uzavření (nemožnost jakékoliv vlastní úpravy nebo možnosti velmi omezené), nebo složitosti, náročnosti řízení/programování nebo vysoké pořizovací ceny a drahého servisu v případě nepovedeného zásahu. Vlastní model pak volí z důvodu absolutní kontroly nad modelem a jeho znalosti. Zde je i benefit v tom, že na tvorbě vlastního modelu se nejvíce naučí a tyto modely mohou napojovat na jakékoliv další systémy, které organizace využívá, nebo je propojovat mezi sebou.

Nevýhodou v případě modelů je nedostupnost informací. Organizace nezveřejňují kompletní informace k modelům a jejich řídicí software. Prezентují buď výsledky, kterých s modely dosáhly (přičemž tyto modely byly vytvořeny pouze za jedním konkrétním účelem, např. pro ověření konkrétního algoritmu), nebo jejich velmi základní popis – tedy např. jaké vybavení a software využívají, ale bez dalších podstatných informací.

V prostudovaných pramenech nebylo nalezeno řešení problematiky návrhu, nebo realizace learning factory s digitálním dvojčtem pro modelování a simulování hlavních myšlenek konceptu Industry 4.0, který by bral v potaz, že je určený pro vzdělávání průmyslového inženýra, využívání samotným průmyslovým inženýrem (bez potřeby dalších pracovníků specifického odborného vzdělání) a zohledňoval by schopnosti a znalosti průmyslového inženýra vyplývající z jeho vzdělání a legislativní omezení vyplývající z tohoto vzdělání.

Provedená rešerše a z ní vyplývající závěry poskytují pohled na problematiku Industry 4.0 ze strany průmyslových inženýrů. Na základě prostudované převážně zahraniční literatury odpovídající současnému stavu poznání lze stanovit tyto dílčí teze disertační práce:

- Průmyslové inženýrství je silně multidisciplinární obor.
- Průmyslový inženýr plánuje, projektuje, řídí a implementuje komplexní integrované výrobní systémy a systémy pro poskytování služeb, zabezpečuje výkonnost a spolehlivost systémů, řídí náklady, zlepšuje procesy, zvyšuje produktivitu práce a efektivitu výroby, vzdělává a trénuje.
- Koncept Industry 4.0 je náročný jak na rozsah znalostí (multioborovost), tak jejich hloubku. Klade tedy vysoké požadavky jak na horizontální, tak vertikální znalosti.
- Jedním z nosných pilířů konceptu Industry 4.0 je modelování a simulace.
- Simulace jsou pro organizace typu SME drahé, resp. nejsou seznámeny s jejím přínosem a výhodami. Organizace vidí pouze počáteční poměrně vysokou investici.
- Organizace typu SME zavádějí koncept Industry 4.0 pomaleji než větší organizace, i když z něj mohou více těžit.
- Industry 4.0 rozšiřuje myšlenku digitálních dvojčat na kyber-fyzikální systém (systémy).
- Průmyslový inženýr bez elektrotechnického vzdělání je v ČR znevýhodněn nemožností získat odbornou způsobilost podle vyhlášky č. 50/1978 Sb. (do 30. 6. 2022), o odborné způsobilosti v elektrotechnice § 5 – Pracovníci znalí, nebo § 6 – Pracovníci pro samostatnou činnost, nebo nově podle nařízení vlády 194/2022 Sb. (od 1. 7. 2022), neměl by PLC obvody zapojovat nebo upravovat.
- V rámci vzdělávání v průmyslovém inženýrství a konceptu Industry 4.0 se využívají learning factories postavené jak na skutečných výrobních strojích a PLC, tak na STEM stavebnicích a řídicích jednotkách typu Arduino apod.
- Řídicích jednotek existuje velké množství stejně jako možností, jak je programovat. Některé jsou omezené pouze na složité jazyky (ve smyslu náročnější na naučení) typu C/C++, jiné podporují Python, Javascript nebo např. programování pomocí skládání vývojových diagramů z grafických bloků. U některých řídicích jednotek je možnost volby z více možností programování.
- Learning factories se využívají jak pro vzdělávání v konceptu Industry 4.0 pracovníků společností podnikajících v okolí univerzit vybavených learning factory, tak vzdělávání studentů těchto univerzit.
- Learning factories využívají akademičtí pracovníci pro ověřování hypotéz a testování nových algoritmů.

## 8 Výzkumný záměr

S ohledem na teoretické závěry, které jsou definovány v předchozí kapitole, je stanoven hlavní cíl i dílčí cíle. Na základě hlavního cíle a jemu odpovídajících dílčích cílů jsou definovány hypotézy.

### 8.1 Cíle práce

Hlavním cílem disertační práce je návrh architektury HW learning factory s digitálním dvojčetem, využívajícím diskrétní simulaci, pro vzdělávání v konceptu Industry 4.0. Tato architektura míří na vzdělávání průmyslových inženýrů v konceptu Industry 4.0 a využití v podnicích typu SME. Navržená architektura umožní průmyslovému inženýrovi modelovat a simulovat hlavní myšlenky konceptu Industry 4.0 s ohledem na jeho schopnosti a znalosti vyplývající ze vzdělání průmyslového inženýra. Architektura tedy musí reflektovat klady ale i zápory a omezení vyplývající z „multioborovosti“ průmyslového inženýra. Těmi jsou často pouze základní schopnosti a znalosti programování a databází, základní znalosti elektrotechniky a s ní související legislativní omezení daná původně vyhláškou č. 50/1978 Sb. (do 1. 7. 2022) a nyní nařízením vlády 194/2022 Sb., podle kterých je průmyslový inženýr silně omezený a bez elektrotechnického vzdělání aktuálně nemůže zapojovat/přepojovat ani PLC a s ním související obvody, nebo má pouze základní znalosti z oblasti kybernetiky a dalších souvisejících oborů, které koncept Industry 4.0 využívá a jejichž znalosti implementuje.

Na základě hlavního cíle byly stanoveny dílčí cíle vedoucí k návrhu, realizaci a ověření navržené architektury. Tyto dílčí cíle jsou stanoveny takto:

1. Specifikovat základní prvky konceptu Industry 4.0 prezentované HW learning factory a jejího digitálního dvojčete, tj. komplexního modelu
2. Zvolit vhodné nástroje pro tvorbu jednotlivých částí komplexního modelu
3. Navrhnout architekturu HW modelu s digitálním dvojčetem
4. Ověřit realizovatelnost a funkčnost jednotlivých částí navržené architektury na HW modelu
5. Vytvořit testy pro programovací jazyky C/C++, Python a Robo Pro Coding a sestavit dotazník pro ověření hypotéz
6. Vybrat vhodný a reprezentativní vzorek respondentů pro provedení výzkumu Provést experimentální testování a dotazníkové šetření na respondentech
7. Provést statistické vyhodnocení získaných dat
8. Verifikovat dané hypotézy a stanovit závěry

### 8.2 Hypotézy

Na základě stanoveného hlavního cíle a jemu odpovídajících dílčích cílů byly definovány hypotézy, které jsou ověřovány v rámci následujícího postupu disertační práce, a to:

#### **Hypotéza číslo 1**

Využitím navržené architektury training factory lze průmyslovým inženýrům zjednodušit realizaci a případnou modifikaci komplexního HW modelu schopného komunikovat se svým digitálním dvojčetem oproti standardnímu přístupu využívajícím pro řízení programovací jazyk C/C++, jenž je využit v případě Fischertechnik Training Factory Industry 4.0.

#### **Vysvětlení hypotézy 1**

Předpokládá se, že využitím grafického způsobu programování modelu dle navržené architektury se průmyslovým inženýrům zjednoduší realizace a případné úpravy chování HW modelu. Ověření hypotézy bude provedeno na průmyslových inženýrech testy zjišťujícími jejich schopnost realizovatelnosti úprav řídicího kódu pro HW modely připravených vzorových příkladů vytvořených v grafickém programovacím jazyce a jazyce C/C++ využívaném ve Fischertechnik Training Factory

Industry, popř. jazyce Python, který na těchto jednotkách lze také použít. Výsledky testů budou statisticky vyhodnoceny.

### **Hypotéza číslo 2**

Navržená architektura zvýší časovou efektivitu průmyslových inženýrů při úpravě řídicího kódu HW modelu se základními čidly využívanými v konceptu Industry 4.0 oproti programování HW standardními programovacími jazyky využitými u Fischertechnik Training Factory Industry 4.0 se zachováním možnosti komunikace HW modelu s okolím.

### **Vysvětlení hypotézy 2**

Využitím grafického způsobu programování modelu v navržené architektuře vzroste časová efektivita průmyslových inženýrů využívajících dané architektury pro potřeby např. testování nových algoritmů nebo optimalizačních metod. Např. Fischertechnik Training Factory Industry 4.0 využívá jen pro řídicí programy programovací jazyky C, C++, skriptování v linuxovém shellu a znalost makefile. Pro cloudovou část je potřeba další minimálně 3 jazyky (HTML, CSS, JavaScript) – toto platí pro vytvoření vlastního cloudu, jinak je cloud od Fischertechnik pevně daný a nelze jej upravovat např. podle úprav modelu. Pro správný překlad řídicího programu je navíc potřeba být seznámený s operačním systémem postaveným na linuxovém jádru a překladem mezi různými architekturami, kdy se kompiluje program na 32/64bitovém procesoru pro ARM, a navíc standardně na Windows systému pro Linuxový systém. Tato architektura je tedy pro zásah/úpravu průmyslovým inženýrem zcela nevhodná a v jejím rámci je navrženo a realizováno takové řešení, které realizaci modelu zjednoduší. V rámci navržené architektury tedy je potřeba nalézt cestu, jak umožnit řídicímu programu vytvořeném v grafickém programovacím jazyce komunikovat s okolím hardwarového modelu, pro což nemusí být a často ani u velké části řídicích jednotek jejich grafický programovací jazyk dostatečně připraven.

Ověření hypotézy bude provedeno na průmyslových inženýrech testy s měřeným časem jednotlivých příkladů a obsahujícími připravené vzorové příklady vytvořené v grafickém programovacím jazyce a jazyce C využívaném ve Fischertechnik Training Factory Industry, popř. jazyce Python, který na těchto jednotkách lze také použít. Výsledky testů budou statisticky vyhodnoceny.

### **Hypotéza číslo 3**

Využitím navržené architektury místo architektury Fischertechnik Training Factory Industry 4.0 dojde k redukci nových potřebných odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčtem.

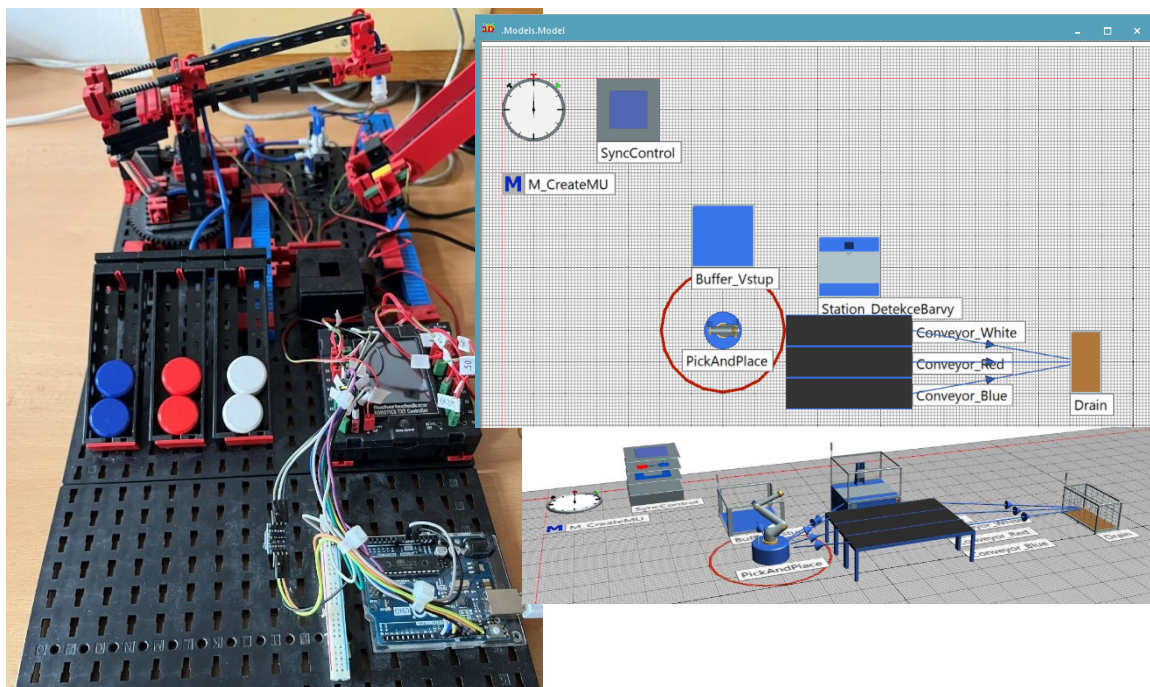
### **Vysvětlení hypotézy 3**

Tato hypotéza bude ověřena porovnáním potřebného množství a typu nových znalostí z oblasti programování a informatiky průmyslových inženýrů pro navrženou architekturu a pro architekturu Fischertechnik Training Factory Industry 4.0.

## 9 Architektura pro training factory s digitálním dvojčetem

Navržená architektura cílí na maximální možné zjednodušení řízení hardware takovým způsobem, aby celek, tedy HW model s digitálním dvojčetem, byl realizovatelný a upravitelný v rámci možností, schopností a znalostí průmyslových inženýrů, kteří nejsou elektrotechniky, kybernetiky ani programátory. Z tohoto důvodu bylo pro programování řídicího kódu hardware zvoleno programování a řízení pomocí vizuálního programovacího jazyka typu Scratch. Podle toho je v případě využití navržené architektury potřeba volit řídicí jednotky tak, aby tento typ vývoje řídicích programů a jejich následné nasazení podporovaly. Kromě samotného řízení musí daná architektura řešit i přenos dat mezi kontroléry řídicími hardware/hardwareový model a digitálním dvojčetem, proto architektura navrhuje i vhodný a v rámci možností nejjednodušší způsob komunikace mezi jednotlivými částmi architektury. Architektura je navržena na základě předchozí rešerše, dlouholeté zkušenosti s prací s průmyslovými inženýry, vývojáři, konzultací s kolegy z jiných univerzit a výuky na FST ZČU.

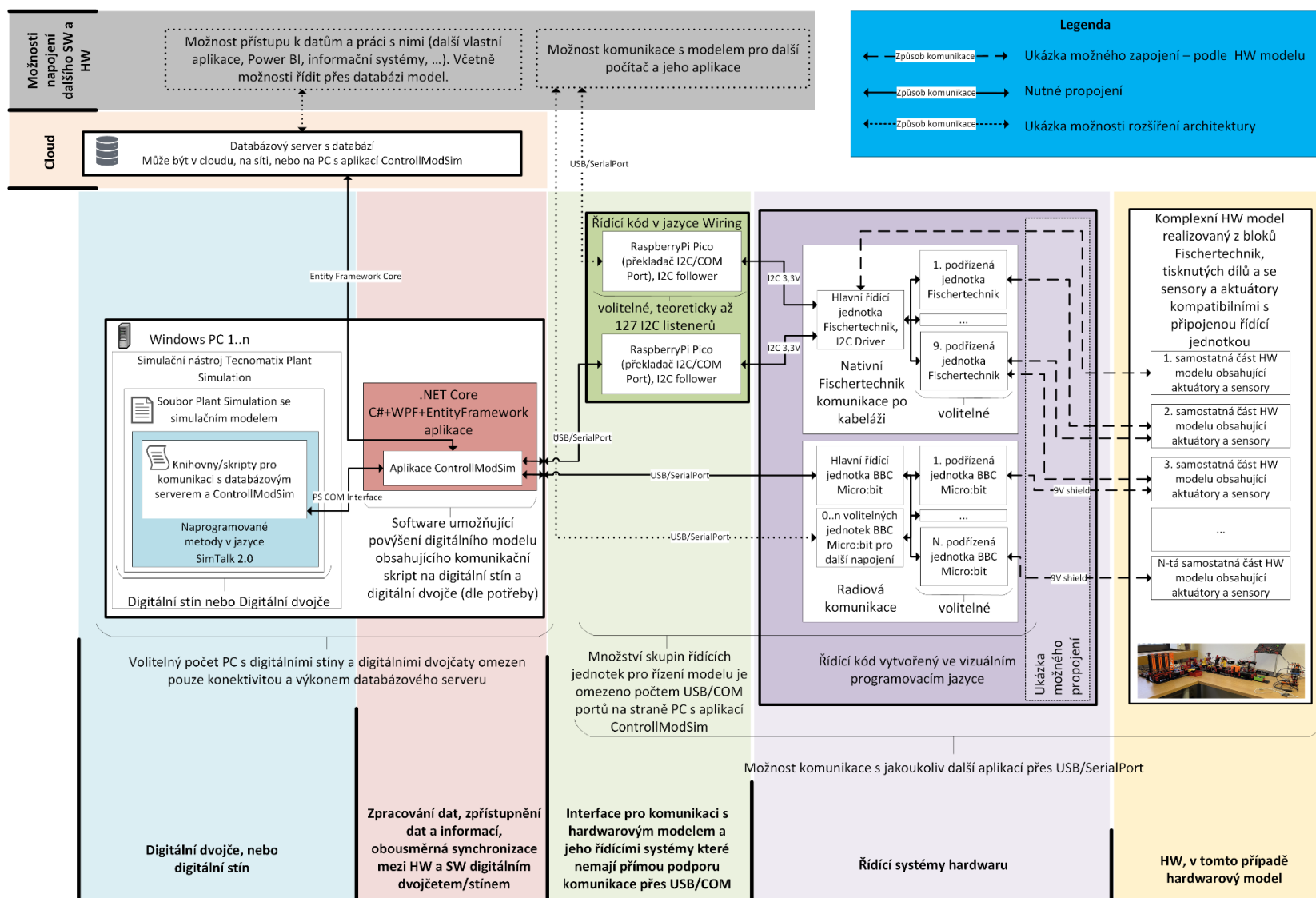
Všechny části navržené architektury pro případ varianty HW modelu s digitálním dvojčetem byly per partes ověřeny na modelu třídičky. Jedná se o model robota, který díky světelné bráně pozná, že dostal materiál, ramenem s podtlakovým uchopovacím mechanismem si materiál vezme, zjistí jeho barvu pomocí analogového barevného sensoru a podle zjištěné barvy materiál zatřídí do správného spádového regálu. Ukázka modelu a jeho digitálního dvojčete v nástroji Tecnomatix Plant Simulation je na následujícím obrázku, viz Obrázek 9-1. Obousměrná komunikace PC – HW model byla ověřena ve spolupráci s Ing. Petrem Švrčulou. S ohledem na rozsah a časové možnosti předmětu „Simulace výrobních systémů a procesů“ je pak tento HW model od roku 2021/2022, ovšem pouze ve variantě s digitálním stínem, využíván jako společné cvičení se studenty, kde se seznamují s myšlenkou digitálního modelu, digitálního stínu a digitálního dvojčete s možnostmi využití nástroje Tecnomatix Plant Simulation v této problematice. Podle navržené architektury je pak aktuálně vytvářena větší training factory, jejíž aktuální stav je popsán dále v rámci této disertační práce.



Obrázek 9-1 – HW model s digitálním dvojčetem v nástroji Tecnomatix Plant Simulation

Schéma navržené architektury training factory s digitálním dvojčetem je na následujícím obrázku, viz Obrázek 9-2.





Obrázek 9-2 – Navržená architektura pro Training factory s digitálním dvojčetem

Na schématu navržené training factory s digitálním dvojčtem je vidět, že se skládá ze sedmi základních částí:

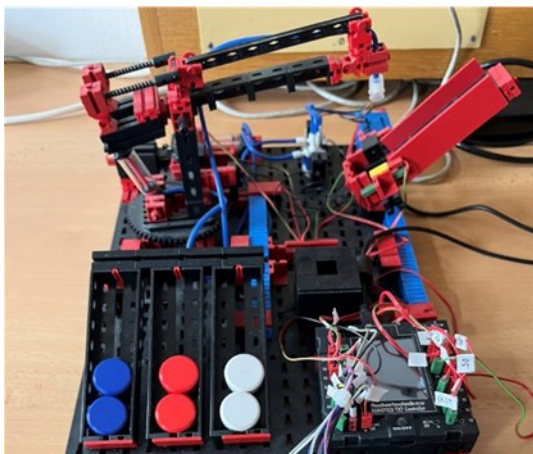
- HW/HW model
- Řídící systémy hardware
- Interface pro komunikaci s HW modelem a jeho řídicími systémy
- Software pro zpracování dat a zpřístupnění informací
- Digitální dvojče, popř. digitální stín
- Cloud – databázový server s databází
- Přípojné body pro připojení dalšího HW a SW

Tyto jednotlivé části jsou podrobněji popsány v následujících podkapitolách.

### 9.1 Hardware/Hardwarový model

První, základní úroveň architektury je HW, resp. hardwarový model systému, pro který je požadováno vytvořit digitální dvojče. Od hardwarového modelu se všeobecně vyžaduje integrace a využití aktuátorů pro simulaci činností a sensorů pro sběr dat z modelu. Může se jednat jak o komplexní stavebnici, kombinaci několika druhů stavebnic, tak o částečně nebo kompletně vytištěný model osazený potřebnými aktuátory a sensory.

V rámci navržené architektury pro training factory byla pro ověření využita stavebnice Fischertechnik s relativně jednoduchou možností rozšířit stavebnici o chybějící nebo atypické díly pomocí 3D tisku. Vlastní vytištěné díly pak mohou sloužit i jako spojovací díly pro různé typy a druhy stavebnic mezi sebou – tím se odstraní problém nekompatibility různých stavebnic. Bylo ověřeno, že díky způsobu spojování dílů stavebnice Fischertechnik lze tisknout vlastní díly s možností velmi pevného a spolehlivého napojení na originální stavební bloky. Ukázka HW modelu, v tomto případě třídačky, která byla využita pro ověření jednotlivých částí navržené architektury, je na následujícím obrázku, viz Obrázek 9-3.



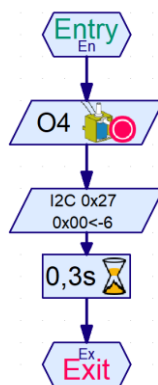
Obrázek 9-3 – Model třídačky

### 9.2 Řídící systémy hardwaru

Kvůli zjednodušení realizace a zachování možnosti úprav průmyslovými inženýry je od řídicích systémů pro řízení hardwarového modelu vyžadována možnost tvorby řídicího kódu pro řídicí jednotku nebo komplex řídicích jednotek pomocí vizuálního grafického programu typu Scratch. Zároveň je potřeba mít možnost z řídicích jednotek nejen data sbírat, ale i data a pokyny řídicím jednotkám předávat, aby bylo možné vytvořit jak digitální stín, tak i digitální dvojče. Takový způsob komunikace pak musí mít podporu ve vizuálním programovacím jazyce/prostředí pro danou řídicí jednotku.

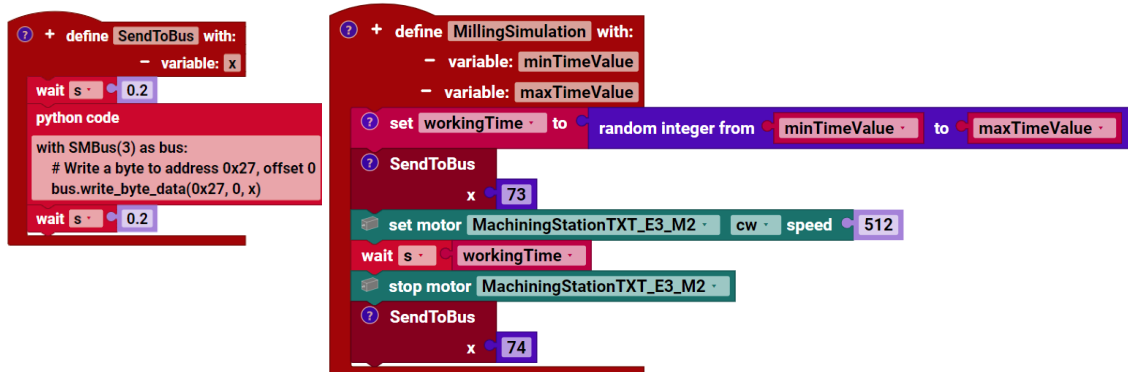
Při ověření této části architektury bylo pro řízení sestaveného hardwarového modelu využito řídicích jednotek, které podporují běh řídicích programů vytvořených pomocí vizuálního programovacího jazyka typu Scratch, který je navrženou architekturou vyžadován. Z tohoto důvodu byly zvoleny řídicí jednotky Fischertechnik TXT 4.0 s řídicím kódem ve vizuálním programovacím jazyce Robo Pro Coding a řídicí jednotky BBC Micro:bit využívající programovací jazyk MakeCode. V případě potřeby lze využít předchůdce Fischertechnik Robotics TXT Controller, se kterým je navržená architektura kompatibilní – jejich starší typ vizuálního programovacího jazyka, tedy programovací jazyk ROBOPro, také podporuje komunikaci po I2C sběrnici. V případě potřeby/požadavku lze využít i jiné řídicí jednotky, jen je potřeba mít na mysli, že zjednodušení realizace training factory a její využití průmyslovými inženýry spočívá v tvorbě řídicích programů HW modelu pomocí vizuálního programovacího jazyka. V takovém případě je potřeba si uvědomit, že jiné řídicí jednotky mohou fungovat na jiné napěťové úrovni a pak jejich kombinace nebo využívání některých aktuátorů a sensorů bude potřeba řešit např. vhodnými převodníky napětí nebo shieldy.

Ukázka části řídicího programu řídicí jednotky Fischertechnik Robotics TXT ve vizuálním programovacím jazyce ROBOPro je na následujícím obrázku, viz Obrázek 9-4. Tento obrázek ukazuje podprogram pro upuštění drženého puku robotického ramene modelu třídičky, která byla využita pro ověření jednotlivých částí navržené architektury. V rámci této sekvence se odesílá po I2C sběrnici na Interface pro komunikaci s HW modelem hodnota 6, což je informace o uvolnění puku pneumatickým obvodem.



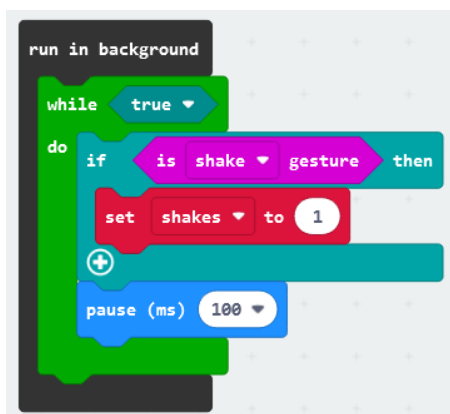
Obrázek 9-4 – Ukázka řídicího kódu jednotky Fischertechnik Robotics TXT

Ukázka části řídicího programu řídicí jednotky Fischertechnik TXT 4.0 ve vizuálním programovacím jazyce ROBO Pro Coding je na následujícím obrázku, viz Obrázek 9-5. Tento obrázek ukazuje metodu pro odesílání dat z HW modelu po I2C sběrnici na Interface pro komunikaci s HW modelem a zavolání této funkce v rámci řídicího kódu. V této konkrétní ukázce předává řídicí jednotka číselné kódy 73 a 74, které prezentují události „Start frézování“ a „Konec frézování“.



Obrázek 9-5 – Ukázka řídicího kódu jednotky Fischertechnik TXT 4.0

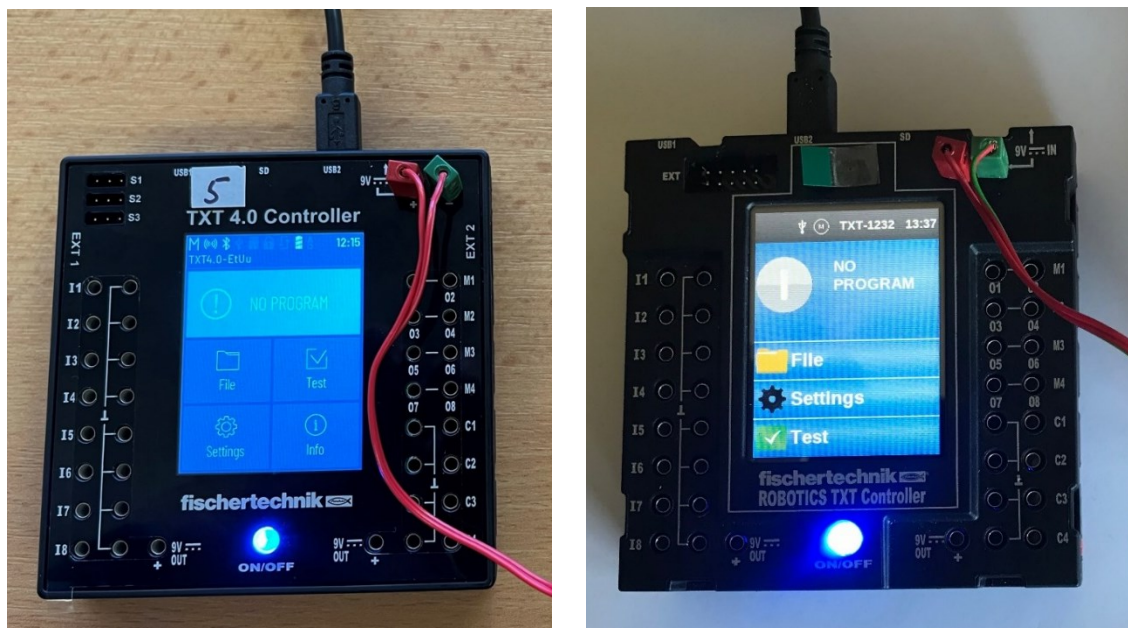
Ukázka části řídicího programu řídicí jednotky BBC Micro:bit je na následujícím obrázku, viz Obrázek 9-6. Tento obrázek ukazuje blok kódu běžícího ve vlákne na pozadí. Tento řídicí kód slouží pro zaznamenávání otřesů sensorovou stanicí.



Obrázek 9-6 – Ukázka řídicího kódu jednotky BBC Micro:bit

### Řídicí jednotka Fischertechnik TXT 4.0

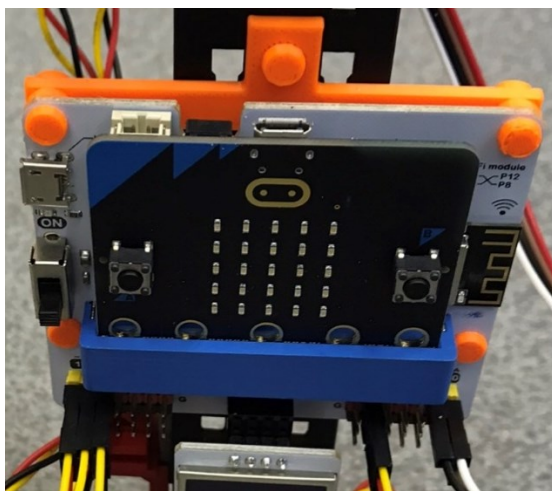
Řídicí jednotka Fischertechnik TXT 4.0 je přizpůsobena pro ovládání a využívání 9V komponent stavebnice Fischertechnik. V případě využívání sensorů a aktuátorů na jiné napěťové úrovni je potřeba použít vhodných převodníků, popř. shieldů. Jedna tato řídicí jednotka může ovládat až 9 dalších řídicích jednotek, které tak fungují jako její rozšíření, připojených na hlavní řídicí jednotku pomocí nativní Fischertechnik komunikace po kabeláži. Volba komunikace pomocí I2C sběrnice byla provedena kvůli zpětné kompatibilitě s předchozí generací jednotek Fischertechnik, nazvaných Fischertechnik Robotics TXT, kterými oddělení také disponuje a u kterých lze předpokládat větší zastoupení mezi uživateli, než se nová řídicí jednotka TXT 4.0 Controller více rozšíří. Díky tomu je možné v rámci architektury využívat obě generace řídicích jednotek. Ukázka řídicích jednotek Fischertechnik TXT 4.0 a Robotics TXT je na následujícím obrázku, viz Obrázek 9-7.



Obrázek 9-7 – Ukázka řídicích jednotek Fischertechnik TXT 4.0 a Fischertechnik Robotics TXT

### Řídicí jednotka BBC Micro:bit

Řídící jednotky BBC Micro:bit podporují nativní radiovou komunikaci a zároveň komunikaci přes USB Sériový port. Proto jsou vhodným doplňkem řídicích jednotek Fischertechnik TXT 4.0. Díky možnosti radiové komunikace lze velmi jednoduše docílit posílání dat z jednoho kontroléru BBC Micro:bit na libovolné množství dalších kontrolérů BBC Micro:bit, přičemž jedinou podmínkou je fyzická dosažitelnost radiovým signálem. Stejně tak lze docílit snadno např. sběru dat z mnoha kontrolérů BBC Micro:bit pomocí radiového signálu na jeden BBC Micro:bit přijímač. Pokud vyvstane potřeba řídit nebo sbírat data pomocí 9V aktuátorů a sensorů stavebnice Fischertechnik, lze to vyřešit přímo shieldem pro BBC Micro:bit, který společnost Fischertechnik dodává, nebo lze použít jakýkoliv vhodný převodník napětí pro 3,3V, na kterých běží kontroléry BBC Micro:bit. V případě využití nativních 3,3V aktuátorů a sensorů v HW modelu se jednoduše vytisknou vlastní 3D díly pro jejich začlenění do HW modelu. Ukázka řídicí jednotky BBC Micro:bit zapojené do sensorové stanice je na následujícím obrázku, viz Obrázek 9-8.

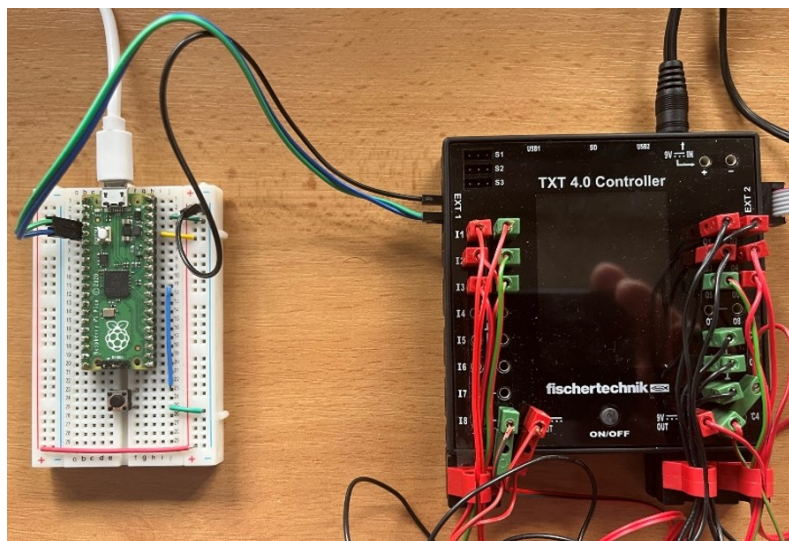


Obrázek 9-8 – Ukázka řídicí jednotky BBC Micro:bit

### 9.3 Interface pro komunikaci s HW modelem a jeho řídicími systémy

Tato část navržené architektury umožňuje využívat i řídicí jednotky, které bez nějakého dalšího rozšíření nebo úpravy nepodporují přímou komunikaci se svým okolím potřebnou pro sdílení dat mezi HW modelem a digitálním dvojčetem. Tento interface tedy není vždy nutnou součástí realizované HW training factory s digitálním dvojčetem. Pokud ovšem zvolené řídicí jednotky nativně nepodporují rozhraní, které lze použít napřímo (v aktuální verzi navržené architektury jde o komunikaci přes USB/ComPort), lze to řešit tímto „překladačem“ mezi dostupnými rozhraními.

Pro komunikaci hlavní řídicí jednotky Fischertechnik TXT 4.0, popř. těchto hlavních řídicích jednotek, s počítačem, kde je zajištěno ukládání a zpřístupnění dat mezi HW a digitálním dvojčetem, je potřeba zpřístupnit hlavní řídicí jednotky tomuto počítači. Pro toto zpřístupnění byla zvolena komunikace přes I2C s nutností zajistit každé hlavní jednotce Fischertechnik TXT 4.0 překladač, který předává data mezi řídicí jednotkou a počítačem s digitálním dvojčetem mezi I2C sběrnici a USB Sériovým portem. Jako překladač byl zvolen mikrokontrolér RaspberryPi Pico, který běží na 3,3V stejně jako I2C na straně řídicích jednotek Fischertechnik. Úkolem tohoto mikrokontroléru tedy není nic jiného než předávání dat mezi I2C na straně řídicí jednotky Fischertechnik a USB/Sériovým portem počítače a naopak. Ukázka RaspberryPi Pico zapojeného a připraveného fungovat jako překladač je na následujícím obrázku, viz Obrázek 9-9.



Obrázek 9-9 – Interface pro komunikaci s HW modelem využívající RaspberryPi Pico

Řídící program (sketch) pro tento překladač je vytvořen v jazyce Wiring, tedy ve stejném jazyce, jako se používá pro mikrokontroléry rodiny Arduino. Důvodem pro využití tohoto jazyka je nutnost pro překladač fungovat jako I2C follower, což není v jazyce Python zatím nativně podporováno (dle autorů to bude v blízké době podporované a už existují uživatelská řešení, viz [130] [131]). Výhodou tohoto jazyka pak je potřeba pouze drobných úprav řídicího programu pro cílový mikrokontrolér v případě využití některého z Arduin místo RaspberryPi Pico. Naprogramovaný překladač nebo překladače lze pak při úpravách hw modelu a jeho digitálního dvojčete v rámci navržené architektury využívat jako Black Box řešení, u kterého pro začátek stačí vědět, že slouží pouze k překladač dat mezi 2 druhy rozhraní. Převodník USB/I2C lze pořídit i jako hotové řešení, např. Převodník USB na I2C FT200XD [132] - v takovém případě není potřeba si toto zařízení sestavovat a programovat. Ukázka řídicího programu pro jednosměrný překlad dat I2C => USB/Sériový Port pro RaspberryPi Pico je na následující obrázku, viz Obrázek 9-10. Řídící program pro RaspberryPi Pico je elektronickou přílohou této disertační práce, viz soubor Data\_translator\_Sketch.ino. Více viz přehled elektronických příloh s jejich popisem, který je uveden v příloze A této disertační práce.

```

DataTranslatorSketch
// University of West Bohemia
// Department of industrial Engineering
//
// usage for communication between HW model and digital twin
// data translator between Fischertechnik model and Computer C# application
// the translator is RaspberryPi Pico as I2C follower, Fischertechnik TXT 4.0 is a Controller
//
// If there is a different voltage between Controller and Follower, the voltage converter must be used
//
// Miroslav Malaga, malaga@kpv.zcu.cz
// November 2022
// version n. 3.1]
//
// tested and used with library: Arduino Mbed OS RP2040 Boards 2.6.1

#include <Wire.h> // using the Wire.h library to enable the i2c bus usage

int msg; // here will be stored a new message from Controller, message is a number
const byte f_addr = 0x27; //the I2C adress of the follower in hexadecimal system
MbedI2C MyWire(p2,p3); //using GP2(SDA) and GP3(SCL) pins for I2C communication

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT); // initialize digital pin for built in LED as an output
  MyWire.begin(f_addr); // join i2c bus with address #39, address must be unique in used controllers
  Serial.begin(9600); // start serial for output, the same baud rate must be on the computer side
  MyWire.onReceive(receiveEvent); // register event
}

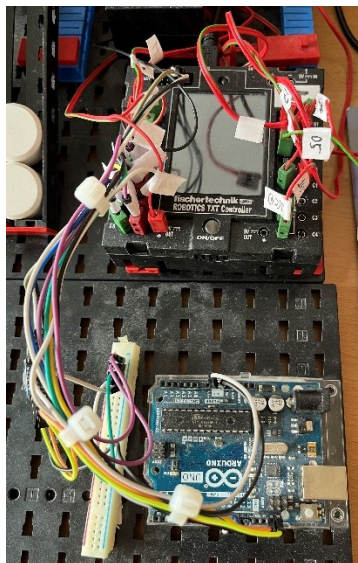
// function that executes whenever data is received from Controller, function is registered as an event, see setup()
void receiveEvent(int numBytes) // parameter tells how many bytes have been received, and consequently how many can be read with Wire.read()
{
  digitalWrite(LED_BUILTIN, HIGH); // LED as visual info, that data transfer is in progress
  MyWire.read(); // reading a first byte - the info that something is comming, dropping it...
  msg = MyWire.read(); // reading a second byte - the new message
  Serial.println(msg); // sending the message to serial
  //delay(1000); // time delay to light up the LED if its necessary
  digitalWrite(LED_BUILTIN, LOW); // LED as visual info, that data transfer is done
}

void loop()
{
  //delay(100); //just to have some pause, protection against congestion, used only when testing
}

```

Obrázek 9-10 – Sketch pro řízení překladače (Interface pro komunikaci s HW modelem)

*Pozn.: Jako prvotní návrh a realizace třídičky bylo využíváno Arduino UNO. Nevýhodou tohoto řešení byla nutnost využít převodní napětí 3,3V – 5V, protože Arduino UNO běží právě na 5V. Další velkou výhodou změny je možnost využití vícejádrového procesoru u RaspberryPi Pico (většina Arduin, bohužel právě těch nejrozšířenějších, využívá čipů rodiny ATmega, které jsou jednovláknové a nepodporují tak běh ani vývoj vícevláknových sketchů) k běhu více procesů najednou a zároveň podpory dvou I2C sběrnic (tedy 2x127 zařízení). To do budoucna otevírá mnoho možností pro rozšiřování funkcionality překladače. Třetí, původně nezamýšlenou výhodou, bylo snížení nákladů. Cena Arduina Uno (nejpopulárnější a široce používaný typ Arduina, zároveň jeden z nejlevnějších a díky této kombinaci je pro něj i nejméně návodů a tutoriálů [133–136]) je aktuálně 715 Kč [137], zatímco cena RaspberryPi Pico je v případě normální verze 119 Kč [138] a v případě verze s Wi-Fi 179 Kč [139]. Ukázka Arduina Uno zapojeného a využívaného v prvotní realizaci digitálního dvojčete pro model třídičky je na následujícím obrázku, viz Obrázek 9-11.*



Obrázek 9-11 – Ukázka Arduino Uno zapojeného a využívaného jako překladače v prvotní realizaci

#### 9.4 Zpracování a zpřístupnění dat

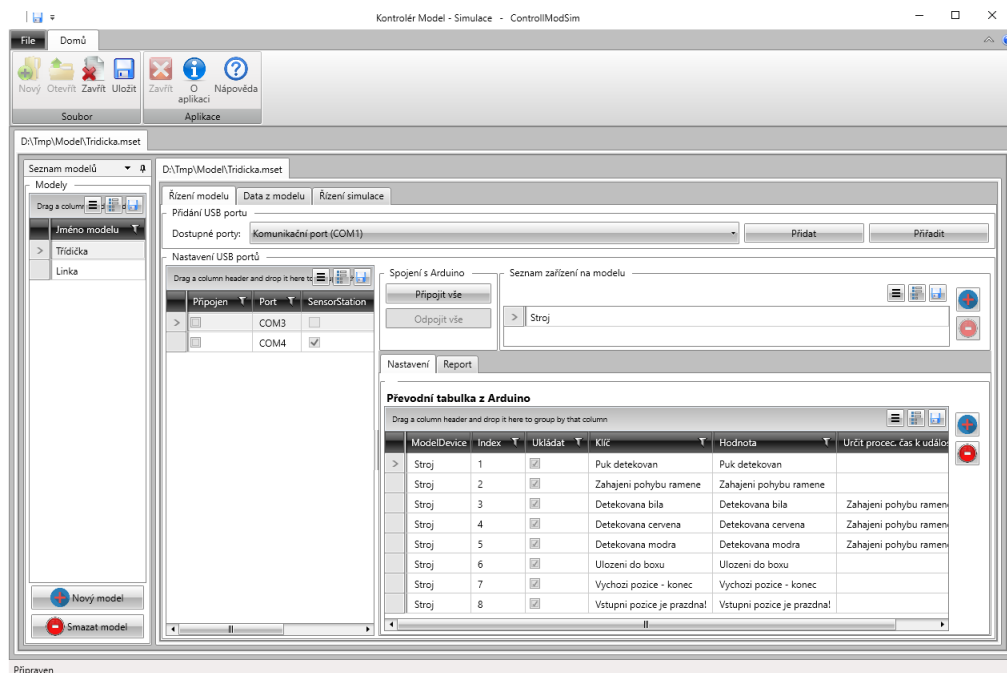
Existuje-li reálný systém (v tomto případě hardwarový model) a jemu odpovídající digitální model, lze jejich propojením vytvořit digitální stín nebo digitální dvojče. Pro toto propojení je potřeba navrhnout a realizovat takové softwarové řešení, které dokáže zajistit pro případ digitálního dvojčete oboustrannou komunikaci mezi HW modelem a simulačním modelem. Pro případ realizace digitálního stínu pak stačí jednostranná komunikace pro předávání dat z HW modelu simulačnímu modelu. Kromě tohoto předávání dat je vhodné, aby tato data byla i ukládána pro případně další využití, popř. pro využití v jiném než reálném čase. Protože data přicházející z modelu a určená digitálnímu dvojčeti, popř. data jdoucí v opačném směru, mohou vypadat téměř jakkoliv (může se jednat např. jen o číselné kódy, které uživateli nic neřeknou), je vhodné, aby na tomto rozhraní byla možnost zadání kódovací/dekódovací tabulky, která by informace dokázala podle potřeby kódovat/dekódovat. Jako příklad lze uvést např. informaci z HW modelu „123“, která může znamenat např. vyřazení aktuálně opracovávaného výrobku jako zmetku. Překlad by měl být umožněný i v opačném směru, aby se minimalizovala velikost dat určených k předání hardwaru/hardwarovému modelu.

Pro ukládání dat i zpřístupnění dat mezi řídicími jednotkami HW modelu a nástrojem Tecnomatix Plant Simulation, ve kterém je digitální dvojče (a obráceně), byla pro ověření architektury a následnou realizaci HW training factory s digitálním dvojčetem vytvořena aplikace ContollModSim. Aplikace byla naprogramována jako .NET Core aplikace pomocí programovacího jazyka C# pro logiku s grafickým rozhraním realizovaným pomocí WPF. Pro ukládání dat a všeobecně práci s databází (v rámci navržené architektury viz Cloud) je využit Entity Framework Core. Pro oboustrannou komunikaci s digitálním dvojčetem v nástroji Tecnomatix Plant Simulation je pak využita dll knihovna Interop.eMPlantLib.dll přímo od tvůrců nástroje Tecnomatix Plant Simulation. Tato knihovna zpřístupňuje aplikaci ControllModSim rozhraní Plant Simulation COM Interface. Díky této knihovně je možné přistupovat zvenčí k simulačnímu modelu (digitálnímu dvojčeti) v nástroji Tecnomatix Plant Simulation. Aplikace zároveň obsahuje naprogramovanou třídu pro řízení a synchronizaci s digitálním dvojčetem.

Aplikace obsahuje možnost vytvoření a naplnění kódovací/dekódovací tabulky (v aplikaci jsou označeny jako převodní tabulky) pro možnost překládání kódu na srozumitelné události/úkony/činnosti a obráceně. Toto podstatně zvyšuje u aplikace ControllModSim její adaptibilitu a možnost komunikace téměř s libovolným hardwarem/hardwarovým modelem, u kterého je potřeba jen využít aktuálně podporovaného rozhraní USB a Seriólové linky pro výměnu dat.

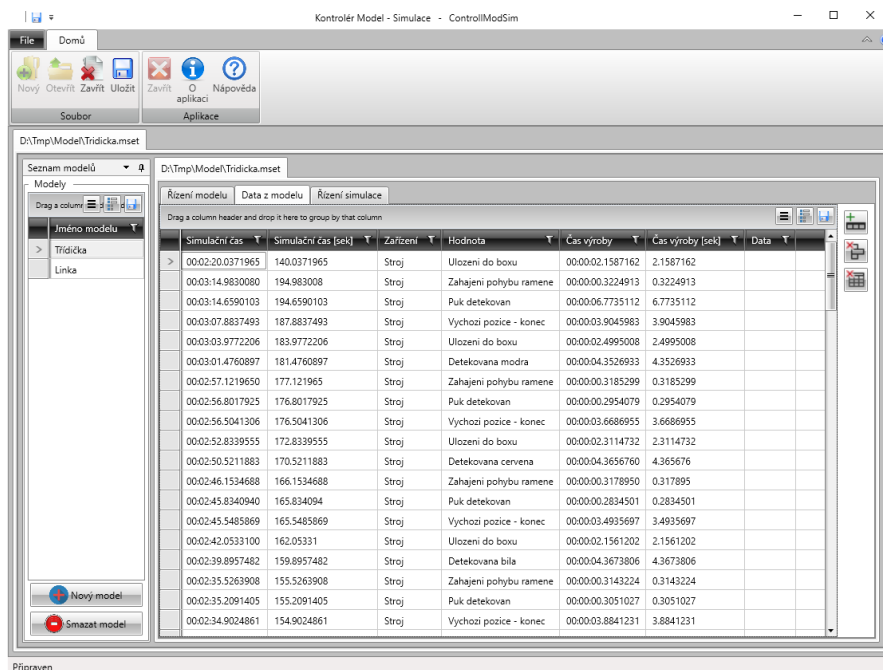


Ukázka aplikace s náhledem na převodní tabulku pro data přicházející z HW modelu třídičky, na kterém byl návrh architektury ověřován, je na následujícím obrázku, viz Obrázek 9-12.



Obrázek 9-12 – Aplikace ControllModSim s náhledem na převodní tabulku HW modelu třídičky

Aplikace ControllModSim pro řízení digitálního dvojčete musí běžet na stejném PC jako Tecnomatix Plant Simulation. Ukázka aplikace ControllModSim s daty z HW modelu třídičky, na kterém byl návrh architektury ověřován, určenými pro digitální dvojče je na následujícím obrázku, viz Obrázek 9-13.



Obrázek 9-13 – Aplikace ControllModSim s náhledem na data z HW modelu třídičky

Do budoucna je plánované rozšíření možností komunikace aplikace ControllModSim s řídicími systémy modelu i na jiná rozhraní a protokoly, jako např. protokol MQTT využívající pro přenos dat

TCP. S tímto rozšíření ovšem vzrostou požadavky na znalosti průmyslových inženýrů, v případě MQTT protokolu např. o základy počítačových sítí, MQTT protokol (broker, klient a další), základy linuxu (v případě využívání např. RaspberryPI jako MQTT brokera apod. Zdrojový kód aplikace ControllModSim je elektronickou přílohou, viz elektronická příloha ControllModSim.zip (více viz přehled elektronických příloh s jejich popisem, který je uvedený v příloze A této disertační práce).

## 9.5 Digitální dvojče/Digitální stín

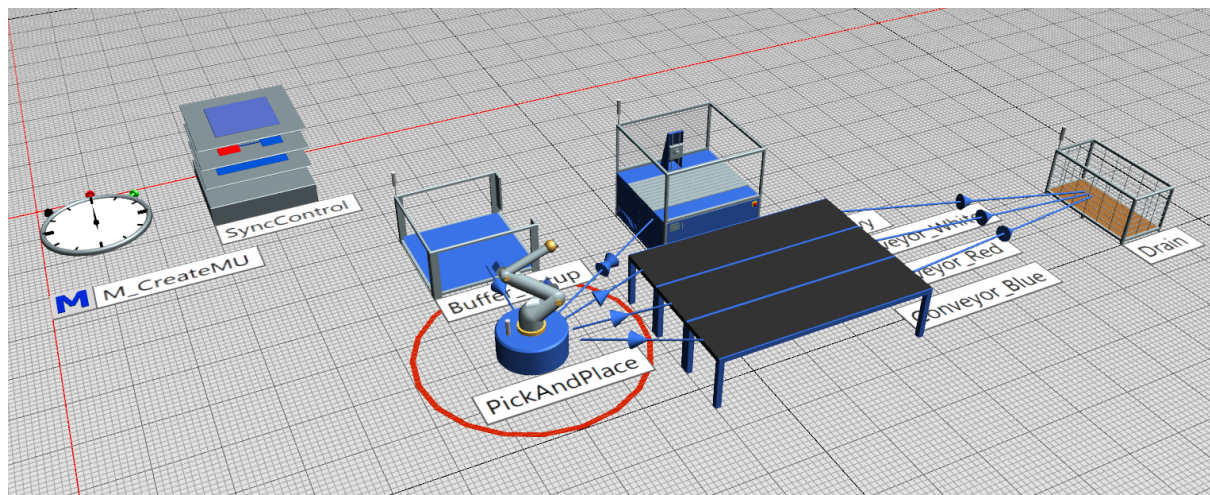
Pro realizaci digitálního dvojčete je potřeba ve vhodném softwarovém nástroji vytvořit digitální model, který by měl mít schopnost obousměrné komunikace se svou reálnou předlohou, umožní vizualizaci nejen aktuálního stavu reálného systému (v tomto případě hardwarového modelu), ale i vizualizaci stavu a simulaci např. pouze teoretických stavů tohoto systému (např. pro ověření dopadu navrhovaných změn v reálném systému). Dalším požadavkem na digitální dvojče (popř. i pouze digitální model nebo digitální stín) je adekvátní blízkost k předloze, jinými slovy adekvátní podobnost a propracovanost vůči reálnému systému.

Digitální dvojče nebo digitální stín (podle potřeby a možností realizátora HW training factory s digitálním dvojčetem) byly v rámci ověření navržené architektury vytvořeny v simulačním nástroji Tecnomatix Plant Simulation, kde pro komunikaci s aplikací ControllModSim musely být vytvořeny metody ve skriptovacím jazyce nástroje Tecnomatix Plant Simulation nazvaném SimTalk 2.0. Tyto metody zajišťují synchronizovanou výměnu dat s aplikací ControllModSim s využitím časových známek a řízení simulačního modelu na základě dat uložených na SQL serveru. Obdobným způsobem musely být naprogramované metody a třídy v aplikaci ControllModSim zajišťující řízení a synchronizaci časů s digitálním dvojčetem. Ukázka metody pro synchronizaci digitálního dvojčete s HW modelem je na následujícím obrázku, viz Obrázek 9-14. Při využívání navržené architektury je mohou průmysloví inženýři využívat jak v rámci svých prací a projektů realizovaných pomocí vlastních HW modelů, tak i ve skutečné výrobě, pokud skutečný systém bude umět posílat data realizovaným způsobem (na USB/ComPort počítače). V případě požadavku je v budoucnu možné způsob sběru dat rozšířit (např. MQTT protokol, WebSockets apod.).

```
//pokud je definovana metoda, tak se zavola
if @ /= void
    if @.CallMethod /= ""
        var pathToMethodCol := splitString( @.CallMethod, ".")
        var o: object
        if pathToMethodCol.Dim = 1
            o := str_to_obj("root." + @.CallMethod)
            o.execute(@.Parameter, @.NextEventTime)
        else
            str_to_method("root." + @.CallMethod).execute(@.Parameter, @.NextEventTime)
        end
    end
end
//ridici program ziska aktualni simulacni cas - z duvodu synchronizace casu mezi aplikaci a modelem
fireSimTalkMessage(to_str(time_to_num(root.eventController.SimTime)))
```

Obrázek 9-14 – Ukázka metody pro synchronizaci digitálního dvojčete s HW modelem v jazyce SimTalk

Ukázka 3D digitálního dvojčete vytvořeného v nástroji Tecnomatix Plant Simulation v ISO pohledu pro hardwarový model třídičky je na následujícím obrázku, viz Obrázek 9-15. Toto digitální dvojče bylo vytvořeno pomocí standardních objektů v Tecnomatix Plant Simulation. Samozřejmě je možné vytvořit si vlastní 3D modely a docílit tak až absolutní vizuální podoby digitálního dvojčete a HW modelu. Soubor digitálního dvojčete pro HW model třídičky je elektronickou přílohou této disertační práce, viz složka Ridici\_programy\_tridicka.zip. Více viz přehled elektronických příloh s jejich popisem, který je uvedený v příloze A této disertační práce.



Obrázek 9-15 – Ukázka 3D digitálního dvojčete vytvořeného v nástroji Tecnomatix Plant Simulation

## 9.6 Cloud

V rámci této části architektury se očekává možnost dlouhodobého uložení dat, které jsou zprostředkovávány mezi reálným systémem a digitálním dvojčetem. V případě potřeby přístupu k datům pro další hardware a softwarové nástroje (jako např. různé informační systémy, PowerBI apod.) by bylo vhodné, aby se jednalo o cloudové úložiště. Taková varianta umožní k datům přistupovat odkudkoliv a popř. je odkudkoliv modifikovat, doplňovat apod. Důležitým aspektem takové varianty je pak i nezávislost reálného systému a digitálního dvojčete na lokalitě – digitální dvojče i hardwarový model jsou nezávislé z pohledu nutnosti umístění ve stejné lokalitě.

Aplikace ControllModSim vytvořená pro ověření architektury a možnost realizace dalších vlastních training factories s digitálními dvojčaty ukládá data (a v případě požadavku je zprostředkovává HW modelu nebo digitálnímu dvojčeti) v databázi. Kvůli často využívanému databázovému serveru MS SQL v reálném prostředí podniků, např. pro ERP, EDMS a další systémy, byla zvolena databáze MS SQL běžící na dostupném serveru. V případě, že je klient s digitálním modelem mimo univerzitu, v tu chvíli to je pro něj Cloud s jedinou podmínkou pro přístup, a to je univerzitní VPN. Díky návrhu využívajícímu EntityFramework Core je možné vytvořit DB kdekoli (od vlastního PC až např. po Amazon Cloud, Azure apod.) včetně možnosti volby potřebné databáze jako např. MS SQL, MySQL, nebo např. lokální SQLite.

## 9.7 Možnost napojení dalšího SW a HW

Training factory s digitálním dvojčetem by měla mít možnost jak dalšího rozšíření jako např. o další hardwarové modely, tak o připojení dalších softwarových nástrojů, které mohou pracovat s daty jak z hardwarové části architektury, tak z části digitálního dvojčete. Toto může být využito např. pro tvorbu mobilních aplikací, vizualizace dat pomocí různých BI nástrojů, tvorbu různých ovládacích pultů a podobně. Navržená architektura umožňuje jednoduché napojení dalšího SW a HW ve dvou přípojných bodech.

Prvním přípojným bodem je možnost připojit se na data posílaná přímo řídicími jednotkami modelu, popř. data napřímou těmto jednotkám s využitím části architektury Interface pro komunikaci s HW modelem. Každá hlavní řídicí jednotka Fischertechnik může mít až 127 I2C listenerů (omezení I2C sběrnice na 128 klientů celkem), přičemž každý listener může sloužit pro komunikaci s jiným softwarem, který může s řízeným modelem pracovat.

Druhým přípojným bodem je databáze, na kterou se lze připojit opět jak za účelem pouze práce s daty (např. PowerBI pro datové analýzy), tak za účelem dodávání dalších dat, která pak může využívat

např. digitální dvojče, popř. lze upravit ControllModSim, aby dodané informace do databáze propagoval do řízeného HW modelu např. formou změny přenastavení.

## 9.8 Kompetence pro využití navržené architektury vs architektury Fischertechnik

V rámci této kapitoly jsou identifikovány IT technologie, znalosti a zkušenosti, které jsou potřeba pro realizaci HW modelu s digitálním dvojčetem při využití architektury, kterou využívá Fischertechnik Training Factory Industry 4.0 a architektura navržená v této práci. Po identifikaci těchto kompetencí proběhla ještě diskuse v rámci již zmíněné tříčlenné expertní skupiny – hlavním důvodem bylo minimalizovat riziko opomenutí důležité kompetence. Tyto identifikované IT technologie jsou pak využity pro sestavení dotazníkového šetření v rámci ověření 3. hypotézy. V rámci zmíněné hypotézy se zjišťují zkušenosti průmyslových inženýrů s těmito IT technologiemi.

### 9.8.1 Kompetence pro využití Fischertechnik Training Factory Industry 4.0 9V

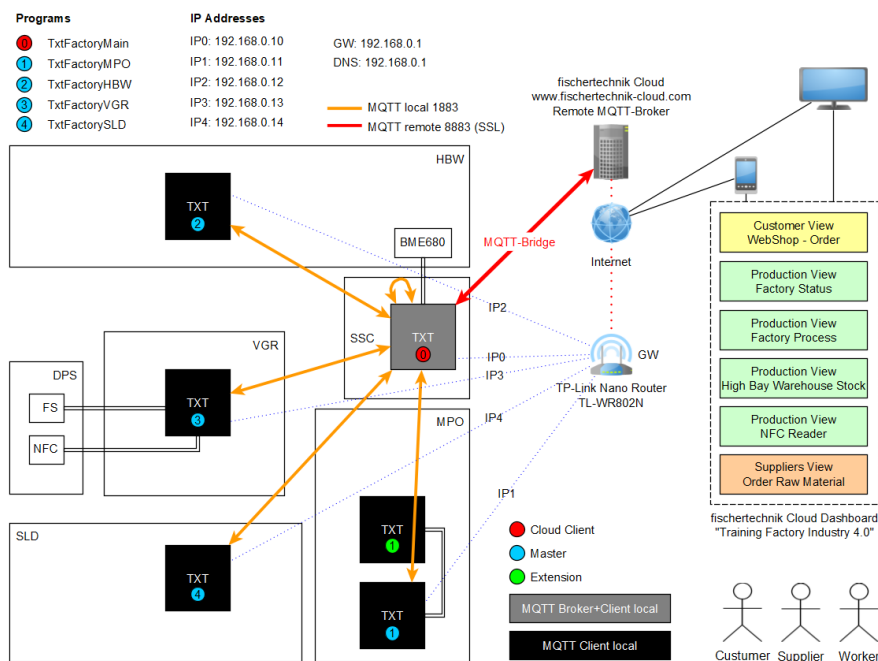
Po seznámení se s modelem Fischertechnik Training Factory Industry 4.0 9V a prostudování zveřejněných řídicích kódů v jazyce C/C++ na githubu společnosti Fischertechnik [140] byly pro realizaci HW modelu s digitálním dvojčetem identifikovány IT kompetence uvedené v následující tabulce, viz Tabulka 9-1. Tyto technologie byly po diskusi v rámci již zmíněné tříčlenné expertní skupiny rozděleny do 7 skupin.

Tyto identifikované kompetence jsou potřebné pro realizaci HW modelu se stejnými technologiemi, jako má HW model Fischertechnik Training Factory Industry 4.0 9V, a navíc technologiemi potřebnými pro komunikaci s digitálním dvojčetem.

Tabulka 9-1– Potřebné kompetence pro využití Fischertechnik Training Factory Industry 4.0 9V

<b>Zkušenosti s programovacími jazyky</b>	C
	C++
	SQL
	Python
	HTML
	PHP
	CSS
	JavaScript
	GTK+ framework
<b>Zkušenosti s vývojovými prostředími</b>	MS Visual Studio
	Eclipse
<b>Ostatní programátorské zkušenosti</b>	JSON
	MQTT protokol
	Komunikace přes COMPort
	Komunikace po síti
	Kompilace programu pro architekturu ARM
	I2C sběrnice
<b>Zkušenosti s uživatelskou prací v operačním systému</b>	Uživatelská práce v MS Windows
	Uživatelská práce v libovolné linuxové distribuci
<b>Zkušenosti se softwarovými nástroji</b>	Node-Red
	DB server (libovolný, např. MySQL, MS SQL, ...)
	Webový server (libovolný, např. Apache, IIS, ...)
<b>Zkušenosti s administrací SW nástrojů</b>	MS Windows Server
	Linuxová distribuce Server (libovolná)
	Node-Red
	DB server (libovolný, např. MySQL, MS SQL, ...)
	Webový server (libovolný, např. Apache, IIS, ...)
<b>Ostatní uživatelské zkušenosti</b>	Komunikace se zařízením přes COM Port
	Komunikace, popř. ovládání zařízení přes SSH
	Příkazový řádek MS Windows
	Shell v Unixu

Ukázka architektury Fischertechnik Training Factory Industry 4.0 9V tak, jak je prezentována společností Fischertechnik, je na následujícím obrázku, viz Obrázek 9-16.



Obrázek 9-16 – Architektura Fischertechnik Training Factory Industry 4.0 9V [140]

Po seznámení se s Fischertechnik Training Factory Industry 4.0 9V byly navíc identifikovány základní charakteristiky, které jsou v rámci expertní skupiny považovány za důležité:

- Ihned po pořízení Training Factory Industry 4.0 9V je možné demonstrovat vypravené aspekty konceptu Industry 4.0.
- Při požadavku na digitální dvojče je potřeba upravit nebo vytvořit nový řídicí kód, který umožní komunikaci Training Factory Industry 4.0 9V s okolím a požadovaným způsobem (v závislosti na digitálním dvojčeti, jeho možnostech a použitých technologiích).
- Přestavba Fischertechnik Training Factory Industry 4.0 9V fakticky znamená rozbití cenově nákladného modelu.
- Cloudová aplikace pro řízení Fischertechnik Training Factory Industry 4.0 9V je neupravitelná a nerozšiřitelná. Nelze tedy po úpravě modelu upravit ovládací cloudovou aplikaci výrobce.
- Nasbíraná data v rámci modelu jsou v cloudové aplikaci výrobce dostupná, pouze pokud je model zapnut, je online a je připojen ke cloudovému účtu.
- Model Fischertechnik Training Factory Industry 4.0 9V může být propojený pouze s jedním uživatelským účtem.
- Při požadavku na změnu logiky v chování modelu jsou vysoké znalostní bariéry způsobené hlavně použitým programovacím jazykem C/C++ a linuxovým operačním systémem, které použité řídicí jednotky využívají.
- Cloud neobsahuje historická data, pouze aktuální stav továrny a skladu.

### 9.8.2 Potřebné kompetence pro využití navržené architektury

Pro realizaci HW modelu s digitálním dvojčetem podle navržené architektury byly identifikovány IT kompetence uvedené v následující tabulce, viz Tabulka 9-2. Tyto technologie byly po diskusi v rámci již zmíněné tříčlenné expertní skupiny rozděleny do 6 skupin. Oproti přístupu Fischertechnik odpadla skupina „Zkušenosti se softwarovými nástroji“, která nebyla naplněna.

Tabulka 9-2 – Potřebné kompetence pro využití navržené architektury

<b>Zkušenosti s programovacími jazyky</b>	VPJ
	C#
	SQL
	EntityFramework
	Python
	XAML (WPF)
<b>Zkušenosti s vývojovými prostředími</b>	MS Visual Studio
	Robo Pro Coding Fischertechnik
<b>Ostatní programátorské zkušenosti</b>	JSON
	Komunikace přes COMPort
	I2C sběrnice
<b>Zkušenosti s uživatelskou prací v operačním systému</b>	Uživatelská práce v MS Windows
<b>Zkušenosti s administrací SW nástrojů</b>	MS Windows Server
<b>Ostatní uživatelské zkušenosti</b>	Komunikace se zařízením přes COM Port

Pro navrženou architekturu byly identifikovány následující základní charakteristiky, které jsou v rámci expertní skupiny považovány za důležité:

- Neexistující referenční HW model, který by byl dostupný na trhu. Toto vyžaduje určité množství kreativity a času na straně realizátorů při stavbě HW training factory s digitálním dvojčetem podle navržené architektury. Toto vede realizaci HW modelu s digitálním dvojčetem, se kterým jsou realizátoři velmi detailně seznámeni a nejsou nijak omezováni v rozšiřování, upravování a vylepšování HW training factory s digitálním dvojčetem.
- Model realizovaný dle navržené architektury není svázaný s žádným účtem ani cloudovou aplikací. Cloud je řešen pomocí SQL Serveru, na který je možné připojovat další potřebné nebo žádané aplikace.
- Pro řízení HW modelu a jeho logiku se předpokládá programování pomocí vizuálních programovacích jazyků (blokového programování).
- Architektura podporuje širší škálu různých typů kontrolérů, což dává větší volnost realizátorům HW modelu. V takovém případě je ale nutné brát ohled na omezení související se zvoleným kontrolérem.
- Historická data jsou dostupná bez ohledu na zapnutý/vypnutý HW model.
- Architektura podporuje snazší rozšiřování o další IT technologie. V závislosti na zvoleném typu řídicí jednotky se nemusí jednat o nárůst složitosti navržené architektury a potřebných kompetencí.

## 10 Vývoj modelu komplexní výrobní linky s digitálním dvojčetem dle navržené architektury

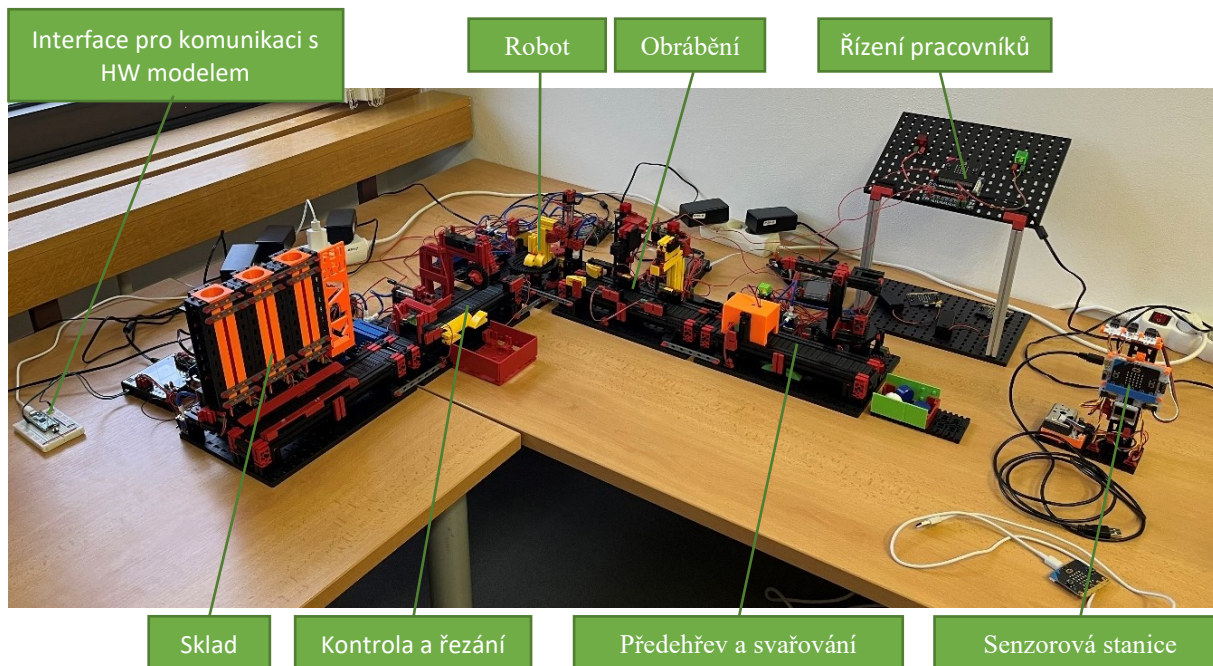
V rámci tvorby větší modelu HW training factory s digitálním dvojčetem pro potřeby katedry byl sestaven a naprogramován základní model výrobní linky ze stavebnice Fischertechnik, která využívá řídicích jednotek Fischertechnik TXT 4.0, BBC Micro:bit a RaspberryPi Pico. Model využívá modulárního principu, kdy lze jednotlivé moduly spojovat v libovolném pořadí a zároveň je tím zajištěna jednoduchá možnost rozšiřování HW modelu do budoucnosti jak v rámci výzkumu, tak např. v rámci semestrálních prací, kvalifikačních prací nebo podle potřeb disertačních prací navazujících na tuto práci, popř. využívajících HW modelu nebo HW modelu s digitálním dvojčetem. Aktuální HW simulační model se skládá ze sedmi modulů:

- Modul skladu
- Modul výrobní a kontrolní úsek se vstupní kontrolou a simulací řezání
- Modul manipulačního robota
- Modul výrobního úseku se simulací frézování a řezání
- Modul systému pro řízení servisního personálu
- Modul předehřevu (s kontrolou barvy výrobku) a svařováním
- Modul sensorové stanice pro měření stavu prostředí

Celý HW model je vytvořený podle navržené architektury. Je tedy naprogramovaný pomocí vizuálních programovacích jazyků, v případě jednotek Fischertechnik TXT 4.0 se jedná o RoboPro Coding, v případě jednotek BBC Micro:bit se jedná o MakeCode. RaspberryPi Pico, které umožňuje komunikaci s PC pomocí sériového portu, využívá jednoduchý sketch vytvořený pomocí jazyka Wiring, který je představený v popisu navržené architektury v rámci této disertační práce, viz kapitola 9.3. Veškeré řídicí programy jsou elektronickou přílohou této disertační práce, viz složka Řidici\_programy\_tovarna.zip (více viz přehled elektronických příloh s jejich popisem, který je uvedený v příloze A této disertační práce). Celá HW training factory, viz Obrázek 10-1, je tak realizována podle navržené architektury, která byla ověřena na modelu třídičky, a v tuto chvíli se pracuje na dotažení úrovně výrobní linky s digitálním stínem. Výrobní linka posílá data (informace o všech událostech v modelu) do aplikace ControllModSim kde jsou dostupná např. pro digitální dvojče. Pan Bc. Lukáš Veszprémi v rámci své diplomové práce zpracovává pro tento model výrobní linky digitální stín v simulační nástroji Tecnomatix Plant Simulation, kterému jsou tato data z HW modelu pomocí aplikace ControllModSim zpřístupněna. Po tomto zpracování digitálního stínu se pak předpokládá povýšení v následujícím akademickém roce na digitální dvojče v kombinaci semestrálních prací studentů vyšších ročníků a bakalářských a diplomových prací.

Zdrojové soubory pro vytištěné díly jsou elektronickou přílohou této disertační práce, viz složka 3D\_tisk.zip. Kompletní přehled údajů, které řídicí linka odesílá je uvedený v elektronické příloze Hardwarovy\_model\_tabulka\_akci.xlsx. Více viz přehled elektronických příloh s jejich popisem, který je uvedený v příloze A této disertační práce.

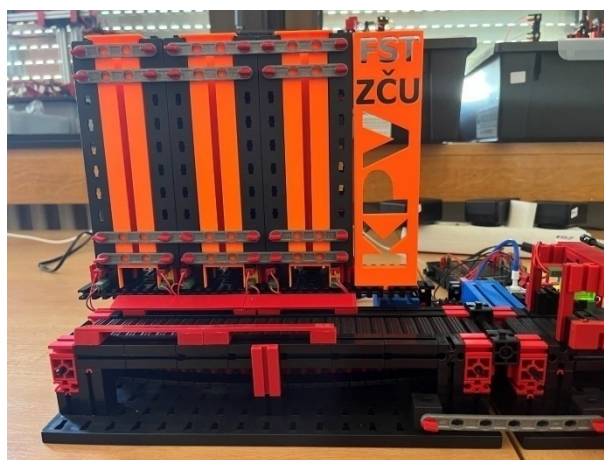




Obrázek 10-1 – Komplexní model HW training factory

## 10.1 Modul sklad

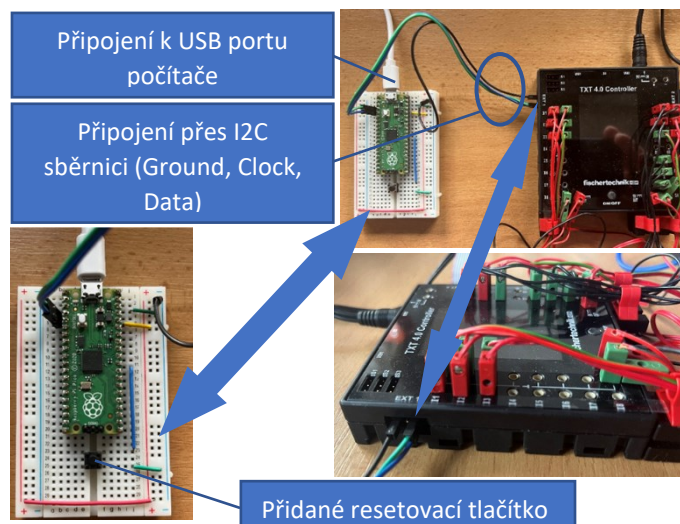
Modul sklad je skladovací systém s dopravníkem postaveným ze stavebnice Fischertechnik v kombinaci s vytištěnými vlastními díly, který slouží k uskladnění a vyskladnění barevných puků představujících materiál (popř. polotovary), v rámci navržené architektury se jedná o část HW/HW model. Systém se skládá ze tří sil a každé silo má svou světelnou závoru, která hlídá jeho obsazenost. Světelná závoru je tvořena vždy fototranzistorem a LED. Každé silo má svůj vlastní výdejní mechanismus, který se skládá z pístu ovládaného 3/2 solenoidovým ventilem a pneumatickým obvodem. Píst vysune puk ze sila na dopravník, který ho přepraví na požadované místo. Stav obsazenosti každého sila je monitorován pomocí světelné závory, která detekuje přítomnost puku ve výdejním otvoru. Pokud jsou všechna tři sila prázdná, řídicí jednotka aktivuje zvukové upozornění, aby uživatel věděl, že je potřeba doplnit sklad. Ukázka modulu skladu je na následujícím obrázku, viz Obrázek 10-2.



Obrázek 10-2 – Modul sklad

Provoz systému je řízen řídicí jednotkou Fischertechnik TXT 4.0 a v rámci navržené architektury je to část Řídicí systémy hardwaru. Tato řídicí jednotka je zároveň hlavní jednotkou celého modelu a

ovládá („managuje“) řídicí jednotky ostatních modulů a zároveň je napojena na překladač (RaspberryPi Pico), který zajišťuje komunikace řídicích jednotek modelu s aplikací ControllModSim. Propojení hlavní řídicí jednotky Fischertechnik TXT 4.0 a překladače postaveného na jednotce RaspberryPi Pico je v rámci navržené architektury část Interface pro komunikaci s hardwarovým modelem. Ukázka tohoto propojení je na následujícím obrázku, viz Obrázek 10-3.



Obrázek 10-3 – Ukázka HW propojení překladače a řídicí jednotky Fischertechnik TXT 4.0

Informace, které aktuálně předává modul skladu do aplikace ControllModSim pro jejich zpřístupnění svému digitálnímu stínu výrobní linky a případnému dalšímu HW a SW, jsou v následující tabulce, viz Tabulka 10-1. V tomto případě se jedná v rámci navržené architektury o Části zpracování a zpřístupnění dat a Cloud (databázový server s databází).

Tabulka 10-1 – Informace předávané z modulu sklad do aplikace ControllModSim

Akce	Číselný kód	Textový popis
Slot1VyskladneniStart	10	Začátek operace vyskladnění materiálu ze skladu, slotu č. 1. Zahájení pohybu pístu pro vyskladnění.
Slot1Vyskladneni	11	Vyskladnění materiálu ze skladu, slotu č. 1 na dopravníkový pás. Dosažení krajní vyskladňovací polohy pístu.
Slot1VyskladneniKonec	12	Ukončení operace vyskladnění materiálu ze skladu, slotu č. 1.
Slot2VyskladneniStart	13	Začátek operace vyskladnění materiálu ze skladu, slotu č. 2. Zahájení pohybu pístu pro vyskladnění.
Slot2Vyskladneni	14	Vyskladnění materiálu ze skladu, slotu č. 2 na dopravníkový pás. Dosažení krajní vyskladňovací polohy pístu.
Slot2VyskladneniKonec	15	Ukončení operace vyskladnění materiálu ze skladu, slotu č. 2.
Slot3VyskladneniStart	16	Začátek operace vyskladnění materiálu ze skladu, slotu č. 3. Zahájení pohybu pístu pro vyskladnění.
Slot3Vyskladneni	17	Vyskladnění materiálu ze skladu, slotu č. 3 na dopravníkový pás. Dosažení krajní vyskladňovací polohy pístu.
Slot3VyskladneniKonec	18	Ukončení operace vyskladnění materiálu ze skladu, slotu č. 3.
VsechnySlotyPrazdne	19	Informace, že všechny sloty skladu jsou prázdné.

Pro některé části tohoto modelu buď nebyly nalezeny vhodné stavební díly zvolené stavebnice Fischertechnik, nebo by se jednalo o velkou spotřebu některých stavebních dílů. Proto bylo v rámci tohoto modelu s přihlédnutím k navržené architektuře využito některých dílů 3D tisku. Tohoto přístupu bylo využito pro vodička určená k vedení puků v sílech a pro reklamní štítek prezentující katedru. Tyto

díly byly vymodelovány v nástroji Autodesk Inventor a vytisknuty na 3D tiskárně. Soubory obsahující modely jsou elektronickou přílohou této práce.

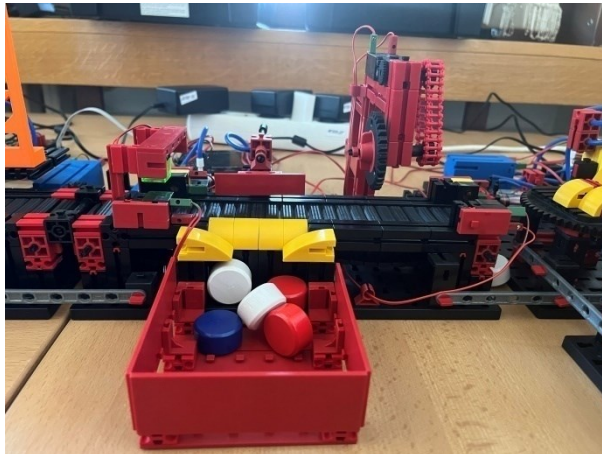
## 10.2 Modul kontroly a řezání

Modul výrobní a kontrolní úsek se vstupní kontrolou a simulací řezání je sestava ze stavebnice Fischertechnik, v rámci navržené architektury se jedná o část HW/HW model. Modul se skládá z dopravníku, který slouží k přepravě materiálu/polotovarů (puky různých barev). Tyto puky jsou přepravovány nejprve na první stanici, kde probíhá simulace rentgenové kontroly puků. Správná pozice puku je detekována pomocí světelné brány, která snímá přerušeni paprsku mezi vysílačem a přijímačem. Pokud je puk v místě kontroly, dopravník se zastaví a LED dioda se na stanovenou dobu rozsvítí – tím se simuluje probíhající kontrola.

Pokud je puk vyhodnocený jako zmetek (na základě generování náhodných čísel), na druhé stanici je vyřazený ze systému pomocí vyřazovacího mechanismu. Toto zařízení se skládá z pneumatického okruhu s kompresorem, solenoidovým ventilem a pístem. Pokud řídicí jednotka detekuje zmetek na dopravníku, pošle signál do solenoidového ventilu, který otevře cestu pro stlačený vzduch do válce s pístem. Píst se pak vysune ven a vystrčí zmetek do vedlejší nádoby.

Pokud puk není vyhodnocený jako zmetek, pokračuje na třetí stanici. Na této stanici proběhne simulace řezání materiálů podle nastavené doby řezání v programu řídicí jednotky. Doba řezání je konstantní pro všechny nevyřazené obrobky. Po řezání dojde obrobek na konec dopravníku, kde je správná pozice hlídána další světelnou bránou, která je využívána následujícím modulem vybaveným robotem s pneumatickým uchopovacím mechanismem.

Ukázka modulu výrobní a kontrolní úsek se vstupní kontrolou a simulací řezání je na následujícím obrázku, viz Obrázek 10-4.



Obrázek 10-4 – Modul kontroly a řezání

Modul je řízený jednou řídicí jednotkou Fischertechnik TXT 4.0, která je followerem hlavní řídicí jednotky. V rámci navržené architektury se jedná o část Řídicí systémy hardwaru.

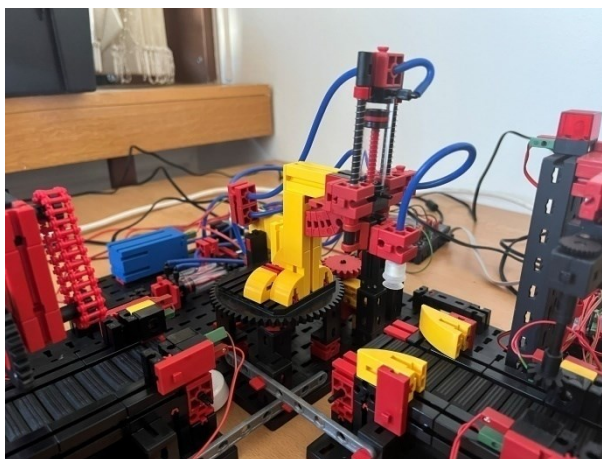
Informace, které aktuálně předává modul výrobní a kontrolní úsek se vstupní kontrolou a simulací řezání do aplikace ControllModSim pro jejich zpřístupnění svému digitálnímu stínu výrobní linky a případnému dalšímu HW a SW, jsou v následující tabulce, viz Tabulka 10-2. V tomto případě se jedná v rámci navržené architektury o části Zpracování a zpřístupnění dat a Cloud (databázový server s databází).

Tabulka 10-2 – Informace předávané z modulu kontroly a řezání do aplikace ControllModSim

Akce	Číselný kód	Textový popisek
<b>RentgenovaKontrolaStart</b>	30	Zahájení rentgenové kontroly obrobku.
<b>MaterialJeZmetek</b>	4	Informace, že materiál je zmetek.
<b>MaterialNeniZmetek</b>	5	Informace, že materiál není zmetek.
<b>RentgenovaKontrolaKonec</b>	31	Ukončení rentgenové kontroly výrobku.
<b>ZmetekVyrazovaniStart</b>	32	Zahájení fyzického vyřazení obrobku ze systému, protože je zmetek.
<b>ZmetekVyrazovaniKonec</b>	33	Ukončení fyzického vyřazení obrobku ze systému, protože je zmetek.
<b>RezaniStart</b>	34	Zahájení operace řezání.
<b>RezaniKonec</b>	35	Ukončení operace řezání.

### 10.3 Modul robot

Modul robota je postavený ze stavebnice Fischertechnik a je součástí komplexní linky na zpracování obrobků, v rámci navržené architektury se jedná o část HW/HW model. Robot pracuje jako spojka mezi dvěma částmi linky, kde přebírá obrobky z jedné větve výrobní linky a předává je k dalšímu zpracování na druhou větev výrobní linky. Robot využívá k uchopování puků vakuový gripper, který vytváří podtlak pomocí solenoidových ventilů a pístů s pneumatickým obvodem, a přísátím se na puk jej uchopí. Robot poté zdvihne rameno s gripperem a otáčí se kolem své osy o určitý úhel. Otáčení robota je řízeno elektromotorem napojeným na ozubený převod. Krajiní polohy robota jsou hlídány koncovými spínači umístěnými na bocích, které lze podle potřeby libovolně posunout pro různé konfigurace linky. Robot pak umístí obrobek na druhou větev výrobní linky. Robot reaguje a spouští svou činnost na základě signálu ze světelné brány umístěné na konci dopravníku připevněné na předchozím modulu. Ukázka modulu robota je na následujícím obrázku, viz Obrázek 10-5.



Obrázek 10-5 – Modul robot

Modul je řízený jednou řídicí jednotkou Fischertechnik TXT 4.0, která je followerem hlavní řídicí jednotky. V rámci navržené architektury se jedná o část řídicího systému hardwaru.

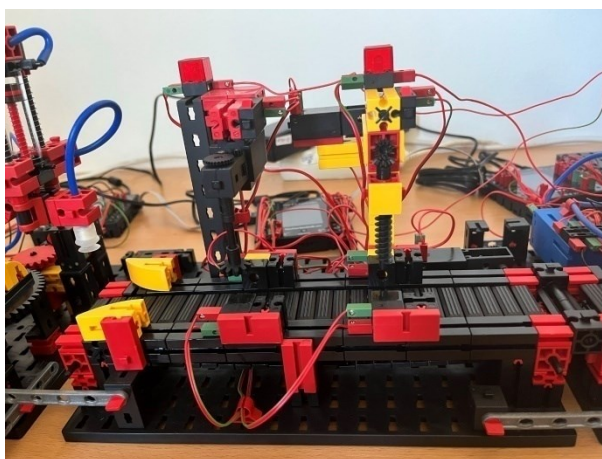
Informace, které aktuálně předává modul robota do aplikace ControllModSim pro jejich zpřístupnění svému digitálnímu stínu výrobní linky a případnému dalšímu HW a SW, jsou v následující tabulce, viz Tabulka 10-3. V tomto případě se jedná v rámci navržené architektury o část Zpracování a zpřístupnění dat a Cloud (databázový server s databází).

Tabulka 10-3 – Informace předávané z modulu robot do aplikace ControllModSim

Akce	Číselný kód	Textový popis
<b>ManipulacíRobotStart</b>	51	Zahájení práce manipulačního robota.
<b>UchopeníVyrobu</b>	52	Uchopení obrobku manipulačním robotem.
<b>UmístěníVyrobu</b>	53	Umístění obrobku na cílovou pozici robotem.
<b>ManipulacíRobotKonec</b>	54	Ukončení práce manipulačního robota.

#### 10.4 Modul obrábění

Modul obrábění je sestaven ze stavebnice Fischertechnik. Modul simuluje proces obrábění obrobků na dvou strojích: frézce a vrtačce a v rámci navržené architektury se jedná o část HW/HW model. Obrobky jsou dopravovány na dopravníku a jejich správná pozice pro obrábění daným strojem je hlídána světelnými bránami. Doba obrábění je jak pro frézování, tak vrtání náhodně generována z intervalů pro tyto dvě operace. Během obrábění mohou nastat chyby, které jsou signalizovány červenou LED diodou na horní části každého stroje. Chyby musí odstranit obsluha stisknutím tlačítka vedle diody na daném stroji, kde chyba nastala, a po jejím odstranění proběhne simulace požadovaného procesu obrábění. Jako rozšíření modelu je připojený na obráběcí stroje modul pro řízení pracovníků, který je popsán v následující kapitole. Ukázka modulu obrábění je na následujícím obrázku, viz Obrázek 10-6.



Obrázek 10-6 – Modul obrábění

Modul je řízený jednou řídicí jednotkou Fischertechnik TXT 4.0, která je followerem hlavní řídicí jednotky. V rámci navržené architektury se jedná o část Řídicí systémy hardwaru.

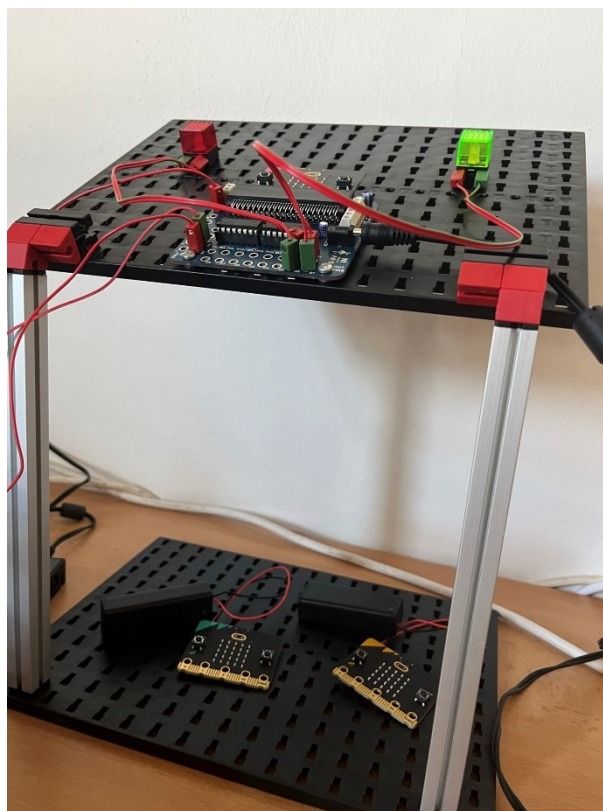
Informace, které aktuálně předává modul obrábění do aplikace ControllModSim pro jejich zpřístupnění svému digitálnímu stínu výrobní linky a případnému dalšímu HW a SW, jsou v následující tabulce, viz Tabulka 10-4. V tomto případě se jedná v rámci navržené architektury o části Zpracování a zpřístupnění dat a Cloud (databázový server s databází).

Tabulka 10-4 – Informace předávané z modulu obrábění do aplikace ControllModSim

Akce	Číselný kód	Textový popis
<b>FrezovaniOpravaStart</b>	71	Výskyt chyby na frézce, kterou je potřeba odstranit uživatelem.
<b>FrezovaniOpravaKonec</b>	72	Ukončení opravy, resp. odstranění chyby na frézce uživatelem.
<b>FrezovaniStart</b>	73	Zahájení operace frézování.
<b>FrezovaniKonec</b>	74	Ukončení operace frézování.
<b>VrtaniOpravaStart</b>	75	Výskyt chyby na vrtačce, kterou je potřeba odstranit uživatelem.
<b>VrtaniOpravaKonec</b>	76	Ukončení opravy, resp. odstranění chyby na vrtačce uživatelem.
<b>VrtaniStart</b>	77	Zahájení operace vrtání.
<b>VrtaniKonec</b>	78	Ukončení operace vrtání.

## 10.5 Modul řízení pracovníků

Modul pro řízení pracovníků na obráběcích strojích je zařízení, které umožňuje efektivněji řešit chyby na frézce a vrtačce, a v rámci navržené architektury se jedná o část HW/HW model. Ukázka modulu obrábění je na následujícím obrázku, viz Obrázek 10-7.



Obrázek 10-7 – Modul řízení pracovníků

Modul se skládá ze tří malých počítačů BBC Micro:bit, které jsou navzájem propojeny bezdrátovou komunikací (rádio) a v rámci navržené architektury jsou částí Řídící systémy hardwaru.

Hlavní mikrokontroler modulu je připojen k Fischertechnik shieldu, který slouží jako adaptér pro napájení a ovládání komponent stavebnice Fischertechnik běžících na 9V (BBC Micro:bit funguje na napětí 3,3V). K tomuto shieldu jsou připojeny dvě LED (jedna červená a jedna zelená), které ukazují

stav systému. Zelená LED svítí, když je systém aktivní, a červená LED svítí, když je systém neaktivní. K shieldu jsou také připojeny další dvě LED diody, které jsou umístěny na frézce a vrtačce – jedná se o LED signalizující poruchy stroje, viz předchozí kapitola. Pokud se některá z nich rozsvítí, znamená to, že stroj potřebuje opravu.

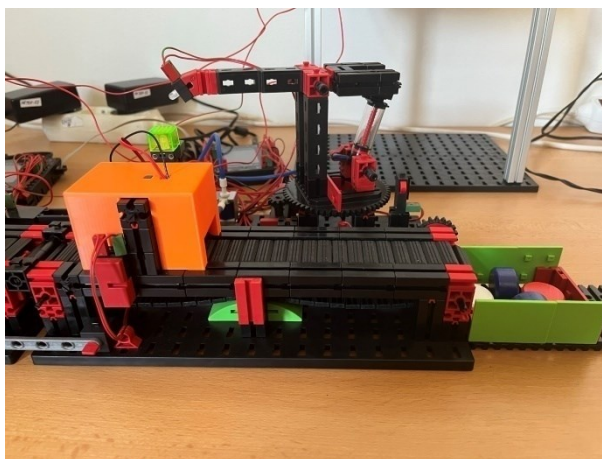
Další dva mikrokontrolery modulu slouží jako nositelnosti pro pracovníky. Tyto nositelnosti mohou představovat např. mobilní telefony nebo chytré hodinky a umožňují pracovníky řídit a dávat jim úkoly a informace od hlavního mikrokontroleru. Když se rozsvítí některá z LED diod na stroji, hlavní mikrokontroler vybere náhodně jednoho ze dvou pracovníků a pošle mu úkol na jeho nositelnost – aby opravil daný stroj. Pracovník je o tom informován na své nositelnosti zobrazením ikony daného stroje na LED displeji a zvoleným zvukem. Pracovník pak musí jít k danému stroji a stisknout tlačítko pro reset chyby. Tím se potvrdí oprava stroje a LED display na nositelnosti zhasne. Nositelnost opět přehraje zvukovou signalizaci jako potvrzení úspěšné opravy.

Modul pro řízení pracovníků na obráběcích strojích je tedy zařízení, které pomáhá zajistit rychlejší a efektivnější řešení chyb na frézce a vrtačce pomocí bezdrátové komunikace mezi jednotkami BBC Micro:bit. Takto navržený a realizovaný modul řízení pracovníků lze nasadit na jakýkoliv stroj nebo stanici v rámci HW training factory. V případě, že je daný objekt vybavený LED signalizující problém, stačí modul řízení pracovníků pouze napojit a není potřeba jakkoliv zasahovat do řídicího kódu modelu. Pokud není vybavený LED informující o problému, lze jej touto LED dovybavit, popř. lze modul pro řízení pracovníků připojit přímo na řídicí jednotku Fischertechnik – v takovém případě je ale potřeba adekvátní úpravy řídicího kódu.

Protože aktuálně veškeré údaje o chybách odesílá modul obrábění, nejsou z modulu pro řízení pracovníků zatím odesílána žádná data. Zatím slouží pouze pro koordinaci servisních pracovníků jako samostatné a v podstatě nezávislé řešení.

## 10.6 Modul předehřevu a svařování

Modul předehřevu a svařování je postavený ze stavebnice Fischertechnik v kombinaci s vlastním vytištěným dílem na 3D tiskárně a v rámci navržené architektury se jedná o část HW/HW model. Soubory obsahující model vytištěného dílu jsou elektronickou přílohou této práce. Ukázka modulu předehřevu a svařování je na následujícím obrázku, viz Obrázek 10-8.



Obrázek 10-8 – Modul předehřevu a svařování

Modul se skládá ze dvou pracovišť: ohřevu a svařování obrobku. Obrobek je mezi stanovišti přepravován dopravníkovým pásem do pece, kde probíhá jeho „ohřev“. Pec je zakrytá krytem vymodelovaným v nástroji Autodesk Inventor a vytištěným na 3D tiskárně. Toto využití 3D tisku je v souladu s navrženou architekturou a v tomto případě slouží k úspoře stavebních dílků stavebnice Fischertechnik. Uvnitř pece jsou umístěny dva senzory: světelná brána a analogový barevný senzor.

Světelná brána zajistí správnou pozici obrobku vůči analogovému barevnému sensoru (aby proběhlo správně zjištění barvy) a vytištěný kryt zajišťuje stále světelné podmínky při zjišťování barvy obrobku. Díky tomu není celý model citlivý vůči okolním světelným podmínkám. Skutečným účelem pece je tedy zjistit barvu právě obráběného obrobku. To, že „ohřev“ probíhá, je znázorněno zelenou LED diodou umístěnou viditelně na horní části pece. Doba ohřevu je aktuálně nastavena jako konstanta. Dopravníkový pás se po ohřevu opět rozjede a přesune obrobek na další pracoviště simulující svařování.

Simulace svařování je realizována svařovacím robotem. Robot se skládá z ramene s elektrodou na konci (LED). Robot má dva stupně volnosti: může otáčet rameno kolem vertikální osy (azimut, pomocí elektromotoru a převodu pomocí ozubených kol) a může pohybovat elektrodou nahoru a dolů (elevace, pomocí pístu, solenoidového ventilu a pneumatického obvodu). Doba svařování se náhodně generuje z intervalů hodnot, přičemž pro každou barvu obrobku je tento interval jiný. Po ukončení svařování je finální výrobek pomocí dopravníkového pásu vyřazen z výrobního systému.

Modul je řízený jednou řídicí jednotkou Fischertechnik TXT 4.0, která je followerem hlavní řídicí jednotky. V rámci navržené architektury jde o část Řídicí systémy hardwaru.

Informace, které aktuálně předává modul předeřevu a svařování do aplikace ControllModSim pro jejich zpřístupnění svému digitálnímu stínu výrobní linky a případnému dalšímu HW a SW, jsou v následující tabulce, viz Tabulka 10-5. V tomto případě se jedná v rámci navržené architektury o části Zpracování a zpřístupnění dat a Cloud (databázový server s databází).

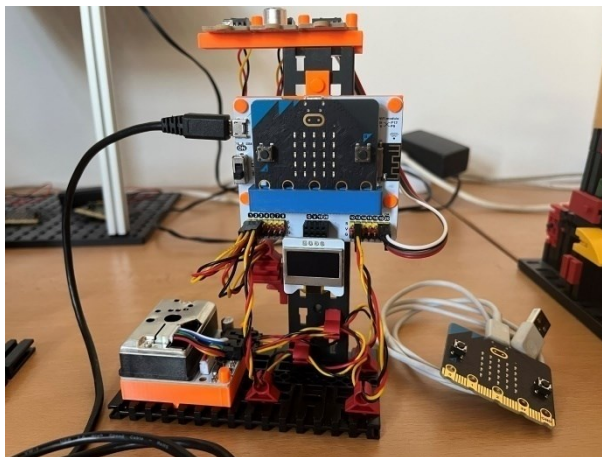
Tabulka 10-5 – Informace předávané z modulu předeřevu a svařování do aplikace ControllModSim

Akce	Číselný kód	Textový popisek
<b>PecStart</b>	90	Začátek ohřevu obrobku.
<b>BilyVyrobek</b>	1	Informace, že obrobek je bílý, resp. že je určený pro bílý finální výrobek.
<b>CervenyVyrobek</b>	2	Informace, že obrobek je červený, resp. že je určený pro červený finální výrobek.
<b>ModryVyrobek</b>	3	Informace, že obrobek je modrý, resp. že je určený pro modrý finální výrobek.
<b>PecKonec</b>	91	Ukončení ohřevu obrobku.
<b>SvarovaniStart</b>	92	Zahájení operace svařování.
<b>SvarovaniKonec</b>	93	Ukončení operace svařování.
<b>VyrazeniHotovehoVyrobkuZeSystemu</b>	100	Vyřazení hotového výrobku z výrobního systému.

## 10.7 Modul sensorové stanice

Modul sensorové stanice je v rámci navržené architektury část HW/HW model a jedná se o zařízení, které dokáže měřit různé fyzikální veličiny v okolním prostředí a posílat je bezdrátově k dalšímu zpracování a díky tomu zasadit chování HW modelu a nasbíraných dat do kontextu stavu okolního prostředí. Kombinuje dvě řídicí jednotky BBC Micro:bit, které jsou v rámci navržené architektury částí Řídicí systémy hardwaru, a dále sensory využívající napětí 3,3V, stavební bloky stavebnice Fischertechnik a vlastní vytištěné díly umožňující kombinaci a sestavení těchto standardně nekompatibilních dílů. Soubory obsahující modely vytištěných dílů jsou elektronickou přílohou této práce. Ukázka sensorové stanice je na následujícím obrázku, viz Obrázek 10-9.





Obrázek 10-9 – Modul sensorové stanice

Senzorová stanice měří pomocí sensorů teplotu, tlak, hluchnost, vlhkost, světelné podmínky, prašnost prostředí, a navíc pomocí pohybového čidla v jednotce BBC Micro:bit i otřesy. Měření probíhá v pravidelných intervalech daných řídicím programem a sada naměřených dat je ihned po naměření zasílána bezdrátově (radiem) jako JSON string druhému BBC Micro:bitu, který je předává na USB/seriálovou linku počítači, kde s daty dále pracuje aplikace ControllModSim – v tomto případě se jedná v rámci navržené architektury o části Zpracování a zpřístupnění dat a Cloud (databázový server s databází). Kromě zasílání dat jsou poslední naměřená data zobrazována i na malém digitálním displeji, který je součástí navržené a postavené sensorové stanice.

## 10.8 Uplatnění training factory realizované podle navržené architektury

Hardwarový model začal být postupně využíván jak ve výuce a semestrálních pracích, tak v diplomových pracích studentů katedry průmyslového inženýrství a managementu FST ZČU. V rámci předmětu KPV/SVS byl v akademickém roce 21/22 realizován první běh cvičení, při kterém se se studenty vytvářelo digitální dvojče HW modelu třídičky. Následně měli studenti jako semestrální práce přidělené jednotlivé moduly HW modelu výrobní linky a pro tento přidělený model vytvářeli digitální stín podle navržené architektury v simulačním nástroji Tecnomatix Plant Simulation.

V akademickém roce 21/22 byly zároveň zadány a úspěšně obhájeny dvě diplomové práce využívající HW model realizovaný podle navržené architektury. První práce byla zaměřena na vytvoření aplikace pro mobilní telefon zobrazující informace a statistiku dat z HW modelu. Druhá práce byla zaměřena na vytvoření modelu ve virtuální realitě synchronizovaného s HW modelem výrobní linky postavené podle navržené architektury. U obou prací se využívají technologie, nástroje, postupy a přístupy, které se používají v reálné praxi, a jsou tedy tyto výstupy do praxe přenositelné, i když byly realizované a ověřené v rámci diplomových prací na training factory. Tyto práce lze považovat za ukázkou životaschopnosti a využitelnosti navržené architektury ve světě průmyslových inženýrů.

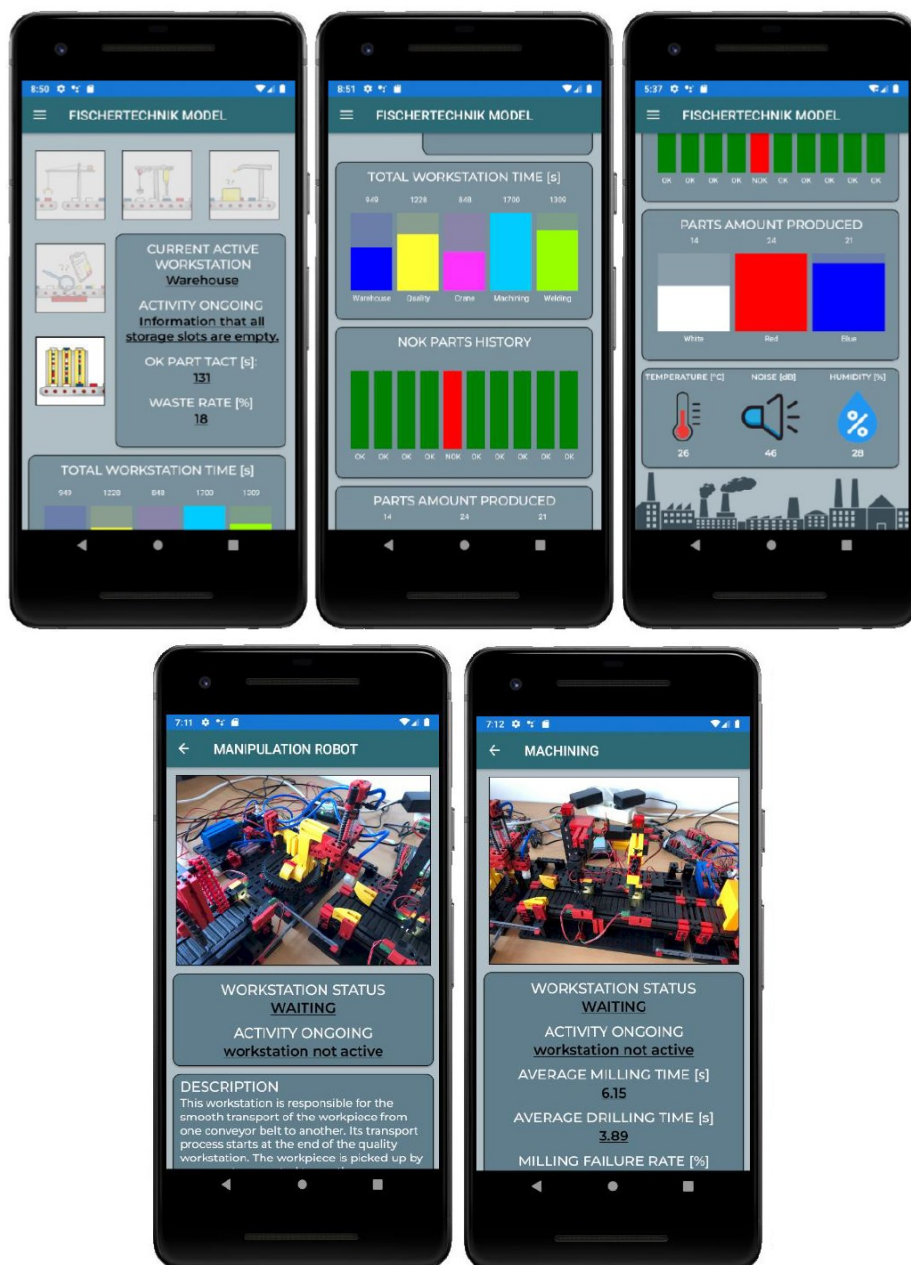
Diplomová práce na téma „Tvorba modulárního interaktivního digitálního dvojčete výrobního úseku pro virtuální realitu“ [141] byla obhájena panem Ing. Matějem Dvořákem. V rámci této práce byl vytvořen digitální stín postaveného modelu výrobní linky (training factory) ve virtuální realitě pomocí Unity 3D, přičemž samotný softwarový model pro využití ve 3D scéně byl vytvořen v modelovacím nástroji Blender. Digitální stín dostává data z HW modelu výrobní linky přes část navržené architektury nazvané Interface pro komunikace s hardwarovým modelem (překladač postavený na RaspberryPi Pico), a to umožňuje synchronizaci modelu ve virtuální realitě s hardwarovým modelem v reálném čase. Ukázka řešení je na následujícím obrázku, viz Obrázek 10-10. Aplikace s digitálním stínem je pak optimalizována pro 3D brýle Oculus Quest 2, na kterých je i provozována. Aktuálně pak probíhá na základě této práce a s využitím HW training factory ve spolupráci s panem Ing. Dvořákem a doc. Hořejším výzkum zaměřený na praktické řešení synchronizace digitálního dvojčete ve virtuální realitě

s realitou v případech, kdy nelze spojitě sledovat a zrcadlit dění v realitě s cílem dosáhnout co nejlepší vizualizace pro pozorovatele ve virtuální realitě. Výsledky budou ověřovány na představené kombinaci training factory s digitálním stínem ve virtuální realitě.



Obrázek 10-10 – Digitální stín training factory ve virtuální realitě [141]

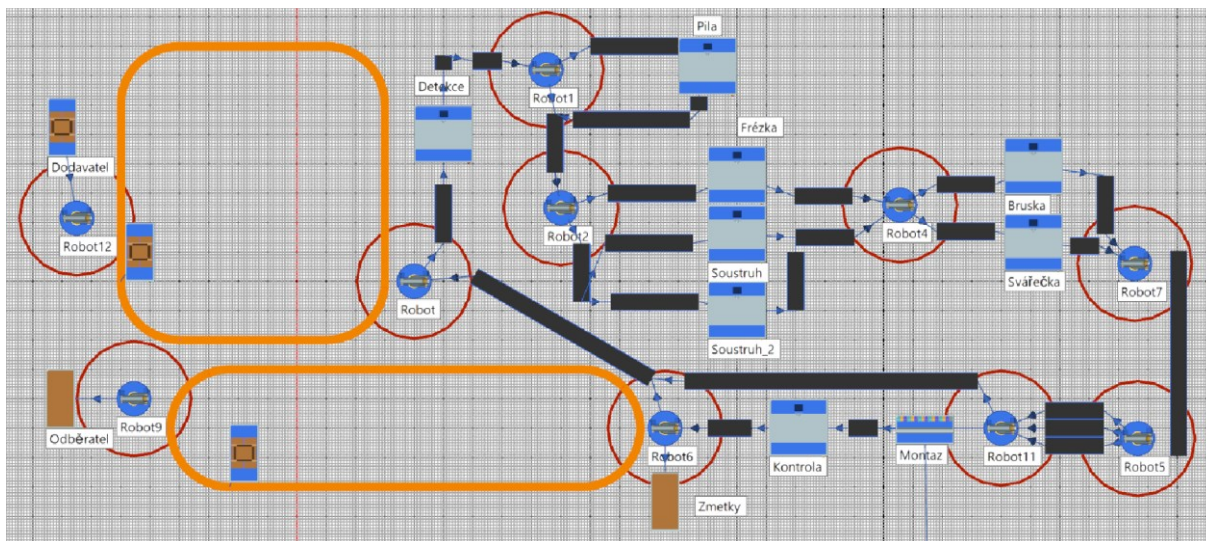
Diplomová práce na téma „Tvorba mobilní aplikace pro zobrazení stavu Fischertechnik modelu“ [142] byla obhájena panem Ing. Markem Velichem. V rámci této práce byla vytvořena aplikace pro mobilní telefony a tablety využívající operační systém Android. Mobilní aplikace využívá části Cloud (databázový server s databází) navržené architektury, pomocí které čerpá data, a zobrazuje jak statistická data ohledně výroby, tak informace o aktuálním dění v HW modelu výrobní linky. Ukázka aplikace je na následujícím obrázku, viz Obrázek 10-11.



Obrázek 10-11 – Mobilní aplikace pro zobrazení stavu a vizualizaci dat HW modelu [142]

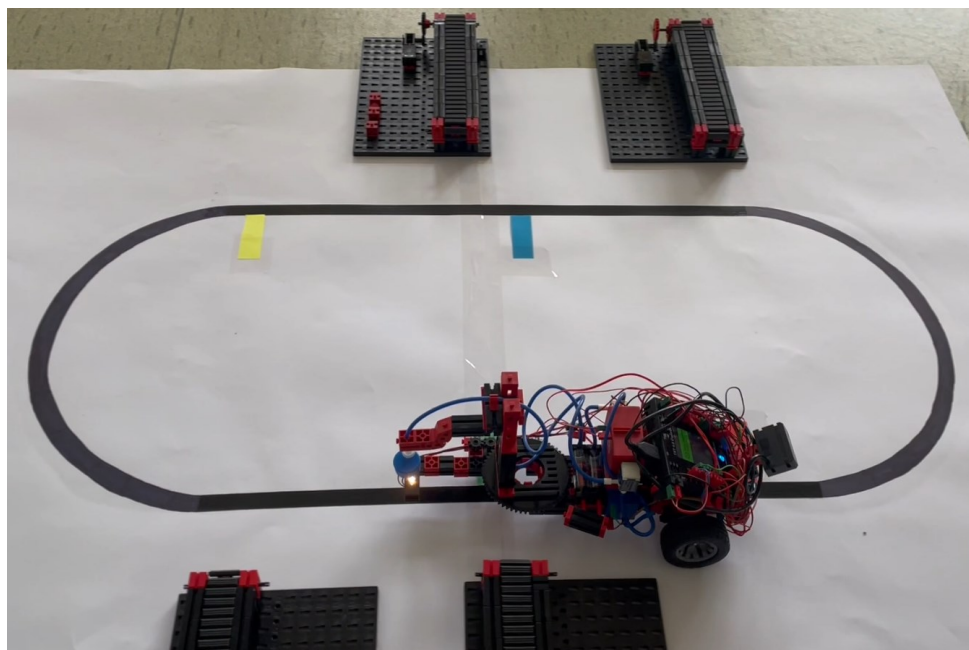
V akademickém roce 22/23 byly zároveň zadány a úspěšně obhájeny dvě diplomové a jedna bakalářská práce využívající HW model realizovaný podle navržené architektury, navrženou architekturu training factory, popř. obojí.

Diplomová práce na téma „Ověření návrhu hardwarového modelu využívajícího metody průmyslového inženýrství pomocí diskrétní simulace“ [143] byla obhájená panem Ondřejem Váňou. Práce je zaměřená na analýzu a návrh HW modelu realizovatelného pomocí navržené architektury s cílem ukázat výhody a dopady implementace vybraných metod průmyslového inženýrství na výrobu (např. Just in Time, Kanban, Theory of Constraints apod.). Ověření navrženého modelu bylo provedeno pomocí simulačního modelu v simulačním nástroji Tecnomatix Plant Simulation a v budoucnu je plánováno realizovat podle navrženého modelu skutečný HW model. Ukázka hlavní části navrženého modelu je na následujícím obrázku, viz Obrázek 10-12.



Obrázek 10-12 – Ukázka hlavní části navrženého a ověřeného SW modelu [143]

Bakalářská práce na téma „Návrh a realizace AGV pro Training Factory s naskladňovacím a vyskladňovacím systémem“ [144] byla obhájena panem Adamem Kohoutem. Práce je zaměřená na návrh, postavení a naprogramování AGV pro transport výrobků v rámci postaveného HW modelu a dalších rozšiřujících pracovišť. Ukázka sestaveného a naprogramovaného prototypu AGV s možností nakládky/vykládky obrobku s rozlišením jeho typu je na následující obrázku, viz Obrázek 10-13. AGV je připravené pro implementaci jako další rozšíření stávající Training Factory.



Obrázek 10-13 – Ukázka realizovaného prototypu AGV pro realizovanou training factory [144]

Aktuálně je pak zadána a zpracovávána diplomová práce zaměřená na vytvoření digitálního stínu kompletního HW modelu výrobní linky v simulačním nástroji Tecnomatix Plant Simulation s možností predikování chování HW modelu v budoucnu. Na tuto práci by v příštím roce měla navázat práce zaměřená na povýšení výstupu práce z digitálního stínu na digitální dvojče.

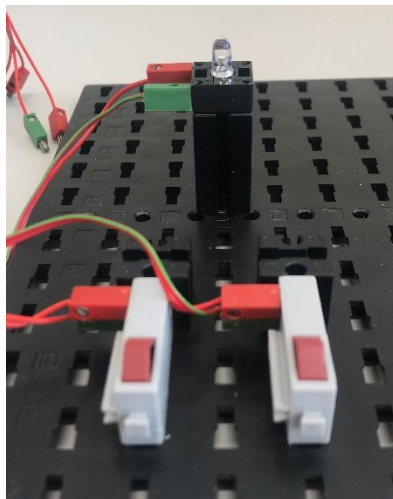
## 11 Způsob testování a sběru dat pro ověření hypotéz

Na základě rešerše a požadavků na cílovou skupinu byla navržena výše uvedená architektura HW learning factory s digitálním dvojčtem využívajícím diskrétní simulaci pro vzdělávání v konceptu Industry 4.0. Tato architektura využívá pro řízení HW modelů vizuálního programovacího jazyka. V případě řídicích jednotek Fischertechnik TXT 4.0 to je jazyk Robo Pro Coding, v případě využití mikrokontroleru BBC Micro:bit to je vizuální programovací jazyk MakeCode. Oba tyto jazyky myšlenkově, strukturou i vizuálem vychází z vizuálního programovacího jazyka Scratch. HW model Fischertechnik Training Factory I 4.0 9 V pro řízení nevyužívá vizuální programovací jazyky, ale jazyk C/C++.

Pro ověření hypotézy č. 1 a č. 2 byly vytvořeny testy s otevřenými otázkami, které ověřují schopnosti probandů pochopit a upravit řídicí kód hardwarových modelů (ve vizuálním programovacím jazyce, jazyce Python a jazyce C/C++). Pro testování byly vytvořeny 4 modely odstupňované podle složitosti řídicího programu a hardwarového modelu:

- **Příklad 1**

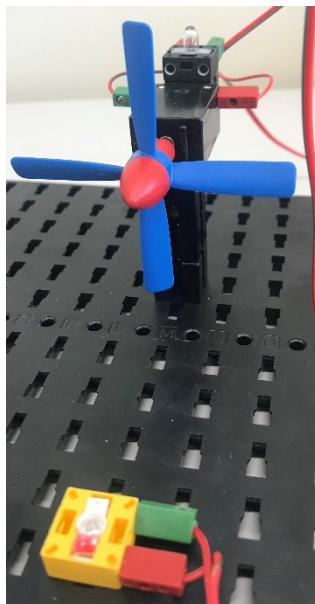
Základní příklad využívající pouze 2 tlačítka a LED. V řídicím kódu není žádná pokročilejší funkce, smyčka nebo myšlenka. Ukázka modelu je na následujícím obrázku, viz Obrázek 11-1.



Obrázek 11-1 – Ukázka modelu pro příklad 1

- **Příklad 2**

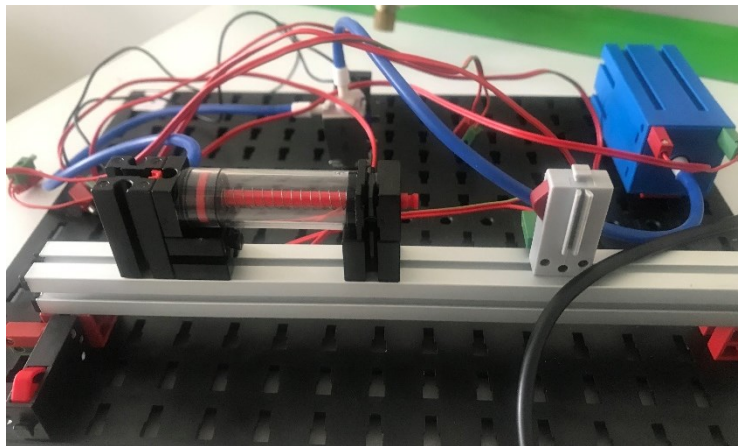
Pokročilejší příklad využívající motor, LED a fototranzistor. V řídicím kódu jsou dvě While smyčky. V jedné smyčce je spouštěný podprogram, v druhé smyčce je využito příkazu Continue. Ukázka modelu je na následujícím obrázku, viz Obrázek 11-2.



Obrázek 11-2 – Ukázka modelu pro příklad 2

- **Příklad 3**

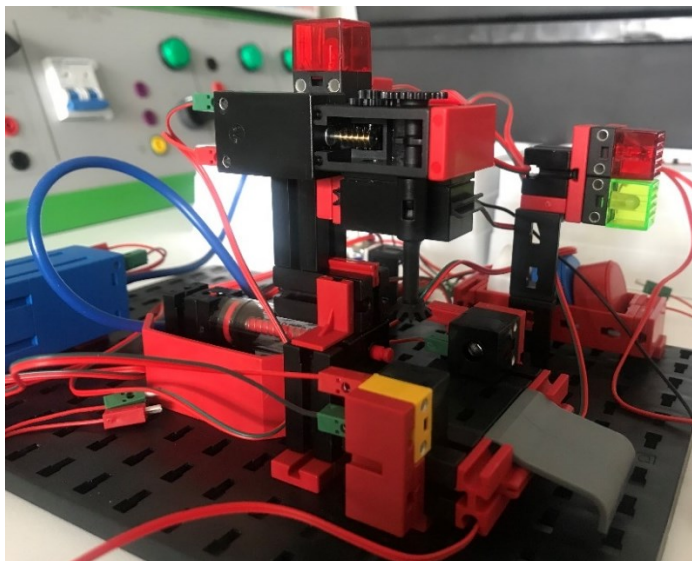
Pokročilejší příklad využívající dvě tlačítka, kompresor a solenoidový ventil ovládající píst. V řídicím kódu je podmínka IF, do které je vnořená druhá podmínka IF. V kódu je použito generování náhodných čísel. Ukázka modelu je na následujícím obrázku, viz Obrázek 11-3.



Obrázek 11-3 – Ukázka modelu pro příklad 3

- **Příklad 4**

Pokročilý příklad využívající motor, tlačítko, 3x LED, fototranzistor, kompresor, solenoidový ventil a optický barevný sensor. V řídicím kódu jsou 3 metody, využívají se cykly While, podmínky IF, IF – ELSE i IF – ELSE IF – ELSE, logické výrazy a generování náhodných čísel. Dvě metody jsou pro volání s parametry. V tomto případě se už jedná o velmi komplexní a složitý program představující řídicí kód pro automatizovanou frézku s automatickým odstraňováním zmetků, automatickým spouštěním obrábění a přizpůsobením obráběcího cyklu podle typu (barvy) obrobku. Ukázka modelu je na následujícím obrázku, viz Obrázek 11-4.



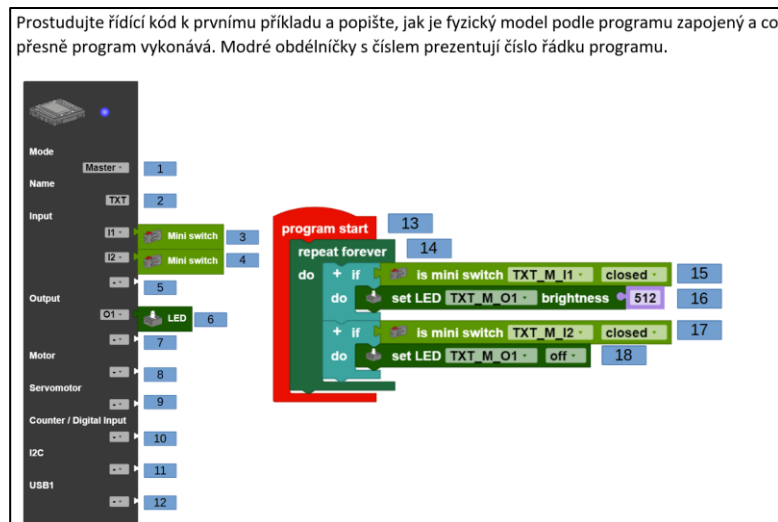
Obrázek 11-4 – Ukázka modelu pro příklad 4

Pro tyto příklady/modely byly stanoveny testovací otázky, které ověřovaly:

1. Porozumění řídicímu kódu a jeho návaznosti na hardwarový model s požadavkem na odvození fyzického zapojení hardwarového modelu a jeho chování s ohledem na předložený řídicí program.
2. Porozumění řídicímu kódu a jeho provázanosti na hardwarový model s požadavkem na přiřazení obrázku nejvhodnějšího hardwarového modelu předloženému řídicímu programu, včetně požadavku na vysvětlení, proč proband považuje právě daný hardwarový model za nejvíce odpovídající řídicímu kódu.
3. Schopnost navrhnout úpravu řídicího kódu podle požadované změny chování hw modelu s požadavkem na přímé označení očíslovaných řádků kódu, kde se má změna provést a jakým způsobem (např. formou pseudokódu).
4. Schopnost identifikovat místa v řídicím kódu, kam by bylo vhodné vložit metodu odesílající informaci o nastalé události, pokud je požadováno informování o naprosto každé události (u nejjednoduššího příkladu), nebo vybraných událostí (u nejkompaktnějšího příkladu), které v modelu nastanou nebo mohou nastat.
5. Na konci celého testu je otázka, jak daný proband vnímal náročnost testu. U této otázky není důležitý strávený čas, cílem je pouze mít možnost porovnat, jak byla probandy vnímána náročnost přiděleného testu/programovacího jazyka.

Hypotézy č. 1 a č.2 jsou zaměřeny na ověření, jestli navržená architektura training factory v maximální možné míře využívající vizuální programovací jazyk pro řízení hardwarového modelu zjednodušuje realizaci, modifikaci komplexního modelu obsahující HW model výrobního systému a zda snižuje čas potřebný průmyslovým inženýrem na úpravu řídicího HW modelu oproti dnes stále ještě standardnímu přístupu řízení hardware a hardwarových modelů, kterým je využíván programovací jazyk C/C++, popř. rychle se rozšiřující programovací jazyk Python. Z tohoto důvodu byly vytvořeny 3 druhy testů, a to pro vizuální programovací jazyk (vytvořený pomocí vývojového prostředí Robo Pro Coding), jazyk Python a jazyk C/C++. Všechny testy obsahují stejné příklady lišící se pouze programovacím jazykem, ve kterém je řídicí kód naprogramovaný. Pro každou otázku jsou řídicí kódy ve všech zvolených programovacích jazycích vytvořeny s důrazem na stejný způsob logiky, stavby, členění a struktury řídicího programu i složitosti (samozřejmě s ohledem na technická a vyjadřovací omezení daných programovacích jazyků).

Ukázka otázky č. 1.1., tedy 1. otázka základního příkladu pro vizuální programovací jazyk, je na následujícím obrázku, viz Obrázek 11-5. Kompletní testy ve formě .docx dokumentů jsou elektronickou přílohou této práce, viz soubory TEST\_CC++.docx, TEST\_Python.docx a TEST\_RoboProCoding.docx (více viz přehled elektronických příloh s jejich popisem je uvedený v příloze A této disertační práce). Způsob testování je popsán dále, viz kapitola 11.1.



Obrázek 11-5 – Ukázka otázky č. 1.1., tedy 1. otázka základního příkladu pro vizuální programovací jazyk

Pro otestování hypotézy č. 3 byl vytvořený dotazník zaměřený na znalosti a zkušenosti s technologiemi, které se využívají jak v této práci představené navržené architektuře pro realizaci hardwarového modelu, jeho řízení a umožnění komunikace s okolím, tak v architektuře training factory Industry 4.0 od společnosti Fischertechnik. Dotazník vyexportovaný do souboru typu PDF je elektronickou přílohou této práce, viz soubor Dotaznik\_Znalosti\_IT\_pro\_realizaci\_TF\_PI.pdf (více viz přehled elektronických příloh s jejich popisem je uvedený v příloze A této disertační práce).

Dotazník byl sestaven na základě tříčlenné expertní skupiny (stejná expertní skupina jako pro tvorbu testů využitých pro ověření hypotéz 1 a 2) a byl rozdělen do 7 skupin:

1. Zkušenosti s programovacími jazyky
2. Zkušenosti s vývojovými prostředími
3. Ostatní programátorské zkušenosti
4. Zkušenosti s uživatelskou prací v operačním systému
5. Zkušenosti se softwarovými nástroji
6. Zkušenosti s administrací SW nástrojů
7. Ostatní uživatelské zkušenosti

V rámci expertní skupiny byly pokryty tyto odbornosti: programování SW, programování HW, databáze a ukládání dat, diskretní simulace, optimalizační algoritmy a Industry 4.0. Všichni tři členové expertní skupiny mají dlouholeté zkušenosti se vzděláváním strojních a průmyslových inženýrů.

Každá skupina v dotazníku zjišťuje zkušenosti respondentů z oblasti programování a informatiky potřebné pro realizaci HW modelu s digitálním dvojčtem realizovaným podle navržené architektury a potřebné pro realizaci HW modelu s digitálním dvojčtem realizovaným dle architektury learning/training factory Fischertechnik Training Factory Industry 4.0 která využívá řídicí jednotky Fischertechnik TXT Controller. U každé možnosti je uvedeno, zda je daná možnost potřebná:

- pro řešení, které bylo vyvinuto v rámci disertační práce (M)
- pro řešení využívané konkurenčním řešením od Fischertechniku (F)
- nebo zda je potřeba pro oba přístupy (MF)



## 11.1 Způsob testování a dotazování

Pro ověření hypotéz č. 1 a č. 2 vzešly z jejich znění a ze struktury navržených testů následující požadavky:

### 1. Nemožnost vracet se k zodpovězeným otázkám.

Tento požadavek vychází ze stavby testovacích příkladů, ve kterých otázky následující po právě položené otázce zodpovídají nebo obsahují odpověď na právě řešenou otázku. Možností vracet se v rámci otázek by tedy byly podstatně ovlivněny odpovědi a nebylo by možné z odpovědi probandů vyvodit relevantní závěry. Tento požadavek prakticky vyřazuje možnost testu vyplňovaného na papír.

### 2. Měřit čas strávený probandem na dané otázce.

Tento požadavek vychází hlavně z hypotézy č. 2. Je potřeba zjistit, jak dlouhý časový úsek věnoval testovaný proband dané otázce. Tento požadavek také vyřazuje možnost testu vyplňovaného na papír z důvodu požadavku na další osobu, která by měřila čas strávený na každé otázce – tím by se podstatně prodloužil čas potřebný pro provedení testů. V případě samoměření probandem by pravděpodobně docházelo ke zkreslení časů samotnou dobou, kterou by musel proband např. práci se stopkami a zapisování času věnovat.

### 3. Měřit čas strávený probandem na dané otázce s přesností na vteřiny.

Tento požadavek vychází z předpokladu, že některé otázky v testech pro určité programovací jazyky by mohly být velmi rychle zodpověditelné. V případě měření odpovědi s přesností na minuty by pak mohlo dojít k zastření nebo minimálně k podstatnému zkreslení rozdílů jak mezi jednotlivými probandy, tak hlavně mezi typy jednotlivých testů pro vizuální programovací jazyk, Python nebo C/C++.

### 4. Nezobrazovat probandu čas strávený na otázce ani celém testu.

Tento požadavek vychází z vytvořených testů a základního přístupu k nim. Cílem není udělat test co nejrychleji, ale co nejlépe. Po probandovi je požadováno pochopit řídicí kód v rámci jeho možností i znalostí a zodpovědět otázky co nejlépe a nej přesněji. Není žádoucí dostat probanda do časové tísně nebo jej vysloveně podporovat k co nejrychlejšímu vyplnění bez ohledu na kvalitu řešení. Délka testu není nijak omezena.

### 5. Náhodně přidělovat variantu testů (Vizuální programovací jazyk, Python a C/C++) s ohledem na v rámci možností rovnoměrný počet vyplněných testů.

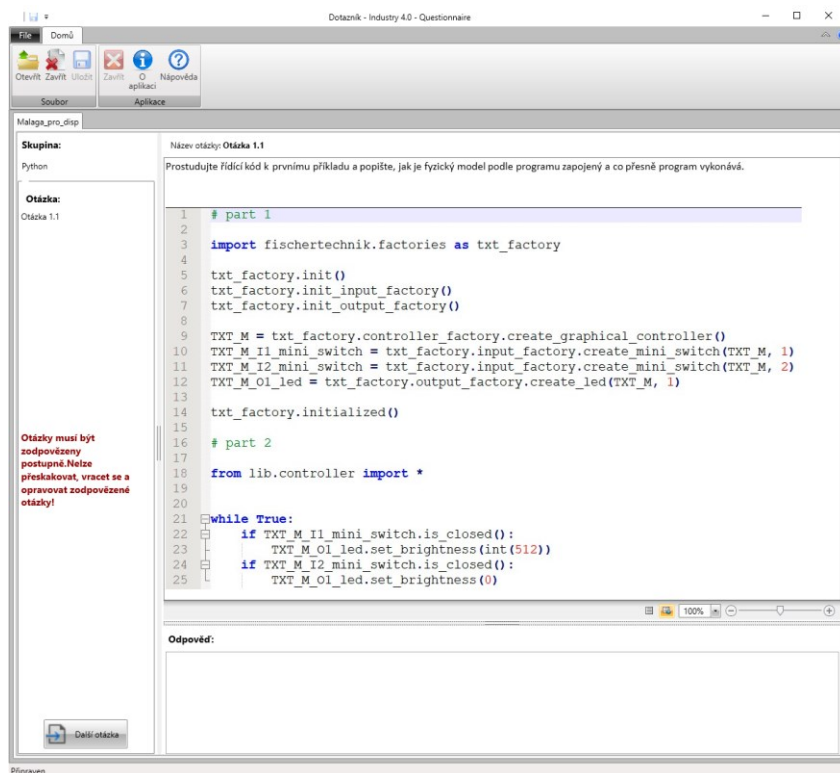
Náhodné přidělování testu vychází z požadavku nemožnosti ovlivnění přiděleného testu probandem např. z toho důvodu, že některý test/programovací jazyk považuje proband např. za jednodušší nebo časově méně náročný. Rovnoměrné rozdělování testů mezi probandy pak zajistí stejný nebo alespoň srovnatelný počet druhů vyplněných testů.

### 6. Testovací software by měl být volně dostupný, popř. by měla univerzita disponovat licenci.

Protože hlavně požadavky číslo 1 a 2 prakticky vyřazují možnost testování formou papírového provedení, vzniká zde požadavek na testovací software, který nejen že výše uvedené požadavky splňuje, ale je i licenčně dostupný.

Protože nebyl nalezen testovací software, který by splňoval výše uvedené požadavky, byl vytvořen, otestován a následně k testování využit testovací program Questionnaire, který vše výše uvedené splňuje. Jedná se o databázovou aplikaci s grafickým rozhraním. Aplikace je vytvořena jako WPF/C#.NET aplikace využívající ORM EntityFramework pro tvorbu a správu databáze a komunikaci s ní. Jako DB server byl pro aplikaci využit server s MS SQL serverem. Aplikace Questionnaire se instaluje pouze do uživatelského profilu, a tedy není potřeba administrátorských oprávnění. Ukázka aplikace v

režimu testování s otázkou 1.1 pro programovací jazyk Python je na následujícím obrázku, viz Obrázek 11-6.

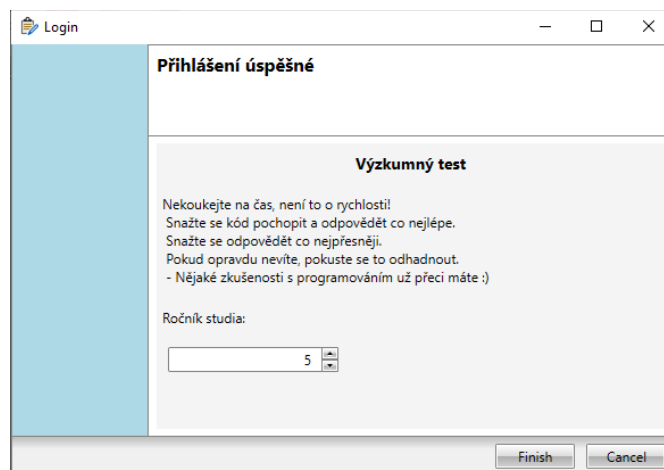


Obrázek 11-6 – Ukázka aplikace Questionnaire v režimu testování pro programovací jazyk Python

Testování probíhalo v počítačových učebnách katedry průmyslového inženýrství a managementu na Fakultě strojní Západočeské univerzity v Plzni. Probandi byli před spuštěním testu vždy poučeni formou krátké prezentace s výkladem v délce 10 minut. V rámci prezentace byly zdůrazněny tyto body:

- Práce s aplikací Questionnaire.
- Upozornění na nemožnost se vracet v otázkách.
- Nesnažit se vyplnit test co nejrychleji, ale co nejlépe.
- Test není nijak časově omezený.
- Test se přiděluje automaticky, proband může dostat test na jazyk C/C++, Python nebo Robo Pro Coding.
- Pokud proband nebude vědět, měl by se na základě svých znalostí a zkušeností snažit odpovědět co nejlépe. Neměl by se bát špatné odpovědi.

Následně byli ještě stručně poučeni i aplikací Questionnaire při přihlášení do testu, viz Obrázek 11-7.



Obrázek 11-7 – Ukázka poučení při přihlášení do testu v aplikaci Questionnaire

Pro ověření hypotézy č. 3, která je postavena jako dotazníkové šetření, byl použit nástroj Google Forms. Tento software je v rámci univerzity k dispozici pro všechny studenty a pracovníky a výhodou je online forma aplikace, která umožňuje probandům vyplnit dotazník kdykoliv a odkudkoliv. Ukázka náhledu části dotazníku je na následujícím obrázku, viz Obrázek 11-8.

Jaké máte znalosti o této technologii? *	Nikdy jsem o něm neslyšel	Vím, že existuje, ale nikdy jsem ho nepoužil	Dokážu naprogramovat jednoduchý program/skript	Dokážu naprogramovat složitější program/skript	Považuji se za zkušeného programátora
Jakýkoliv vizuální programovací jazyk (RoboPro Fischart, MakeCode, Scratch,...) (M)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C# (M)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C (F)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C++ (F)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SQL (MF)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
EntityFramework (M)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Python (MF)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Obrázek 11-8 – Ukázka náhledu části dotazníku v Google Forms

## 11.2 Průběh bodového hodnocení odpovědí pro H1 a H2

V rámci testů vytvořených pro hypotézy č. 1 a č. 2 byly všechny otázky otevřené a testování probandů proběhlo pomocí softwarového nástroje Questionnaire. Ohodnocení, tedy obodování jednotlivých otázek, bylo stanoveno v bodovém rozsahu 0–100, kde 0 bodů je žádná nebo naprosto mylná odpověď a 100 bodů je zcela správná, kompletní odpověď. Rozsah bodů 0–100 byl zvolen s ohledem na případný jednoduchý a přesný převod na procenta bez vytvoření zkreslení, např. zaokrouhlováním a možností podchycení a zohlednění i malých rozdílů v úrovni správnosti a kvality odpovědí. Hodnocení odpovědí proběhlo nezávisle třemi odborníky (stejní odborníci kteří sestavovali

dotazník) a poté následovalo osobní setkání této expertní skupiny hodnotitelů. U odpovědí, při kterých se jejich hodnocení neshodovalo, proběhla diskuse s cílem dosáhnout konsensu v počtu udělených bodů dané odpovědi. Při bodování odpovědí nebylo přihlíženo k typu a náročnosti programovacího jazyka v testu, pouze ke správnosti a kvalitě odpovědi. Tedy pokud 3 probandi, každý s jiným typem testu, měli stejně správnou a kvalitní odpověď, získali stejný počet bodů.

### 11.3 Výzkumný vzorek

S ohledem na zaměření disertační práce jsou populací neboli základním výzkumným vzorkem studenti oboru průmyslové inženýrství nebo jeho obdoby. Během průběhu stanovování výzkumného vzorku bylo zjištěno, že je hodně blízkých oborů průmyslovému inženýrství hlavně na ekonomických fakultách v rámci České republiky. U těchto studentů se předpokládá jiná základna znalostí, proto jsou v rámci této práce z výzkumného vzorku studentů takových oborů vyřazeni. Do budoucna by ale bylo vhodné provést stejné testování i na nich a zjistit tak vhodnost navržené architektury pro průmyslové inženýry jiného než strojírenského typu a jejich porovnání právě s těmito studenty.

Testování a dotazníkové šetření proběhlo na studentech průmyslového inženýrství, kteří byli v akademickém roce 2021/2022 konci 1. ročníku/na začátku 2. ročníku navazujícího magisterského studia a v akademickém roce 2022/2023 na konci 1. ročníku/na začátku 2. ročníku navazujícího magisterského studia (testování probíhalo před letními prázdninami a případně po letních prázdninách pro ty co se nemohli účastnit před létem). Díky šetření ve dvou letech po sobě se podařilo zvýšit počet otestovaných průmyslových inženýrů, kde všichni byli v podobné fázi studia a znalostí v rámci svého studia průmyslového inženýrství. Na konci prvního ročníku, resp. na začátku 2. ročníku navazujícího magisterského studia se pak dá očekávat orientace v odbornosti a po rešerši studijních plánů už ve druhém ročníku nemají zásadní informatické předměty. Také se dá předpokládat, že danou architekturu mohou využívat ve 2. ročníku navazujícího magisterského studia pro své diplomové práce a následně ve svém dalším pracovním působení nebo při své vědecké kariéře. Bylo zjištěno, že za tyto dva roky se jednalo o 146 potenciálních průmyslových inženýrů vhodných k testování, viz Tabulka 11-1. Z tohoto počtu je viditelné, že se jedná o malou skupinu. Nebylo rozlišováno, zda se jedná o studenty denního studia, nebo studenty kombinovaného studia.

Tabulka 11-1 – Velikost populace, na kterou se zaměřuje výzkum

Univerzita	Fakulta	Katedra	Studijní obor	Počet studentů 5. ročníku 2021/2022	Počet studentů 5. ročníku 2022/2023	Celkový počet studentů 5. ročníku za oba roky
VŠB – Technická univerzita Ostrava	Fakulta strojní	Katedra mechanické technologie	Průmyslové inženýrství	27	13	40 <sup>2</sup>
TUL – Technická univerzita v Liberci	Fakulta strojní	Katedra výrobních systémů a automatizace	Inovační a průmyslové inženýrství	6	9	15 <sup>3</sup>
ZČU – Západočeská univerzita v Plzni	Fakulta strojní	Katedra průmyslového inženýrství a managementu	Průmyslové inženýrství a management	25	20	45
ČVUT – České vysoké učení technické v Praze	Fakulta strojní	Ústav řízení a ekonomiky podniku	Řízení průmyslových systémů	24	22	46 <sup>4</sup>
					<b>Celkem</b>	<b>146</b>

Pro experimenty, tedy testování schopností tvorby a úprav řídicích programů pro HW model, byl zvolen počet 30 studentů průmyslového inženýrství. Tato hodnota byla stanovena expertním odhadem jako výchozí s ohledem na počty dostupných studentů vhodných pro zařazení do výzkumného vzorku, protože se porovnávají výsledky mezi jednotlivými jazyky (C/C++, Python a Robo Pro Coding jako zástupce vizuálních programovacích jazyků) a nebylo možné dopředu odhadnout velikost rozptylu u jednotlivých programovacích jazyků.

K vyplnění dotazníků bylo osloveno 33 studentů oboru průmyslového inženýrství. Po oslovení jednotlivých kateder českých univerzit, kde takové obory existují, byla velikost základního souboru, tj. celkový počet takových studentů, stanovena na 146 osob. Výběrový vzorek průmyslových inženýrů, s kterým bylo pracováno v rámci tohoto dotazníkového šetření, tak čítá 22,6 % základního souboru, což je pro potřeby disertační práce s ohledem na velikost populace považováno za dostatečné a reprezentativní množství.

<sup>2</sup> Informace poskytli pan prof. Ing. Radek Čada, CSc., vedoucí katedry mechanické technologie (FS VŠB) a paní Ing. Ivany Šajdlerové, Ph.D., tajemnice katedry mechanické technologie (FS VŠB)

<sup>3</sup> Informace poskytl pan Ing. František Koblasa, Ph.D., OA katedry výrobních systémů a automatizace (FS TUL)

<sup>4</sup> Informace poskytl pan Ing. Miroslav Žilka, Ph.D., vedoucí ústavu řízení a ekonomiky podniku (FS ČVUT)

## 12 Vyhodnocení a výsledky výzkumu

V rámci této kapitoly je popsána metodika statistického posouzení hypotéz a jejich vyhodnocení. Data pro vyhodnocení jsou el. přílohou této disertační práce, viz soubory Testy\_data.xlsx pro H1 a H2 a Dotaznik\_odpovedi.xlsx pro H3 (více viz přehled elektronických příloh s jejich popisem, který je uvedený v příloze A této disertační práce).

### 12.1 Metodika statistického vyhodnocení

Statistické vyhodnocení dat bylo realizováno v programu RStudio verze 2022.12.0+353 s jazykem R ve verzi 4.2.3. Statistické testy byly vždy vyhodnocovány na hladině významnosti 5 %. Rozhodnutí o zamítnutí nebo nezamítnutí jejich nulové hypotézy bylo vždy činěno na základě  $p$ -hodnoty, která pro zamítnutí musela být nižší než hladina významnosti [145].

Kromě statistických testů byly pro základní orientaci v datech vypočítány některé popisné statistiky jako např. absolutní četnosti nebo charakteristiky polohy datového souboru (průměr, medián, 1. a 3. kvartil, minimum a maximum). V případě identifikace odlehlých pozorování byla tato pozorování odstraněna.

#### 12.1.1 Vyhodnocení normality

Pro zvolení správného statistického testu bylo třeba ověřit předpoklad normality, který parametrické testy vyžadují. Tento předpoklad byl otestován pomocí Anderson-Darlingova testu normality [146]. Výsledky testu byly následně zkontrolovány graficky pomocí histogramů. Ve všech případech (data pro H1, H2 i H3) byla zamítnuta normalita. Z toho důvodu byly v rámci celého vyhodnocení používány neparametrické testy, které tento předpoklad nevyžadují.

#### 12.1.2 Testování tří a více skupin a mnohonásobné porovnávání

V případě testování rozdílů mezi třemi a více skupinami byl použit Kruskal-Wallisův test [145]. Nulová hypotéza tohoto testu předpokládá, že měření ve skupinách mají stejné mediány, tedy neliší se od sebe. Formálně jsou hypotézy tohoto testu stanoveny následovně:

1.  $H_0$ : Mediány všech souborů dat jsou stejné.
2.  $H_1$ : Mediány alespoň dvou souborů dat jsou odlišné.

Tento test byl využit např. v případě H1 a H2 pro testování rozdílů mezi výsledky pro tři programovací jazyky, tedy C/C++, Python a Robo Pro Coding. V případě třetí hypotézy byl využit pro testování rozdílů hodnocení zkušeností tří a více nástrojů/technologií.

Při zamítnutí nulové hypotézy Kruskal-Wallisova testu je třeba zjistit, které dvojice skupin se od sebe statisticky odlišují. K tomu je určena skupina post-hoc testů pro tzv. mnohonásobné testování, kdy jsou testovány všechny dvojice skupin zvlášť. Pro tento účel byl použit Dunnův test s Holmovo korekcí [147]. Jedná se o neparametrický test srovnávající mediány, který se používá v post-hoc testech pro identifikaci skupin, které se od sebe statisticky významně odlišují.

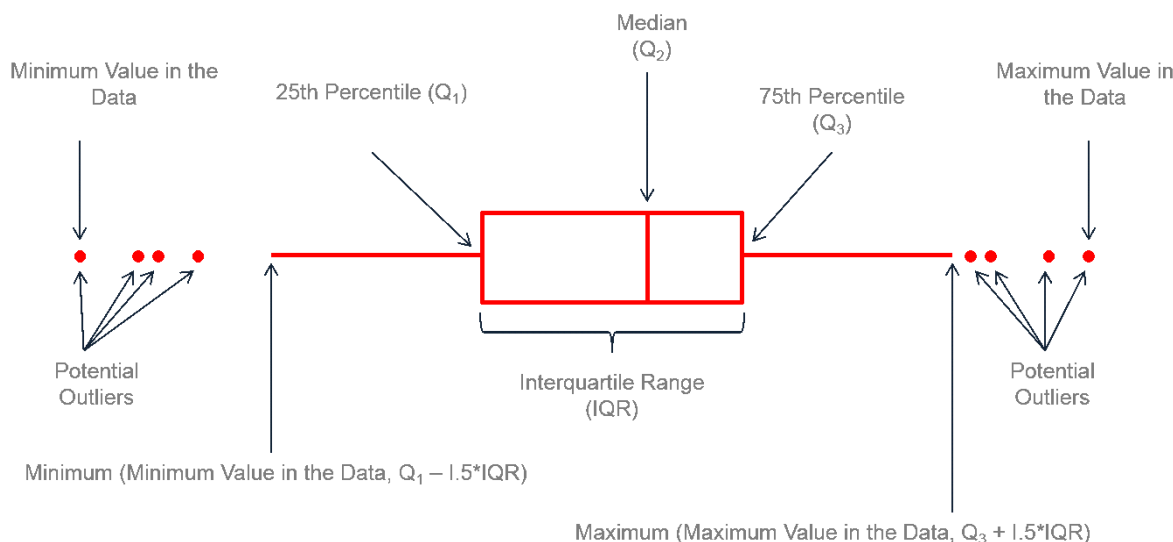
#### 12.1.3 Testování dvou skupin

V případě testování pouze dvou skupin byl použit Wilcoxonův test [145]. Wilcoxonův test byl využit u H3, kde se zjišťovaly znalosti a zkušenosti se dvěma operačními systémy. Tento neparametrický test je zjednodušením Kruskal-Wallisova testu, kde jsou proti sobě testovány pouze dva mediány. Test má stanovené hypotézy následovně:

- $$H_0: \text{Mediány obou souborů dat jsou stejné.}$$
- $$H_1: \text{Mediány obou souborů dat jsou odlišné.}$$

### 12.1.4 Grafické srovnání skupin

Pro grafické srovnání výsledků několika skupin dat byly použity tzv. krabicové grafy/ boxploty [145]. S jejich pomocí lze porovnat rozložení dat včetně jejich rozptýlenosti a jsou vhodné i pro identifikaci odlehlých pozorování. Základní krabice boxplotu obsahuje 50 % dat a je rozdělena mediánem na dvě části. Další podrobnosti ke konstrukci boxplotů lze vidět na následujícím obrázku, viz Obrázek 12-1.



Obrázek 12-1 – Popis boxplotu – krabicového grafu [148]

## 12.2 Příklady na programování HW – test

Na základě testu ověřujícího znalosti průmyslových inženýrů prostřednictvím vzorových příkladů s použitím tří různých programovacích jazyků (C/C++, Python, Robo Pro Coding) byly vyhodnocovány hypotézy H1 a H2. Výsledky jsou rozděleny podle jednotlivých okruhů otázek (příklady č. 1 – 4).

V jednotlivých kapitolách byly pro testování statistických hypotéz použity neparametrické testy, protože u testovaných dat byla na základě testu normality a vyhodnocení histogramu zamítnuta normalita dat na hladině významnosti 5 %.

Pro vyhodnocení hypotéz H1 a H2 byl testován vliv využitého programovací jazyka na následující veličiny:

- procentuální hodnocení, tedy na kolik procentních bodů respondent splnil zadání
- čas, který tímto řešením strávil
- dopočtená hodnota z výše uvedených =  $\frac{\% \text{ hodnocení}}{\text{čas}}$ , tedy procentuální hodnocení přepočtené na 1 minutu. Toto lze vnímat i jako počet získaných bodů za jednotku času, přičemž max. množství bodů bylo 100.

### 12.2.1 Okruh otázek k příkladu č. 1

Pro vyhodnocení okruhu otázek k příkladu č. 1 byly použity výsledky otázek 1.1–1.6, ze kterých bylo vypočítáno průměrné % hodnocení a průměrný čas vynaložený pro zodpovězení otázky.

#### 12.2.1.1 Hodnocení splnění úkolů

Následující tabulka, viz Tabulka 12-1, ukazuje základní charakteristiky pro procentuální hodnocení dle posuzovaných programovacích jazyků. Lze z ní vidět, že v průměru dosahovali respondenti nejlepších výsledků v případě technologie Robo Pro Coding, kde je průměrné hodnocení 68,25 %. U

jazyka C/C++ je to pouze 25,58 % a u Pythonu 48,25 %. Zajímavý je pohled i na minimální hodnoty a hodnoty 1. kvartilu. Z minimálních hodnot lze vidět, že u programovacího jazyka Robo Pro Coding byla tato hodnota 30 %, zatímco u ostatních dvou to bylo 0 %, tedy našli se tací probandi, kteří tento úkol v C/C++ a v Pythonu nedokázali vůbec vyřešit. V případě těchto dvou programovacích jazyků je také velmi nízké hodnocení spodních 25 % respondentů, v případě C/C++ to je 17,71 % a v případě Pythonu 18,33 %. U Pythonu a C/C++ tedy 25 % respondentů nedosáhlo v hodnocení ani 20 %. V případě Robo Pro Coding je hodnota 1. kvartilu 65,42 %, tedy 75 % respondentů dosáhlo minimálně tohoto hodnocení a spodních 25 % mělo hodnocení mezi 30 % a 65,42 %. I z hodnot 3. kvartilu a maxima je viditelné, že respondenti dosahovali nejlepších výsledků v případě programovacího jazyka Robo Pro Coding.

Tabulka 12-1 – Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 1

Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,00	17,71	20,00	25,58	40,42	50,83
<b>Python</b>	0,00	18,33	60,42	48,25	69,58	95,00
<b>Robo Pro Coding</b>	30,00	65,42	73,33	68,25	76,25	95,83

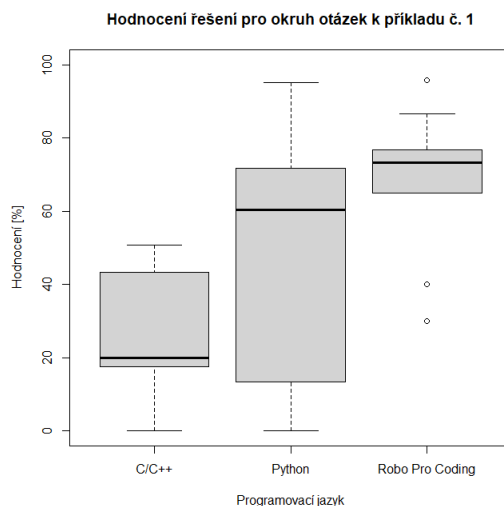
Na základě Kruskal-Wallisova testu byla zamítnuta nulová hypotéza o shodě distribucí 3 testovaných datových sad na hladině významnosti 5 % (p-hodnota: 0,0052), tedy existuje statisticky významný rozdíl alespoň mezi 2 programovacími jazyky. Pro identifikaci, mezi kterými jazyky je tento významný statistický rozdíl, byla využita metoda mnohonásobného porovnávání. Výsledky metody mnohonásobného porovnávání (p-hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-2. Lze vidět, že je p-hodnota <5 % pouze v případě dvojice C/C++ a Robo Pro Coding, kde byl tímto prokázán statisticky významný rozdíl a lze tvrdit, že respondenti v 1. okruhu otázek dosáhli v Robo Pro Coding statisticky významně vyššího hodnocení než při použití jazyka C/C++. Naopak mezi jazykem Python a dalšími dvěma jazyky nebyl ani v jednom případě prokázán statisticky významný rozdíl.

Tabulka 12-2 – P-hodnoty kombinací způsobů programování pro hodnocení řešení úkolů k příkladu č. 1

p-hodnota dané kombinace	C/C++	Python
<b>Python</b>	0,1397	--
<b>Robo Pro Coding</b>	0,0039	0,1397

Pro grafické znázornění výsledků jsou zobrazeny distribuce dat v boxplotech, viz Obrázek 3-1. Lze vidět, že výsledky testu v jazyce Robo Pro Coding jsou výrazně lepší než v případě zbylých dvou využívaných programovacích jazyků. Největší rozpětí hodnot lze vidět v případě jazyka Python, kde právě toto velké rozpětí v kombinaci se vzorkem o velikosti 10 respondentů mohlo způsobit zamítnutí rozdílu mezi výsledky v něm a v dalších dvou programovacích jazycích.





Obrázek 12-2 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 1

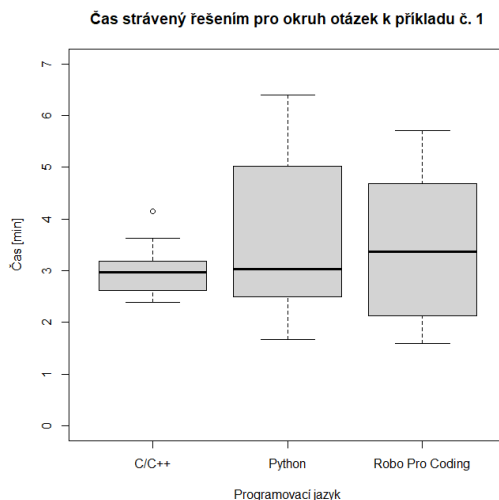
#### 12.2.1.2 Čas strávený řešením úkolů

Následující tabulka, viz Tabulka 12-3, ukazuje základní charakteristiky pro čas, který respondenti strávili s plněním úkolů, rozdělené dle posuzovaných programovacích jazyků. Lze vidět, že průměrné časy se zde výrazně neliší, v průměru strávili respondenti řešením úkolu nejvíce času při využití Pythonu, následně v Robo Pro Coding a nejméně času strávili u řešení úkolů v jazyce C/C++. Toto pořadí se příliš neliší ani v jiných charakteristikách s výjimkou minimálního času.

Tabulka 12-3 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 1

Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	2,386	2,695	2,971	3,043	3,143	4,154
<b>Python</b>	1,668	2,568	3,026	3,541	4,659	6,393
<b>Robo Pro Coding</b>	1,601	2,297	3,365	3,419	4,509	5,712

Na základě Kruskal-Wallisova testu nebyla zamítnuta nulová hypotéza o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % ( $p$ -hodnota: 0,8489), tedy neexistuje statisticky významný rozdíl alespoň mezi dvěma testovanými programovacími jazyky. Výsledky lze rovněž vidět graficky znázorněné v následujících boxplotech, viz Obrázek 12-3.



Obrázek 12-3 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 1

### 12.2.1.3 Časová efektivita splnění úkolů

Následující tabulka, viz Tabulka 12-4, ukazuje základní charakteristiky pro procentuální hodnocení přepočtené na 1 minutu stráveného času dle posuzovaných programovacích jazyků. Lze vidět, že po přepočtu dosahovali respondenti nejlepších výsledků při použití jazyka Robo Pro Coding, kdy za jednu minutu vyřešili v průměru 22,1 % úloh, zatímco v případech ostatních jazyků to bylo méně. V případě C/C++ to bylo pouze 9,06 %. Je zde tedy vidět, že ačkoliv respondenti řešili úlohy v C/C++ rychleji než v Pythonu nebo Robo Pro Coding (viz předchozí kapitola), neřešili tyto úlohy správně. Časově nejefektivnější je tedy z tohoto pohledu Robo Pro Coding.

Tabulka 12-4 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 1

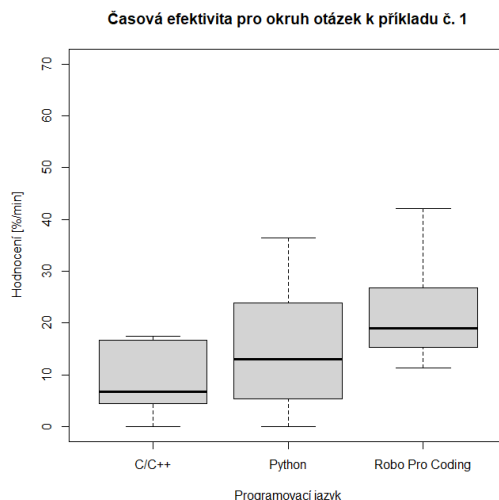
Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,000	4,810	6,731	9,058	15,863	17,491
<b>Python</b>	0,000	6,470	12,970	14,740	21,450	36,460
<b>Robo Pro Coding</b>	11,380	16,160	19,010	22,050	26,630	42,070

Na základě Kruskal-Wallisovo testu byla zamítnuta nulová hypotéza o shodě distribucí 3 testovaných datových sad na hladině významnosti 5 % (p-hodnota: 0,0167), tedy existuje statisticky významný rozdíl alespoň mezi 2 programovacími jazyky. Pro identifikaci, mezi kterými konkrétně, byla využita metoda mnohonásobného porovnávání, jejíž výsledky (p-hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-5. Z té lze vidět, že statisticky významný rozdíl byl na hladině významnosti 5 % potvrzen v případě dvojice C/C++ a Robo Pro Coding.

Tabulka 12-5 – P-hodnoty kombinací způsobů programování pro hodnocení časovou efektivitu k příkladu č. 1

p-hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,4720	-
<b>Robo Pro Coding</b>	0,0045	0,1816

Výsledky lze vidět i graficky znázorněné na následujících boxplotech, viz Obrázek 12-4. Lze vidět, že v případě jazyka Robo Pro Coding je přepočtené % skóre vyšší než v případě C/C++ i Pythonu.



Obrázek 12-4 – Boxplot zobrazující časovou efektivitu k příkladu č. 1

### 12.2.2 Okruh otázek k příkladu č. 2

Pro vyhodnocení okruhu otázek k příkladu č. 2 byly použity výsledky otázek 2.1–2.3, ze kterých bylo vypočítáno průměrné procentuální hodnocení a průměrný čas vynaložený pro zodpovězení otázky.

#### 12.2.2.1 Hodnocení splnění úkolů

Následující tabulka, viz Tabulka 12-6, ukazuje základní charakteristiky pro procentuální hodnocení dle posuzovaných programovacích jazyků. Průměrný procentuální výsledek je nejvyšší v případě programovacího jazyka Robo Pro Coding, stejně tak jako hodnota mediánu. Tedy 50 % respondentů měla v případě použití Robo Pro Coding 75 % a více bodů. V případě Pythonu je hodnota mediánu 58,3 % a u jazyka C/C++ 15,8 %, což je méně než minimální procentuální výsledek v případě Robo Pro Coding. Maximální hodnota v případě C/C++ dosahuje 66,7 %, zatímco u Pythonu je to 96,7 % a v případě Robo Pro Coding dokonce 100 %.

Tabulka 12-6 – Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 2

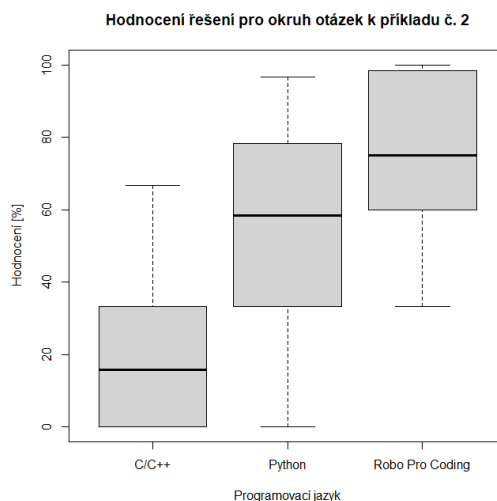
Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,00	0,00	15,83	19,50	32,50	66,67
<b>Python</b>	0,00	35,83	58,33	52,17	77,08	96,67
<b>Robo Pro Coding</b>	33,33	61,67	75,00	74,67	96,25	100,00

Na základě Kruskal-Wallisovo testu byla zamítnuta nulová hypotéza o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % (p-hodnota: 0,0015). Tedy existuje statisticky významný rozdíl alespoň mezi dvěma programovacími jazyky. Pro identifikaci, mezi kterými konkrétně, byla využita metoda mnohonásobného porovnávání, jejíž výsledky (p-hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-7. Lze vidět, že p-hodnota je nižší než hladina významnosti 5 % v případě dvojic C/C++; Python a C/C++; Robo Pro Coding. Statisticky významný rozdíl nebyl identifikován v případě dvojice programovacích jazyků Python a Robo Pro Coding.

Tabulka 12-7 – P-hodnoty kombinací způsobů programování pro hodnocení řešení úkolů k příkladu č. 2

p-hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,0431	--
<b>Robo Pro Coding</b>	0,0021	0,1119

Pro grafické znázornění výsledků jsou zobrazeny distribuce dat v boxplotech, viz Obrázek 12-5. Lze vidět, že nejvyšší procentuální hodnocení dosahovali respondenti s použitím programovacího jazyka Robo Pro Coding, nejnižší naopak v případě jazyka C/C++. Distribuce procentuálního hodnocení úkolů řešených v Pythonu má stejně jako v případě okruhu úkolů k příkladu č. 1 největší rozpětí hodnot, ovšem i z boxplotů lze vidět, že v tomto případě jsou hodnoty více vzdáleny od C/C++, což potvrdily i statistické testy.



Obrázek 12-5 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 2

#### 12.2.2.2 Čas strávený řešením úkolů

Následující tabulka, viz Tabulka 12-8, ukazuje základní charakteristiky pro čas, který respondenti strávili s plněním úkolů, rozdělené dle posuzovaných programovacích jazyků. Z té lze vidět, že v případě 2. okruhu otázek se již časy v závislosti na programovacích jazycích lišily. Průměrný čas strávený u úkolů byl v případě C/C++ 1,9 minuty, u Pythonu to bylo 2,6 minuty a nejvíce času trávili při využití Robo Pro Coding, tj. 3,3 minuty. Ostatní charakteristiky kopírují toto pořadí.

Tabulka 12-8 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 2

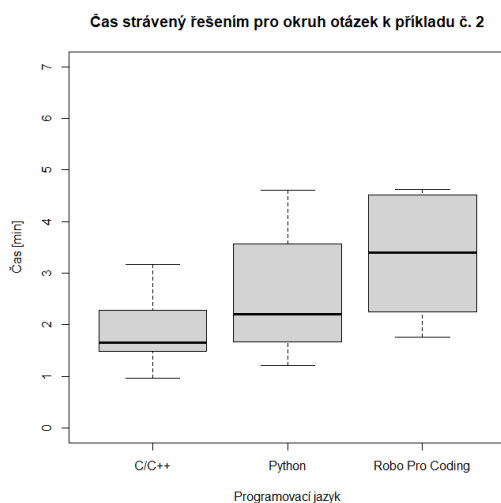
Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,963	1,482	1,6523	1,899	2,280	3,171
<b>Python</b>	1,205	1,737	2,200	2,620	3,423	4,608
<b>Robo Pro Coding</b>	1,759	2,337	3,396	3,318	4,372	4,629

Na základě Kruskal-Wallisovo testu byla zamítnuta nulová hypotéza o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % (p-hodnota: 0,0211). Byl tedy potvrzen statisticky významný rozdíl alespoň mezi dvěma testovanými programovacími jazyky. Pro identifikaci, mezi kterými jazyky je tento významný statistický rozdíl, byla využita metoda mnohonásobného porovnávání. Výsledky (p-hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-9. Lze vidět, že p-hodnota je nižší než hladina významnosti 5 % v případě dvojice C/C++ vs Robo Pro Coding. V případě dalších dvou dvojic programovacích jazyků nebyl potvrzen statisticky významný rozdíl mezi nimi.

Tabulka 12-9 – P-hodnoty kombinací způsobů programování čas strávený řešením úkolů k příkladu č. 2

p-hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,165	--
<b>Robo Pro Coding</b>	0,017	0,165

Pro grafické znázornění výsledků jsou zobrazeny distribuce dat v následujících boxplotech, viz Obrázek 12-6. Z boxplotů je viditelné, že v jazyce C/C++ byl čas k řešení výrazně nižší než u jazyka Robo Pro Coding.



Obrázek 12-6 – Boxplot zobrazující čas strávený řešením úkolů k příkladu č. 2

### 12.2.2.3 Časová efektivita splnění úkolů

Následující tabulka, viz Tabulka 12-10, ukazuje základní charakteristiky pro procentuální hodnocení přepočtené na 1 minutu stráveného času rozdělené dle posuzovaných programovacích jazyků. Stejně jako v případě 1. okruhu otázek i zde lze vidět, že v průměru nejlepších výsledků dosahovali respondenti s použitím jazyka Robo Pro Coding, konkrétně 26,1 %/min. Ačkoliv jim tedy řešení úloh trvalo nejdelší dobu, viz předchozí kapitola, bylo řešení v Robo Pro Coding časově nejefektivnější. V případě jazyka C/C++, ve kterém strávili respondenti nejméně času, dokázali respondenti v přepočtu na minutu získat pouze 6,221 %.

Tabulka 12-10 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 2

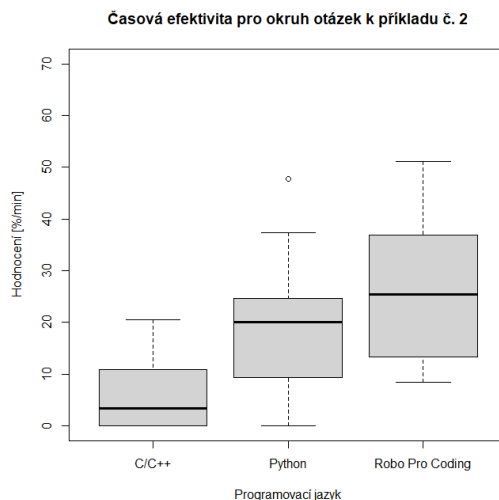
Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,000	0,000	3,373	6,221	10,852	20,469
<b>Python</b>	0,00	11,39	20,10	19,98	24,28	47,74
<b>Robo Pro Coding</b>	8,469	14,822	25,482	26,114	35,575	51,164

Na základě charakteristik lze vidět rozdíly, které potvrdil i výsledek Kruskal-Wallisovo testu, který zamítl nulovou hypotézu o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % ( $p$ -hodnota: 0,0094). Tedy byl potvrzen statisticky významný rozdíl alespoň mezi dvěma testovanými programovacími jazyky. Pro identifikaci, mezi kterými jazyky konkrétně, byla využita metoda mnohonásobného porovnávání, jejíž výsledky ( $p$ -hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-11. Z té lze vidět, že statisticky významný rozdíl je mezi C/C++ a Robo Pro Coding, kde je  $p$ -hodnota menší než hladina významnosti 5 % V případě dalších dvojic nebyl na této hladině významnosti potvrzen statisticky významný rozdíl.

Tabulka 12-11 –  $P$ -hodnoty kombinací způsobů programování pro hodnocení časovou efektivitu k příkladu č. 2

$p$ -hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,052	--
<b>Robo Pro Coding</b>	0,011	0,427

Výsledky lze vidět i graficky znázorněné na následujících boxplotech, viz Obrázek 12-7, které rovněž potvrzují závěry předchozích výpočtů. Lze vidět, že v případě jazyka C/C++ je přepočtené procentuální skóre nižší než při použití jazyků Python a Robo Pro Coding. V případě dvojice C/C++ a Robo Pro Coding je tento rozdíl výrazný, jak potvrdil i Kruskall-Wallisův test.



Obrázek 12-7 – Boxplot zobrazující časovou efektivitu k příkladu č. 2

### 12.2.3 Okruh otázek k příkladu č. 3

Pro vyhodnocení okruhu otázek k příkladu č.3 byly použity výsledky otázek 3.1–3.3, ze kterých bylo vypočítáno průměrné procentuální hodnocení a průměrný čas vynaložený pro zodpovězení otázky.

#### 12.2.3.1 Hodnocení splnění úkolů

Následující tabulka, viz Tabulka 12-12, ukazuje základní charakteristiky pro procentuální hodnocení dle posuzovaných programovacích jazyků. Z vypočítaných hodnot lze na první pohled vidět významný rozdíl mezi programovacím jazykem C/C++ a zbylými dvěma jazyky. V případě C/C++ měla dle mediánu polovina studentů 0 bodů a průměrné hodnocení pak bylo 5,3 %. U Pythonu byla průměrná hodnota 54,3 %, ale dle minima lze vidět, že i zde byl některý respondent, který z této oblasti nevyřešil žádný z úkolů. U Robo Pro Coding je průměrné hodnocení rovno 80,2 %, polovina respondentů dosáhla alespoň 83,3 % a nejnižší procentuální hodnocení je u tohoto programovacího jazyka 41,7 %. V případě Robo Pro Coding tedy každý respondent dokázal vyřešit úlohy alespoň částečně.

Tabulka 12-12 – Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 3

Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,000	0,000	0,000	5,333	1,667	36,667
<b>Python</b>	0,00	15,83	69,17	54,33	79,17	100,00
<b>Robo Pro Coding</b>	41,67	68,75	83,33	80,17	95,83	100,00

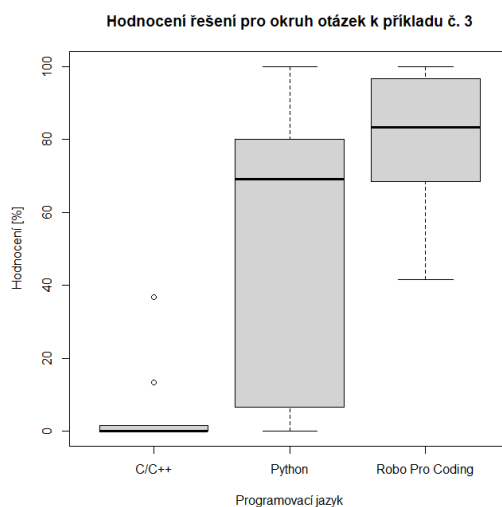
Na základě Kruskall-Wallisovo testu byla zamítnuta nulová hypotéza o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % (p-hodnota: 0,004) Tedy existuje statisticky významný rozdíl alespoň mezi 2 programovacími jazyky. Pro identifikaci, mezi kterými konkrétními jazyky je statisticky významný rozdíl, byla využita metoda mnohonásobného porovnávání, jejíž výsledky (p-hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-13. Z té je zjevné, že existuje statisticky významný rozdíl mezi programovacími jazyky C/C++ a Pythonem a také mezi

programovacími jazyky C/C++ a Robo Pro Coding, což bylo patrné již z vypočtených základních charakteristik. Mezi Pythonem a Robo Pro Coding nebyl potvrzen statisticky významný rozdíl.

Tabulka 12-13 – P-hodnoty kombinací způsobů programování pro hodnocení řešení úkolů k příkladu č. 3

p-hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,0133	--
<b>Robo Pro Coding</b>	0,0004	0,2251

Pro grafické znázornění výsledků jsou zobrazeny distribuce dat v boxplotech, viz Obrázek 12-8. Lze vidět, že nejvyšší procentuální hodnocení dosahovali respondenti v případě jazyka Robo Pro Coding a že i ti respondenti, kteří dosáhli horších výsledků v rámci tohoto jazyka, byli stále úspěšnější než v případě jazyka C/C++ a částečně i v případě Pythonu.



Obrázek 12-8 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 3

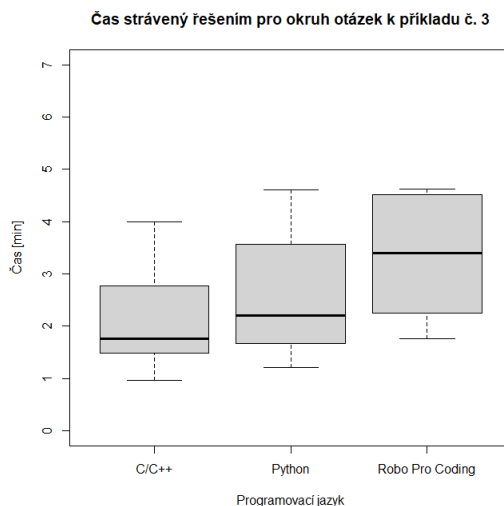
### 12.2.3.2 Čas strávený řešením úkolů

Pro vyhodnocení vlivu použitého programovacího jazyka na čas vyřešení úlohy muselo být odstraněno jedno odlehlé pozorování, protože jednomu ze studentů trvalo vyřešení úlohy v Pythonu téměř 25 minut. Jak lze vidět z tabulky níže, viz Tabulka 12-14, která ukazuje základní charakteristiky pro čas, který respondenti strávili s plněním úkolů, rozdělených dle posuzovaných programovacích jazyků, u ostatních studentů byl maximální čas v případě použití Pythonu necelé 3,5 minuty. Průměrně nejvyšší čas pro řešení byl třeba u jazyka Robo Pro Coding, naopak nejnižší u jazyka C/C++. Tomuto trendu odpovídá i většina dalších vypočtených charakteristik. Rozdíly mezi jednotlivými programovacími jazyky jsou ale minimální a ani ne v řádu minut.

Tabulka 12-14 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 3

Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	1,024	1,354	1,583	1,810	1,906	4,279
<b>Python</b>	0,9955	1,376	1,809	2,051	2,826	3,375
<b>Robo Pro Coding</b>	1,426	1,847	3,018	3,004	3,571	5,214

Na základě Kruskal-Wallisovo testu nebyla zamítnuta nulová hypotéza o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % (p-hodnota: 0,0949). Tedy neexistuje statisticky významný rozdíl alespoň mezi dvěma testovanými programovacími jazyky. Výsledky lze rovněž vidět graficky znázorněné v následujících boxplotech, viz Obrázek 12-9.



Obrázek 12-9 – Boxplot zobrazující čas strávený řešením úkolů k příkladu č. 3

### 12.2.3.3 Časová efektivita splnění úkolů

Následující tabulka, viz Tabulka 12-15, ukazuje základní charakteristiky pro procentuální hodnocení přepočtené na 1 minutu rozdělené dle posuzovaných programovacích jazyků. V případě průměrné hodnoty vychází v tomto případě nejlépe jazyk Python. Při pohledu na hodnotu mediánu vychází nejlépe Robo Pro Coding, kdy více než 50 % respondentů dosáhlo v přepočtu na minutu alespoň 28,6 %, zatímco v Pythonu 23,1 %. Naopak v jazyce C/C++ polovina respondentů nevyřešila žádnou z úloh. Maximální hodnota byla dosažena s použitím Pythonu, což je pravděpodobně i důvod, proč je průměrná hodnota u Pythonu vyšší než u Robo Pro Coding.

Tabulka 12-15 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 3

Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,000	0,000	0,000	3,200	1,484	22,661
<b>Python</b>	0,000	3,98	23,10	50,29	50,29	62,76
<b>Robo Pro Coding</b>	7,827	21,065	28,609	29,266	37,942	52,718

Na základě Kruskal-Wallisova testu byla zamítnuta nulová hypotéza o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % ( $p$ -hodnota: 0,0017). Tedy existuje statisticky významný rozdíl alespoň mezi dvěma programovacími jazyky. Pro identifikaci, mezi kterými jazyky konkrétně je statistický významný rozdíl, byla využita metoda mnohonásobného porovnávání, jejíž výsledky ( $p$ -hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-16. Z ní lze vidět, že statisticky významný rozdíl lze najít mezi jazyky Python a C/C++ a také mezi jazyky C/C++ a Robo Pro Coding. V případě úloh ze 3. okruhu se tedy významně liší právě jazyk C/C++, kde respondenti dosahovali výrazně horších výsledků v přepočtu na strávený čas.

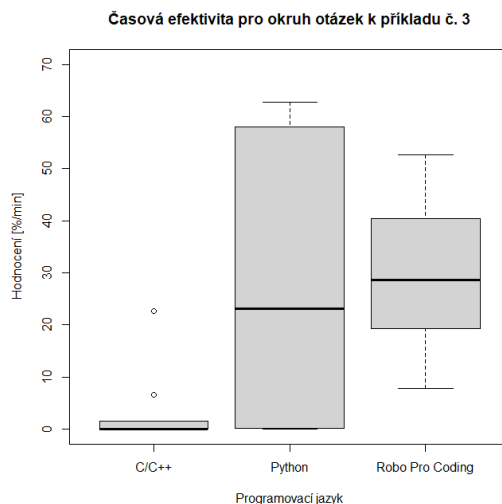
Tabulka 12-16 –  $P$ -hodnoty kombinací způsobů programování pro hodnocení časovou efektivitu k příkladu č. 3

$p$ -hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,0258	-
<b>Robo Pro Coding</b>	0,0011	0,5204

Graficky znázorněné výsledky lze vidět i na boxplotech viz Obrázek 12-10. Lze vidět, že v případě Pythonu někteří respondenti dosahovali opravdu lepších výsledků než v Robo Pro Coding, ovšem rozpětí hodnot je od 0 do téměř 63 %. Tedy našlo se velké množství respondentů, kteří si s úkoly



v jazyce Python neporadili. Naopak v případě Robo Pro Coding každý z respondentů úlohy alespoň částečně vyřešil.



Obrázek 12-10 – Boxplot zobrazující časovou efektivitu k příkladu č. 3

#### 12.2.4 Okruh otázek k příkladu č. 4

Pro vyhodnocení okruhu otázek k příkladu č. 4 byly použity výsledky otázek 4.1–4.6, ze kterých bylo vypočítáno průměrné procentuální hodnocení a průměrný čas vynaložený pro zodpovězení otázky.

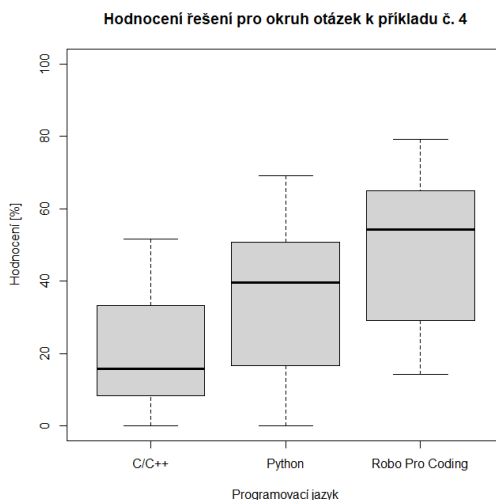
##### 12.2.4.1 Hodnocení splnění úkolů

Následující tabulka, viz Tabulka 12-17, ukazuje základní charakteristiky pro procentuální hodnocení dle posuzovaných programovacích jazyků. Z vypočtených hodnot lze vidět, že nejvyššího hodnocení dosahovali respondenti v případě použití jazyka Robo Pro Coding, kde lze pozorovat nejvyšší hodnotu mediánu, průměru i maxima. Zároveň pouze v případě tohoto jazyka všichni respondenti vyřešili úlohy alespoň částečně, minimální hodnocení je 14,2 %. Naopak u jazyka C/C++ a Pythonu je minimum 0 % bodů. V případě jazyka C/C++ dosáhla polovina respondentů pouze 15,8 % a průměrně 22,9 %. V případě Pythonu byla hodnota mediánu 39,6 % a průměru 34,8 %.

Tabulka 12-17– Základní statistické charakteristiky pro hodnocení řešení úkolů k příkladu č. 4

Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,00	10,00	15,83	22,92	32,29	51,67
<b>Python</b>	0,00	17,50	39,58	34,75	48,96	69,17
<b>Robo Pro Coding</b>	14,17	30,62	54,17	48,58	64,17	79,17

I přes tyto rozdíly nepotvrdil Kruskal-Wallisův test statisticky významný rozdíl mezi výsledky v závislosti na použití tří různých programovacích jazyků ( $p$ -hodnota: 0,0687). Přesto lze ovšem i v grafickém zobrazení níže, viz Obrázek 12-11, vidět, že nejlepší výsledků dosahovali respondenti v případě jazyka Robo Pro Coding, nejhorších pak v případě C/C++.



Obrázek 12-11 – Boxplot zobrazující hodnocení řešení pro okruh otázek k příkladu č. 4

#### 12.2.4.2 Čas strávený řešením úkolů

Následující tabulka, viz Tabulka 12-18, ukazuje základní charakteristiky pro čas, který respondenti strávili s plněním úkolů, rozdělené dle posuzovaných programovacích jazyků. Lze vidět, že stejně jako v předchozích okruzích otázek i zde nejvíce času trávili respondenti s řešením v jazyce Robo Pro Coding, naopak nejméně v jazyce C/C++. Tomu odpovídají všechny níže vypočtené charakteristiky. Průměrná hodnota v případě jazyka C/C++ je o minutu kratší než u Pythonu a o 1,5 minuty kratší než při řešení v Robo Pro Coding.

Tabulka 12-18 – Základní statistické charakteristiky pro čas strávený řešením úkolů k příkladu č. 4

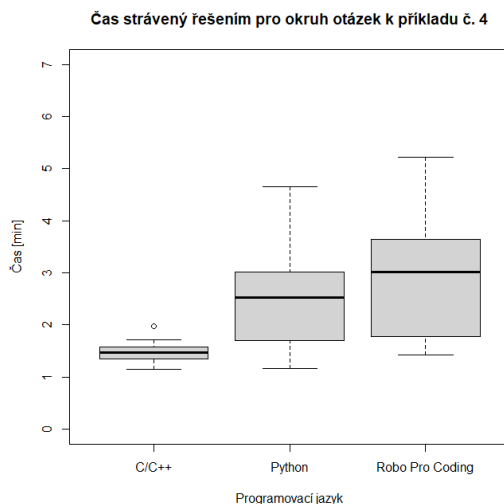
Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	1,150	1,356	1,473	1,499	1,571	1,972
<b>Python</b>	1,163	1,826	2,525	2,535	2,936	4,655
<b>Robo Pro Coding</b>	1,426	1,847	3,018	3,004	3,571	5,214

Tento rozdíl potvrdil i Kruskal-Wallisův test, díky kterému zamítáme nulovou hypotézu o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % (p-hodnota: 0,0028). Tedy existuje statisticky významný rozdíl alespoň mezi dvěma testovanými programovacími jazyky. Pro identifikaci, mezi kterými konkrétně, byla využita metoda mnohonásobného porovnávání, jejíž výsledky (p-hodnoty testovaných dvojic) obsahuje následující tabulka, viz Tabulka 12-19. Z té lze vidět, že na hladině významnosti 5 % byl potvrzen statisticky významný rozdíl mezi jazyky C/C++ a Robo Pro Coding, kdy v C/C++ je čas řešení statisticky nižší než při použití Robo Pro Coding. Pro ostatní dvojice je dle testu rozdíl statisticky nevýznamný.

Tabulka 12-19 – P-hodnoty kombinací způsobů programování čas strávený řešením úkolů k příkladu č. 4

p-hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,220	--
<b>Robo Pro Coding</b>	0,0015	0,3527

Pro grafické znázornění výsledků jsou zobrazeny distribuce dat v boxplotech, viz Obrázek 12-12. Z boxplotů je vidět, že v jazyce C/C++ byl čas k řešení výrazně nižší než u ostatních dvou jazyků. V případě použití Robo Pro Coding byl čas pro řešení úlohy nejvyšší.



Obrázek 12-12 – Boxplot zobrazující čas strávený řešením úkolů k příkladu č. 4

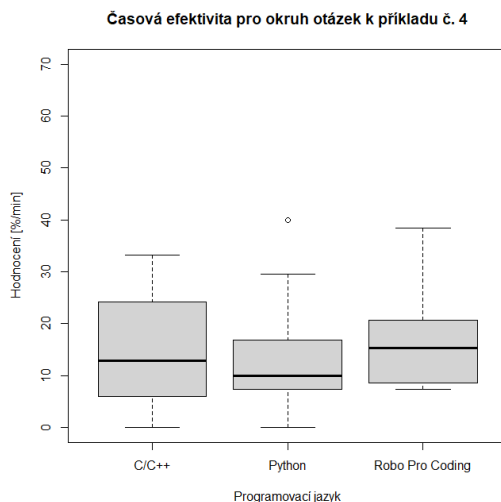
#### 12.2.4.3 Časová efektivita řešení úkolů

Následující tabulka, viz Tabulka 12-20, ukazuje základní charakteristiky pro procentuální hodnocení přepočtené na 1 minutu stráveného času rozdělené dle posuzovaných programovacích jazyků. V případě 4. okruhu otázek lze vidět, že jak průměrná hodnota, tak medián je nejvyšší při použití programovacího jazyka Robo Pro Coding. Zajímavé je, že v tomto okruhu otázek je vyšší průměrné procentuální hodnocení přepočtené na minuty vyšší u jazyka C/C++ než u Pythonu a stejně tomu tak je u hodnoty mediánu.

Tabulka 12-20 – Základní statistické charakteristiky pro časovou efektivitu k příkladu č. 4

Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	0,000	6,884	12,867	14,885	21,874	33,173
<b>Python</b>	0,000	7,459	9,937	13,730	16,657	39,989
<b>PRobo Pro Coding</b>	7,338	9,786	15,405	18,342	20,468	38,495

Rozdělení hodnot je ale v případě nejtěžšího okruhu otázek tak podobné, že Kruskal-Wallisův test nezamítl nulovou hypotézu o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % ( $p$ -hodnota: 0,525). Tedy neexistuje statisticky významný rozdíl alespoň mezi dvěma testovanými programovacími jazyky. Výsledky lze rovněž vidět graficky znázorněné v následujících boxplotech, viz Obrázek 12-13.



Obrázek 12-13 – Boxplot zobrazující časovou efektivitu k příkladu č. 4

### 12.2.5 Otázka 5 - hodnocení náročnosti testu

V závěru respondenti hodnotili náročnost řešených úloh v závislosti na posuzovaných programovacích jazycích. Následující tabulka, viz Tabulka 12-21, ukazuje základní charakteristiky jejich odpovědí, kdy 100 % znamená největší obtížnost a 0 % nejnižší. Lze vidět, že odpovědi respondentů odpovídají výsledkům v jednotlivých okruzích testu. Nejnáročnější bylo pro respondenty řešení v jazyce C/C++, které hodnotili od 80 – 100 %, průměrně 95 % a více jako polovina z nich ho hodnotila plnými 100 %. Python i Robo Pro Coding byly hodnoceny v rozmezí od 50 do 100 % s tím, že Robo Pro Coding byl pro respondenty v průměru méně náročný než Python, stejně tak medián hodnocení je v případě Robo Pro Coding nižší.

Tabulka 12-21 – Základní statistické charakteristiky pro hodnocení náročnosti testu

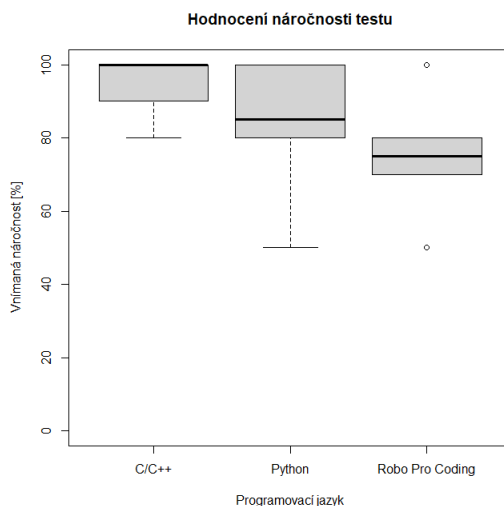
Způsob programování	Minimum	1. kvartil	Medián	Průměr	3. kvartil	Maximum
<b>C/C++</b>	80	90	100	95	100	100
<b>Python</b>	50	80	85	85	97.5	100
<b>Robo Pro Coding</b>	50	70	75	73	80	100

Na základě Kruskal-Wallisova testu byla zamítnuta nulová hypotéza o shodě distribucí tří testovaných datových sad na hladině významnosti 5 % ( $p$ -hodnota: 0,0029), tedy existuje statisticky významný rozdíl alespoň mezi dvěma programovacími jazyky. Pro identifikaci, mezi kterými jazyky konkrétně je statisticky významný rozdíl, byla využita metoda mnohonásobného porovnávání, jejíž výsledky ( $p$ -hodnoty testovaných dvojic) obsahuje následující tabulka, viz tab Tabulka 12-22. Z té lze vidět, že statisticky významný rozdíl na hladině významnosti 5 % byl potvrzen v případě dvojice C/C++ a Robo Pro Coding. U dalších dvojic nebyl statisticky významný rozdíl na této hladině potvrzen.

Tabulka 12-22 –  $P$ -hodnoty kombinací způsobů programování pro hodnocení náročnosti testu

$p$ -hodnoty testovaných dvojic	C/C++	Python
<b>Python</b>	0,0831	-
<b>Robo Pro Coding</b>	0,0047	0,0737

Rozdíly mezi hodnocením náročnosti testu v jednotlivých jazycích lze vidět i na následujících boxplotech, viz Obrázek 12-14.



Obrázek 12-14– Boxplot zobrazující hodnocení náročnosti testů

### 12.2.6 Shrnutí

Pro testování hypotéz H1 a H2 byla analyzována data od 30 respondentů, kteří plnili úkoly celkem ve 4 okruzích, a to ve 3 různých programovacích jazycích. Okruh/otázka č. 5 byl doplňující dotaz zaměřený na zjištění vnímání náročnosti vyplňovaného testu respondentem. Během testu byly respondentům zaznamenávány odpovědi a čas, který strávili řešením jednotlivých úloh. Pro vyhodnocení byly použity průměrné hodnoty skóre (hodnocení) a průměrný čas vždy za všechny testové otázky, které spadaly do daného okruhu. Zároveň byla dopočítána veličina odpovídající přepočtenému skóre na 1 minutu, aby pro posouzení času stráveného na řešení úlože bylo zohledněno i výsledné skóre (záleží nám na tom, jestli zadání respondent vyřešil rychle a celé chybně, nebo pomaleji, ale celé správně).

#### 12.2.6.1 Hypotéza 1

**H1: Využitím navržené architektury training factory lze průmyslovým inženýrům zjednodušit realizaci a případnou modifikaci komplexního HW modelu schopného komunikovat se svým digitálním dvojčetem oproti standardnímu přístupu využívajícím pro řízení programovací jazyk C/C++, jenž je využit v případě Fischertechnik Training Factory Industry 4.0.**

Pro ověření první hypotézy byla analyzována data o procentuálním hodnocení řešených úloh. Krátké shrnutí výsledků obsahuje následující tabulka, viz Tabulka 12-23. První čtyři sloupce obsahují název programovacího jazyka, u kterého byla daná charakteristika pro otázky k příkladům 1–4 nejvyšší, tj. nejvyšší minimum, medián, průměr a maximum. Z těchto charakteristik je viditelné, že kromě maximální hodnoty u okruhu otázek k příkladu 3, kde byla stejná nejvyšší hodnota pro Python i Robo Pro Coding (100 %), dosahovali respondenti vždy nejvyšších hodnocení s využitím jazyka Robo Pro Coding. Tento rozdíl byl statisticky testován pomocí Kruskal-Wallisova testu a následně pomocí mnohonásobného porovnávání, které v případě identifikace významného rozdílu mezi třemi programovacími jazyky testoval všechny dvojice zvlášť. Poslední dva sloupce obsahují rozhodnutí na základě mnohonásobného porovnávání, zda byl rozdíl mezi jazykem Robo Pro Coding a dalšími dvěma jazyky statisticky významný. Lze vidět, že v případě dvojice C/C++ a Robo Pro Coding byl tento rozdíl statisticky významný v případě prvních 3 okruhů, v tom poslední již ne. To je pravděpodobně způsobeno vyšší náročností řešených úloh, kde již rozdíly mezi dosaženým hodnocením v těchto dvou jazycích nebyly tak velké a zároveň se dá očekávat, že studenti byli při řešení posledního (nejnáročnějšího) příkladu již mentálně unaveni. V případě dvojice Python a Robo Pro Coding nebyl tento rozdíl potvrzen jako statisticky významný ani v jednom případě, ačkoliv rozdíly jsou z dat patrné.

To je pravděpodobně způsobeno menšími rozdíly, které v tomto počtu respondentů nebylo možné pomocí statistických testů identifikovat.

Tabulka 12-23 – Shrnutí výsledků pro H1

H1	Minimum	Medián	Průměr	Maximum	Robo Pro Coding x C/C++	Robo Pro Coding x Python
1	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Potvrzen	Nepotvrzen
2	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Potvrzen	Nepotvrzen
3	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding + Python	Potvrzen	Nepotvrzen
4	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Nepotvrzen	Nepotvrzen

Přesto lze z dosažených výsledků konstatovat, že v případě procentuálního hodnocení řešených úloh dosahovali respondenti nejlepších výsledků při použití Robo Pro Coding, naopak nejhorších výsledků dosahovali při použití C/C++. V případě Pythonu byly výsledky často různorodé a hodnoty procentuálního hodnocení měly široké rozpětí. Našli se respondenti, kteří Python ovládali velmi dobře, ale zároveň se našli tací, kteří si s ním vůbec neporadili. I to může být jeden z důvodů, proč statistické testy neoznačily rozdíl mezi jazyky Python a Robo Pro Coding jako statisticky významný – čím vyšší je rozptyl dané veličiny, tím větší je třeba počet pozorování pro identifikaci rozdílu. Velikost rozptylu ovšem nebylo před samotným sběrem dat a analýzou, jak odhadnout. Ovšem i tento velký rozptyl poukazuje na to, že využití Robo Pro Coding je pro využití průmyslovými inženýry vhodnější než Python. Důvodem je, že v případě Robo Pro Coding měli respondenti schopnosti daný úkol vždy alespoň částečně vyřešit (žádné průměrné hodnocení úloh řešených v Robo Pro Coding se nerovnálo 0 %), na rozdíl od C/C++ a Pythonu, kde se ve všech 4 okruzích našel alespoň 1 respondent, který nebyl schopný vyřešit ani část úkolů (tj. měl 0 %). Z tohoto důvodu považujeme hypotézu 1 za potvrzenou. Pro další výzkum doporučujeme vyjít z našich výsledků a rozšířit počet respondentů tak, aby bylo možné důkladněji otestovat rozdíl mezi programovacími jazyky Python a Robo Pro Coding.

#### 12.2.6.2 Hypotéza 2

**H2: Navržená architektura zvýší časovou efektivitu průmyslových inženýrů při úpravě řídicího kódu HW modelu se základními čidly využívanými v konceptu Industry 4.0 oproti programování HW standardními programovacími jazyky využitými u Fischertechnik Training Factory Industry 4.0 se zachováním možnosti komunikace HW modelu s okolím.**

Pro otestování druhé hypotézy byly původně využity naměřené hodnoty času, tedy jak dlouho respondenti strávili řešením úlohy. Z této analýzy ovšem ve většině případů vycházelo, že respondenti strávili nejméně času s jazykem C/C++, a naopak nejdéle trávili řešením úloh v programovacím jazyce Robo Pro Coding. Protože jsme z předchozí analýzy věděli, že procentuální hodnocení vypovídají o tom, že nejlepší výsledky měli respondenti v Robo Pro Coding, rozhodli jsme se testovat hodnoty nové proměnné, konkrétně přepočtu bodů na jednotku času (1 min), což jsme spočítali jako podíl procentuálního hodnocení a času stráveného řešením. Získat opačnou hodnotu, tedy přepočet času na počet bodů, nebylo možné z toho důvodu, že procentuální hodnocení bylo často 0 % a nulou dělit nelze.

Stejně jako v případě hypotézy 1 i zde je uvedena souhrnná tabulka, viz Tabulka 12-24. Lze vidět, že v případě minima a mediánu byly ve všech případech dosaženy respondenty nejvyšší hodnoty v případě použití Robo Pro Coding. Průměrná hodnota byla v případě okruhu otázek č. 3 nejvyšší u Pythonu, kde bylo zároveň i maximum, které pravděpodobně ovlivnilo i průměrnou hodnotu. Potvrzen byl statisticky významný rozdíl mezi jazykem C/C++ a Robo Pro Coding u všech okruhů otázek. U dvojice Robo Pro Coding a Python nebyl potvrzen statisticky významný rozdíl ani v jednom případě. Důvody

jsou podobné jako v případě hodnocení hypotézy H1. Hodnoty v případě jazyka Python jsou příliš rozptýlené, a tedy by na otestování rozdílu mezi nimi bylo třeba více pozorování stejně jako v případě předchozí hypotézy H1. Stejně jako v případě H1 jsou zde důležité hodnoty minima, kdy ve všech případech byla největší hodnota u Robo Pro Coding. Stejně tak u mediánu dosahoval Robo Pro Coding nejvyšší hodnoty. Z toho lze soudit, že v jazyce Robo Pro Coding jsou všichni respondenti schopni dané úlohy alespoň částečně vyřešit, což je pro posouzení navržené architektury klíčové. Jazyk Robo Pro Coding je tedy z tohoto pohledu v rámci zvolených programovacích jazyků nejefektivnějším řešením z hlediska poměru procentuálního hodnocení a času potřebného na řešení. Lze tedy konstatovat, že hypotéza 2 byla rovněž potvrzena.

Tabulka 12-24 – Shrnutí výsledků pro H2

H2	Minimum	Medián	Průměr	Maximum	Robo Pro Coding x C/C++	Robo Pro Coding x Python
1	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Potvrzeno	Nepotvrzeno
2	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Potvrzeno	Nepotvrzeno
3	Robo Pro Coding	Robo Pro Coding	Python	Python	Potvrzeno	Nepotvrzeno
4	Robo Pro Coding	Robo Pro Coding	Robo Pro Coding	Python	Potvrzeno	Nepotvrzeno

### 12.3 Hodnocení znalostí – dotazník

**H3: Využitím navržené architektury místo architektury Fischertechnik Training Factory Industry 4.0 dojde k redukci potřebných nových odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčtem.**

Tato hypotéza je ověřena porovnáním potřebného množství a typu znalostí z oblasti programování a informatiky pro navrženou architekturu a pro architekturu Fischertechnik Training Factory Industry 4.0. Dotazník vyplnilo 33 respondentů. Otázky byly rozřazeny do 7 podtémat a každé podtéma bylo vyhodnocováno zvlášť. U každé položky je uvedeno, zda je tento přístup pro řešení, které bylo vyvinuto v rámci disertační práce (M), zda tento přístup využívá konkurenční řešení od Fischertechnik (F) nebo zda je potřeba pro oba přístupy (MF).

#### 12.3.1 Zkušenosti s programovacími jazyky

V rámci tohoto podtématu byli respondenti tázáni na zkušenosti s 13 technologiemi, odpovídat mohli na 5bodové stupnici, viz Tabulka 12-25. Slovní odpovědi byly pro vyhodnocení překódovány na číselné jak pro jednodušší práci s nimi, tak zároveň pro využití statistických testů, které lze realizovat jen pro číselné údaje.

Tabulka 12-25 – Stupnice pro hodnocení zkušeností s programovacími jazyky

Bod	Význam
1	Nikdy jsem o něm neslyšel
2	Vím, že existuje, ale nikdy jsem ho nepoužil
3	Dokážu naprogramovat jednoduchý program/ skript
4	Dokážu naprogramovat složitější program/ skript
5	Provažuji se zkušeného programátora

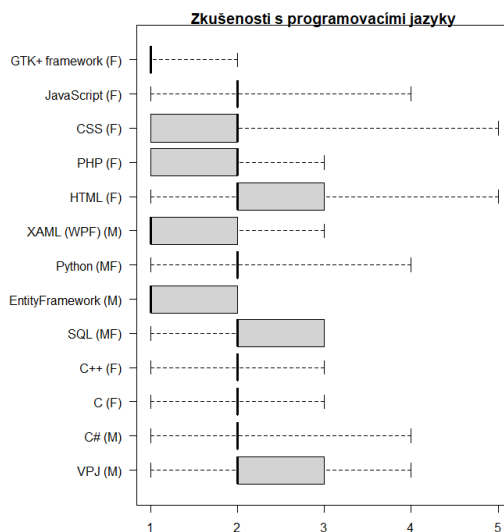
Následující tabulka, viz Tabulka 12-26, obsahuje četnosti odpovědí pro jednotlivé technologie a zároveň základní charakteristiky souboru, tj. hodnotu mediánu odpovědí a jejich průměrnou hodnotu. Dle průměrných hodnot lze vidět, že nejvíce zkušenosti mají studenti s HTML, SQL a VPJ, kde je průměrná hodnota nejvyšší. Zároveň si lze z četností všimnout, že pouze u technologií VPJ, C#, Python, CSS a HTML studenti odpověděli, že dokážou podle svého sebehodnocení naprogramovat složitější program/skript. Nejhůře vycházejí dle průměrných odpovědí technologie GTK+ framework, EntityFramework a XAML (WPF), kde většina studentů o těchto technologiích nikdy neslyšela.



Tabulka 12-26 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s programovacími jazyky“

Programovací jazyky a frameworky	Četnosti odpovědí					Charakteristiky odpovědí	
	1	2	3	4	5	Medián	Průměr
VPJ (M)	7	15	9	2	0	2,00	2,18
C# (M)	8	18	6	1	0	2,00	2,00
C (F)	7	22	4	0	0	2,00	1,91
C++ (F)	4	24	5	0	0	2,00	2,03
SQL (MF)	2	17	14	0	0	2,00	2,36
EntityFramework (M)	23	10	0	0	0	1,00	1,30
Python (MF)	8	21	3	1	0	2,00	1,91
XAML (WPF) (M)	21	11	1	0	0	1,00	1,39
HTML (F)	2	17	12	1	1	2,00	2,46
PHP (F)	13	16	4	0	0	2,00	1,73
CSS (F)	13	14	4	1	1	2,00	1,88
JavaScript (F)	3	24	5	1	0	2,00	2,12
GTK+ framework (F)	27	6	0	0	0	1,00	1,18

Distribuce odpovědí respondentů pro jednotlivé technologie jsou ukázány i v následujícím grafu zobrazujícím boxploty, viz Obrázek 12-15. Rozdíly mezi znalostmi těchto technologií byly testovány pomocí Kruskal-Wallisova testu, na jehož základě byla zamítnuta shoda v množství znalostí ( $p$ -hodnota  $< 0,001$ ). Znalosti respondentů se tedy od sebe navzájem statisticky významně liší alespoň pro jednu dvojici technologií. Pro identifikaci, které z dvojic to jsou, byla využita metoda mnohonásobného porovnávání.



Obrázek 12-15 – Boxplot zobrazující zkušenosti respondentů s programovacími jazyky

Výsledky této metody (pro lepší přehlednost pouze dvojice, kde byl identifikován statisticky významný rozdíl, tj.  $p$ -hodnota  $<$  hladina významnosti = 5 %) lze vidět v následující tabulce, viz Tabulka 12-27. Tučně jsou označeny ty dvojice, u kterých se liší jejich využití v hodnocených řešeních (M/MF/F).

Tabulka 12-27 – P-hodnoty dvojic se statisticky významným rozdílem pro okruh „Zkušenosti s programovacími jazyky“

Zkušenosti s technologiemi	p-hodnota	Závěr
GTK+ framework (F) - HTML (F)	<0,001	
<b>GTK+ framework (F) - SQL (MF)</b>	<b>&lt;0,001</b>	<b>MF</b>
<b>EntityFramework (M) - HTML (F)</b>	<b>&lt;0,001</b>	<b>F</b>
<b>EntityFramework (M) - SQL (MF)</b>	<b>&lt;0,001</b>	<b>MF</b>
<b>HTML (F) - XAML (WPF) (M)</b>	<b>&lt;0,001</b>	<b>F</b>
<b>SQL (MF) - XAML (WPF) (M)</b>	<b>&lt;0,001</b>	<b>MF</b>
<b>GTK+ framework (F) - VPJ (M)</b>	<b>&lt;0,001</b>	<b>M</b>
GTK+ framework (F) - JavaScript (F)	<0,001	
C++ (F) - GTK+ framework (F)	<0,001	
EntityFramework (M) - VPJ (M)	<0,001	
<b>EntityFramework (M) - JavaScript (F)</b>	<b>&lt;0,001</b>	<b>F</b>
<b>C# (M) - GTK+ framework (F)</b>	<b>&lt;0,001</b>	<b>M</b>
C (F) - GTK+ framework (F)	<0,001	
<b>C++ (F) - EntityFramework (M)</b>	<b>&lt;0,001</b>	<b>F</b>
<b>GTK+ framework (F) - Python (MF)</b>	<b>0,001</b>	<b>MF</b>
VPJ (M) - XAML (WPF) (M)	0,001	
<b>JavaScript (F) - XAML (WPF) (M)</b>	<b>0,001</b>	<b>F</b>
C# (M) - EntityFramework (M)	0,004	
<b>C++ (F) - XAML (WPF) (M)</b>	<b>0,007</b>	<b>F</b>
HTML (F) - PHP (F)	0,012	
CSS (F) - GTK+ framework (F)	0,013	
<b>C (F) - EntityFramework (M)</b>	<b>0,013</b>	<b>F</b>
<b>PHP (F) - SQL (MF)</b>	<b>0,015</b>	<b>MF</b>
<b>EntityFramework (M) - Python (MF)</b>	<b>0,023</b>	<b>MF</b>
C# (M) - XAML (WPF) (M)	0,030	
CSS (F) - HTML (F)	0,046	

Na základě těchto výsledků lze konstatovat, že v případě, že mezi technologiemi existuje statisticky významný rozdíl, je tento rozdíl většinou ve prospěch řešení od Fischertechniku (7x lepší znalosti než druhá technologie ve dvojici). V šesti případech byla respondentům statisticky významně více známá technologie, kterou využívají obě řešení, a pouze ve dvou případech se ukázala jako známější technologie využívaná v navrženém řešení, a to vždy ve srovnání s GTK+ frameworkem, který téměř žádný z respondentů nezná.

Při vyhodnocení tohoto tématu je ale zároveň třeba brát v potaz, že navržená architektura v této disertační práci vyžaduje po uživateli znalost pouze 6 technologií, kde 2 z nich jsou pro většinu respondentů neznámé (EntityFramework a XAML(WPF)). Architektura Fischertechnik Training Factory Industry 4.0 vyžaduje po uživateli znalost 9 technologií, z nichž 1 je pro většinu respondentů zcela neznámá (GTK+ framework s nejhorším průměrným výsledkem). Na základě toho lze konstatovat, že celkové množství potřebných odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčetem se v případě tohoto podtématu reálně zredukuje, úroveň znalostí průmyslových inženýrů je ale pro obě řešení velmi podobná.

### 12.3.2 Zkušenosti s vývojovými prostředími

V rámci tohoto podtématu byli respondenti tázáni na zkušenosti se třemi vývojovými prostředími. Opět mohli pro hodnocení využít 5bodovou škálu, která je včetně převodu na číselné hodnoty uvedena v následující tabulce, viz Tabulka 12-28.

Tabulka 12-28 – Stupnice pro hodnocení zkušeností s vývojovými prostředími

Bod	Význam
1	Neznám
2	Znám, ale nemám praktickou zkušenost
3	Již jsem někdy vyzkoušel
4	Znám jen základní funkce
5	Považuji se za pokročilého uživatele

Následující tabulka, viz Tabulka 12-29, obsahuje četnosti odpovědí pro jednotlivá vývojová prostředí a zároveň základní charakteristiky souboru, tj. hodnotu mediánu odpovědí a jejich průměrnou hodnotu. Dle průměrné hodnoty lze vidět, že největší zkušenosti mají respondenti s prostředím MS Visual Studio, kde se jeden z nich označil dokonce za pokročilého uživatele. Další dvě vývojová prostředí jsou na tom velmi podobně, přičemž o něco lepší hodnocení má prostředí Robo Pro Coding Fischertechnik oproti Eclipse.

Tabulka 12-29 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s vývojovými prostředími“

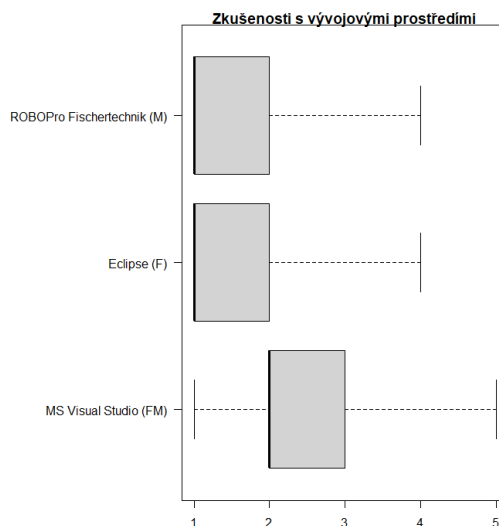
Vývojová prostředí	Četnosti odpovědí					Charakteristiky odpovědí	
	1	2	3	4	5	Medián	Průměr
MS Visual Studio (FM)	7	11	7	7	1	2,00	2,52
Eclipse (F)	22	5	3	3	0	1,00	1,61
Robo Pro Coding Fischertechnik (M)	20	6	6	1	0	1,00	1,64

Distribuce odpovědí respondentů pro jednotlivá vývojová prostředí lze vidět v následujících boxplotech, viz Obrázek 12-16. Rozdíly mezi znalostmi byly testovány pomocí Kruskal-Wallisova testu, na jehož základě byla zamítnuta shoda v množství znalostí jednotlivých prostředí (p-hodnota <0,001). Znalosti respondentů se tedy od sebe navzájem statisticky významně liší alespoň pro jednu dvojici technologií. Pro identifikaci, které z dvojic to jsou, byla využita metoda mnohonásobného porovnávání.

Pomocí této metody byl nalezen statisticky významný rozdíl mezi MS Visual Studio a ostatními dvěma prostředími:

- MS Visual Studio (FM) – Eclipse (F), p-hodnota: 0,0010
- MS Visual Studio (FM) - Robo Pro Coding Fishertechnik (M), p-hodnota: 0,0019

Mezi prostředím Robo Pro Coding Fishertechnik a prostředím Eclipse neexistuje statisticky významný rozdíl.



Obrázek 12-16 – Boxplot zobrazující zkušenosti respondentů s vývojovými prostředími

Na základě toho lze konstatovat, že obě architektury mají na uživatele stejné požadavky a využitím navržené architektury se nezredukuje množství nových potřebných odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojitým. Množství těchto znalostí se ale ani nezvyšuje.

### 12.3.3 Ostatní programátorské zkušenosti

V rámci třetího podtématu byli respondenti tázáni na další programátorské zkušenosti, kterých bylo definováno v rámci dotazníku šest. V tomto případě odpovídali na 4bodové škále, která je i s číselným převodem k nahlédnutí v následující tabulce, viz Tabulka 12-30.

Tabulka 12-30 – Štupnice pro hodnocení ostatních programátorských zkušeností

Bod	Význam
1	Nevím, co to znamená
2	Teoreticky znám, ale ve vlastním kódu jsem to nikdy nepoužil
3	Zkusil jsem si ukázkovou implementaci
4	Mám zkušenosti s využitím ve vlastním programu

Následující tabulka, viz Tabulka 12-31, obsahuje četnosti odpovědí pro jednotlivé položky a zároveň základní charakteristiky souboru, tj. hodnotu mediánu odpovědí a jejich průměrnou hodnotu. Z hodnot četností, ale i z hodnot mediánů, které jsou ve všech případech rovny 1, tj. nejnižší možné hodnocení (nevím, co to znamená), je zřejmé, že většina respondentů se s většinou z těchto technologií nikdy nesešla. Nejvíce zkušeností mají respondenti s komunikací po síti, dále pak s JSONem a s I2C sběrnici. Ve všech třech případech se ale jedná o pouhé jednotky respondentů, kteří mají větší než pouze teoretické znalosti.

Tabulka 12-31 – Tabulka četnosti odpovědí pro okruh „Ostatní programátorské zkušenosti“

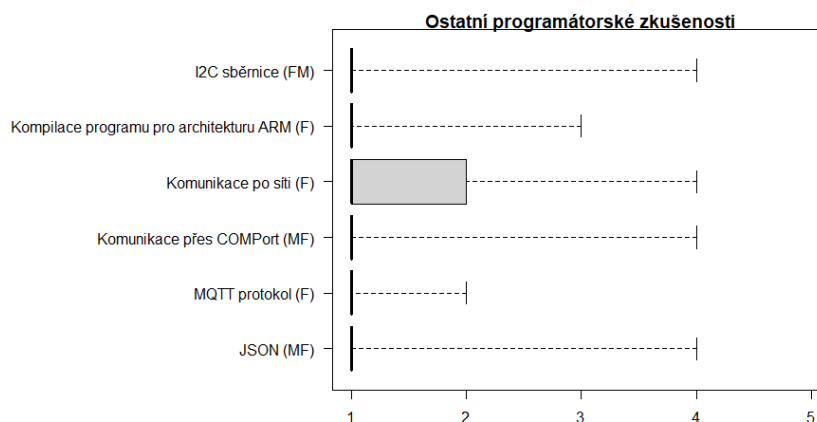
Ostatní programátorské zkušenosti	Četnosti odpovědí				Charakteristiky odpovědí	
	1	2	3	4	Medián	Průměr
JSON (MF)	29	1	2	1	1,00	1,24
MQTT protokol (F)	32	1	0	0	1,00	1,03
Komunikace přes COMPort (MF)	30	1	1	1	1,00	1,18
Komunikace po síti (F)	21	8	3	1	1,00	1,52
Kompilace programu pro architekturu ARM (F)	30	2	1	0	1,00	1,12
I2C sběrnice (FM)	28	4	0	1	1,00	1,21

Distribuce odpovědí respondentů pro jednotlivé hodnocené položky lze vidět v následujících boxplotech, viz Obrázek 12-17. Rozdíly mezi znalostmi byly testovány pomocí Kruskal-Wallisova testu, na jehož základě byla zamítnuta shoda v množství znalostí ( $p$ -hodnota=0,003). Znalosti respondentů se tedy od sebe navzájem statisticky významně liší alespoň pro jednu dvojici technologií.

Na základě mnohonásobného porovnávání byl nalezen statisticky významný rozdíl mezi těmito dvojicemi:

- Kompilace programu pro architekturu ARM (F) - Komunikace po síti (F),  $p$ -hodnota: 0,0242
- Komunikace po síti (F) - Komunikace přes COMPort (MF),  $p$ -hodnota: 0,0277
- Komunikace po síti (F) - MQTT protokol (F),  $p$ -hodnota: 0,0018

Tyto rozdílné dvojice jsou způsobené tím, že s komunikací po síti mají respondenti nejvíce zkušeností, zatímco zbylé tři technologie většina respondentů vůbec nezná (viz výše).



Obrázek 12-17 – Boxplot zobrazující ostatní programátorské zkušenosti

Na základě vyhodnocení tohoto podtématu lze konstatovat především to, že většina respondentů nezná ani jednu z dotazovaných technologií více než teoreticky. Nejlépe je hodnocena komunikace po síti, kterou využívá architektura Fischertechnik Training Factory Industry 4.0, zároveň však tato architektura vyžaduje po uživatelích znalost dalších 5 technologií, které respondenti neznají. Naopak navrhovaná architektura vyžaduje znalost pouze 3 technologií, které jsou zároveň požadovány i u konkurenčního produktu. Na základě toho lze konstatovat, že využitím navržené architektury místo architektury Fischertechnik Training Factory Industry 4.0 dojde k redukci nových potřebných odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčtem.

### 12.3.4 Zkušenosti s uživatelskou prací v operačním systému

V rámci tohoto tématu byli respondenti tázáni na zkušenosti se dvěma operačními systémy – MS Windows a Linux<sup>5</sup>. Pro hodnocení mohli respondenti využít 5bodovou škálu, která je včetně převodu na číselné hodnoty uvedena v následující tabulce, viz Tabulka 12-32.

Tabulka 12-32 – Stupnice pro hodnocení zkušeností s uživatelskou prací v operačním systému

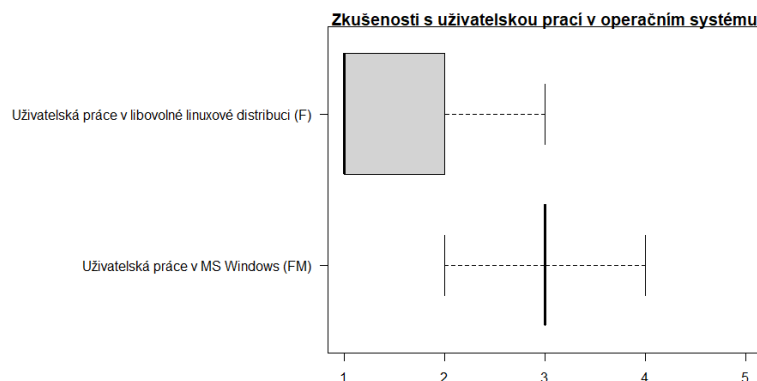
Bod	Význam
1	Nikdy jsem tento operační systém nepoužil(a)
2	Operační systém jsem si zkusil(a)
3	Jsem běžný uživatel
4	Jsem Power User
5	Považuji se za experta

Následující tabulka, viz Tabulka 12-33, obsahuje četnosti odpovědí pro jednotlivé OS a zároveň základní charakteristiky souboru, tj. hodnotu mediánu odpovědí a jejich průměrnou hodnotu. Z těchto hodnot lze vidět, že respondenti mají větší zkušenosti s operačním systémem MS Windows, kde se více než polovina označila za běžné uživatele, 8 z nich pak za Power Usery. V případě Linuxu více než polovina respondentů systém nikdy nepoužila, část z nich ho jen vyzkoušela a pouze 1 respondent se označil za běžného uživatele. Ani u jednoho z OS se žádný z respondentů neoznačil za experta.

Tabulka 12-33 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s uživatelskou prací v operačním systému“

Zkušenosti s uživatelskou prací v operačním systému	Četnosti odpovědí					Charakteristiky odpovědí	
	1	2	3	4	5	Medián	Průměr
<b>Uživatelská práce v MS Windows (FM)</b>	0	5	20	8	0	3,00	3,09
<b>Uživatelská práce v libovolné linuxové distribuci (F)</b>	20	12	1	0	0	1,00	1,42

Distribuce odpovědí respondentů pro jednotlivá vývojová prostředí lze vidět v boxplotech, viz Obrázek 12-18. Rozdíly mezi znalostmi byly testovány pomocí Wilcoxonova testu, na jehož základě byla zamítnuta shoda v množství znalostí uživatelské práce v testovaných OS (p-hodnota <0,001)., Znalosti respondentů se tedy od sebe navzájem statisticky významně liší. Z boxplotů i hodnot v tabulce výše lze konstatovat, že znalosti respondentů v uživatelské práci v OS MS Windows jsou statisticky vyšší než v libovolné linuxové distribuci.



Obrázek 12-18 – Boxplot zobrazující zkušenosti s uživatelskou prací v operačním systému

<sup>5</sup> V tomto případě je Linuxem myšlena jakákoliv linuxová distribuce

Na základě tohoto vyhodnocení lze konstatovat, že využitím navržené architektury místo architektury Fischertechnik Training Factory Industry 4.0 dojde k redukci nových potřebných odborných znalostí průmyslového inženýra v oblasti práce s operačním systémem pro realizaci HW modelu s napojeným digitálním dvojčtem.

### 12.3.5 Zkušenosti se softwarovými nástroji

V rámci tohoto podtématu byli respondenti tázáni na zkušenosti se třemi různými SW nástroji. Odpovídali opět na 5bodové škále, kterou lze vidět včetně číselného převodu v následující tabulce, viz Tabulka 12-34.

Tabulka 12-34 – Stupnice pro hodnocení zkušeností se softwarovými nástroji

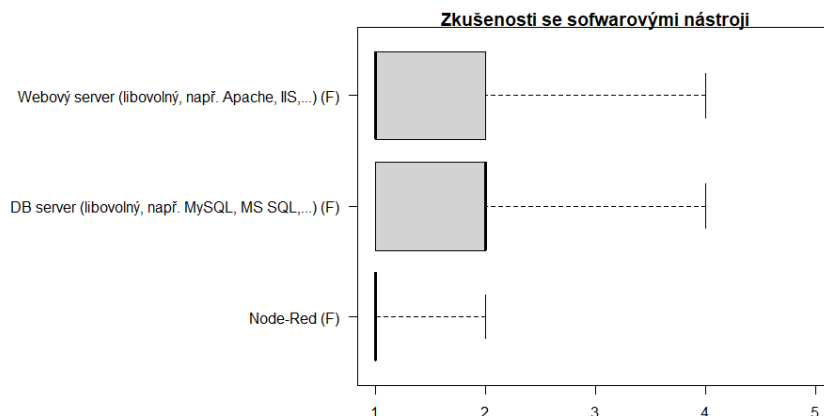
Bod	Význam
1	Nevím, co to znamená
2	Teoreticky znám, ale nikdy jsem nepoužil
3	Zkusil jsem si ukázkovou implementaci
4	Mám zkušenosti s využitím ve vlastním řešení
5	Považuji se za experta

Následující tabulka, viz Tabulka 12-35, obsahuje četnosti odpovědí pro jednotlivé SW nástroje a zároveň základní charakteristiky souboru, tj. hodnotu mediánu odpovědí a jejich průměrnou hodnotu. Z hodnot v tabulce lze vidět, že nejznámějším nástrojem jsou DB servery, kdy většina respondentů má alespoň teoretické znalosti, případně zkušenosti s ukázkovou implementací. V případě webových serverů více než polovina respondentů neví, co to znamená. Node-Red nezná většina respondentů s výjimkou dvou respondentů, kteří nástroj znají alespoň teoreticky.

Tabulka 12-35 – Tabulka četnosti odpovědí pro okruh „Zkušenosti se softwarovými nástroji“

Softwarové nástroje	Četnosti odpovědí					Charakteristiky odpovědí	
	1	2	3	4	5	Medián	Průměr
Node-Red (F)	31	2	0	0	0	1,00	1,06
DB server (libovolný, např. MySQL, MS SQL apod.) (F)	10	15	7	1	0	2,00	1,97
Webový server (libovolný, např. Apache, IIS apod.) (F)	20	9	3	1	0	1,00	1,55

Distribuce odpovědí respondentů pro jednotlivé SW nástroje lze vidět v níže zobrazených boxplotech, viz Obrázek 12-19. Pomocí Kruskalova-Wallisova testu byla zamítnuta shoda znalostí jednotlivých nástrojů ( $p < 0,001$ ). Znalosti respondentů se od sebe tedy navzájem statisticky významně liší. Pomocí metody mnohonásobného porovnání byl identifikován statisticky významný rozdíl mezi všemi dvojicemi ( $p < 0,001$ ), tj. nejvíce znalostí mají respondenti o DB serverech, dále následují webové servery a nejméně znalostí mají s Node-Red. Vzhledem k tomu, že žádný z těchto nástrojů není potřeba pro práci s navrhovanou architekturou, ale je potřeba pro architekturu Fischertechnik Training Factory Industry 4.0, lze konstatovat, že v tomto případě dojde k redukci nových potřebných odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčtem.



Obrázek 12-19 – Boxplot zobrazující zkušenosti se softwarovými nástroji

### 12.3.6 Zkušenosti s administrací SW nástrojů

V rámci tohoto podtématu byly hodnoceny zkušenosti s administrací několika SW nástrojů. Respondenti mohli odpovídat na 5bodové škále, kterou lze vidět včetně číselného převodu v následující tabulce, viz Tabulka 12-36.

Tabulka 12-36 – Stupnice pro hodnocení zkušeností s administrací SW nástrojů

Bod	Význam
1	Nemám žádnou zkušenost
2	Mám zkušenost s defaultní instalací
3	Mám zkušenost s instalací a konfigurací
4	Mám zkušenosti s instalací, konfigurací a zabezpečením
5	Považuji se za experta

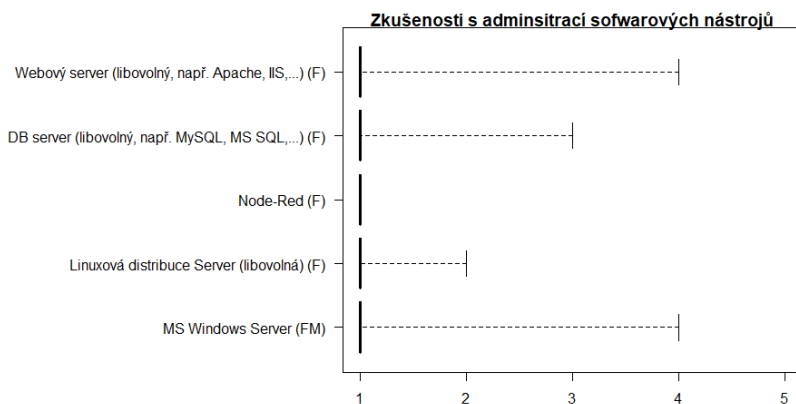
Následující tabulka, viz Tabulka 12-37, obsahuje četnosti odpovědí pro administraci jednotlivých SW nástrojů a zároveň základní charakteristiky souboru, tj. hodnotu mediánu odpovědí a jejich průměrnou hodnotu. Hodnoty v tabulce odpovídají výsledkům předchozího podtématu, u kterého z odpovědí vyplynulo, že respondenti obecně nemají příliš zkušeností s dotazovanými SW nástroji. V případě jejich administrace mají respondenti největší zkušenosti s MS Windows Server, kde dva respondenti mají zkušenosti s instalací, konfigurací i zabezpečením. Většina z respondentů ale ani s tímto nástrojem nemá žádnou zkušenost. Dalším více známým SW nástrojem jsou DB servery, s kterými má nějakou zkušenost alespoň sedm respondentů z 33. Naopak úplně neznámá je administrace Node-Red, se kterou nemá zkušenost ani jeden z respondentů.

Tabulka 12-37 – Tabulka četnosti odpovědí pro okruh „Zkušenosti s administrací SW nástrojů“

SW nástroj	Četnosti odpovědí					Charakteristiky odpovědí	
	1	2	3	4	5	Medián	Průměr
MS Windows Server (FM)	25	6	0	2	0	1,00	1,36
Linuxová distribuce Server (libovolná) (F)	31	2	0	0	0	1,00	1,06
Node-Red (F)	33	0	0	0	0	1,00	1,00
DB server (libovolný, např. MySQL, MS SQL apod.) (F)	26	5	2	0	0	1,00	1,27
Webový server (libovolný, např. Apache, IIS apod.) (F)	29	3	0	1	0	1,00	1,18



Distribuce odpovědí respondentů pro jednotlivé hodnocené položky lze vidět v následujících boxplotech, viz Obrázek 12-20. Rozdíly mezi znalostmi byly testovány pomocí Kruskal-Wallisova testu, na jehož základě byla zamítnuta shoda v množství znalostí ( $p$ -hodnota=0,015). Znalosti respondentů se tedy od sebe navzájem statisticky významně liší alespoň pro jednu dvojici SW nástrojů.



Obrázek 12-20 – Boxplot zobrazující zkušenosti s administrací softwarových nástrojů

Následné mnohonásobné porovnávání prokázalo statisticky významný rozdíl pouze mezi MS Windows Server s nejvyšším průměrným hodnocením znalostí a Node-Red s nejnižším průměrným hodnocením ( $p$ -hodnota: 0,0314). Mezi znalostmi administrace dalších SW nástrojů neexistuje statisticky významný rozdíl.

Vzhledem k tomuto vyhodnocení můžeme konstatovat, že využitím navržené architektury místo architektury Fischertechnik Training Factory Industry 4.0 dojde k redukci nových potřebných odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčtem. Jednak se sníží množství SW nástrojů, které respondent musí znát (z pěti na jeden), a zároveň administraci SW nástroje, který je v navržené architektuře používán, znají respondenti dle jejich odpovědí nejlépe.

### 12.3.7 Ostatní uživatelské zkušenosti

V případě tohoto podtématu hodnotili respondenti své další uživatelské zkušenosti s vybranými nástroji. Hodnocení probíhalo opět na 5bodové škále, kterou lze vidět včetně převodu na čísla v následující tabulce, viz Tabulka 12-38.

Tabulka 12-38 – Štupnice pro hodnocení ostatních uživatelských zkušeností

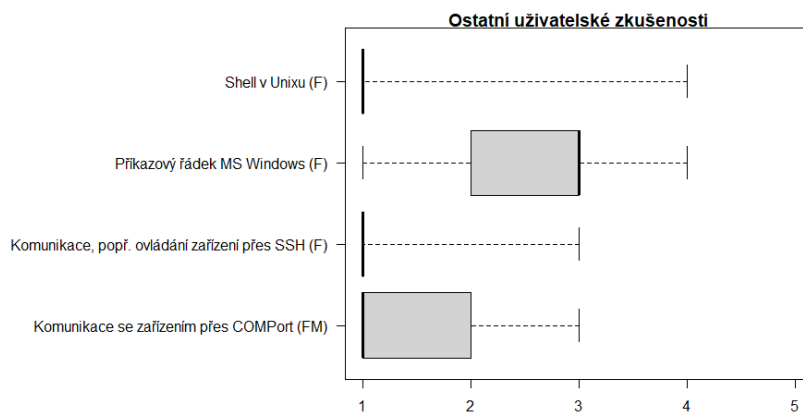
Bod	Význam
1	Nevím, co to znamená
2	Teoretický znám, ale prakticky jsem to nikdy nepoužil
3	Zkusil jsem si to
4	Běžně používám
5	Považuji se za experta

Následující tabulka, viz Tabulka 12-39, obsahuje četnosti odpovědí pro jednotlivé nástroje a zároveň základní charakteristiky souboru, tj. hodnotu mediánu odpovědí a jejich průměrnou hodnotu. Z hodnot v tabulce je patrné, že nejvíc zkušeností mají respondenti s příkazovým řádkem v MS Windows, který dva respondenti běžně užívají a více než polovina z nich si práci s ním vyzkoušela. Další tři nástroje jsou na tom v hodnocení podobně. Komunikace se zařízením přes COM Port je o něco známější než Shell v Unixu a nejméně známá je pro respondenty komunikace, popř. ovládání zařízení přes SSH.

Tabulka 12-39 – Tabulka četnosti odpovědí pro okruh „Ostatní uživatelské zkušenosti“

Ostatní uživatelské zkušenosti	Četnosti odpovědí					Charakteristiky odpovědí	
	1	2	3	4	5	Medián	Průměr
Komunikace se zařízením přes COM Port (FM)	23	6	4	0	0	1,00	1,42
Komunikace, popř. ovládání zařízení přes SSH (F)	27	5	1	0	0	1,00	1,21
Příkazový řádek MS Windows (F)	5	9	17	2	0	3,00	2,48
Shell v Unixu (F)	25	5	2	1	0	1,00	1,36

Distribuce odpovědí respondentů pro jednotlivé hodnocené položky lze vidět v následujících boxplotech, viz Obrázek 12-21. Rozdíly mezi znalostmi byly testovány pomocí Kruskal-Wallisova testu, na jehož základě byla zamítnuta shoda v množství znalostí ( $p$ -hodnota  $<0,001$ ). Znalosti respondentů se tedy od sebe navzájem statisticky významně liší alespoň pro jednu dvojici technologií.



Obrázek 12-21– Boxplot zobrazující ostatní uživatelské zkušenosti

Na základě mnohonásobného porovnávání byl nalezen statisticky významný rozdíl mezi příkazovým řádkem MS Windows a všemi ostatními možnostmi ( $p$ -hodnoty  $<0,001$ ). Příkazový řádek tedy znají respondenti statisticky významně lépe než všechny ostatní technologie/nástroje, jejichž znalost se navzájem dle testů neliší.

V tomto případě mají uživatelé největší znalost v nástroji, který využívá architektura Fischertechnik Training Factory Industry 4.0, ta ale vyžaduje zároveň znalost dalších 3 technologií/nástrojů. Navrhovaná architektura vyžaduje znalost pouze komunikace se zařízením přes COM Port, ta je ale potřeba i pro konkurenční architekturu. Z toho důvodu můžeme říct, že i přes nižší hodnocení požadované znalosti dojde v případě použití navržené architektury k redukci nových potřebných odborných znalostí průmyslového inženýra z oblasti těchto nástrojů/technologií ze světa informatiky pro realizaci HW modelu s napojeným digitálním dvojčtem.

### 12.3.8 Shrnutí hypotézy 3

V rámci statistického vyhodnocení bylo posuzováno množství a typ znalostí z oblasti programování a informatiky pro navrženou architekturu a pro architekturu Fischertechnik Training Factory Industry 4.0. V rámci toho bylo vyhodnoceno, zda respondenti mají vyšší znalosti pro využití navržené architektury, nebo pro konkurenční architekturu. Zároveň bylo při vyhodnocení hypotézy bráno v potaz množství znalostí v jednotlivých podtématech pro jednotlivé architektury. Shrnutí vyhodnocení H3 pro jednotlivá podtémata jsou v následující tabulce, viz Tabulka 12-40.

Tabulka 12-40 – Shrnutí výsledků pro H3

Zaměření skupiny otázek	H3	Odůvodnění
Zkušenosti s technologiemi	Nepotvrzena	Podobné výsledky pro obě architektury, nelze jednoznačně rozhodnout.
Vývojová prostředí	Nepotvrzena	Obě architektury vyžadují stejné množství nových znalostí.
Ostatní programátorské zkušenosti	Potvrzena	Nižší množství požadovaných nových znalostí.
Uživatelská práce v OS	Potvrzena	Lepší znalosti pro navrhované řešení.
SW nástroje	Potvrzena	Znalosti z této oblasti navrhovaná architektura nevyžaduje.
Administrace SW nástrojů	Potvrzena	Nižší množství požadovaných nových znalostí, lepší znalosti pro navrhované řešení.
Ostatní uživatelské zkušenosti	Potvrzena	Nižší množství požadovaných nových znalostí.

Na základě výše uvedeného lze říct, že hypotéza 3 byla potvrzena. Důvodem je, že navrhovaná architektura vyžaduje buď stejné množství nových znalostí jako konkurenční řešení, nebo v některých oblastech jich vyžaduje významně méně. Úroveň znalostí respondentů byla v některých případech sice vyšší při hodnocení technologií, které využívá architektura Fischertechnik Training Factory Industry 4.0, ovšem ta vyžaduje zároveň další velké množství nových nástrojů a technologií, které respondenti neznají.

## 13 Použité vědecké metody

Pro teze disertační práce byly využívány vědecké metody, které napomáhaly dosažení teoretických i praktických cílů. Podle Hendla [149] jsou tyto vědecké metody zařaditelné do 3 skupin: empirické, logické a specifické vědecké metody.

### 13.1 Empirické vědecké metody

Tyto metody jsou postaveny na bezprostředním živém obrazu reality. Zahrnují takové metody, ve kterých se odraz jevů uskutečňuje prostřednictvím smyslových požitků a vjemů zdokonalovaných úrovní techniky. Těmito metodami je možné zjistit konkrétní jedinečné vlastnosti objektu či jevu v realitě. [150]

#### **Pozorování**

Sledování výuky předmětů zaměřených na informatiku, simulace a koncept Industry 4.0, přístupů k výuce ze strany vyučujících i studentů, reakce studentů. Sledování vnitřního i vnějšího prostředí výrobních podniků, jejich možností, omezení.

#### **Experimentování**

Proběhlo experimentování s modelem Fischertechnik Training Factory Industry 4.0 9V, se stavebnicí Fischertechnik Stem Engineering Kit a s řídicími jednotkami z rodiny Arduino, RaspberryPi, BBC Micro:bit, ... a různými programovacími jazyky (např. Wiring, Python, C/C++, programování pomocí grafických bloků). Experiment lze označit jako stěžejní metodu celého výzkumu. Nejdůležitější experimenty byly provedeny formou testování pro verifikaci a validaci navržené architektury jak z pohledu její realizovatelnosti a funkčnosti (ověřeno na modelu třídičky), tak vhodnosti pro průmyslové inženýry. Tyto experimenty zaměřené na vhodnost architektury pro využití průmyslovými inženýry byly následně statisticky vyhodnoceny.

Hendl [149] u empirických vědeckých metod uvádí i rešerši dostupné literatury. V rámci teoretické části disertační práce byla provedena rešerše, kterou představuje prostudování hlavně zahraniční literatury se zaměřením na výzkumné a aplikační aktivity výzkumných a akademických pracovníků. Dále byla provedena i rešerše (resp. analýza trhu) STEM produktů a produktů vztahujících se k learning factory a vzdělávání v konceptu Industry 4.0.

### 13.2 Logické vědecké metody

Tyto metody využívají principy logiky a logického myšlení, jsou to párové metody. Jednotlivé logické metody se vzájemně doplňují, kombinují a ve svém účinku překrývají – tím vytvářejí určitou synergii. Používají se pro pochopení stavů, jevů a vazeb. [150]

#### **Analýza – Syntéza**

Analýza je proces faktického, nebo myšlenkového rozčlenění celku (jevu, objektu) na části. Syntéza znamená postupovat od části k celku, dovoluje poznávat objekt jako jediný celek spojováním dílčích poznatků získaných analytickým přístupem. [150]

Tato kombinace metod se prolíná celou touto prací, která začíná definováním jednotlivých částí až po celky, jež tyto jednotlivé prvky využívají a kombinují, a vede k definování tezí a hypotéz.

#### **Abstrakce – konkretizace**

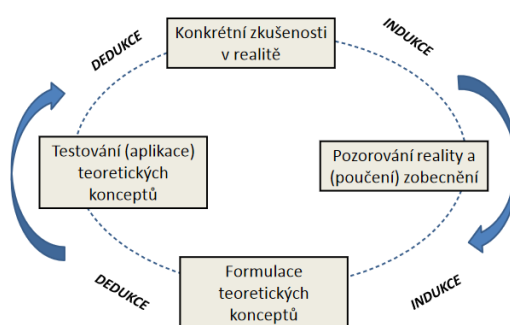
Abstrakce je myšlenkový proces, v jehož rámci se u různých objektů vydělují pouze jejich podstatné charakteristiky a nepodstatné se neuvažují. Konkretizace je opačný proces k abstrakci, kdy se vyhledávají konkrétní výskyty určitého objektu z určité třídy objektů a je snaha aplikovat charakteristiky platné pro tuto třídu objektů. [150]

Tato kombinace metod umožnila skutečně prostudovanou množinu zdrojů zúžit pouze na množinu využitelnou pro tuto disertační práci a zvolit směr, kterým se má práce dále ubírat. Zároveň umožňuje stanovovat důležité prvky např. pro porovnání a toto aplikovat při vynechávání nepodstatného.

### Indukce – dedukce

Induktivní přístup pracuje především s kvalitativními daty a sběr dat využívá k získání několika různých pohledů na daný problém. Na základě zjištěných poznatků se vytváří hypotézy, které se následně testují za účelem zobecnění a tím vzniku nové teorie. Deduktivní přístup umožňuje na základě teorie formulovat hypotézy a pomocí logického uvažování a známých faktů testuje tuto hypotézu. K tomu využívá hlavně kvantitativní data. [150]

Znalost této kombinace metod umožnila snáze definovat hypotézy, které budou dále ověřovány. Kolbův experimentální cyklus popisující kombinaci Indukce a Dedukce je na následujícím obrázku, viz Obrázek 13-1.



Obrázek 13-1 – Kolbův experimentální cyklus [150]

### 13.3 Specifické vědecké metody

V práci byly použity kromě empirických a logických vědeckých metod i specifické vědecké metody a to:

Aplikace systémového a holistického přístupu, který umožňuje na předmět zájmu nahlížet jako na systém a brát v úvahu jeho děje a části v souvislostech. Systémem je neprázdná množina prvků a množina vazeb mezi nimi, kde vlastnosti těchto prvků a vazby mezi nimi určují vlastnosti celku [149]. Tato metoda je důležitá pro logický a funkční návrh architektury HW learning factory s digitálním dvojčtem, který je potřeba vnímat jako komplexní systém, ale zároveň si být vědom jednotlivých procesů, které představuje a které se v něm odehrávají, a vztahů mezi nimi.

Tvůrčí metody, které vyžadují využití kreativity a určité dávky intuice autora. Zvyšují pravděpodobnost úspěšného vyřešení problému v průběhu tvůrčího procesu [149]. Tvůrčí metody byly důležité hlavně při experimentech s neznámými řídicími jednotkami.

Metoda porovnávání, která umožňuje nacházet společné rysy dvou jevů, nebo naopak to, co je odlišuje [149]. Tato metoda byla využita při experimentování s řídicími jednotkami a jejich programováním a byla využita i v následujících experimentech potřebných pro ověření navržené architektury.

## 14 Přínosy disertační práce

Obsahem této kapitoly je souhrn teoretických a praktických přínosů výzkumu zaměřeného na návrh architektury HW learning factory s digitálním dvojčetem a doporučení pro další výzkum v dané problematice.

### 14.1 Teoretický přínos práce

Teoretickým přínosem disertační práce je návrh architektury hardwarového modelu s digitálním dvojčetem ve formě learning/training factory, který kombinuje aktuálně dostupné technologie, nástroje a postupy s ohledem na cílovou skupinu průmyslových inženýrů strojírenského typu. Navržená architektura zohledňuje schopnosti a znalosti této cílové skupiny v oblastech, jako jsou programování, elektronika, kybernetika apod. To průmyslovým inženýrům umožňuje realizaci, úpravy a využití hw modelu s digitálním dvojčetem s ohledem na koncept Industry 4.0.

I když výzkumné týmy po celém světě často využívají různé learning/training factories, jejich výzkumníci se zaměřují na konkrétní výzkum, pro který si svou learning/training factory sestavili, ale nekladou žádný důraz na popis learning/training factory jako takové a její možnou reprodukci na jiném pracovišti. Stejně tak se nevěnují návrhu learning/training factory se zaměřením na určitou skupinu studentů nebo výzkumníků a jejich specifické nebo omezené znalosti. Tato práce se zaměřuje na uvedenou problematiku, tedy problematiku návrhu a možností realizace learning/training factory se zaměřením na studenty, pracovníky a výzkumníky z oboru průmyslového inženýrství strojírenského typu se zohledněním jejich znalostí programování, elektrotechniky, kybernetiky/robotiky a dalších oborů, které je potřeba kombinovat pro realizaci takovýchto studijních a výzkumných komplexních nástrojů typu learning/training factory. Navržená architektura bere v potaz i národní legislativu ČR omezující možnosti průmyslových inženýrů strojírenského typu, kteří nedisponují odbornou způsobilostí v elektrotechnice – touto způsobilostí mohou disponovat pouze lidé s elektrotechnickým vzděláním.

Výzkum v rámci disertační práce potvrdil, že navržená architektura hw learning factory s digitálním dvojčetem vyžaduje méně nových znalostí realizátorů z oblasti IT oproti standardnímu přístupu využívajícímu nejčastěji programovací jazyky pro řízení HW C/C++ a na to navázané další technologie. V případě realizace hw modelu pak bylo v práci prokázáno, že tvorba a případná úprava řídicího kódu pro hw model pomocí vizuálního programovacího jazyka je pro průmyslové inženýry snazší – tedy je vyšší úspěšnost při tvorbě nebo úpravě řídicího kódu. Zároveň bylo prokázáno, že tvorba/úprava řídicího kódu zabere průmyslovým inženýrům strojírenského typu méně času s lepšími výsledky oproti standardnímu využití jazyka C/C++, tedy průmysloví inženýři dosahují vyšší časové efektivity díky využití navržené architektury.

### 14.2 Praktický přínos práce

Navržená architektura pro training factory s digitálním dvojčetem využitelná průmyslovými inženýry strojírenského typu, kteří nemají pokročilé znalosti programování, elektrotechniky nebo kybernetiky umožňuje vytvářet si vlastní hw modely, popř. hw modely s digitálními dvojčaty a využívat je jak pro vzdělávání a lektoring, tak pro pracovní i vědeckou činnost např. pro testování hypotéz, nových postupů, algoritmů apod.

Jako první důležitý praktický přínos práce je aktuálně realizovaný komplexní model výrobní linky podle navržené teoretické architektury, který již prokázal svou využitelnost a vhodnost pro využití průmyslovými inženýry jak v některých semestrálních pracích studentů, tak hlavně v představených diplomových pracích, kdy průmysloví inženýři byli schopni pro danou hw learning/training factory realizovat mobilní aplikaci pro průběžné sledování stavu modelu výrobní linky nebo pro ni realizovali digitální stín ve formě 3D modelu ve virtuální realitě se schopností stínovat hw model výrobní linky v reálném čase.

Druhým důležitým praktickým přínosem je možnost využití části navržené a realizované architektury pro napojení na jiný i reálný systém a zpřístupnění digitálního dvojčete vytvořeného v nástroji Siemens Texnomatix Plant Simulation reálnému systému. Tento praktický přínos je zajištěný aplikací ControllModSim, která organizuje komunikaci s digitálním dvojčetem a pro komunikaci s reálnou částí (hw model nebo reálná výrobní linka) aktuálně využívá komunikaci přes USB serial port (COM Port), ale je možné aplikaci v budoucnu rozšířit i o další způsoby komunikace, pokud by to bylo potřeba.

Vedlejším a původně neplánovaným praktickým přínosem práce je naprogramovaná aplikace Questionnaire, která byla využita pro testování probandů při ověřování hypotéz 1 a 2. Tato aplikace umožňuje testovat libovolné množství probandů s náhodným přidělováním testů, přičemž se probandům počítá čas strávený na každé otázce a není jim umožněno se vracet k předchozím – to je potřebné chování v situacích, kdy následující otázky mohou dávat odpovědi na předchozí otázky, ale není žádoucí, aby probandí podle toho mohli předcházející odpovědi modifikovat. Tato aplikace je tedy využitelná v případech s obdobnými požadavky na průběh a způsob testování.

### 14.3 Doporučení pro další výzkum v dané problematice

V rámci výzkumu vhodnosti a realizovatelnosti navržené architektury hw learning factory s digitálním dvojčetem byla porovnávána vhodnost tří programovacích jazyků (C/C++, Python, vizuální programovací jazyk), přičemž nejdůležitější byl pro navrženou architekturu rozdíl mezi jazyky C/C++ a vizuálním programovacím jazykem z toho důvodu, že jazyk C/C++ je standardním přístupem pro programování HW a vizuální programovací jazyk byl považován za jazyk s největším potenciálem zjednodušení realizovatelnosti HW modelu průmyslovými inženýry a byl proto zvolen v rámci navržené architektury. V tomto případě bylo potvrzeno, že při využívání mezi nimi existuje v rámci navržené architektury statisticky významný rozdíl na hladině významnosti 5 %. V případě dvojice Python a Robo Pro Coding nebyl tento rozdíl potvrzen jako statisticky významný, ačkoliv rozdíly jsou z dat patrné. To je pravděpodobně způsobeno menšími rozdíly, které v tomto počtu respondentů nebylo možné pomocí statistických testů identifikovat. Bylo by tedy vhodné v budoucnu rozšířit testování na více respondentů se zaměřením na tuto kombinaci programovacích jazyků pro průmyslové inženýry strojírenského typu.

Navržená architektura cílí na průmyslové inženýry strojírenského typu, ovšem v dnešní době se rozmáhají také obory zaměřené na průmyslové inženýrství i na ekonomických vysokých školách. Dá se předpokládat, že tito průmysloví inženýři budou mít jinou základní sadu znalostí, ovšem i u nich může být HW training factory důležitým nástrojem jak ve vzdělávání, tak ve výzkumu. Proto by mohlo být vhodné ověřit vhodnost navržené architektury i na průmyslových inženýrech ekonomického typu a zároveň tak zjistit, jaké jsou rozdíly znalostí a schopností v dané problematice v porovnání s průmyslovými inženýry strojírenského typu.

Jedním z velkých trendů v poslední době využívaných nejen v rámci konceptu Industry 4.0 jsou umělá inteligence a machine learning. I v tomto případě se jedná o technologie náročné na znalosti pokročilého programování, pokročilé matematiky a elektrotechniky (např. klasické sensory se nahrazují kamerami s pokročilými funkcemi jako např. automatické ostření, noční vidění nebo vidění v různých spektrech apod). S postupem času je viditelná snaha společností jako Google, Microsoft a další zpřístupnit nástroje využívající funkcionality machine learning a umělé inteligence veřejnosti. S tím se otevírají možnosti zpřístupnění i průmyslovým inženýrům, kteří by mohli tyto technologie využívat a v určité míře a složitosti i implementovat. Když už pro průmyslové inženýry strojírenského typu tedy existuje architektura umožňující jim realizovat hardwarový model s digitálním dvojčetem, bylo by vhodné navrženou architekturu rozšířit o tyto technologie s obdobným přístupem reflektujícím jejich rozsah znalostí v oborech, jako je programování, elektrotechnika, kybernetika apod.

## 15 Diskuse

Hlavním cílem disertační práce je návrh architektury HW learning factory s digitálním dvojčtem, využívajícím diskrétní simulaci, pro vzdělávání v konceptu Industry 4.0. Tato architektura míří na vzdělávání průmyslových inženýrů v konceptu Industry 4.0 a využití v podnicích typu SME. Navržená architektura umožní průmyslovému inženýrovi modelovat a simulovat hlavní myšlenky konceptu Industry 4.0 s ohledem na jeho schopnosti a znalosti vyplývající ze vzdělání průmyslového inženýra. Architektura tedy musí reflektovat klady, ale i zápory a omezení vyplývající z „multioborovosti“ průmyslového inženýra. Těmi jsou často pouze základní schopnosti a znalosti programování a databází, základní znalosti elektrotechniky a s ní související legislativní omezení daná původně vyhláškou č. 50/1978 Sb. (do 1. 7. 2022) a nyní nařízením vlády 194/2022 Sb., podle kterých je průmyslový inženýr silně omezený a bez elektrotechnického vzdělání aktuálně nemůže zapojovat/přepojovat ani PLC a s ním související obvody, nebo má pouze základní znalosti z oblasti kybernetiky a dalších souvisejících oborů, které koncept Industry 4.0 využívá a jejichž znalosti implementuje.

Na základě stanoveného hlavního cíle a jemu odpovídajících dílčích cílů byly definovány hypotézy, které jsou ověřovány v rámci následujícího postupu disertační práce, a to:

### Hypotéza číslo 1

*Využitím navržené architektury training factory lze průmyslovým inženýrům zjednodušit realizaci a případnou modifikaci komplexního HW modelu schopného komunikovat se svým digitálním dvojčtem oproti standardnímu přístupu využívajícím pro řízení programovací jazyk C/C++, jenž je využit v případě Fischertechnik Training Factory Industry 4.0.*

### Hypotéza číslo 2

*Navržená architektura zvýší časovou efektivitu průmyslových inženýrů při úpravě řídicího kódu HW modelu se základními čidly využívanými v konceptu Industry 4.0 oproti programování HW standardními programovacími jazyky využitými u Fischertechnik Training Factory Industry 4.0 se zachováním možnosti komunikace HW modelu s okolím.*

### Hypotéza číslo 3

*Využitím navržené architektury místo architektury Fischertechnik Training Factory Industry 4.0 dojde k redukci nových potřebných odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčtem.*

Tyto hypotézy byly využity jako základ pro sběr a analýzu dat, které byly dále použity k vyhodnocení efektivity navržené architektury ve srovnání s tradičním přístupem.

Pro testování hypotéz H1 a H2 byla analyzována data od 30 respondentů, kteří plnili úkoly ve 4 okruzích a ve 3 programovacích jazycích. Výsledky ukazují, že jazyk Robo Pro Coding dosahoval nejlepších výsledků ve smyslu procentuálního hodnocení úloh a přepočtu bodů na jednotku času. Rozdíl mezi jazykem Robo Pro Coding a C/C++ byl statisticky významný ve většině případů. Rozdíl mezi jazykem Robo Pro Coding a Pythonem nebyl statisticky významný, výsledky ukazují vysokou variabilitu dovedností mezi respondenty. Tyto výsledky potvrzují hypotézy H1 a H2.

Pro testování hypotézy H3 byly analyzovány dotazníky od 33 respondentů. Hypotéza 3, týkající se znalostí a zkušeností v oblasti programování a informatiky pro navrženou architekturu a architekturu Fischertechnik Training Factory Industry 4.0, byla potvrzena. V některých oblastech vyžaduje navržená architektura stejné množství nových znalostí jako konkurenční řešení, ale v jiných oblastech vyžaduje méně nových znalostí. Respondenti vykazovali vyšší znalosti technologií používaných v architektuře Fischertechnik Training Factory Industry 4.0, ale tato architektura zároveň vyžaduje další nové nástroje a technologie, které respondenti neznají. Závěrem lze konstatovat, že znalosti a zkušenosti respondentů jsou klíčovým faktorem pro posouzení vhodnosti a přínosu navržené architektury.



Výsledky tohoto výzkumu ukazují, že navržená architektura hw learning factory s digitálním dvojčtem má potenciál usnadnit práci průmyslovým inženýrům, zejména ve srovnání s tradičním přístupem využívajícím jazyk C/C++. Statisticky významný rozdíl mezi jazykem C/C++ a vizuálním programovacím jazykem podporuje tvrzení, že vizuální programovací jazyk může být přístupnější a efektivnější pro realizaci hw modelu s digitálním dvojčtem. I když pro jazyk Python nebyl prokázán statisticky významný rozdíl, výsledky ukazují, že existují individuální preference a velmi variabilní úroveň dovedností mezi respondenty. Výsledky dotazníkového šetření potvrdily, že navržená architektura přináší redukci potřebných kompetencí ve srovnání s konkurenční architekturou Fischertechnik Training Factory Industry 4.0. Toto je důležité pro efektivní a rychlou implementaci a modifikaci komplexního HW modelu. Průmysloví inženýři tak nemusí získávat rozsáhlé nové znalosti z oblasti informatiky, což jim umožňuje rychleji a efektivněji využívat technologie Industry 4.0 ve své praxi.

V rámci této disertační práce je důležité zmínit některé limity výzkumu. Prvním limitem je omezení dané zaměřením na průmyslové inženýry strojírenského typu. Bylo by vhodné provést srovnání s průmyslovými inženýry ekonomického typu, kteří mají odlišný teoretický základ. Druhým limitem výzkumu byl počet respondentů v případě jazyka Python. Výsledky pro jazyk Python vykazovaly vysokou variabilitu, která by mohla být lépe vyjasněna s větším testovaným vzorkem. Navíc navržené rozšíření architektury o technologie umělé inteligence a machine learningu vyžaduje další výzkum pro určení nejvhodnějšího přístupu a implementace těchto technologií v rámci systému. Při interpretaci výsledků a plánování dalšího výzkumu v této oblasti je tedy důležité brát v úvahu tyto limity.

V rámci výzkumu byla porovnávána vhodnost tří programovacích jazyků (C/C++, Python, vizuální programovací jazyk) pro navrženou architekturu hw learning factory s digitálním dvojčtem. Byl zjištěn statisticky významný rozdíl mezi jazyky C/C++ a vizuálním programovacím jazykem, což naznačuje, že vizuální programovací jazyk má větší potenciál usnadnit práci průmyslovým inženýrům. V budoucnu by bylo vhodné rozšířit testování na více respondentů a zahrnout i průmyslové inženýry ekonomického typu. Dále by mohlo být užitečné zahrnout do navržené architektury technologie umělé inteligence a machine learningu, které jsou v současnosti rozšiřujícím se trendem a mohly by být přínosné pro průmyslové inženýry s různými znalostmi v programování, elektrotechnice a kybernetice.

Je také důležité si uvědomit, že navržená architektura nikdy nebude ve svém konečném stavu. S ohledem na neustále se rozvíjející a inovující HW a SW prostředky je nezbytné architekturu neustále aktualizovat a udržovat. V budoucnu můžeme očekávat nové možnosti a nástroje, které přinesou další možná zjednodušení a zároveň rozšíření a nárůst jejich možností využití a schopností. Proto je důležité sledovat vývoj těchto technologií a adaptovat navrženou architekturu s cílem zachovat její efektivitu a relevanci pro průmyslové inženýry v rámci konceptu Industry 4.0 a následujících budoucích trendů. Průběžné aktualizace a inovace architektury umožní využít nové příležitosti a zůstat v čele rychle se rozvíjejících technologických trendů.

Přínos navržené architektury pro HW learning factory s digitálním dvojčtem je hlavně v oblasti vzdělávání průmyslových inženýrů v konceptu Industry 4.0. Umožňuje jim efektivní modelování, simulaci a řízení hardwarových modelů bez nutnosti hlubších znalostí programování a elektrotechniky. HW model může být nahrazen i reálnou výrobní linkou, což umožní i sběr a zpracování reálných dat z reálné výrobní linky. Architektura obecně podporuje monitorování a optimalizaci výrobních procesů. Tím přispívá k zjednodušení, zvýšení efektivity a aplikaci konceptů Industry 4.0 v praxi i na akademické půdě a dále zlepšuje vzdělání průmyslových inženýrů a jejich schopnosti v moderním průmyslu.

## 16 Závěr

Hlavním cílem disertační práce je návrh architektury HW learning factory s digitálním dvojčetem, využívajícím diskretní simulaci, pro vzdělávání v konceptu Industry 4.0. Tato architektura cílí na vzdělávání průmyslových inženýrů v konceptu Industry 4.0 a využití v podnicích typu SME průmyslovými inženýry. Navržená architektura umožňuje průmyslovému inženýrovi modelovat a simulovat hlavní myšlenky konceptu Industry 4.0 s ohledem na jeho schopnosti a znalosti vyplývající ze vzdělání průmyslového inženýra strojírenského typu. Architektura tedy reflektuje klady, ale i omezení vyplývající z „multioborovosti“ průmyslového inženýra. Těmi jsou často pouze základní schopnosti a znalosti programování a databází, základní znalosti elektrotechniky a s ní související legislativní omezení daná původně vyhláškou č. 50/1978 Sb. (do 1. 7. 2022) a nyní nařízením vlády 194/2022 Sb., podle kterých by průmyslový inženýr bez elektrotechnického vzdělání neměl ani zapojovat/přepojovat PLC a s ním související obvody hojně využívané v Industry 4.0. Dalším omezením jsou pouze základní znalosti z oblasti kybernetiky a dalších souvisejících oborů, které koncept Industry 4.0 využívá, a jejichž znalosti implementuje.

Cíl práce byl definován s ohledem na teoretické závěry rešeršní části práce. Byla zjištěna skutečnost, že v prostudovaných odborných pramenech nebyla doposud řešena problematika návrhu nebo realizace learning/training factory s digitálním dvojčetem pro modelování a simulování hlavních myšlenek konceptu Industry 4.0, který by bral v potaz, že je určený pro vzdělávání průmyslového inženýra, nebo využívání samotným průmyslovým inženýrem (bez potřeby dalších pracovníků specifického odborného vzdělání) a zohledňoval by schopnosti a znalosti průmyslového inženýra vyplývající z jeho vzdělání a legislativní omezení vyplývající z tohoto vzdělání.

Po provedení rešerší, stanovení tezí a celkovém bádání v oblastech návrhu a realizace hardwarových modelů, digitálních dvojčat, simulací, všeobecně průmyslového inženýrství apod. byla navržena architektura, která cílí na maximální možné zjednodušení řízení hardware takovým způsobem, aby celek, tedy HW model s digitálním dvojčetem, byl realizovatelný a upravitelný v rámci možností, schopností a znalostí průmyslových inženýrů, kteří nejsou elektrotechniky, kybernetiky ani programátory, a z toho vyplynuly určité hypotézy. Z důvodu cílení architektury na skupinu průmyslových inženýrů (neprogramátorů) bylo pro programování řídicího kódu hardware zvoleno programování a řízení hardware pomocí vizuálního programovacího jazyka typu Scratch. Kromě samotného řízení HW daná architektura řeší i přenos dat mezi kontroléry řídicími hardware/hardwarový model a digitálním dvojčetem, proto architektura navrhuje i vhodný a v rámci možností nejjednodušší způsob komunikace mezi jednotlivými částmi architektury. Všechny části navržené architektury pro případ varianty HW modelu s digitálním dvojčetem byly per partes ověřeny na modelu třídičky. Navržená architektura má 7 základních částí, a to:

1. HW model
2. Řídící systémy hardware
3. Interface pro komunikaci s HW modelem a jeho řídicími systémy, které nemají přímou podporu komunikace přes USB/COM
4. Zpracování dat, zpřístupnění dat a informací, obousměrná synchronizace mezi HW a SW digitálním dvojčetem nebo stínem
5. Digitální dvojče nebo stín
6. Cloud (cloudový DB server s databází)
7. Možnosti napojení dalšího SW a HW

Po provedení návrhu architektury a ověření funkčnosti a realizovatelnosti HW modelu s digitálním dvojčetem vytvořeným dle navržené architektury byla ověřena její vhodnost pro využití průmyslovými inženýry na základě experimentálních testů a dotazníkového šetření. Tento výzkum pak byl zaměřen na jejich relevantní znalosti a zkušenosti s řízením HW. V rámci metodiky statistického vyhodnocení

byly hypotézy testovány vždy na hladině významnosti 5 %. Pro vizualizaci dat byly využívány boxploty. Experimentálních testů se zúčastnilo 30 probandů, dotazníkového šetření 33 probandů.

První hypotéza se zaměřila na schopnost průmyslových inženýrů realizovat řídicí program HW modelu v jazycích C/C++, Python a Robo Pro Coding (vizuální programovací jazyk), přičemž v rámci architektury se používá pro řízení HW modelu právě zmíněný vizuální programovací jazyk, popř. lze využívat i další vizuální programovací jazyky, které se používají stejným způsobem (např. pro řídicí jednotky BBC Micro:bit). Na základě dosažených výsledků lze konstatovat, že v případě procentuálního hodnocení řešených úloh dosahovali respondenti nejlepších výsledků při použití RoboPro Coding, naopak nejhorších výsledků dosahovali při použití C/C++. V případě Pythonu byly výsledky často různorodé a hodnoty procentuálního hodnocení měly široké rozpětí – našli se respondenti, kteří Python ovládali velmi dobře, ale zároveň se našli tací, kteří s ním neuměli vůbec pracovat. I to může být jeden z důvodů, proč statistické testy neoznačily rozdíl mezi jazyky Python a RoboPro Coding jako statisticky významný. Velikost rozptylu před samotným sběrem dat a analýzou nebylo možné odhadnout. Ovšem i tento velký rozptyl poukazuje na to, že využití RoboPro Coding je pro využití průmyslovými inženýry vhodnější než Python, protože v případě Robo Pro Coding měli respondenti schopnosti daný úkol vždy alespoň částečně vyřešit (žádné průměrné hodnocení úloh řešených v RoboPro Coding se nerovnálo 0 %), na rozdíl od C/C++ a Python, kde se ve všech 4 okruzích našel alespoň 1 respondent, který nebyl schopný vyřešit ani část úkolů (tj. měl 0 %). Na základě nasbíraných dat a jejich statistického vyhodnocení byla potvrzena první hypotéza ve znění: *Využitím navržené architektury training factory lze průmyslovým inženýrům zjednodušit realizaci a případnou modifikaci komplexního HW modelu schopného komunikovat se svým digitálním dvojčetem oproti standardnímu přístupu využívajícím pro řízení programovací jazyk C/C++, jenž je využit v případě Fischertechnik Training Factory Industry 4.0.*

Druhá hypotéza se zaměřila na časovou efektivitu průmyslových inženýrů při úpravě řídicího kódu HW modelu v jazycích C/C++, Python a Robo Pro Coding (vizuální programovací jazyk). Tato časová efektivita při využití navržené architektury byla testována jako přepočtení bodů hodnocení úkolů na jednotku času (1 min). V případě minima a mediánu časové efektivity byly ve všech případech dosaženy respondenty nejvyšší hodnoty v případě použití RoboPro Coding. Průměrná hodnota časové efektivity byla v případě okruhu otázek č. 3 nejvyšší u Pythonu, kde bylo zároveň i maximum, které pravděpodobně ovlivnilo i průměrnou hodnotu. Potvrzen byl statisticky významný rozdíl mezi C/C++ a RoboPro Coding u všech okruhů otázek. U dvojice RoboPro Coding a Python nebyl potvrzen statisticky významný rozdíl ani v jednom případě. Důvodem jsou v případě jazyka Python příliš rozptýlené hodnoty, a tedy by na otestování rozdílu mezi nimi bylo třeba více pozorování jako u předchozí hypotézy. Z výše uvedeného lze soudit, že v jazyce RoboPro Coding jsou všichni respondenti schopni dané úlohy alespoň částečně vyřešit, což je pro posouzení navržené architektury klíčové. Jazyk RoboPro Coding je tedy nejefektivnějším řešením z hlediska poměru procentuálního hodnocení a času na to potřebného. Na základě nasbíraných dat a jejich statistického vyhodnocení byla potvrzena druhá hypotéza ve znění: *Navržená architektura zvýší časovou efektivitu průmyslových inženýrů při úpravě řídicího kódu HW modelu se základními čidly využívanými v konceptu Industry 4.0 oproti programování HW standardními programovacími jazyky využitými u Fischertechnik Training Factory Industry 4.0 se zachováním možnosti komunikace HW modelu s okolím.*

Třetí hypotéza se zaměřila na porovnání potřebných nových znalostí z oblasti informatiky při využití navržené architektury nebo při využití architektury Fischertechnik Training Factory Industry 4.0 využívající řídicí jednotky Fischertechnik TXT Controller naprogramované v jazyce C/C++. V rámci statistického vyhodnocení bylo posuzováno množství a typ znalostí z oblasti programování a informatiky pro navrženou architekturu a pro architekturu Fischertechnik Training Factory Industry 4.0. V rámci toho bylo vyhodnoceno, zda respondenti mají více potřebných znalostí pro využití navržené architektury, nebo pro konkurenční architekturu. Zároveň bylo ale při vyhodnocení hypotézy bráno v potaz množství znalostí v jednotlivých podtématech pro jednotlivé architektury. Na základě

dotazníkového šetření a jeho statistického vyhodnocení byla potvrzena třetí hypotéza ve znění: *Využitím navržené architektury místo architektury Fischertechnik Training Factory Industry 4.0 dojde k redukci potřebných nových odborných znalostí průmyslového inženýra z oblasti informatiky pro realizaci HW modelu s napojeným digitálním dvojčetem.*

Navržená architektura HW modelu s digitálním dvojčetem umožňuje realizovat a upravovat HW modely s digitálním dvojčetem, popř. pouze její jednotlivé části podle potřeb průmyslových inženýrů nejen pro jejich vzdělávání, ale i pro ověřování nových technik, technologií, postupů, algoritmů apod. Např. při ověřování nového IT systému komunikujícího s reálnou výrobou lze minimálně ve vývojové etapě ověřovat a testovat funkčnost na HW modelu simulujícím skutečnou výrobu jak z pohledu posílaných, tak přijímaných dat a příkazů. Při dodržení podporovaných způsobů komunikace aplikace ControllModSim lze tuto aplikaci s digitálním dvojčetem napojit i na jiný reálný systém, než je HW model. V případě potřeby lze pak aplikaci ControllModSim rozšířit i o další způsoby komunikace. Využitelnost learning factory realizované podle navržené architektury se podařilo prokázat nejen výzkumem v rámci disertační práce, ale i studenty průmyslového inženýrství v rámci jejich diplomových prací. Ti pro komplexní HW model výrobní linky vytvořili mobilní aplikaci pro sledování jejího stavu a stavu výroby nebo digitální stín ve formě modelu ve virtuální realitě, který stínuje model linky v reálném čase. Tyto diplomové práce byly úspěšně obhájeny. V rámci předmětu simulace ve strojírenském podniku je model třídičky s digitálním stínem využíván v rámci cvičení a v rámci semestrálních prací jsou jednotlivé části výrobní linky využívány pro tvorbu digitálních stínů v prostředí Tecnomatix Plant Simulation.

## Seznam literatury

- [1] MAŠÍN, Ivan a Milan VYTLAČIL. *Cesty k vyšší produktivitě: strategie založená na průmyslovém inženýrství*. Liberec: Institut průmyslového inženýrství, 1996. ISBN 978-80-902235-0-9.
- [2] Slovník průmyslového inženýrství. *BusinessInfo.cz* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.businessinfo.cz/navody/slovník-prumysloveho-inzenyrstvi/>
- [3] KOSKY, Philip, Robert BALMER, William KEAT a George WISE. *Exploring engineering*. 5. vyd. Philadelphia: Elsevier Inc, 2020. ISBN 978-0-12-815073-3.
- [4] *Průmyslový inženýr/ NSP.CZ* [online]. [vid. 2023-02-20]. Dostupné z: <https://nsp.cz/jednotka-prace/prumyslovy-inzenyr>
- [5] CULOT, Giovanna, Guido NASSIMBENI, Guido ORZES a Marco SARTOR. Behind the definition of Industry 4.0: Analysis and open questions. *International Journal of Production Economics* [online]. 2020, **226**, 107617. ISSN 09255273. Dostupné z: doi:10.1016/j.ijpe.2020.107617
- [6] THOBEN, Klaus-Dieter, Stefan WIESNER a Thorsten WUEST. “Industrie 4.0” and Smart Manufacturing – A Review of Research Issues and Application Examples. *International Journal of Automation Technology* [online]. 2017, **11**(1), 4–16. ISSN 1883-8022, 1881-7629. Dostupné z: doi:10.20965/ijat.2017.p0004
- [7] OZTEMEL, Ercan a Samet GURSEV. *Literature review of Industry 4.0 and related technologies* [online]. VAN GODEWIJCKSTRAAT 30, 3311 GZ DORDRECHT, NETHERLANDS: SPRINGER. leden 2020. ISSN 0956-5515. Dostupné z: doi:10.1007/s10845-018-1433-8
- [8] VAIDYA, Saurabh, Prashant AMBAD a Santosh BHOSLE. Industry 4.0 – A Glimpse. *Procedia Manufacturing* [online]. 2018, **20**, 233–238. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2018.02.034
- [9] LAMPROPOULOS, Georgios, Kerstin SIAKAS a Theofylaktos ANASTASIADIS. INTERNET OF THINGS IN THE CONTEXT OF INDUSTRY 4.0: AN OVERVIEW. *International Journal of Entrepreneurial Knowledge* [online]. 2019, **7**(1), 4–19. ISSN 2336-2960, 2336-2952. Dostupné z: doi:10.37335/ijek.v7i1.84
- [10] HOFFMANN, Max. *Smart agents for the Industry 4. 0: enabling machine learning in industrial production* [online]. Wiesbaden: Springer Vieweg, 2019. ISBN 978-3-658-27742-0. Dostupné z: <https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=5896709>
- [11] ANGELOPOULOS, Angelos, Emmanouel T. MICHAILIDIS, Nikolaos NOMIKOS, Panagiotis TRAKADAS, Antonis HATZIEFREMIDIS, Stamatis VOLIOTIS a Theodore ZAHARIADIS. Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects. *Sensors* [online]. 2020, **20**(1). ISSN 1424-8220. Dostupné z: doi:10.3390/s20010109
- [12] OJRA, Ahmad. Revisiting Industry 4.0: A New Definition. In: Kohei ARAI, Supriya KAPOOR a Rahul BHATIA, ed. *Intelligent Computing* [online]. Cham: Springer International Publishing, 2019, Advances in Intelligent Systems and Computing, s. 1156–1162. ISBN 978-3-030-01173-4. Dostupné z: doi:10.1007/978-3-030-01174-1\_88

- [13] HAN, Soonhung. A review of smart manufacturing reference models based on the skeleton meta-model. *Journal of Computational Design and Engineering* [online]. 2020, **7**(3), 323–336. ISSN 2288-5048. Dostupné z: doi:10.1093/jcde/qwaa027
- [14] IEC PAS. *IEC PAS 63088:2017 Smart Manufacturing – Reference Architecture Model Industry 4.0 (RAMI4.0)*. 2017
- [15] KREJCI, Jan a Michal AMBLER. Průmysl 4.0: národní, firemní a akademické přístupy. *Contemporary Europe*. 2017, **2017**(2), 46–62. ISSN 18041280.
- [16] YÁÑEZ, F. *The Goal is Industry 4.0: Technologies and Trends of the Fourth Industrial Revolution*. B.m.: Independent, 2017. ISBN 978-1-973413-17-2.
- [17] MAŘÍK, Vladimír. *Průmysl 4.0: výzva pro Českou republiku*. Vydání 1. Praha: Management Press, 2016. ISBN 978-80-7261-440-0.
- [18] BARTODZIEJ, C. J. *The concept industry 4.0*. New York, NY: Springer Berlin Heidelberg, 2016. ISBN 978-3-658-16501-7.
- [19] GILCHRIST, Alasdair. *Industry 4.0: The Industrial Internet of Things* [online]. 1st ed. 2016. Berkeley, CA: Apress: Imprint: Apress, 2016. ISBN 978-1-4842-2047-4. Dostupné z: doi:10.1007/978-1-4842-2047-4
- [20] VRCHOTA, Jaroslav, Tomas VOLEK a Martina NOVOTNÁ. Factors Introducing Industry 4.0 to SMES. *Social Sciences* [online]. 2019, **8**(5), 130. ISSN 2076-0760. Dostupné z: doi:10.3390/socsci8050130
- [21] SOMMER, Lutz. Industrial revolution - industry 4.0: Are German manufacturing SMEs the first victims of this revolution? *Journal of Industrial Engineering and Management* [online]. 2015, **8**(5), 1512–1532. ISSN 2013-0953, 2013-8423. Dostupné z: doi:10.3926/jiem.1470
- [22] ŠOLTÉS, Michal. *Průmysl 4.0: Jaká je startovní pozice České republiky? - Roklen24.cz*. 28. říjen 2016.
- [23] Industry 5.0. *European Economic and Social Committee* [online]. 16. listopad 2018 [vid. 2023-02-20]. Dostupné z: <https://www.eesc.europa.eu/en/agenda/our-events/events/industry-50>
- [24] SACHSENMEIER, Peter. Industry 5.0—The Relevance and Implications of Bionics and Synthetic Biology. *Engineering* [online]. 2016, **2**(2), 225–229. ISSN 2095-8099. Dostupné z: doi:10.1016/J.ENG.2016.02.015
- [25] DEMIR, Kadir a Halil CICIBAŞ. INDUSTRY 5.0 AND A CRITIQUE OF INDUSTRY 4.0. In: *4th International Management Information Systems Conference*. 2017.
- [26] NAHAVANDI, Saeid. Industry 5.0—A Human-Centric Solution. *Sustainability* [online]. 2019, **11**(16), 4371. ISSN 2071-1050. Dostupné z: doi:10.3390/su11164371
- [27] ÖZDEMİR, Vural a Nezih HEKİM. Birth of Industry 5.0: Making Sense of Big Data with Artificial Intelligence, “The Internet of Things” and Next-Generation Technology Policy. *OMICS: A Journal of Integrative Biology* [online]. 2018, **22**(1), 65–76. Dostupné z: doi:10.1089/omi.2017.0194

- [28] BANKS, Jerry, ed. *Handbook of simulation: principles, methodology, advances, applications, and practice*. New York : [Norcross, Ga.]: Wiley, ; Co-published by Engineering & Management Press, 1998. ISBN 978-0-471-13403-9.
- [29] SUMMERS, Richard L. Computer Simulation Studies and the Scientific Method. *Journal of Applied Animal Welfare Science* [online]. 1998, 1(2), 119–131. ISSN 1088-8705. Dostupné z: doi:10.1207/s15327604jaws0102\_3
- [30] PEGDEN, C. Dennis, Robert E. SHANNON a Randall P. SADOWSKI. *Introduction to simulation using SIMAN*. New York: McGraw-Hill, 1990. ISBN 978-0-07-049217-2.
- [31] PELÁNEK, Radek. *Modelování a simulace komplexních systémů: Jak lépe porozumět světu*. 2011. ISBN 978-80-210-5807-1.
- [32] BANKS, Jerry. *Discrete-event system simulation*. 5. ed., new internat. ed. Harlow: Pearson, 2014. Always learning. ISBN 978-1-292-02437-0.
- [33] ELIZANDRO, David a Hamdy A. TAHA. *Simulation of industrial systems: discrete event simulation using Excel/VBA*. New York: Auerbach Publications, 2008. Engineering and management innovation series. ISBN 978-1-4200-6744-6.
- [34] LAW, Averill M. *Simulation modeling and analysis*. Fifth edition. Dubuque: McGraw-Hill Education, 2013. McGraw-Hill series in industrial engineering and management science. ISBN 978-0-07-340132-4.
- [35] SIEMENS PRODUCT LIFECYCLE MANAGEMENT SOFTWARE INC. *SimTalk Reference* [online]. [vid. 2023-02-20]. Dostupné z: [https://docs.plm.automation.siemens.com/content/plant\\_sim\\_help/15/plant\\_sim\\_all\\_in\\_one\\_html/en\\_US/tecnomatix\\_plant\\_simulation\\_help/simtalk\\_reference/simtalk\\_reference.html](https://docs.plm.automation.siemens.com/content/plant_sim_help/15/plant_sim_all_in_one_html/en_US/tecnomatix_plant_simulation_help/simtalk_reference/simtalk_reference.html)
- [36] DEBNÁR, Róbert, Ján KOŠTURIK a Ivan KURIC. *SIMULÁCIA ako nástroj pre zvyšovanie produktivity a zisku v podniku*.
- [37] LANG, Sebastian, Tobias REGGELIN, Marcel MÜLLER a Abdulrahman NAHHAS. Open-source discrete-event simulation software for applications in production and logistics: An alternative to commercial tools? *Procedia Computer Science* [online]. 2021, 180, 978–987. ISSN 18770509. Dostupné z: doi:10.1016/j.procs.2021.01.349
- [38] ROSEN, Roland, Georg VON WICHERT, George LO a Kurt D. BETTENHAUSEN. About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC-PapersOnLine* [online]. 2015, 48(3), 567–572. ISSN 24058963. Dostupné z: doi:10.1016/j.ifacol.2015.06.141
- [39] WANG, Zongyan. Digital Twin Technology. In: Tamás BÁNYAI a Antonella PETRILLOAND FABIO DE FELICE, ed. *Industry 4.0 - Impact on Intelligent Logistics and Manufacturing* [online]. B.m.: IntechOpen, 2020. ISBN 978-953-51-6996-3. Dostupné z: doi:10.5772/intechopen.80974
- [40] KRITZINGER, Werner, Matthias KARNER, Georg TRAAR, Jan HENJES a Wilfried SIHN. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* [online]. 2018, 51(11), 1016–1022. ISSN 24058963. Dostupné z: doi:10.1016/j.ifacol.2018.08.474

- [41] KIM, Yong-Woon. Digital Twin maturity model [online]. 2020. Dostupné z: doi:10.13140/RG.2.2.28750.48967
- [42] HARTMANN, Dirk a Herman VAN DER AUWERAER. Digital Twins. In: Manuel CRUZ, Carlos PARÉS a Peregrina QUINTELA, ed. *Progress in Industrial Mathematics: Success Stories* [online]. Cham: Springer International Publishing, 2021, SEMA SIMAI Springer Series, s. 3–17. ISBN 978-3-030-61843-8. Dostupné z: doi:10.1007/978-3-030-61844-5\_1
- [43] FRYER, T. Digital Twin - Introduction. This is the age of The Digital Twin. *Engineering & Technology* [online]. 2019, **14**(1), 28–29. ISSN 1750-9637, 1750-9645. Dostupné z: doi:10.1049/et.2019.0125
- [44] ANDRONAS, Dionisis, George KOKOTINIS a Sotiris MAKRIS. On modelling and handling of flexible materials: A review on Digital Twins and planning systems. *Procedia CIRP* [online]. 2021, **97**, 447–452. ISSN 22128271. Dostupné z: doi:10.1016/j.procir.2020.08.005
- [45] CENTOMO, Stefano, Nicola DALL'ORA a Franco FUMMI. The Design of a Digital-Twin for Predictive Maintenance. In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA): 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* [online]. Vienna, Austria: IEEE, 2020, s. 1781–1788. ISBN 978-1-72818-956-7. Dostupné z: doi:10.1109/ETFA46521.2020.9212071
- [46] RAO, Samvith Vasudeva. Using a Digital Twin in Predictive Maintenance. *Journal of Petroleum Technology* [online]. 2020, **72**(08), 42–44. ISSN 0149-2136, 1944-978X. Dostupné z: doi:10.2118/0820-0042-JPT
- [47] LILJANIEMI, Antti a Heikki PAAVILAINEN. Using Digital Twin Technology in Engineering Education – Course Concept to Explore Benefits and Barriers. *Open Engineering* [online]. 2020, **10**(1), 377–385. ISSN 2391-5439. Dostupné z: doi:10.1515/eng-2020-0040
- [48] SUH, Sang C., U. John TANIK, John N. CARBONE a Abdullah EROGLU, ed. *Applied Cyber-Physical Systems* [online]. New York, NY: Springer New York, 2014. ISBN 978-1-4614-7335-0. Dostupné z: doi:10.1007/978-1-4614-7336-7
- [49] LV, Zhihan, Dongliang CHEN, Ranran LOU a Ammar ALAZAB. Artificial intelligence for securing industrial-based cyber–physical systems. *Future Generation Computer Systems* [online]. 2021, **117**, 291–298. ISSN 0167739X. Dostupné z: doi:10.1016/j.future.2020.12.001
- [50] VOGEL-HEUSER, Birgit, Jay LEE a Paulo LEITÃO. Agents enabling cyber-physical production systems. *at - Automatisierungstechnik* [online]. 2015, **63**(10). ISSN 0178-2312, 2196-677X. Dostupné z: doi:10.1515/auto-2014-1153
- [51] TAO, Fei, Qinglin QI, Lihui WANG a A.Y.C. NEE. Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering* [online]. 2019, **5**(4), 653–661. ISSN 20958099. Dostupné z: doi:10.1016/j.eng.2019.01.014
- [52] UHLEMANN, Thomas H.-J., Christian LEHMANN a Rolf STEINHILPER. The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0. *Procedia CIRP* [online]. 2017, **61**, 335–340. ISSN 22128271. Dostupné z: doi:10.1016/j.procir.2016.11.152
- [53] LU, Yuqian, Chao LIU, Kevin I-Kai WANG, Huiyue HUANG a Xun XU. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and*



- Computer-Integrated Manufacturing* [online]. 2020, **61**, 101837. ISSN 07365845. Dostupné z: doi:10.1016/j.rcim.2019.101837
- [54] LEE, Jay, Behrad BAGHERI a HUNG-AN KAO. Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics [online]. 2014. Dostupné z: doi:10.13140/2.1.1464.1920
- [55] GOODWIN, Melvin, Jacqueline HEALY, Kristen JACKSA a James WHITEHAIR. Strategies to address major obstacles to STEM-based education. In: *2017 IEEE Integrated STEM Education Conference (ISEC): 2017 IEEE Integrated STEM Education Conference (ISEC)* [online]. Princeton, NJ, USA: IEEE, 2017, s. 156–158. ISBN 978-1-5090-5379-7. Dostupné z: doi:10.1109/ISECon.2017.7910233
- [56] PAIGE, Kathryn, Lisa O'KEEFFE, Ruth GEER, Denise MACGREGOR a Debra PANIZZON. Using artefacts to articulate teachers' perceptions of STEM. *Teaching Science*. 2019, **65**(1), 48–54. ISSN 1449-6313.
- [57] NITE, Sandra B., Mary MARGARET, Robert M. CAPRARO, Jim MORGAN a Cheryl Ann PETERSON. Science, technology, engineering and mathematics (STEM) education: A longitudinal examination of secondary school intervention. In: *2014 IEEE Frontiers in Education Conference (FIE): 2014 IEEE Frontiers in Education Conference (FIE) Proceedings* [online]. Madrid, Spain: IEEE, 2014, s. 1–7. ISBN 978-1-4799-3922-0. Dostupné z: doi:10.1109/FIE.2014.7044214
- [58] CLARK, Clayton J., Tiffany W. ARDLEY a Jason T. BLACK. The program of excellence in STEM: Involvement of traditionally underrepresented students in STEM education through research and mentoring at Florida A&M University. In: *2015 IEEE Integrated STEM Education Conference (ISEC): 2015 IEEE Integrated STEM Education Conference* [online]. Princeton, NJ, USA: IEEE, 2015, s. 159–163. ISBN 978-1-4799-1829-4. Dostupné z: doi:10.1109/ISECon.2015.7119915
- [59] MIORELLI, Jonathan, Nicholas STAMBACH, Barbara MOSKAL a Jerry DWYER. Improving faculty perception of and engagement in STEM education. In: *2015 IEEE Frontiers in Education Conference (FIE): 2015 IEEE Frontiers in Education Conference (FIE)* [online]. Camino Real El Paso, El Paso, TX, USA: IEEE, 2015, s. 1–6. ISBN 978-1-4799-8454-1. Dostupné z: doi:10.1109/FIE.2015.7344220
- [60] KNEY, Arthur D., J. Christian TATU, Matthew MARLIN a Xiaoyan MENG. Transforming STEM to STEAM (Work in Progress): How a traditionally run STEM camp successfully incorporated the arts into its framework. In: *2016 IEEE Integrated STEM Education Conference (ISEC): 2016 IEEE Integrated STEM Education Conference (ISEC)* [online]. Princeton, NJ, USA: IEEE, 2016, s. 1–4. ISBN 978-1-4673-9773-5. Dostupné z: doi:10.1109/ISECon.2016.7457470
- [61] CONRADTY, Cathérine a Franz X. BOGNER. From STEM to STEAM: Cracking the Code? How Creativity & Motivation Interacts with Inquiry-based Learning. *Creativity Research Journal* [online]. 2019, **31**(3), 284–295. ISSN 1040-0419, 1532-6934. Dostupné z: doi:10.1080/10400419.2019.1641678
- [62] BICER, Ali, Sandra B. NITE, Robert M. CAPRARO, Luciana R. BARROSO, Mary M. CAPRARO a Yujin LEE. Moving from STEM to STEAM: The effects of informal STEM learning on students' creativity and problem solving skills with 3D printing. In: *2017 IEEE Frontiers in Education*

- Conference (FIE): 2017 IEEE Frontiers in Education Conference (FIE)* [online]. Indianapolis, IN: IEEE, 2017, s. 1–6. ISBN 978-1-5090-5920-1. Dostupné z: doi:10.1109/FIE.2017.8190545
- [63] STEM vs. STEAM vs. STREAM: What's the Difference? *Niche* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.niche.com/blog/stem-vs-steam-vs-stream/>
- [64] *Koncept STEM, Národní pedagogický institut České republiky (dříve Národní ústav pro vzdělávání)* [online]. [vid. 2023-02-20]. Dostupné z: <http://www.nuv.cz/p-kap/koncept-stem>
- [65] AMR, Mostaafaa. How to Build a DIY Arduino-Based Smart Home Hub with 1Sheeld. *Arduino Project Hub* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.hackster.io/amrmostaafaa/how-to-build-a-diy-arduino-based-smart-home-hub-with-1sheeld-79d405>
- [66] BACHMANNOVÁ, Jarmila. *Naše řeč – Bastlovat, bastlit* [online]. [vid. 2023-02-20]. Dostupné z: <http://nase-rec.ujc.cas.cz/archiv.php?art=6725>
- [67] IT SLOVNÍK.CZ. *Kdo je to Bastlíř? - IT Slovník* [online]. [vid. 2023-02-20]. Dostupné z: <https://it-slovník.cz/pojem/bastlir>
- [68] SUSILO, Ekawahyu, Jianing LIU, Yasmin ALVARADO RAYO, Ashley Melissa PECK, Justin MONTENEGRO, Mark GONYEA a Pietro VALDASTRI. STORMLab for STEM Education: An Affordable Modular Robotic Kit for Integrated Science, Technology, Engineering, and Math Education. *IEEE Robotics & Automation Magazine* [online]. 2016, **23**(2), 47–55. ISSN 1070-9932. Dostupné z: doi:10.1109/MRA.2016.2546703
- [69] WANG, Wei-hsing. A mini experiment of offering STEM education to several age groups through the use of robots. In: *2016 IEEE Integrated STEM Education Conference (ISEC): 2016 IEEE Integrated STEM Education Conference (ISEC)* [online]. Princeton, NJ, USA: IEEE, 2016, s. 120–127. ISBN 978-1-4673-9773-5. Dostupné z: doi:10.1109/ISECon.2016.7457516
- [70] KORKMAZ, Özgen. The Effect of Lego Mindstrom Eve 3 Based Design Activities on Students' Attitudes towards Learning Computer Programming, Self-efficacy Beliefs and Levels of Academic Achievement. *Baltic Journal of Modern Computing* [online]. 2016, **4**(4). ISSN 22558950. Dostupné z: doi:10.22364/bjmc.2016.4.4.24
- [71] GENSHEIMER, Matthias, Erich MÜLLER a Georg HAGEL. A hands-on approach on software engineering education using fischertechnik hardware. In: *ECSEE '20: European Conference on Software Engineering Education: Proceedings of the 4th European Conference on Software Engineering Education* [online]. Seeon/Bavaria Germany: ACM, 2020, s. 71–75. ISBN 978-1-4503-7752-2. Dostupné z: doi:10.1145/3396802.3396812
- [72] MERKURTOYS. *Novinky - Produkty - Svět Merkur* [online]. [vid. 2023-02-20]. Dostupné z: <https://eshop.merkurtoys.cz/k7-robotika>
- [73] *SmartLab Toys | SmartLab Toys Books | Book Publisher SmartLab Toys* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.smartlabtoys.com/>
- [74] MICRO:BIT EDUCATIONAL FOUNDATION. *Micro:bit Lessons* [online]. [vid. 2023-02-20]. Dostupné z: <https://microbit.org/>

- [75] RASPBERRY PI FOUNDATION. RaspberryPi Online learning. *Raspberry Pi* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.raspberrypi.org/courses/learn-python>
- [76] ARDUINO AG. *Arduino Tutorials* [online]. [vid. 2023-02-20]. Dostupné z: <https://docs.arduino.cc/tutorials/>
- [77] MEET EDISON. Robot programming software- free software for the Edison robot. *Meet Edison* [online]. [vid. 2023-02-20]. Dostupné z: <https://meetedison.com/robot-programming-software/>
- [78] HEATH, Nick. Raspberry Pi 4 Model B review: This board really can replace your PC. *TechRepublic* [online]. 23. červen 2019 [vid. 2023-02-20]. Dostupné z: <https://www.techrepublic.com/article/raspberry-pi-4-model-b-review-this-board-really-can-replace-your-pc/>
- [79] REHMAN, Yawar, GhouS Bakhsh NAREJO a Shoaib ZAIDI. An experiment for testing efficiency of effective teaching model and comprehensive use of limited resources. In: *2012 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE): Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) 2012* [online]. Hong Kong, China: IEEE, 2012, s. W2B-1-W2B-3. ISBN 978-1-4673-2418-2. Dostupné z: doi:10.1109/TALE.2012.6360294
- [80] ZOKOWSKI, Peg, Katharine GERAMITA, Jonathan ASHDOWN, Brenda BROOKS a Amanda THOMPkins. Connecting Kids to STEM through entrepreneurship and innovation. In: *2016 IEEE Integrated STEM Education Conference (ISEC): 2016 IEEE Integrated STEM Education Conference (ISEC)* [online]. Princeton, NJ, USA: IEEE, 2016, s. 71–74. ISBN 978-1-4673-9773-5. Dostupné z: doi:10.1109/ISECon.2016.7457557
- [81] KOBLASA, František, Eva ŠÍROVÁ a Růžena KRÁLÍKOVÁ. The Use of Process Thinking in the Industrial Practice – Preliminary Survey. *Tehnicki vjesnik - Technical Gazette* [online]. 2019, **26**(3). ISSN 13303651, 18486339. Dostupné z: doi:10.17559/TV-20150617135306
- [82] VOTRUBEC, Radek a Frantisek KOBLASA. Control System of Vehicle for Smart Factory Model with Principles of Industry 4.0. In: Branko KATALINIC, ed. *DAAAM Proceedings* [online]. 1. vyd. B.m.: DAAAM International Vienna, 2019, s. 0261–0267. ISBN 978-3-902734-22-8. Dostupné z: doi:10.2507/30th.daaam.proceedings.034
- [83] SANFILIPPO, Filippo a Kolbjorn AUSTRENG. Enhancing Teaching Methods on Embedded Systems with Project-Based Learning. In: *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE): 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* [online]. Wollongong, NSW: IEEE, 2018, s. 169–176. ISBN 978-1-5386-6522-0. Dostupné z: doi:10.1109/TALE.2018.8615221
- [84] ANDRAMUNO, Joselyne, Nino VEGA a Pablo PARRA. Industry 4.0 Embedded Systems Network. In: *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON): 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)* [online]. Valparaiso, Chile: IEEE, 2019, s. 1–6. ISBN 978-1-72813-185-6. Dostupné z: doi:10.1109/CHILECON47746.2019.8987620
- [85] GARCIA, Carlos A., Franklin SALAZAR L., Christian MARINO, Williams R. VILLALBA a Marcelo V. GARCIA. Design of Flexible Cyber-Physical Production Systems Architecture for Industrial

- Robot Control. In: *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM): 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)* [online]. Cuenca: IEEE, 2018, s. 1–6. ISBN 978-1-5386-6657-9. Dostupné z: doi:10.1109/ETCM.2018.8580338
- [86] FESTO DIDACTIC SE. *CP Factory – The Cyber-Physical Factory - CP Factory - Learning systems for Industry 4.0 - - Výukové systémy - Odborné vzdělávání a výukové systémy Festo Didactic* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.festo-didactic.com/cz-cs/vyukove-systemy/learning-systems-for-industry-4.0/cp-factory/cp-factory-the-cyber-physical-factory.htm?fbid=Y3ouY3MuNTQzLjMxLjE4LjE3ODUuNzY0Mw>
- [87] FESTO DIDACTIC SE. *CP Lab 404-1, with four application modules - CP Lab systems (I4.0) - Learning factories - Factory automation & Industry 4.0 - Výukové systémy - Odborné vzdělávání a výukové systémy Festo Didactic* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.festo-didactic.com/int-en/learning-systems/factory-automation-industry-4.0/learning-factories/cp-lab-systems-i4.0/cp-lab-404-1,with-four-application-modules.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC4xNzg3LjEwMjkxMQ>
- [88] FISCHERTECHNIK GMBH. *Digitized learning and simulation model for industry 4.0 applications* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.fischertechnik.de/en/products/simulating/training-models/551584-sim-training-factory-industry-4-0-9v-simulation>
- [89] FISCHERTECHNIK GMBH. *Factory Simulation 9V - Simulation - fischertechnik* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.fischertechnik.de/en/products/simulating/training-models/551584-sim-training-factory-industry-4-0-9v>
- [90] FISCHERTECHNIK GMBH. *Factory Simulation 24V - Simulation - fischertechnik* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.fischertechnik.de/en/products/simulating/training-models/536634-sim-factory-simulation-24v-simulation>
- [91] STAVEBNICE PRO CHYTRÉ DĚTI. Programovatelná minilinka s MG. *Stavebnice pro chytré děti* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.stavebniceprochytredeti.cz/cs/merkur-112227-programovatelna-minilinka-450.html>
- [92] LEGISLATIVA ČR. 50/1978 Sb. Vyhláška o odborné způsobilosti v elektrotechnice. *Zákony pro lidi* [online]. [vid. 2023-02-20]. Dostupné z: <https://www.zakonyprolidi.cz/cs/1978-50>
- [93] LEGISLATIVA ČR. 194/2022 Sb. Nařízení vlády o požadavcích na odbornou způsobilost k výkonu činnosti na elektrických zařízeních a ... *Zákony pro lidi* [online]. [vid. 2023-03-16]. Dostupné z: <https://www.zakonyprolidi.cz/cs/2022-194>
- [94] NÁRODNÍ SOUSTAVA POVOLÁNÍ. *Programátor PLC| NSP.CZ* [online]. [vid. 2023-02-20]. Dostupné z: <https://nsp.cz/jednotka-prace/programator-plc>
- [95] ATORF, Linus a Jurgen ROSSMANN. Interactive Analysis and Visualization of Digital Twins in High-Dimensional State Spaces. In: *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV): 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)* [online]. Singapore: IEEE, 2018, s. 241–246. ISBN 978-1-5386-9582-1. Dostupné z: doi:10.1109/ICARCV.2018.8581126

- [96] JAENSCH, Florian, Akos CSISZAR, Christian SCHEIFELE a Alexander VERL. Digital Twins of Manufacturing Systems as a Base for Machine Learning. In: *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP): 2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)* [online]. Stuttgart: IEEE, 2018, s. 1–6. ISBN 978-1-5386-7544-1. Dostupné z: doi:10.1109/M2VIP.2018.8600844
- [97] ZARTE, Maximilian, Ude WUNDER a Agnes PECHMANN. Concept and First Case Study for a Generic Predictive Maintenance Simulation in AnyLogic (TM). In: *lecon 2017 - 43rd Annual Conference of the Ieee Industrial Electronics Society*. New York: Ieee, 2017, s. 3372–3377. ISBN 978-1-5386-1127-2.
- [98] TEUFL, Sabine, Kayode OWA, Dirk STEINHAEUER, Elkin CASTRO, Graham HERRIES, Robert JOHN a Svetan RATCHEV. OPTIMISED - Developing a State of the Art System for Production Planning for Industry 4.0 in the Construction Industry Using Simulation - Based Optimisation. In: M. PERUZZINI, M. PELLICCIARI, C. BIL, J. STJEPANDIC a N. WOGNUM, ed. *Transdisciplinary Engineering Methods for Social Innovation of Industry 4.0* [online]. Amsterdam: Ios Press, 2018, s. 731–740. ISBN 978-1-61499-898-3. Dostupné z: doi:10.3233/978-1-61499-898-3-731
- [99] FANTINI, Paola, Marta PINZONE a Marco TAISCH. Placing the operator at the centre of Industry 4.0 design: Modelling and assessing human activities within cyber-physical systems. *Computers & Industrial Engineering* [online]. 2020, **139**, 105058. ISSN 03608352. Dostupné z: doi:10.1016/j.cie.2018.01.025
- [100] DENKENA, B. a F. WINTER. Simulation-based Planning of Production Capacity through Integrative Roadmapping in the Wind Turbine Industry. *Procedia CIRP* [online]. 2015, **33**, 105–110. ISSN 22128271. Dostupné z: doi:10.1016/j.procir.2015.06.020
- [101] YANG, Wenhe a Soemon TAKAKUWA. Simulation-based dynamic shop floor scheduling for a flexible manufacturing system in the industry 4.0 environment. In: *2017 Winter Simulation Conference (WSC): 2017 Winter Simulation Conference (WSC)* [online]. Las Vegas, NV: IEEE, 2017, s. 3908–3916. ISBN 978-1-5386-3428-8. Dostupné z: doi:10.1109/WSC.2017.8248101
- [102] CONSTANTINESCU, C.L., E. FRANCALANZA a D. MATARAZZO. Towards Knowledge Capturing and Innovative Human-system Interface in an Open-source Factory Modelling and Simulation Environment. *Procedia CIRP* [online]. 2015, **33**, 23–28. ISSN 22128271. Dostupné z: doi:10.1016/j.procir.2015.06.006
- [103] KRENCZYK, D, W M KEMPA, K KALINOWSKI, C GRABOWIK a I PAPROCKA. Integration of manufacturing operations management tools and discrete event simulation. *IOP Conference Series: Materials Science and Engineering* [online]. 2018, **400**, 022037. ISSN 1757-899X. Dostupné z: doi:10.1088/1757-899X/400/2/022037
- [104] CAGGIANO, Alessandra a Roberto TETI. Digital factory technologies for robotic automation and enhanced manufacturing cell design. *Cogent Engineering* [online]. 2018, **5**(1), 1426676. ISSN 2331-1916. Dostupné z: doi:10.1080/23311916.2018.1426676
- [105] AVVENTUROSO, G., R. FORESTI, M. SILVESTRI a E. Morosini FRAZZON. Production paradigms for additive manufacturing systems: A simulation-based analysis. In: *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC): 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)* [online]. Funchal: IEEE, 2017, s. 973–981. ISBN 978-1-5386-0774-9. Dostupné z: doi:10.1109/ICE.2017.8279987

- [106] GOODALL, Paul, Richard SHARPE a Andrew WEST. A data-driven simulation to support remanufacturing operations. *Computers in Industry* [online]. 2019, **105**, 48–60. ISSN 01663615. Dostupné z: doi:10.1016/j.compind.2018.11.001
- [107] RODIČ, Blaž. Industry 4.0 and the New Simulation Modelling Paradigm. *Organizacija* [online]. 2017, **50(3)**, 193–207. ISSN 1581-1832. Dostupné z: doi:10.1515/orga-2017-0017
- [108] GRUBE, David, Ali A. MALIK a Arne BILBERG. SMEs can touch Industry 4.0 in the Smart Learning Factory. *Procedia Manufacturing* [online]. 2019, **31**, 219–224. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2019.03.035
- [109] UHLEMANN, Thomas H.-J., Christoph SCHOCK, Christian LEHMANN, Stefan FREIBERGER a Rolf STEINHILPER. The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems. *Procedia Manufacturing* [online]. 2017, **9**, 113–120. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2017.04.043
- [110] MOURTZIS, Dimitris, Anastasios VASILAKOPOULOS, Evagoras ZERVAS a Nikoletta BOLI. Manufacturing System Design using Simulation in Metal Industry towards Education 4.0. *Procedia Manufacturing* [online]. 2019, **31**, 155–161. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2019.03.024
- [111] LANG, Sebastian, Tobias REGGELIN, Motasem JOBRAN a Wladimir HOFMANN. Towards a Modular, Decentralized and Digital Industry 4.0 Learning Factory. In: *2018 Sixth International Conference on Enterprise Systems (ES): 2018 Sixth International Conference on Enterprise Systems (ES)* [online]. Limassol: IEEE, 2018, s. 123–128. ISBN 978-1-5386-8388-0. Dostupné z: doi:10.1109/ES.2018.00026
- [112] KÁDÁR, Botond, Péter EGRI, Gianfranco PEDONE a Takafumi CHIDA. Smart, simulation-based resource sharing in federated production networks. *CIRP Annals* [online]. 2018, **67(1)**, 503–506. ISSN 00078506. Dostupné z: doi:10.1016/j.cirp.2018.04.046
- [113] KAIHARA, Toshiya, Yoshiteru KATSUMURA, Yuuichi SUGINISHI a Botond KADAR. Simulation model study for manufacturing effectiveness evaluation in crowdsourced manufacturing. *CIRP Annals* [online]. 2017, **66(1)**, 445–448. ISSN 00078506. Dostupné z: doi:10.1016/j.cirp.2017.04.094
- [114] TÜRKER, Ahmet Kürşad, Adem GÖLEÇ, Adnan AKTEPE, Süleyman ERSÖZ, Mümtaz İPEK a Gültekin ÇAĞIL. Siparişlerin gecikme tahmini için veri madenciliğine dayalı gerçek zamanlı bir sistem tasarımı ve uygulaması. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi* [online]. 2019. ISSN 1300-1884. Dostupné z: doi:10.17341/gazimmfd.478648
- [115] KŁOS, Sławomir. Knowledge Acquisition Using Computer Simulation of a Manufacturing System for Preventive Maintenance. In: Robertas DAMAŠEVIČIUS a Giedrė VASILJEVIENĖ, ed. *Information and Software Technologies* [online]. Cham: Springer International Publishing, 2018, Communications in Computer and Information Science, s. 29–40. ISBN 978-3-319-99971-5. Dostupné z: doi:10.1007/978-3-319-99972-2\_3
- [116] SHIN, Hyun-Jun, Kyoung-Woo CHO a Chang-Heon OH. SVM-Based Dynamic Reconfiguration CPS for Manufacturing System in Industry 4.0. *Wireless Communications and Mobile Computing* [online]. 2018, **2018**, 1–13. ISSN 1530-8669, 1530-8677. Dostupné z: doi:10.1155/2018/5795037

- [117] LAURINDO, Quézia Manuela Gonçalves, Túlio Almeida PEIXOTO a João José de Assis RANGEL. Communication mechanism of the discrete event simulation and the mechanical project softwares for manufacturing systems. *Journal of Computational Design and Engineering* [online]. 2019, **6**(1), 70–80. ISSN 2288-5048. Dostupné z: doi:10.1016/j.jcde.2018.02.005
- [118] TURNER, Christopher J., Windo HUTABARAT, John OYEKAN a Ashutosh TIWARI. Discrete Event Simulation and Virtual Reality Use in Industry: New Opportunities and Future Trends. *IEEE Transactions on Human-Machine Systems* [online]. 2016, **46**(6), 882–894. ISSN 2168-2291, 2168-2305. Dostupné z: doi:10.1109/THMS.2016.2596099
- [119] DELBRÜGGER, Tim, Matthias MEISSNER, Andreas WIRTZ, Dirk BIERMANN, Johanna MYRZIK, Jürgen ROSSMANN a Petra WIEDERKEHR. Multi-level simulation concept for multidisciplinary analysis and optimization of production systems. *The International Journal of Advanced Manufacturing Technology* [online]. 2019, **103**(9–12), 3993–4012. ISSN 0268-3768, 1433-3015. Dostupné z: doi:10.1007/s00170-019-03722-1
- [120] KERN, Stefan a Johannes SCHOLZ. Agent-Based Simulation for Indoor Manufacturing Environments—Evaluating the Effects of Spatialization. In: Phaedon KYRIAKIDIS, Diofantos HADJIMITSIS, Dimitrios SKARLATOS a Ali MANSOURIAN, ed. *Geospatial Technologies for Local and Regional Development* [online]. Cham: Springer International Publishing, 2020, Lecture Notes in Geoinformation and Cartography, s. 309–324. ISBN 978-3-030-14744-0. Dostupné z: doi:10.1007/978-3-030-14745-7\_17
- [121] PECHMANN, Agnes, Jeffrey WERMANN, Armando Walter COLOMBO a Maximilian ZARTE. Using a semi-automated job-shop production system model to prepare students for the challenges of Industrial Cyber-Physical Systems. *Procedia Manufacturing* [online]. 2019, **31**, 377–383. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2019.03.059
- [122] HOFMANN, Wladimir, Sebastian LANGER, Sebastian LANG a Tobias REGGELIN. Integrating Virtual Commissioning Based on High Level Emulation into Logistics Education. *Procedia Engineering* [online]. 2017, **178**, 24–32. ISSN 18777058. Dostupné z: doi:10.1016/j.proeng.2017.01.055
- [123] FREEMAN, S., S. L. EDDY, M. MCDONOUGH, M. K. SMITH, N. OKOROAFOR, H. JORDT a M. P. WENDEROTH. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* [online]. 2014, **111**(23), 8410–8415. ISSN 0027-8424, 1091-6490. Dostupné z: doi:10.1073/pnas.1319030111
- [124] PHILLIPS, Kaylin, Bartek GRUSZKA, Johnson CARROLL, Margret BAUER a Oomesh MAHARAJ. Connecting industrial automation software to a discrete manufacturing plant model for research and education. In: *AFRICON 2013: 2013 Africon* [online]. Pointe-Aux-Piments, Mauritius: IEEE, 2013, s. 1–5. ISBN 978-1-4673-5943-6. Dostupné z: doi:10.1109/AFRCON.2013.6757722
- [125] PADILLA, Vladimir Sanchez, Ronald A. PONGUILLO, Arnaldo A. ABAD a Lissette E. SALAS. Cyber-physical system based on image recognition to improve traffic flow: A case study. *International Journal of Electrical and Computer Engineering (IJECE)* [online]. 2020, **10**(5), 5217. ISSN 2722-2578, 2088-8708. Dostupné z: doi:10.11591/ijece.v10i5.pp5217-5226
- [126] ZARTE, Maximilian, Agnes PECHMANN, Jeffrey WERMANN, Frederik GOSEWEHR a Armando Walter COLOMBO. Building an Industry 4.0-compliant lab environment to demonstrate connectivity between shop floor and IT levels of an enterprise. In: *IECON 2016 - 42nd Annual*

- Conference of the IEEE Industrial Electronics Society: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society* [online]. Florence, Italy: IEEE, 2016, s. 6590–6595. ISBN 978-1-5090-3474-1. Dostupné z: doi:10.1109/IECON.2016.7792956
- [127] LYU, Guolin, Alireza FAZLIRAD a Robert W. BRENNAN. Multi-Agent Modeling of Cyber-Physical Systems for IEC 61499 Based Distributed Automation. *Procedia Manufacturing* [online]. 2020, **51**, 1200–1206. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2020.10.168
- [128] ILC, Nejc a Uros LOTRIC. FTsim: A 3D Tool for Teaching Automation Concepts. In: *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO): 2018 13th APCA International Conference on Control and Soft Computing (CONTROLO)* [online]. Ponta Delgada: IEEE, 2018, s. 31–36. ISBN 978-1-5386-5346-3. Dostupné z: doi:10.1109/CONTROLO.2018.8514259
- [129] FALLER, Clemens a Dorothee FELDMÜLLER. Industry 4.0 Learning Factory for regional SMEs. *Procedia CIRP* [online]. 2015, **32**, 88–91. ISSN 22128271. Dostupné z: doi:10.1016/j.procir.2015.02.117
- [130] RASPBERRY PI FOUNDATION. *Use Pico as an I2C slave? - Raspberry Pi Forums* [online]. [vid. 2023-07-12]. Dostupné z: <https://forums.raspberrypi.com/viewtopic.php?t=302978>
- [131] PYTHON BEGINNERS. How to set up Raspberry Pi Pico as I2C slave. *Python Beginners* [online]. 26. říjen 2022 [vid. 2023-07-12]. Dostupné z: <https://python-academia.com/en/raspberry-pi-pico-slave/>
- [132] LASKAKIT. Převodník USB na I2C, FT200XD | LaskaKit. *laskakit.cz* [online]. 7. září 2021 [vid. 2023-07-12]. Dostupné z: <https://www.laskakit.cz/prevodnik-usb-na-i2c--ft200xd/>
- [133] LONG, Moe. *Best Arduino Board for Beginners, IoT Projects, and More* [online]. [vid. 2023-04-12]. Dostupné z: <https://www.electromaker.io/blog/article/best-arduino-board-2019-for-beginners-iot-projects-and-more>
- [134] CHRIS. Best Arduino for Beginners (Costs, Ease of Use compared). *Chip Wired* [online]. [vid. 2023-04-12]. Dostupné z: <https://chipwired.com/best-arduino-for-beginners/>
- [135] JOBIT, Joseph. *Types of Arduino Boards – Quick Comparison on Specification and Features* [online]. [vid. 2023-04-12]. Dostupné z: <https://circuitdigest.com/article/different-types-of-arduino-boards>
- [136] GUPTA, Pooja. What is Arduino UNO? | Amazing 11 Features of Arduino UNO. *EDUCBA* [online]. 13. srpen 2020 [vid. 2023-04-12]. Dostupné z: <https://www.educba.com/what-is-arduino-uno/>
- [137] RPISHOP.CZ. *Arduino Uno Rev3, originál* [online]. [vid. 2023-04-12]. Dostupné z: <https://rpishop.cz/arduino-uno/1356-arduino-uno-r3-7630049200050.html>
- [138] RPISHOP.CZ. *Raspberry Pi Pico* [online]. [vid. 2023-04-12]. Dostupné z: <https://rpishop.cz/raspberry-pi-pico/5117-raspberry-pi-pico-0617588405587.html>
- [139] RPISHOP.CZ. *Raspberry Pi Pico W* [online]. [vid. 2023-04-12]. Dostupné z: <https://rpishop.cz/raspberry-pi-pico/5073-raspberry-pi-pico-w-5056561803173.html>



- [140] FISCHERTECHNIK GMBH. *fischertechnik Training Factory Industry 4.0 (en)* [online]. C++. B.m.: fischertechnik GmbH. 10. červenec 2023 [vid. 2023-07-17]. Dostupné z: [https://github.com/fischertechnik/txt\\_training\\_factory](https://github.com/fischertechnik/txt_training_factory)
- [141] DVOŘÁK, Matěj. *Tvorba modulárního interaktivního digitálního dvojčete výrobního úseku pro virtuální realitu* [online]. B.m., 2022 [vid. 2023-08-09]. Diplomová práce. Západočeská univerzita v Plzni. Dostupné z: <http://dspace5.zcu.cz/handle/11025/49318>
- [142] VELICH, Marek. *Tvorba mobilní aplikace pro zobrazení stavu Fischertechnik modelu* [online]. Plzeň, 2022 [vid. 2023-08-09]. Diplomová práce. Západočeská univerzita v Plzni. Dostupné z: <http://dspace5.zcu.cz/handle/11025/49320>
- [143] VÁŇA, Ondřej. *Ověření návrhu hardwarového modelu využívajícího metody průmyslového inženýrství pomocí diskrétní simulace* [online]. B.m., 2023 [vid. 2023-08-10]. Diplomová práce. Západočeská univerzita v Plzni. Dostupné z: <http://dspace5.zcu.cz/handle/11025/53611>
- [144] KOHOUT, Adam. *Návrh a realizace AGV pro Training Factory s naskladňovacím a vyskladňovacím systémem* [online]. B.m., 2023 [vid. 2023-08-10]. Diplomová práce. Západočeská univerzita v Plzni. Dostupné z: <http://dspace5.zcu.cz/handle/11025/53049>
- [145] HENDL, Jan. *Přehled statistických metod: analýza a metaanalýza dat*. Páté, rozšířené vydání. Praha: Portál, 2015. ISBN 978-80-262-0981-2.
- [146] R: Anderson-Darling test for normality. *R Documentation* [online]. [vid. 2023-05-08]. Dostupné z: <https://search.r-project.org/CRAN/refmans/nortest/html/ad.test.html>
- [147] dunn.test: Dunn's Test - RDocumentation. *RDocumentation* [online]. [vid. 2023-05-08]. Dostupné z: <https://www.rdocumentation.org/packages/dunn.test/versions/1.3.5/topics/dunn.test>
- [148] PARKER, Michael. *leansigmacorporation.com. What is a Box Plot?* [online]. [vid. 2023-05-08]. Dostupné z: <https://www.leansigmacorporation.com/box-plot-with-minitab/>
- [149] HENDL, Jan. *Kvalitativní výzkum: základní teorie, metody a aplikace*. 2016. ISBN 978-80-262-0982-9.
- [150] MOLNÁR, Zdeněk, Stanislava MILDEOVÁ, Hana ŘEZANKOVÁ, Radim BRXÍ a Jaroslav KALINA. *Pokročilé metody vědecké práce*. Zeleneč: Profess Consulting, 2012. ISBN 978-80-7259-064-3.

## Publikační činnost studenta doktorské formy studia

### Publikované práce související s tématem disertační práce

ULRYCH, Z. MALAGA, M. Vzdělávání budoucích strojních a průmyslových inženýrů v oblastech důležitých pro Industry 4.0. In MEDZINÁRODNÁ KONFERENCIA O VYUČOVANÍ INFORMATIKY DidInfo 2023 28. ROČNÍK KONFERENCIE. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied v Banskej Bystrici, Slovensko, 2023. s. 160-165. ISBN: 978-80-557-2038-8 , ISSN: 2454-051X (mentální podíl 50 %)

ULRYCH, Z. MALAGA, M. *Základy robotiky – programování hardwarových modelů*. 1. vyd. Plzeň : Západočeská univerzita v Plzni, 2023, 112 s. ISBN: 978-80-261-1144-3 (mentální podíl 50 %)

BROUM, T. HOŘEJŠÍ, P. MALAGA, M. GRZONA, P. Competencies of Industrial Engineers for Implementing Augmented Reality Metadata Systems. *Sustainability*, 2023, roč. 15, č. 1, s. nestránkováno. ISSN: 2071-1050 (mentální podíl 30 %)

MÜLLER, J. BROUM, T. MALAGA, M. Use of machine vision systems in industrial robotics. In *Engineering Sciences and Production Management 2022*. Košice: Pro Enviro, 2022. s. 87-94. ISBN: 978-80-89295-96-8 (mentální podíl 33,3 %)

DVOŘÁK, M. KUBR, J. BEDNÁŘ, M. KRÁKORA, D. MALAGA, M. HOŘEJŠÍ, P. ŠIMON, M. User Acceptance of A Modular Interactive Digital Twin of The Production Department for Virtual Reality. In *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2022. (mentální podíl 14,3 %)

MALAGA, M. BROUM, T. ULRYCH, Z. Cost-related industrial project design using simulation, layout design and an ERP system. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2022. s. 1-4. ISBN: 978-1-66546-835-0 (mentální podíl 33,3 %)

MALAGA, M. ŠVRČULA, P. ULRYCH, Z. Connection and Communication Between Computer And Hardware Simulation Model for Industrial Engineers. In *Proceedings of the 39th International Business Information Management Association (IBIMA)*. NORRISTOWN: INT BUSINESS INFORMATION MANAGEMENT ASSOC-IBIMA, 2022. s. 89-100. ISBN: 978-0-9998551-9-5 , ISSN: 2767-9640 (mentální podíl 33,3 %)

MALAGA, M. BROUM, T. ŠIMON, M. FRONĚK, M. Industrial robotics as an important part of modern production automation. *Acta Mechatronica – International Scientific Journal about Mechatronics*, 2022, roč. 7, č. 4, s. 31-36. ISSN: 2453-7306 (mentální podíl 40 %)

MALAGA, M. BROUM, T. ULRYCH, Z. Extended process of Industry 4.0 knowledge transfer using automation elements and hardware simulation. In *Proceedings of the 39th International Business Information Management Association (IBIMA)*. NORRISTOWN: INT BUSINESS INFORMATION MANAGEMENT ASSOC-IBIMA, 2022. s. 29-35. ISBN: 978-0-9998551-9-5 , ISSN: 2767-9640 (mentální podíl 33,3 %)

HUCL, M. MALAGA, M. ULRYCH, Z. Řízení a sběr dat z hardwarového simulačního modelu pomocí RaspberryPi a MQTT protokolu. In *Průmyslové inženýrství 2022: Mezinárodní studentská vědecká konference*. Plzeň: Západočeská univerzita v Plzni, 2022. s. 43-60. ISBN: 978-80-261-1114-6 (mentální podíl 33,3 %)

BROUM, T. MALAGA, M. ŠIMON, M. FRONĚK, M. SENSOR SYSTEMS FOR AUTOMATION IN INDUSTRY 4.0. *Transfer inovací*, 2022, roč. Neuveden, č. 46, s. 124-127. ISSN: 1337-7094 (mentální podíl 40 %)

MALAGA, M. ULRYCH, Z. Implementace nové generace řídicí jednotky Fischertechnik TXT 4.0 do hardwarového simulačního modelu představujícího koncept Industry 4.0. In *Průmyslové inženýrství*

2022: *Mezinárodní studentská vědecká konference*. Plzeň: Západočeská univerzita v Plzni, 2022. s. 130-141. ISBN: 978-80-261-1114-6 (mentální podíl 50 %)

DVOŘÁK, M. HOŘEJŠÍ, P. MALAGA, M. KUBR, J. Tvorba modulárního interaktivního digitálního dvojčete výrobního úseku pro virtuální realitu. In *Průmyslové inženýrství 2022: Mezinárodní studentská vědecká konference*. Plzeň: Západočeská univerzita v Plzni, 2022. s. 13-25. ISBN: 978-80-261-1114-6 (mentální podíl 14,3 %)

BROUM, T. MALAGA, M. ŠIMON, M. FRONĚK, M. Přehled průmyslové automatizace v Industry 4.0. In *Proceedings of the XXV. International Scientific Conference TIABP*. Košice: Technická univerzita v Košiciach, 2022. s. 104-113. ISBN: 978-80-553-4177-4 (mentální podíl 25 %)

MALAGA, M. ULRYCH, Z. Training factories s principy Industry 4.0 v průmyslovém inženýrství. In *Průmyslové inženýrství 2021: Mezinárodní studentská vědecká konference*. Plzeň: Západočeská univerzita v Plzni, 2021. s. 133-149. ISBN: 978-80-261-0792-7 (mentální podíl 50 %)

RAŠKA, P. ULRYCH, Z. MALAGA, M. Data Reduction of Digital Twin Simulation Experiments Using Different Optimisation Methods. *Applied Sciences*, 2021, roč. 11, č. 16, s. nestránkováno. ISSN: 2076-3417 (mentální podíl 33,3 %)

ŠVRČULA, P. MALAGA, M. ULRYCH, Z. Napojení na STEM model pro potřeby průmyslových inženýrů. In *Průmyslové inženýrství 2021: Mezinárodní studentská vědecká konference*. Plzeň: Západočeská univerzita v Plzni, 2021. s. 203-219. ISBN: 978-80-261-0792-7 (mentální podíl 33,3 %)

MALAGA, M. ULRYCH, Z. Industry 4.0 And Training Factories For Data Generation And Verification. In *Proceedings of the 38th International Business Information Management Association Conference (IBIMA)*. NORRISTOWN: International Business Information Management Association (IBIMA), 2021. s. 1542-1550. ISBN: 978-0-9998551-7-1 , ISSN: 2767-9640 (mentální podíl 50 %)

MALAGA, M. ULRYCH, Z. Physical modelling of the Industry 4.0 concept. In *Proceedings of the 35th International Business Information Management Association Conference*. Seville, Spain: International Business Information Management Association (IBIMA), 2020. s. 17540-17549. ISBN: 978-0-9998551-4-0 (mentální podíl 50 %)

MALAGA, M. ULRYCH, Z. Diskrétní simulace v konceptu Industry 4.0. In *Průmyslové inženýrství 2020: Mezinárodní studentská vědecká konference*. Plzeň: Západočeská univerzita v Plzni, 2020. s. 138-153. ISBN: 978-80-261-0969-3 (mentální podíl 50 %)

MALAGA, M. BROUM, T. ULRYCH, Z. Industry 4.0 basic knowledge transfer. In 2020. In *Proceedings of the 36th International Business Information Management Association Conference*. Granada, Spain: International Business Information Management Association (IBIMA), 2020. ISBN: 978-0-9998551-5-7 (mentální podíl 33,3 %)

MALAGA, M. ULRYCH, Z. Trends in discrete simulation in the era of Industry 4.0. In *Proceedings of the 36th International Business Information Management Association Conference*. Granada, Spain: International Business Information Management Association (IBIMA), 2020. ISBN: 978-0-9998551-5-7 (mentální podíl 50 %)

MALAGA, M. ULRYCH, Z. *Základy řízení robotů pro strojní inženýrství*. 1. vyd. Plzeň : Západočeská univerzita v Plzni, 2020, 145 s. ISBN: 978-80-261-0486-5 (mentální podíl 50 %)

MALAGA, M. ULRYCH, Z. Koncept STEM se zaměřením na problematiku Industry 4.0. In *Průmyslové inženýrství 2019: Mezinárodní studentská vědecká konference*. Plzeň: Západočeská univerzita v Plzni, 2019. s. 94-100. ISBN: 978-80-261-0894-8 (mentální podíl 50 %)

MALAGA, M. ULRYCH, Z. STEM concept in the era of industry 4.0. In *Vision 2025: Education Excellence and Management of Innovations through Sustainable Economic Competitive Advantage*.

NORRISTOWN: INT BUSINESS INFORMATION MANAGEMENT ASSOC-IBIMA, 2019. s. 9669-9678. ISBN: 978-0-9998551-3-3 (mentální podíl 50 %)

### **Ostatní publikované práce**

KOPEČEK, P. MALAGA, M. *Plánování a řízení výroby s podporou ERP Helios iNuvio*. 1. vyd. Plzeň : Západočeská univerzita v Plzni, 2022, 338 s. ISBN: 978-80-261-1096-5 (mentální podíl 50 %)

MALAGA, M. BROUM, T. KOPEČEK, P. Kvalitní ERP potřebují i technické obory budoucnosti. *Techmagazín*, 2021, roč. 12, č. 09, s. 20. ISSN: 1804-5413 (mentální podíl 33,3 %)

MALAGA, M. BROUM, T. KOPEČEK, P. Technické obory budoucnosti potřebují kvalitní ERP. *Reflex*, 2021, roč. 32, č. 38, s. 6. ISSN: 0862-6634 (mentální podíl 33,3 %)

KOPEČEK, P. MALAGA, M. ERP ve vzdělávání průmyslového inženýra. *MM Průmyslové spektrum*, 2020, roč. Neuveden, č. 6,7,8, s. 80-81. ISSN: 1212-2572 (mentální podíl 50 %)

KOPEČEK, P. MALAGA, M. *Plánování a řízení výroby s podporou ERP Helios*. 1. vyd. Plzeň : SmartMotion s.r.o., 2013, ISBN: 978-80-87539-41-5 (mentální podíl 50 %)

KOPEČEK, P. MALAGA, M. *Interní příručka Helios Orange*. 1. vyd. SmartMotion s.r.o., 2013, ISBN: 978-80-87539-42-2 (mentální podíl 50 %)

KOPEČEK, P. MALAGA, M. *Plánování a řízení výroby a DP*. 1. vyd. Plzeň : SmartMotion, 2012, ISBN: 978-80-87539-14-9 (mentální podíl 50 %)

### **Recenzní posudky**

MALAGA, M. Socio-technical aspects of the enterprise transformation in the conditions of Industry 4.0. Seville, Spain : International Business Information Management Association (IBIMA), 2021.

MALAGA, M. Threats in international logistics – study during covid-19 pandemic time. Cordoba, Spain : International Business Information Management Association (IBIMA), 2021.

MALAGA, M. *The elements of technical support for integrated safety management in the Industry 4.0*. Granada, Spain : International Business Information Management Association (IBIMA), 2020.

### **Editorství**

BROUM, T. MALAGA, M. BUREŠ, M. *Sborník příspěvků mezinárodní studentské vědecké konference – Průmyslové inženýrství 2022*. 1. vyd. Plzeň : Západočeská univerzita v Plzni, 2022, 179 s. ISBN: 978-80-261-1114-6

### **Vystoupení na vědeckých setkáních**

2022 16<sup>th</sup> Scientific Conference MET2022: “Management, Economy and Technology – Contemporary Challenges and Perspectives of Economies and Societies Development”

2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)

2022 Průmyslové inženýrství 2022: Mezinárodní studentská vědecká konference

2022 Erasmus+ Training Week 2022 – Smart Production Systems Modelling Laboratory

2022 39<sup>th</sup> International Business Information Management Association (IBIMA) – 2 vystoupení

2019 Průmyslové inženýrství 2019: Mezinárodní studentská vědecká konference

### Projekty související s tématem disertační práce

1. 1. 2024 – 31. 12. 2026	101135948, PROSPECTS5.0_PROGRESS TOWARDS INDUSTRY 5-0: A SMART STUDY ON ANALYSIS AND IDENTIFICATION OF PRACTICES, DRIVERS, SUCCESS FACTORS AND OBSTACLES OF TRANSITIONS TOWARDS INDUSTRY 5.0., European Commission, Horizon EU
1. 4. 2023 – 31. 12. 2023	PRVA-23-042, Inovace a modernizace předmětů KPV/PI – Průmyslové inženýrství a KPV/APS – Aplikace průmyslu 4.0 ve strojírenství
1. 4. 2021 – 31. 12. 2023	SGS-2021-028, Vývojové a tréninkové prostředky pro interakci člověka a kyber-fyzického výrobního systému
1. 6. 2021 – 31. 12. 2022	IDEG-ING_2021-005, Data collecting from hardware simulation for use by digital twins
1. 4. 2022 – 31. 12. 2022	PRVA-22-027, Podpora rozvoje vzdělávacích aktivit zaměřených na vybrané prvky konceptu Industry 4.0 v rámci předmětu Počítačová podpora ve strojírenství
1. 4. 2021 – 31. 12. 2021	VS-21-021, Tvorba výukových materiálů pro simulaci s využitím digitálního dvojčete a principy Industry 4.0
1. 4. 2018 – 31. 12. 2020	SGS-2018-031, Optimalizace parametrů udržitelného výrobního systému

## **PŘÍLOHA A**

### **Přehled elektronických příloh disertační práce**

Soubor/ZIP Složka	Popis	Software pro práci se souborem/soubory
3D_tisk.zip	<p>Soubory 3D modelů a pro 3D tisk dílů tisknutých pro realizovanou Training Factory:</p> <ul style="list-style-type: none"> <li>• Držáky sensorů</li> <li>• Držák BBC Micro:bit</li> <li>• Kryt světelného sensoru</li> <li>• Logo pro sklad</li> <li>• Vložka pro kulaté puky do skladu</li> </ul>	<p>*.ipt – Autodesk Inventor</p> <p>*.3mf – PrusaSlicer</p> <p>*.stl – Autodesk Inventor, PrusaSlicer</p> <p>*.gcode - soubor pro tisk 3D modelu pro Průša Mini+</p>
Řidici_programy_tovarna.zip	<p><b>Malaga_KPV_Factory_final_disp.ft</b> Řídící program pro realizovanou Training Factory</p> <p><b>Data_Translator_Sketch.ino</b> Řídící program pro překladač RaspberryPi Pico</p> <p><b>microbit-SRP_client.hex</b> Řídící program pro BBC Micro:bit sloužící jako klient systému řízení pracovníků</p> <p><b>microbit-SRP_driver.hex</b> Řídící program pro BBC Micro:bit sloužící jako driver systému řízení pracovníků</p> <p><b>microbit-Sensor_station.hex</b> Řídící program pro senzorovou stanici</p> <p><b>microbit-TF_Broker.hex</b> Řídící program pro bezdrátový přijímač senzorové stanice</p>	<p>*.ft – Robo PPro Coding</p> <p>*.hex – MakeCode</p> <p>*.ino – Arduino IDE</p>
Řidici_programy_tridicka.zip	<p><b>Sorter.spp</b> Ukázkový model třídičky (digitálního dvojčete) v Tecnomatix Plant Simulation pro napojení na ControllModSim</p> <p><b>Sorter.mset</b> Ukázkové nastavení pro HW model třídičky pro aplikaci ControllModSim</p> <p><b>SyncControl.psobj</b> Předpřipravená knihovna objektu SyncControl, která zajišťuje přenášení dat z řídicího programu do simulačního modelu a spouštění jednotlivých metod ve správný čas.</p> <p><b>SorterController</b> Aplikace s grafickým rozhraním pro řízení HW modelu</p> <p><b>Color sorting robot_i2c.rpp</b> Řídící program pro model třídičky</p>	<p>*.spp – Tecnomatix Plant Simulation v 16.1</p> <p>*.psobj – Tecnomatix Plant Simulation v 16.1</p> <p>*.mset – soubor s nastavením pro import do aplikace ControllModSim</p> <p>SorterController – Visual Studio 2022</p> <p>*.rpp – ROBOPro</p> <p>*.ino – Arduino IDE</p>

	<b>UNO_Main.ino</b> Řídící program pro Arduino UNO umožňující obousměrnou komunikaci	
<b>Hardwarovy_model_tabulka_akci.xlsx</b>	Kompletní přehled číselných hodnot, jejich textového názvu a popisu, které jsou sbírány z realizované Training Factory	Microsoft Excel
<b>ControllModSim.zip</b>	Zdrojové kódy aplikace ControllModSim	Visual Studio 2022
<b>Dotaznik_Znalosti_IT_pro_realizaci_TF_Pl.pdf</b>	Dotazník exportovaný z Google Forms využitý pro sběr dat elektronickou formou dotazníkového šetření	Adobe Acrobat
<b>Dotaznik_odpovedi.xlsx</b>	Data pro ověření hypotézy 3 získaná formou dotazníkového šetření	Microsoft Excel
<b>TEST_CC++.docx</b>	Varianta testu zaměřená na programovací jazyk C/C++	Microsoft Word
<b>TEST_Python.docx</b>	Varianta testu zaměřená na programovací jazyk Python	Microsoft Word
<b>TEST_RoboProCoding.docx</b>	Varianta testu zaměřená na vizuální programovací jazyk Robo Pro Coding	Microsoft Excel
<b>Testy_data.xlsx</b>	Data pro ověření hypotéz 1. a 2. získaná testováním probandů	Microsoft Excel