

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**Využití výukového kitu TinyLab v přípravě
budoucích učitelů**
DIPLOMOVÁ PRÁCE

Jiří Noska
Učitelství pro 2. stupeň ZŠ

Vedoucí práce: PhDr. Tomáš Jakeš, Ph.D.

Plzeň, 2022

Prohlašuji, že jsem diplomovou práci vypracoval samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 27. června 2022

.....
vlastnoruční podpis

MÉ PODĚKOVÁNÍ PATŘÍ PHDr. TOMÁŠI JAKEŠOVI, PH.D., ZA
ODBORNÉ VEDENÍ, TRPĚLIVOST A OCHOTU, KTEROU MI V PRŮBĚHU
ZPRACOVÁNÍ DIPLOMOVÉ PRÁCE VĚNOVAL

OBSAH

ÚVOD	6
1 VÝVOJ KITU TINYLAB.....	8
1.1 VARIANTY VÝUKOVÉHO KITU TINYLAB	8
2 TINYLAB.....	11
2.1 POTENCIOMETR	12
2.2 PIEZOELEKTRICKÝ MĚNIČ.....	12
2.3 FOTOREZISTOR.....	13
2.4 TEPLTNÍ ČIDLO	13
2.5 RELÉ	14
2.6 TLAČÍTKA	14
2.7 LED DIODY	15
2.8 ROTAČNÍ ENKODÉR	16
2.9 7SEGMENTOVÝ ZOBRAZOVACÍ PRVEK	17
2.10 LCD DISPLEJ	18
3 PROGRAMOVÁNÍ VÝUKOVÉHO KITU TINYLAB.....	19
3.1 ARDUINO WEB EDITOR.....	19
3.2 ARDUINO IDE.....	19
3.3 ALTERNATIVY PRO PROGRAMOVÁNÍ.....	20
3.4 SCRATCH.....	21
3.5 SIMULAČNÍ PROGRAMY	21
4 VÝUKOVÝ MATERIÁL.....	23
4.1 SADA ÚLOH DEMONSTRUJÍCÍ VYUŽITÍ VÝUKOVÉHO KITU TINYLAB	25
4.2 SEZNÁMENÍ SE S VÝUKOVÝM KITEM TINYLAB.....	26
4.3 LED DIODY	29
4.4 POTENCIOMETR	32
4.5 TLAČÍTKA	36
4.6 RELÉ, PIEZOELEKTRICKÝ MĚNIČ	38
4.7 FOTOREZISTOR, TEPLTNÍ ČIDLO	42
4.8 ROTAČNÍ ENKODÉR	45
4.9 7 SEGMENTOVÝ DISPLEJ.....	48
4.10 LCD DISPLEJ, FUNKCE, CYKLY	51
4.11 KOMPLEXNÍ ÚLOHY	53
ZÁVĚR	59
RESUMÉ.....	60
SEZNAM LITERATURY	62
SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ	64
PŘÍLOHY.....	I

Úvod

Hlavním cílem mé diplomové práce je poskytnout přehled o možnostech využití výukového kitu TinyLab (dále jen „TinyLab“) v přípravě budoucích učitelů. V rámci této práce poskytnu sadu úloh, kterou lze využít při výuce. Tato sada úloh však nebude sloužit pouze pro účely výuky budoucích pedagogů na vysoké škole, ačkoliv je pro ně primárně připravována, ale bude moci sloužit i pro výuku informatiky na školách základních, jako pomocný materiál pro pedagoga. Díky narůstající obtížnosti a vhodně zvoleným příkladům tak mohou mít tyto úlohy široké využití. Vytvořením samotných úloh však proces tvorby této práce nekončí. Bude vytvořen i pomocný materiál, který bude opět primárně určen pro výuku budoucích učitelů, ale stejně jako v případě předešlém, lze jej použít i pro výuku na základních školách.

Důležité je předem uvést, že veškerý materiál a úlohy nejsou určeny k tomu, aby naučily budoucí pedagogy bezchybně programovat v TinyLabu, ale budou určeny jako základní penzum znalostí, ze kterého lze dále vycházet a v případě nutnosti a potřeby se i zdokonalovat, což by mělo být jedním ze základních vlastností budoucích pedagogů. Příklady a aktivity budou reprezentovat základní možnosti, na základě kterých lze dále zvyšovat své programátorské dovednosti.

Materiál, který bude v rámci této práce vytvořen, bude volně dostupný ve vhodném úložišti tak, aby mohl být co nejvíce využíván, a to nejen v přípravě budoucích učitelů, ale aby byla možnost jej dále používat i na budoucích pracovištích těchto pedagogů. Vytvořenými materiály si tak mohou výrazně usnadnit počáteční vstup do pedagogického života.

Proč se tato práce zabývá právě možnostmi TinyLabu? Díky změně školského vzdělávacího plánu se proměňuje i náplň hodin. Větší důraz je kladen na prohlubování informatického myšlení. Právě díky programování tohoto kitu lze prohloubit i výše uvedené dovednosti. Výhoda TinyLabu je pak jednoznačně v již zapojených komponentech, kdy ve vyučovací hodině odpadá často zdlouhavé zapojování jednotlivých součástí k deskám, jako je například Arduino. Minimalizuje se zde i riziko chybného zapojení a možnosti nechtěné destrukce dané desky. Programováním TinyLabu se tak prohlubují i klíčové kompetence žáků, které jsou potřebné pro běžný život či další profesní kariéru.

V úvodu drobně představím historii TinyLabu, od vzniku programovacího jazyka wiring, přes desky zvané Arduino, ke crowdfundingové kampani, která měla za cíl představit projekt, v

jehož závěru byl samotný TinyLab. Ačkoliv Arduino desek je velká řada, TinyLabu se dostaly pouze tři varianty, které ale plně dostačují pro pochopení systému programování.

V další kapitole představím jednotlivé komponenty samotné základové desky TinyLabu s drobným popisem jejich funkce. Vysvětlení popisu funkce těchto komponentů se váže na jejich využití, a proto se nebudu zabývat těmito komponenty do hloubky.

Závěr teoretické části bude vyhrazen softwarovému vybavení. Představíme programy, ve kterých lze TinyLab, popřípadě Arduino samotné, naprogramovat. Cílem této části nebude jednotný závěr, ve kterém softwaru programovat, ale spíše představení možností a výběr již bude na samotném uživateli. V této práci bude použit jeden software, a to z důvodu jednoduchosti jeho použití. Také zde popíšu vznik výukového materiálu, který bude kopírovat tuto diplomovou práci.

Praktická část by se již plně věnovala samotnému představení aktivit. Tyto aktivity budou rozděleny dle využití jednotlivých komponent TinyLabu a budu se snažit o vzrůstající obtížnost úloh. V samotném závěru této sady se tak budou nacházet úlohy, které by měly propojit získané znalosti, a hlavně mít návaznost na reálný život. Programovat tedy budeme zařízení, které lze využít v běžném životě a nebude se jednat pouze o jakési teoretické představy. V rámci metodických doporučení pro pedagogy zazní i možné variace pro zlehčení, popřípadě pro ztížení dané úlohy.

V rámci této práce bych chtěl ukázat budoucím, popřípadě i současným pedagogickým pracovníkům, kteří vyučují informatiku na základních školách, že se nemusejí obávat vstoupit do světa praktického programování.

1 VÝVOJ KITU TINYLAB

Ačkoliv práce pojednává o možnostech TinyLabu, tak nelze nezmínit příběh zvaný Arduino. Ačkoliv projekt Arduina nebyl technologicky velmi inovátorský, tak proti svým předchůdcům měl velmi výrazný pokrok v propagaci. Díky komunitě, která se vytvořila okolo této platformy, se z něj stal celosvětový fenomén.

Nejsou přesně známá čísla prodeje u platformy Arduino, ale dle Maxe Maxfielda se platformy RapsberryPi prodalo okolo 50 miliónů kusů. Pokud by se jednalo o prodej oficiálních systémů Arduino, pak by toto číslo zůstalo nepokořeno. Arduino má ale mnoho klonů, které pracují na stejném principu. Zde již Maxfield odhaduje, že Arduino má v prodejnosti silně navrch. (MAXFIELD, 2020)

Arduino vzniklo díky diplomové práci Hernanda Barragána v roce 2003. Tuto práci vedli Massimo Banzi a Casey Reas. Výstupem bylo zařízení označované jako Wiring. V průběhu několika málo let Wiring končí, ale na něj plynule navazuje koncept Arduino, které převzali Massimo Banzi a David Mellis. (SELECKÝ, 2016 stránky 16-17)

První zmínky o TinyLabu pochází z jeho vývoje. Nejsou jedinou firmou, která k propagaci a podpoře svého výrobku využila tzv. crowdfunding¹. V červnu roku 2015 dochází k prvnímu hardwarovému návrhu, v srpnu toho roku byl vyroben první prototyp. Kampaň, která probíhala v prosinci roku 2015 a měla za cíl vybrat 25 tisíc USD, nakonec překročila stanovený cíl o 58 tisíc USD. Na této crowdfundingové kampani se podílelo celkem 874 přispívajících. (Sixfab Inc., 2015)

1.1 VARIANTY VÝUKOVÉHO KITU TINYLAB

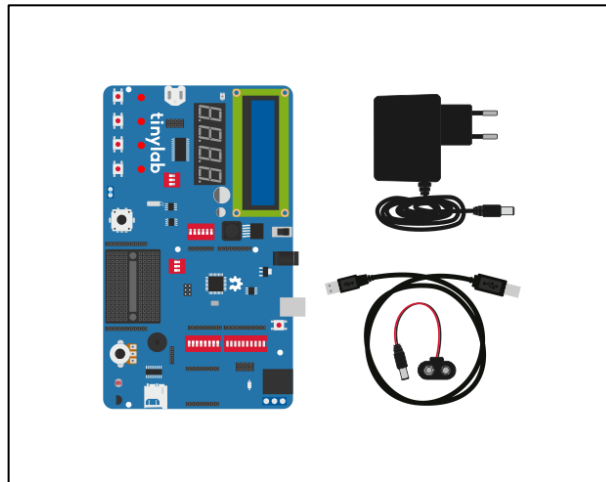
Společnost TinyLab nabízí k prodeji celkem tři varianty tohoto výukového kitu. Tyto varianty se liší, jak si dále ukážeme, především v rozšiřujících modulech. První varianta, která je variantou základní, nese pojmenování Maker kit. Druhá varianta pak nese název IoT kit a nejvyšší řada je pojmenována Exclusive kit.

První, nejnižší a samozřejmě nejlevnější, varianta TinyLab Maker kit obsahuje:

- samotnou stavebnici TinyLab,

¹ Crowdfundingová kampaň slouží k alternativnímu získávání finančních prostředků

- adaptér pro napájení,
- bateriový napaječ.

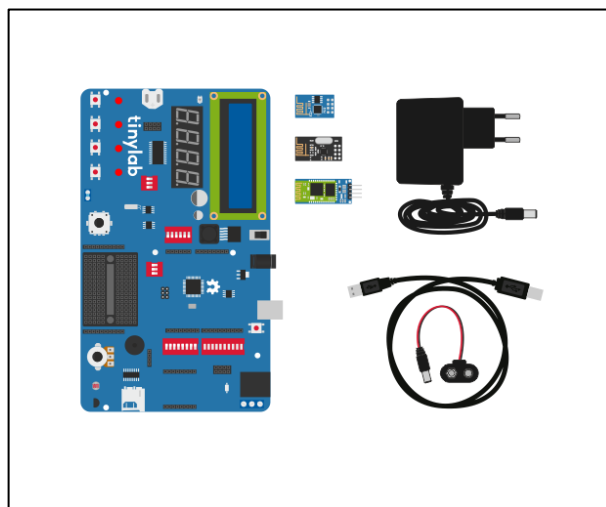


Obrázek 1 - TinyLab Maker Kit, zdroj: <https://www.tinylab.cc/>

Vyšší varianta zvaná IoT kit, kromě výše zmíněného, obsahuje moduly:

- NRF24L01P,
- ESP8266,
- HC06.

Modul NRF24L01P je Wi-Fi modul běžící na frekvenci 2,4GHz a s přenosovou rychlostí 2 Mbps. Modul ESP8266 je Wi-Fi mikročip s vestavěným TCP/IP síťovým protokolem. Poslední modul HC06 využívá technologie Bluetooth. Všechny tyto moduly slouží k tomu, aby mohl TinyLab komunikovat s ostatními zařízeními.

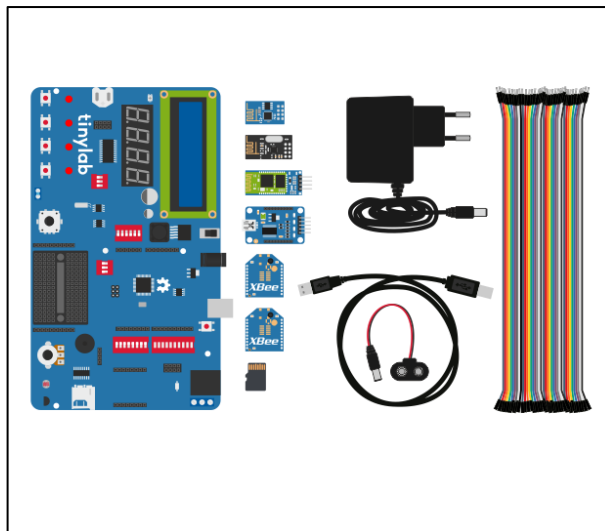


Obrázek 2 - TinyLab IoT Kit, zdroj: <https://www.tinylab.cc/>

Nejdražší varianta Exclusive kit obsahuje ještě navíc:

- 2x XBee
- XBee explorer
- mikroSD kartu
- sadu jumper kabelů

XBee produkty jsou bezdrátové moduly určené pro použití s Arduino deskami. (TinyLab, 2022)



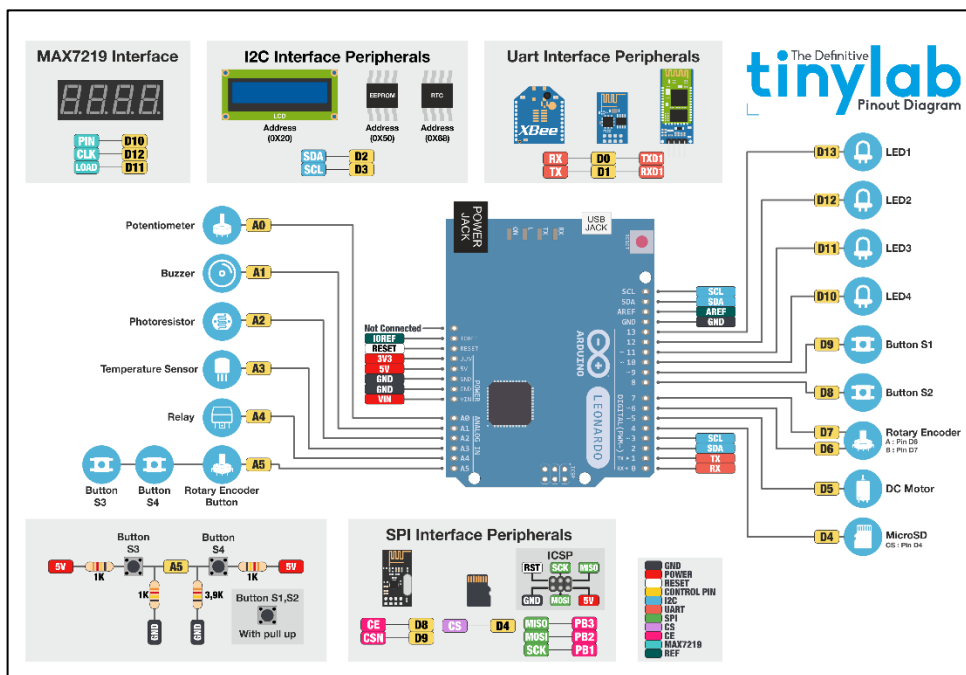
Obrázek 3 - TinyLab Exclusive Kit, zdroj: <https://www.tinylab.cc/>

2 TINYLAB

Samotný TinyLab obsahuje množství integrovaných komponent, které jsou pevně spojeny se základní deskou kitu. Výhodou oproti použití ostatních Arduino desek je odpadnutí nutnosti zapojování těchto komponent a tím i možnost chyby v rámci tohoto zapojení nebo nechtěné destrukce desky. Jistou nevýhodou však může být, že studenti plně nepochopí princip fungování jednotlivých komponent.

Základem Tynylabu je základní deska. Tou je Arduino deska s označením Leonardo, jejímž základem je mikro kontrolér ATmega32u4. Tento 8bitový kontrolér obsahuje 2560 KB paměť RAM, 1024 bytovou paměť EEPROM a rychlost 16 MIPS. (Microchip Technology Inc., 2021)

Deska Leonardo obsahuje 20 vstupně-výstupních pinů. Sedm z těchto pinů lze ovládat pomocí pulsně šířkové modulace (piny 3, 5, 6, 9, 10, 11, 13). Napájecí část obsahuje piny s výstupním napětím 5 V (označení 5V) a 3,3 V (označení 3V3), zemnicí pin (označení GND) a pin s napětím, se kterým pracují vstupně výstupní piny (označení IOREF). Další z pinů, které výše uvedená deska má je referenční napětí (označení AREF) a pin ovládající reset mikro kontrolérů (označení Reset). (Arduino, 2021)



Obrázek 4 - Zapojení jednotlivých komponentů, zdroj: <https://www.tinylab.cc/wp-content/uploads/2016/05/tinylab-pinout-diagram2.png>

Následné vyjmenování a popsání komponent se bude řídit dle schématu zapojení uvedeného na obrázku výše (Obrázek 4). Dále by mohlo být rozdělení například na vstupní a výstupní komponenty, ale právě propojení s tímto obrázkem, který lze najít přímo na stránkách TinyLabu, je názornější, a v případě výuky i praktičtější.

2.1 POTENCIOMETR

Potenciometr je proměnným typem rezistoru. (DOLEČEK, 2007 str. 258) Potenciometr má vývody označované jako E (vstup), S (jezdec) a A (výstup). (TKOTZ, 2001 str. 29) Díky integraci potenciometru na desce odpadá zapojování potenciometru, konkrétně vývodů odporové vložky. (BURGER, a další, 1989 str. 419)

Z jezdcy lze regulované napětí odečítat rovnou z analogového pinu A0. V případě TinyLabu se hodnoty na pinu A0 pohybují v krajních polohách mezi 0 až 1023. Tato hodnota je dána funkcí `analogRead`, která má v případě desky Arduino Leonardo rozlišení 10 bitů. Při přepočtu tohoto rozlišení na dekadickou hodnotu je rovno hodnotě 1024. Jelikož se hodnota počítá od nuly, pak se dostaneme k rozmezí 0 až 1023. (Arduino, 2021)

2.2 PIEZOELEKTRICKÝ MĚNIČ

Tato součástka využívá nepřímý piezoelektrický jev. Při přivedení napětí na měnič dochází na vnitřním krystalu k deformacím, které lze následně slyšet díky převedení této deformace na sluchový vjem. (VODA, 2017)

V případě bzučáku² lze měnit frekvenci. Frekvence tónu A je 440 Hz, to znamená, že bzučák musí být zapnut a vypnut 440krát za sekundu. Pomocí jednoduchého výpočtu se zjistí, že perioda zapnutí a vypnutí musí být 2272 mikrosekund. V případě, že se bzučák tedy na polovinu hodnoty periody, tj. 1134 mikrosekundy zapne, a poté na stejný časový okamžik vypne, pak bude piezoelektrický měnič vydávat tón A neboli zvuk o frekvenci 440 Hz.

V TinyLabu je připojen bzučák na pin A1.

² bzučák je odlišný název pro piezoelektrický měnič zapříčiněný charakteristickým zvukem

2.3 FOTOREZISTOR

Fotorezistor je součástka, která mění automaticky svůj odpor v závislosti na dopadajícím osvětlení. Když roste dopadající osvětlení, dochází ve fotorezistoru k narůstající vodivosti a klesajícímu odporu. (TKOTZ, 2001 str. 181)

Fotorezistor využívá vnitřního fotoelektrického jevu. (BURGER, a další, 1989 str. 349) Vlivem optického záření dochází k ionizaci atomů polovodičů a vzniká zde fotoelektrická vodivost. (BURGER, a další, 1989 str. 332)

Tato součástka je vyvedena na analogový pin A2. Stejně jako v případě potenciometru, i zde je výstupní proměnná v rozmezí 0 až 1023. V případě, že se na pinu objeví hodnota 0, pak je hodnota intenzity osvětlení maximální, v případě, že se objeví na daném pinu hodnota 1023, pak se jedná o minimální intenzitu osvětlení.

2.4 TEPLOTNÍ ČIDLO

Součástí kitu TinyLab je i teplotní čidlo s výstupem na analogovém pinu A3. Na tomto výstupu můžeme odečítat hodnoty, které jsou závislé na okolní teplotě. Pro přesný přepočítání na stupně Celsia musíme znát parametry této součástky. Do TinyLabu je integrováno teplotní čidlo LM35, které má pracovní rozmezí od -55 °C do +150 °C. V případě změny teploty o 1 °C dojde ke změně napětí na tomto pinu o 10mV. (Texas Instruments, 2017)

Pro správné určení teploty je nutný provést vždy přepočítání. Proměnná hodnota, která se nachází na analogovém pinu A3, se musí převést na milivolyty.

$$U (mV) = \frac{A3}{1024} * 5000$$

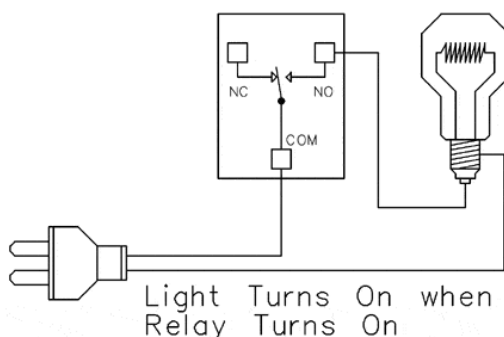
To lze provést prostým vydělením hodnoty pinu A3 hodnotou 1024 (rozmezí, které lze číst z tohoto pinu) a následně vynásobit vstupním napětím (5 V, pro lepší výpočet ale použijeme 5000 mV). Následně tento výsledek přepočítat na stupně Celsia. Předchozí výsledek vydělit hodnotou 10 (1 °C se rovná 10mV).

$$teplota = \frac{U (mV)}{10}$$

2.5 RELÉ

Relé je součástka sloužící k oddělení dvou obvodů. Jedním obvodem dochází prostřednictvím relé k sepnutí obvodu druhého. (VODA, 2018)

Toto relé snese zatížení střídavým proudem 7 A při napětí 240 V, popř. 10 A při napětí 125 V. V případě stejnosměrného proudu lze zatížit relé 10 A při 28 V. Relé je řízeno přivedením digitální hodnoty LOW nebo HIGH na pin A4. Na základě uvedení pinu A4 do stavu LOW nebo HIGH dochází k změně na konektoru. Tento konektor má vývody NC, COM a NO. Princip fungování relé je znázorněn níže.



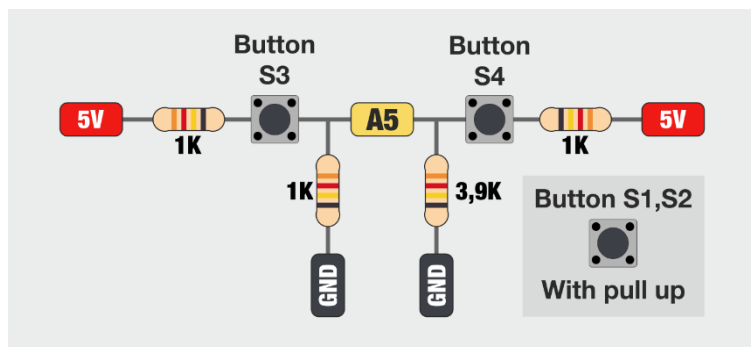
Obrázek 5 - Schéma zapojení relé, zdroj: <https://ncd.io/relay-logic/>

Relé můžeme spínat i analogovou hodnotou. Tato hodnota v případě konkrétního kitu TinyLab využívaného pro zpracování této práce byla 127 a vyšší.

Při nastavení pinu se musí uvést celý název daného pinu, tedy pinu A4. V případě, že se inicializuje pouze pin 4, relé nereaguje na případnou změnu hodnot.

2.6 TLAČÍTKA

Tlačítka v případě TinyLab jsou připojena na několik pinů, ze kterých lze číst jejich hodnoty. Tlačítka S1 a S2 jsou připojena na digitální piny D8 a D9. V tomto případě při stisku dochází k přechodu do hodnoty LOW, jinak jsou v hodnotě HIGH. Když je hodnota úrovně napětí logické 0 vyšší než hodnota logické hodnoty 1. Tyto piny jsou chráněny pull-rezistory zabraňujícím zkratu při sepnutí tlačítka.



Obrázek 6 - schéma zapojení tlačítek S3, S4, zdroj: <https://www.tinylab.cc/wp-content/uploads/2016/05/tinylab-pinout-diagram2.png>

Na analogový pin A5 jsou připojeny další tlačítka S3 a S4. Dále je na tento pin připojeno i tlačítko rotačního enkodéru. Jelikož se jedná o analogový pin, hodnoty zde mohou být 0 až 1023. Při stisku jednotlivých tlačítek se musí nejprve zjistit hodnota, která značí sepnuté tlačítko. V následující tabulce je patrné, že na analogovém pinu A5 lze číst celkem 8 hodnot. Je samozřejmostí, že hodnoty na pinu A5 mohou být drobně odlišné vlivem tolerance použitých součástek.

S3 stisknuto	S4 stisknuto	S5 stisknuto	hodnota pinu A5
ne	ne	ne	0
ne	ne	ano	90
ano	ne	ne	206
ano	ne	ano	266
ne	ano	ne	510
ne	ano	ano	535
ano	ano	ne	568
ano	ano	ano	588

Vysvětlivky: S5 = tlačítko rotačního enkodéru

Tabulka 1 - tabulka dosažitelných hodnot z pinu A5

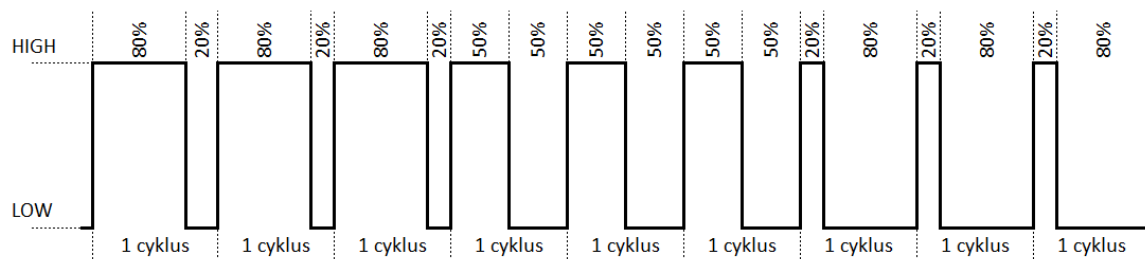
2.7 LED DIODY

LED dioda neboli světlo emitující dioda, je součástí, která díky přeskoku elektronů dochází k emitování fotonů. Díky tomu dochází ke vzniku světla. (KESL, 2003 str. 32)

V případě TinyLab jsou k dispozici v řadě pod 7segmentovým displejem 4 LED diody. Tyto diody jsou připojeny na piny D10 až D13. Ještě je možnost využít 1 vestavěnou diodu, která není připojena k žádnému pinu, ale je ovládána pomocí příkazu `LED_BUILTIN`. Integrované

diody jsou pouze červené barvy, ale je možné ovládat jejich jas prostřednictvím PWM (pulzně šířková modulace).

Pulsně šířková modulace využívá střidu, což je poměr aktivních a neaktivních stavů. Na obrázku pod textem jsou patrné rozdíly mezi jednotlivými střídami. V případě regulace svitu LED diod dochází k přechodu do aktivního stavu v případě 50% střídý jen na polovinu času dané periody. (SELECKÝ, 2016 str. 107)



Obrázek 7 - Střída, autor: Jiří Noska

Piny lze regulovat pomocí PWM modulace v rozsahu hodnot 0 až 255. Pro přesný výpočet regulace lze použít vzorec:

$$\frac{\text{procentuální hodnota stavu } 1}{100} * \text{rozsah hodnot.}$$

Pro výpočet 20% jasu LED diody se dosadí tato hodnota do vzorce a vypočítá hodnota, kterou následně zapíšeme do příkazu `analogWrite()`.

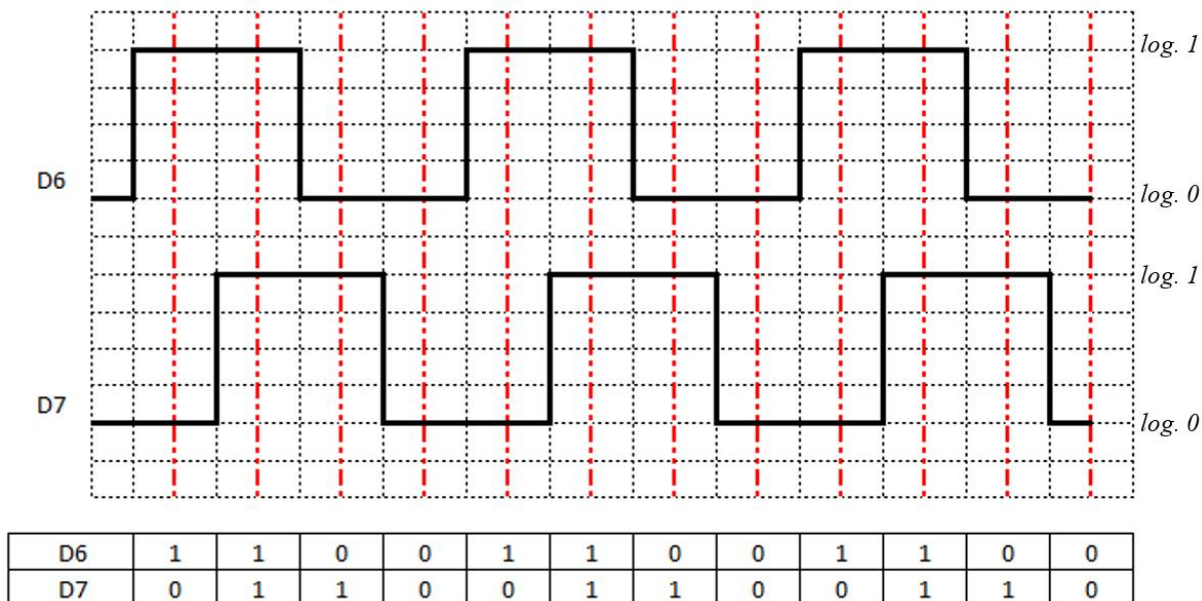
$$\frac{20}{100} * 256 = 51,2$$

Tuto hodnotu je nutno dále zaokrouhlit na celá čísla, takže výsledný zápis pro rozsvícení LED diody na pinu D13 na hodnotu 20% svítivosti pomocí PWM modulace je `analogWrite(13, 52)`.

2.8 ROTAČNÍ ENKODÉR

Rotační enkodér je zařízení, které dokáže převést rotační pohyb na sled impulsů. Díky tomuto zařízení v kitu lze odečítat pohyb a jeho směr. (VOJÁČEK, 2017)

Z jednotlivých pinů D6 a D7 lze číst digitální hodnotu na základě otáčení hřídele. Tyto hodnoty jsou oproti sobě posunuty, a tak reprezentují nejen samotný pohyb, ale také směr tohoto pohybu.



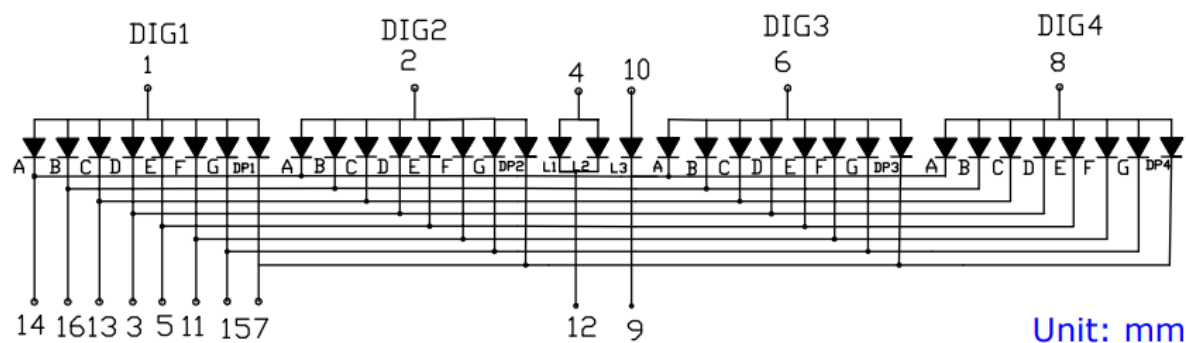
Obrázek 8 - ukázka principu snímání pohybu, autor: Jiří Noska

Pokud bychom snímali pouze signál z digitálního pinu D6 nebo samotný D7, pak můžeme rozeznat rychlost rotace hřídele enkodéru. Vzájemným porovnáním stavů, které jsou generovány na pinech D6 a D7, rozpoznáváme směr rotace hřídele. (SCHMACHTL CZ, 2005)

2.9 7SEGMENTOVÝ ZOBRAZOVACÍ PRVEK

Jedná se o zobrazovací prvek, který se skládá ze segmentů majících určitý tvar. Přivedením napětí na určitý segment dochází k jeho rozsvícení. Rozsvícením různých segmentů můžeme vytvořit námi požadovaný obrazec. (BURGER, et al., 1989 pp. 483-484)

Na desce TinyLab jsou čtyři 7segmentové zobrazovací prvky. Pokud chceme rozsvítit daný segment, pak na pin daného segmentu musíme přivést napětí. V takovém případě by ale tento modul měl 16 vývodů.

Obrázek 9 - zapojení 7segmentového displeje se čtyřmi členy, zdroj: <https://z9q5c2a5.stackpathcdn.com/wp-content/uploads/2018/08/SH5461AS.pdf>

Díky ovladači MAX7219 (VODA, 2018), který je součástí kitu TinyLab, je možné připojit a řídit displej jen pomocí trojice pinů D10, D11 a D12. Tyto piny jsou totožné s piny LED diod umístěných na desce, takže při aktivaci 7segmentového displeje dojde i k rozsvěcování LED diod. Pro zobrazování na 7segmentovém displeji musí být přepnut příslušný přepínač do pozice „To Control 7-segment“.

Knihovna LedControl, která se využívá pro ovládání displeje, poskytuje základní přednastavená písmena a čísla. (FAHLE, 2015)

2.10 LCD DISPLEJ

Jedná se o zobrazovací jednotku pracující na principu kapalných krystalů. Díky přivedenému elektrickému signálu dochází ke změně orientace těchto krystalů a displej mění na dané pozici barvu. (MALINA, 2001 str. 59)

Ve vývojovém kitu TinyLab je použit znakový LCD displej. Tento displej je schopen, jak již z názvu vyplývá, pouze znaky. Velikost tohoto displeje je 16x2. To znamená, že na displeji lze zobrazit 16 znaků ve dvou řádcích, tedy celkem 32 znaků. (VODA, 2018)

Displej je připojen přes I2C sběrnici. Tato sběrnice umožňuje mít připojeno na dvou vodičích až 128 zařízení. U I2C komunikace je pouze jedno zařízení řídicí, ostatní jsou vždy řízené. Řízení zařízení pak musí mít přidělenou vlastní adresu o velikosti 7bitů. Adresa LCD displeje umístěného na TinyLabu je 0x20. Komunikace mezi displejem a deskou probíhá pomocí SCL a SDA vodiče. První vodič udává takt komunikace, druhý pak předává data. (VODA, a další, 2018 str. 115)

3 PROGRAMOVÁNÍ VÝUKOVÉHO KITU TINYLAB

Na webových stránkách TinyLabu lze najít odkazy software pro programování kitu TinyLab. Přímo stránky Arduino.cc nabízejí 3 nástroje pro programování. Při výběru vhodného softwaru velmi často záleží jen na osobních preferencích jako například cena nebo znalost daného softwaru.

3.1 ARDUINO WEB EDITOR

Arduino Web editor je způsob online programování a ukládání sketchů (programů) v cloudové službě. Podmínkou pro možnost programování je vytvoření účtu. Dále je možné pokračovat zdarma, popřípadě si vybrat placenou verzi tohoto nástroje. Na výběr je možnost platby měsíčně nebo ročně, zde se mění cena poskytované služby a také drobně její obsah. Cena se pohybuje od 1,99\$ za měsíc po 23,99\$ za měsíc. (Arduino, 2022)

Tento nástroj je dostupný na adrese: <https://create.arduino.cc/editor/>.

3.2 ARDUINO IDE

Aplikace Arduino IDE je volně ke stažení na stránkách Arduino.cc. K 5. lednu 2022 se jedná o build s označením 1.18.19 a je dostupný pro nejrozšířenější platformy (Windows, MacOS, Linux). Tato aplikace prochází rovněž vývojem a je již ke stejnému datu dostupná aplikace Arduino IDE 2.0 ve verzi RC3.



```

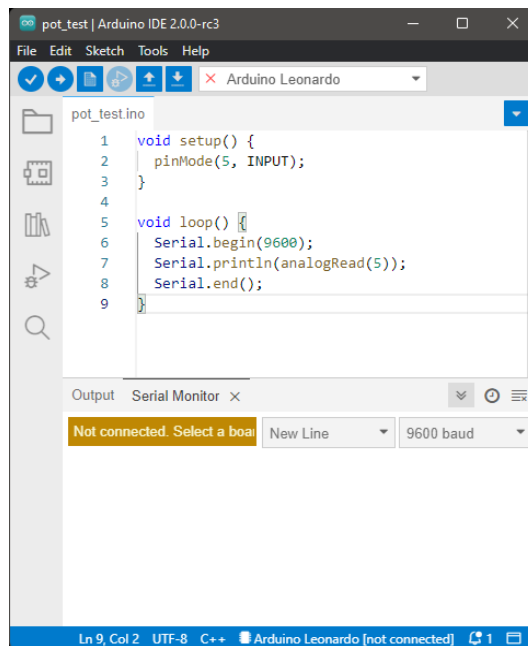
analog_button | Arduino 1.8.16
Soubor Úpravy Projekt Nástroje nápověda

analog_button
void setup() {
  pinMode (5, INPUT);
  for (int i = 11; i <= 13; i++){
    pinMode (i, OUTPUT);
  }
}

void loop() {
  Serial.begin(9600);
  Serial.println(analogRead(5));
  if (analogRead(5) > 200 && analogRead(5) < 210){
    digitalWrite (13, HIGH);
  }
  if (analogRead(5) > 505 && analogRead(5) < 515 )
    digitalWrite (11, HIGH);
}

```

Obrázek 10 - Arduino, verze 1.8.16



```

pot_test | Arduino IDE 2.0.0-rc3
File Edit Sketch Tools Help
pot_test.ino
1 void setup() {
2   pinMode(5, INPUT);
3 }
4
5 void loop() {
6   Serial.begin(9600);
7   Serial.println(analogRead(5));
8   Serial.end();
9 }

```

Obrázek 11 - Arduino IDE, verze 2.0.0-rc3

Arduino IDE jsou dostupné na adrese: <http://arduino.cc/en/software>.

3.3 ALTERNATIVY PRO PROGRAMOVÁNÍ

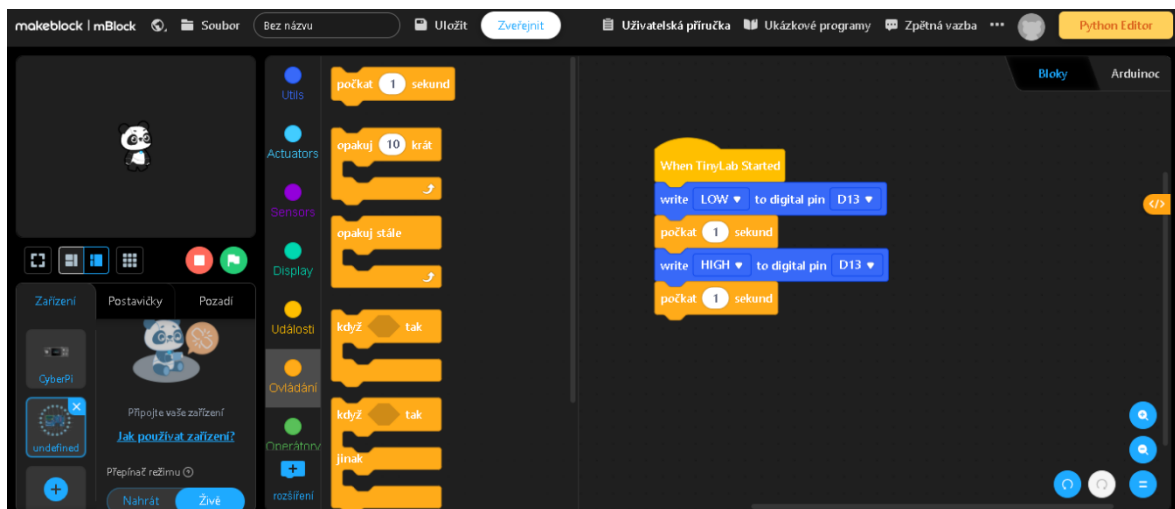
Další z možností softwaru pro programování je v dalších často používaných programech jako Microsoft Visual Studio, ve kterém lze například využít nadstavbu Atmel studio, nebo Eclipse.

3.4 SCRATCH

Programovat Arduino, stejně jako TinyLab, lze i v programech pracujících na bázi programu Scratch, který je speciálně designovaný pro dětské programátory. Mimo programování v programu Scratch lze využít i aplikaci *mBlock* nebo *Ardublock*, kde programování probíhá v předpřipravených blocích, které se dle požadované funkce spojují a vytvoří komplexní program. Tyto programy obsahující blokové programování jsou vhodné pro méně zkušené programátory.

Tyto nástroje jsou dostupné na adrese:

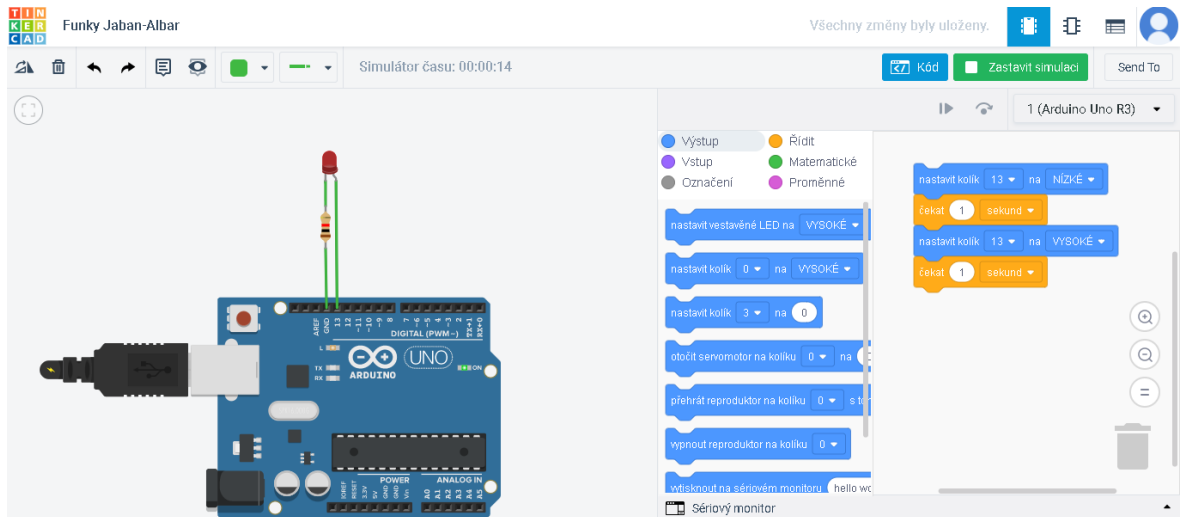
- Scratch: <http://scratch.mit.edu>,
- mBlock: <http://mblock.makeblock.com>,
- Ardublock: <http://blog.ardublock.com>.



Obrázek 12 - Ukázka z mBlock - blokového programování Arduina, zdroj: <https://ide.mblock.cc/>

3.5 SIMULAČNÍ PROGRAMY

Programovat můžeme také bez nutnosti mít připojenou Arduino desku. Dá se říci, že se jedná o takové Arduino simulátory. Mezi tyto programy patří například Tinkercad, XODIDE, Visuino. Tyto programy jsou vhodné na seznámení či rychlý test programu, ale velmi často nenabízejí plnohodnotnou náhradu programování TinyLabu.



Obrázek 13 - Ukázka práce v Tinkercad, zdroj: <https://www.tinkercad.com/things/3wm1LExCrK7-funky-jaban-albar/editel?tenant=circuits>

Tyto programy jsou dostupné na adresách:

- Tinkercad: <https://www.tinkercad.com>
- XOD IDE: <https://xod.io>
- Visuino: <https://visuino.com/>

4 VÝUKOVÝ MATERIÁL

Součástí této práce je i vlastní navržený kurz v prostředí Moodle (dále jen „kurz“). E-learningová platforma Moodle je LMS (Learning Management Systém), která umožňuje využívat širokou škálu nástrojů. (MANĚNA, 2015 str. 34)

Cílová skupina

Kurz je určen pro výuku budoucích učitelů informatiky na katedře výpočetní a didaktické techniky Fakulty pedagogické Západočeské univerzity v Plzni. Primárně je zařazen do druhého ročníku prezenční formy bakalářského studia. V případě potřeby se po úpravách může stát i základem pro realizaci výuky v kombinované formě studia.

Zaměření kurzu

Cílem tohoto kurzu je poskytnout cílové skupině potřebné znalosti pro zvládnutí programování v TinyLabu a také prověřovat pomocí autotestů a úkolů získané poznatky tak, aby bylo možné zareagovat na případné nedostatky.

Tento kurz bude u žáků postupně rozvíjet znalost programování v Tinylabu tak, aby na jeho konci byli žáci schopni splnit komplexní úlohy a vytvořit tak funkční program.

Předpokládané využití kurzu

Tento kurz by měl být využíván v rámci prezenční výuky. Žákům budou z tohoto kurzu zadávány aktivity a budou ověřovány získané znalosti. Kurz může také sloužit jako opora v případě zrušení prezenční výuky nebo v případech, kdy se žák nemůže výuky zúčastnit. Teoretické části kurzu budou sloužit pro doplnění teoretických znalostí.

Koncepce kurzu

Kurz je složen z teoretických a praktických částí. Rozdíl mezi kurzem a materiály z praktické části této práce je ve větším zaměření na teoretické znalosti.

Teoretická část vždy obsahuje základní informace a znalosti, které student potřebuje k dalšímu postupu kurzem. Teoretické znalosti jsou prověřovány v autotestech, které jsou vždy součástí dané kapitoly. Praktickou část tvoří úlohy v autotestech a dále úkoly, které musí být studentem vypracovány a následně odevzdány ke kontrole.

Kurz je koncipován od nejjednodušších úkolů po komplexnější, kterými žáci prokážou své znalosti a dovednosti v programování TinyLabu. Jedním z cílů tohoto kurzu je vytvořit prostředek pro zdokonalení programátorských dovedností daných žáků.

Kurz kopíruje úlohy uvedené v praktické části této práce. Je tedy rozdělen na 11 částí, kdy prvních devět vytváří teoreticko-praktický základ pro následující komplexní úlohy. V rámci kurzu je k dispozici pedagogovi odkaz na materiály, které ho odkážou na úložiště GitHub⁴ s potřebnými materiály. Dále jsou v kurzu skryté soubory, které vidí pouze pedagog obsahující výsledky testů či úloh, které kurz obsahuje. Pokud žáci odevzdají vytvořené programy, pak je nutno je individuálně hodnotit, jelikož každé zadání lze splnit několika způsoby a záleží vždy jen na znalostech a dovednostech jednotlivého žáka. Důležitý tedy bude především výsledek dané úlohy, který má splňovat požadavky v zadání. Postup může být pro žáka zpětnou vazbou o konkrétních chybách či nevhodně řešených postupech.

Úkoly na začátku kurzu mají vytvořit přípravu pro progres dovedností a nastavením vyšší obtížnosti by mohlo dojít k neúměrnému zatížení a tím i zmaření výukových cílů. Příklady v prvních hodinách kurzu nejsou ničím převratně novým, ale mají za cíl motivovat vhodně žáka k dalšímu objevování možností TinyLabu. Dalším cílem těchto základních úloh je poskytnutí schémat pro pochopení ovládnutí jednotlivých komponentů. Právě díky pochopení činnosti každé komponenty mohou žáci překročit ke komplexnějším, a na žáka i náročnějším aktivitám. S přibývajícemi znalostmi jsou na žáka kladeny i větší nároky v podobě komplexnějších úloh, díky kterým si žák prohlubuje programátorské dovednosti. Komplexní úlohy v podobě kuchyňské minutky, převodníku z dekadické soustavy a vysílače Morseovy abecedy, mohou plně sloužit pedagogovi k zpětné vazbě a získání přehledu o dovednostech jednotlivých studentů.

Tento kurz by mohl sloužit i jako opora pro kombinované či distanční studium. V tomto případě je ale nutno žákům poskytnout potřebné vybavení v podobě TinyLabu. Možnosti simulačních programů, které jsou k dispozici a které jsou uvedeny v předchozí kapitole, nemusí plně pokrývat nároky, které kurz klade na žáka. V případě, že nebudou mít žáci

⁴ GitHub je jedna z vývojových platforem, nabízející verzovací nástroje, bezplatný webhosting

k dispozici žádný z kitu TinyLab, pak tento kurz může studentům poskytnout schéma znalostí, které by měli obsáhnout.

4.1 SADA ÚLOH DEMONSTRUJÍCÍ VYUŽITÍ VÝUKOVÉHO KITU TINYLAB

Sada úloh, které jsou představeny v této kapitole, tvoří základ pro praktické úkoly v kurzu. V rámci 13 hodin, které na Západočeské fakultě, Katedře výpočetní a didaktické výuky, vyplívají z časové dotace pro výuku programování, se budoucí učitelé seznámí se základy programování TinyLabu. Díky postupně vzrůstající náročnosti lze takto seznámit budoucí pedagogy s programováním TinyLabu tak, aby mohli následně své získané znalosti použít ve výuce.

Předpokladem je alespoň částečná znalost programování, které studenti již měli v rámci dosavadního studia na vysoké škole. Celková vzrůstající úroveň aktivit však není velmi náročná. Aktivity mají za cíl seznámit budoucího učitele s jednou z možných alternativ výuky programování.

Veškeré podklady vytvořené v rámci této práce jsou dostupné na online úložišti Github:

<http://www.github.com/JiriNoska/tinylab> (dále jen „úložiště“).

Rozcestník podkladů určených pro pedagoga je v podobě webových stránek založených na šablonách platformy GitHub je dostupný na adrese:

<https://jirinoska.github.io/tinylab/> (dále jen „rozcestník“).

Rozcestník slouží pro rychlejší orientaci v aktivitách a napomůže zrychlenému vyhledávání vhodného materiálu.

V úložišti je dostupný ke každé hodině i metodický manuál. Ten je, v případě, že jsou aktivity vytvářeny pro 1 vyučovací hodinu, nastaven na 40 minut čistého času. Zbývající čas výukové jednotky je určen na další aktivity (příchod, pozdrav, závěrečná reflexe).

Využití komponent, příkazů a prekonceptů je uvedeno v tabulce (Tabulka 3 - Tabulka aktivit), díky které se může pedagog snadněji orientovat v potřebných a již získaných znalostech.

V rámci aktivit jsou dostupné i diagnostické aktivity, které nutně nemusí proběhnout v danou hodinu, ale mohou sloužit pedagogovi jako diagnostika vědomostí, které je nutno

prohloubit a fixovat. V případě, že aktivity na ověření znalostí chybí, jsou ale nahrazeny komplexnější úlohou, který slouží obdobným způsobem. Úlohy jsou poskládány od nejjednodušších po složitější. Komplexní úlohy mají za cíl prověřit znalost problematiky programování v TinyLabu. Ačkoliv jsou k dispozici ukázková vypracování zadání, tak výsledky žáků nemusí nutně vypadat totožně. Důležitý je však výsledek, kterým žáci budou muset prezentovat své splnění zadání.

Aktivity jsou vytvářeny převážně pro explorační metodu, kdy žáci, kteří jsou minimálně na základní úrovni ve znalosti programování, objevují dané postupy týkající se programování TinyLabu.

Program, ve kterém byly úlohy vypracovány je Arduino IDE. Tento program je ke stažení a používání zdarma, chybí složitější nastavování či dohrávání dalších komponent. Pro školské prostředí a seznámení se s TinyLabem se mi proto právě tento program zdál nejvhodnější.

V rámci této práce byl vytvořen i Moodle kurz, který svým složením kopíruje následující kapitoly. Kurz je však více zaměřen na teoretické znalosti a je tedy možné jej využívat v rámci nestandardní výuky (online výuka). Součástí tohoto kurzu jsou autotesty a úkoly pro vypracování.

4.2 SEZNÁMENÍ SE S VÝUKOVÝM KITEM TINYLAB

Sada následujících aktivit je určena pro jednu vyučovací hodinu. V ní se žáci seznámí s TinyLabem, nahrají a spustí první program.

Potřebné pracovní pomůcky ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita01/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Aktivita 1.1 Seznámení se s komponenty výukového kitu TinyLab

Vypracuj pracovní list.

Cíle aktivity:

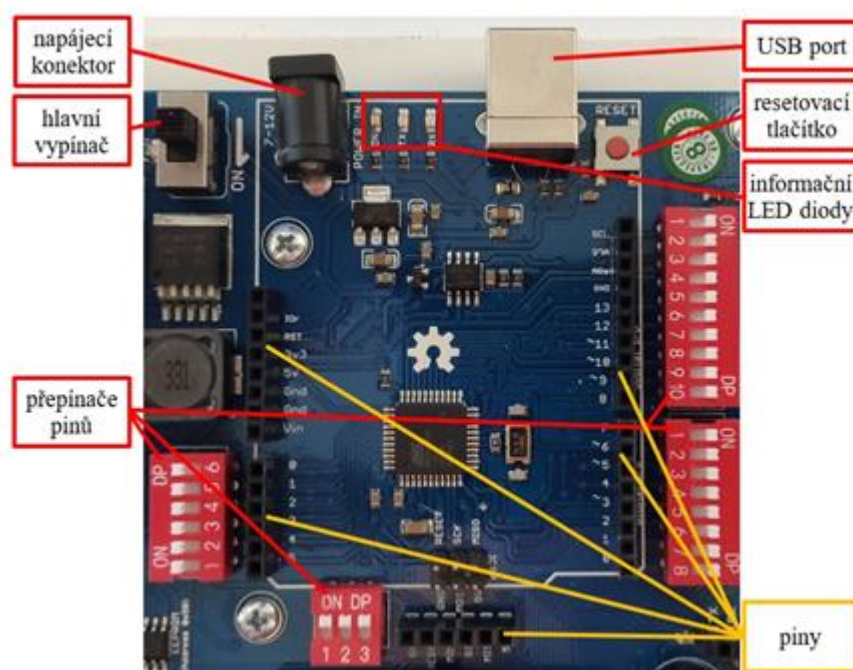
- žák vyhledá jednotlivé komponenty,

- žák vyplní v pracovním listu další informace o komponentech.

Metodika pro pedagoga

V rámci první aktivity budou žáci s pedagogem diskutovat o komponentech na desce TinyLabu. V rámci výše uvedeného pracovního listu jsou uvedeny nápovědné texty, které odhalí pozici komponenty na desce. Tuto pozici následně žáci vepíší do připravených koleček, tak, jak je patrné u první komponenty (LED diody). Díky této aktivitě žáci získají orientaci na desce a vypíší i piny, které jsou k těmto komponentům přiřazené.

V rámci diskuse a odhalování jednotlivých komponent pedagog vysvětlí a ukáže nejenom jednotlivé komponenty (2 TinyLab), ale i prvky na samotné základní desce (Obrázek 14).



Obrázek 14 - Základní deska kitu TinyLab, autor: Jiří Noska

Zde je důležité upozornit žáky na napájecí konektor, který bychom používali v případě samostatného využití desky TinyLab. TinyLab lze napájet i přes USB konektor, díky kterému také TinyLab komunikuje s počítačem a prostřednictvím kterého nahráváme program do TinyLabu. Základní deska má na sobě piny, které jsou v případě přepínačů pinů v poloze „off“ aktivní, takže na tyto piny lze umístit vlastní přídavný komponent.

Podrobnější představení komponent proběhne až v kapitolách, kde budou tyto komponenty využívány.

Aktivita 1.2 Připojení kitu TinyLab

Připoj TinyLab k počítači, nastav software a pomocí programu rozsviť diodu L1.

Cíle aktivity

- žák zprovozní TinyLab na počítači,
- žák spustí svůj první program v TinyLabu.

Metodika pro pedagoga

Žáci budou pracovat metodou explorační. Žák má, v případě varianty TinyLab Maker kit (kapitola 1.1), k dispozici USB kabel sloužící k propojení kitu s počítačem, dále napájecí kabel do sítě a napájecí konektor na 9 V baterii. Žáci, kteří připojí TinyLab prostřednictvím USB kabelu a zároveň jej budou napájet síťovým nebo bateriovým napájením mohou zkusit toto napájení odebrat. Díky tomuto žáci zjistí, že napájet TinyLab lze jen prostřednictvím USB kabelu.

Dalším krokem bude nastavení TinyLabu v počítači. V případě, že program není nainstalován, provedou žáci stažení a instalaci⁶. Instalaci provedeme v záložce nástroje. Nejprve připojíme kit pomocí USB kabelu k počítači. V záložce *Nástroje* -> *Vývojová deska* zvolíme vhodnou desku. V případě TinyLabu je základní deskou Arduino Leonardo. Zvolíme tedy tuto desku a přesuneme se do kolonky *Port*. Zde bychom již měli vidět využívané porty. Označíme ten, u kterého je uveden název Arduino desky.

Při spuštění programu si již žákům ukáže základní program. Ten tvoří dvě základní funkce. Vhodnou otázkou je zde:

Vysvětli, co mohou v programu vykonávat funkce `setup` a `loop`.

Vzhledem k základní znalosti anglického jazyka lze odvodit, že funkce `setup` nastavuje a funkce `loop` slouží k opakování.

⁶ Program Arduino IDE ke stažení na adrese: <https://www.arduino.cc/en/software>

V pracovním listu mají žáci k dispozici příkazy, které mají použít. Žáci poskládáním vytvoří program a spustí.

Pedagog bude pokládat otázky tak, aby žáci vysvětlili princip programu.

Vhodné otázky

Kterým příkazem aktivuji daný pin?

Pin mohu nastavit jako vstupní nebo výstupní, jak to poznám?

Jakým příkazem zapisuji hodnotu na daný pin?

Jakou logickou hodnotou rozsvítím LED diodu?

Aktivita 1.3 Úprava programu

Rozsviť jinou diodu než L1.

Cíle aktivity

- žák využije znalosti funkce konstanty.

Metodika pro pedagoga

Tato aktivita slouží jako diagnostická, ve které žák prokazuje pochopení principu konstant. Pro správné splnění zadání lze pouze změnit hodnotu konstanty. LED diody

4.3 LED DIODY

Sada aktivit s využitím LED diod je koncipována pro jednu vyučovací hodinu, během které dojde k vytvoření několika modifikací základního zadání. Žáci již umí vytvořit základní program, kompilovat ho a nahrát do TinyLabu. Vhodnou výukovou metodou této hodiny bude metoda explorační, kdy žáci budou postupně objevovat nové postupy programování TinyLabu.

Potřebné materiály ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita02/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Aktivita 2.1 Program v TinyLabu

Odpověz na otázky z pracovního listu.

Cíle aktivity

- žák analyzuje program,
- žák argumentuje, jak umístění příkazu *delay* v programu ovlivní chod programu.

Metodika pro pedagoga

V této aktivizační části žáci vypracují pracovní list. V něm mají ukázky 3 programů a musí označit odpovědi jejich funkci. Zároveň musí při dotazu pedagoga argumentovat, proč zvolili právě tuto odpověď.

Aktivita 2.2 Blikající LED dioda

Vytvoř program s blikající LED diodou L1. Jaké minimální zpoždění může být nastaveno, aby lidské oko zachytilo blikání LED diody?

Cíle aktivity

- žák vytvoří program s blikající LED diodou L1,
- žák upraví hodnotu zpoždění, aby dosáhl efektu blikání pozorovatelného okem,
- žák pohledem otestuje funkčnost programu.

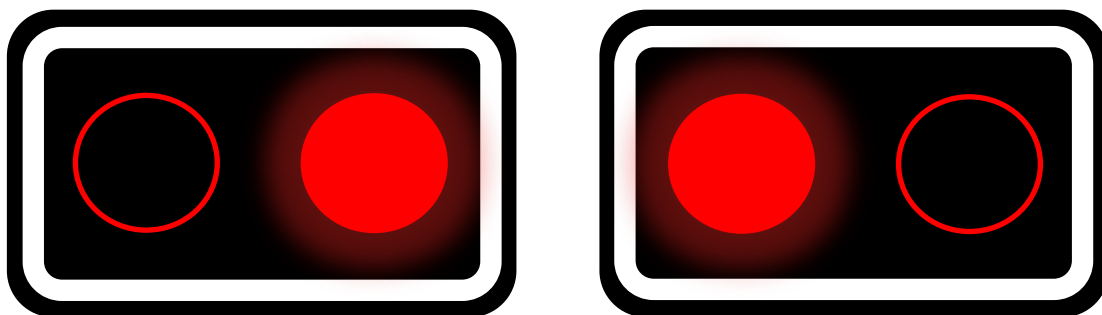
Metodika pro pedagoga

V pracovním listu, který žáci vyplnili předchozí aktivitu (Aktivita 2.1 **Chyba! Nenalezen zdroj odkazů.**), je tento program již řešen. Tento pracovní list po vyplnění žáci odevzdají, aby nemohlo dojít k pouhému přepisu. Žáci tedy již viděli řešení, ale pravděpodobně si ho celý nezapamatovali. Program tedy budou muset sami vytvořit.

Otestování hotového programu dochází vizuální kontrolou. V případě, že LED dioda bliká, je program vytvořen správně. Program budou poté upravovat tak, aby dosáhli efektu blikání s nejmenším možným zpožděním.

Aktivita 2.3 Blikající semafor

Vytvoř blikající semafor, který vidíš na železničním přejezdu.



Cíle aktivity

- žák navrhne program pro blikání dvou LED diod,
- žák vytvoří funkční program a otestuje ho.

Metodika pro pedagoga

Žáci v této úloze aplikují své poznatky z předešlých aktivit. Zde budou vytvářet nové konstanty sloužící k zapojení dalších diod do výsledného programu. Pokud žáci nesprávně navrhnu systém pro blikání diod, může se tento moment stát kritickým úsekem. K tomuto slouží pomůcka zobrazení tohoto semaforu, na kterém pedagog může vysvětlit celý postup.

Vytvoření tohoto programu se skládá ze tří zásadních kroků. Prvním je vytvoření konstant s piny, na kterých jsou připojeny LED diody. Druhým je aktivace pinů a třetí vytvoření samotného blikání LED diod. Rozšíření tohoto zadání může být stanovení konkrétních LED diod a jejich následná výměna za LED diody jiné.

Aktivita 2.4 Běžící had

Vytvoř z LED diod „běžícího hada“. LED diody se budou postupně po jedné rozsvěcovat a poté zhasínat. Výměnou hodnoty zpoždění zrychluj nebo zpomaluj běh hada.

Cíle aktivity

- žák navrhne jednotlivé kroky programu,
- žák vytvoří a otestuje program.

Metodika pro pedagoga

Tato úloha bude klást důraz na představivost žáka. Žák musí postupně zapínat a vypínat jednotlivé LED diody. Stejně jako předchozí aktivitu (Aktivita 2.3) i tuto je nutné členit na jednotlivé kroky. Úlohu můžeme rozdělit na jednotlivé kroky:

- vytvoření konstant (pinů LED diod),
- aktivace pinů prostřednictvím příkazu `pinMode`,
- funkci `loop`, kde bude postupné zapnutí a vypnutí diod.

Pro lepší znázornění může pedagog použít zobrazení „běžícího hada“, které je v úložišti. Tato pomůcka pomůže odbourat možné kritické místo této hodiny, kterým bude chybné určení logických hodnot. Proto tato pomůcka rozděluje proces běžícího hada na jednotlivé kroky, které žákům pomohou lépe si celý proces představit.

Rozšířením této aktivity může být změna rychlosti běžícího hada.

Aktivita 2.5 Ověření znalostí

Vyplň pracovní list.

Cíle aktivity

- žák odhalí chybu v programu,
- žák opraví chybu,
- žák vytvoří funkční program.

Metodika pro pedagoga

Žáci mají zadaný program a zobrazenou chybovou hlášku, která se zobrazila při kompilaci programu. Žáci díky tomu mohou chybu odhalit a odstranit. Chyba je zde na řádce 2, kde chybí středník za příkazem. Toto je jedna z nejčastějších syntaktických chyb, která je však snadno odhalitelná díky chybovému hlášení při kompilaci programu.

Dalším úkolem je odpověď na otázku o fungování programu. LED dioda se v tomto programu nerozsvítí. Posledním úkolem je úprava tohoto programu tak, aby výsledkem byla blikající LED dioda.

4.4 POTENCIOMETR

Tato úloha je koncipována na jednu vyučovací hodinu. Prekoncepty pro tuto hodinu budou znalosti datových typů, proměnných a konstant.

Potřebné materiály ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita03/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Cíle této hodiny jsou:

- žák přečte hodnotu ze vstupního zařízení,
- žák vypíše hodnotu ze sériové komunikace.

Aktivita 3.1 Sériová komunikace

Otevři pracovní list. Vytvoř program z uvedených příkazů.

Cíle aktivity

- žák zkompiluje program se sériovou komunikací,
- žák vysvětlí rozdíl mezi umístěním ve funkci `setup` nebo ve funkci `loop`,
- žák definuje rozdíl mezi příkazem `print` a `println`,
- žák spustí a otestuje program.

Metodika pro pedagoga

Žákům ještě nebylo uvedeno, kde lze sériovou komunikaci přečíst. Po zkompilování programu tedy žáci nebudou nikde vidět jejich sériovou komunikaci. Nejlepší možností je nechat žáky pátrat, jak mají zobrazit sériovou komunikaci. Tu zobrazí pomocí nástroje sériový monitor. V případě, že by hledání bylo delší, pak je možné žáky navést kladením otázek.

Vhodné otázky:

*Jakým **nástrojem** si zobrazíte sériovou komunikaci vyslanou z TinyLabu do počítače?*

Jakou klávesovou zkratkou lze vyvolat okno se zobrazením sériové komunikace?

Žáci explorací objeví nástroj *Sériový monitor* a zde také vypíší komunikaci.

Aktivita 3.2 Výpis hodnoty jezdce potenciometru

Vypiš hodnotu z jezdce potenciometru.

Cíle aktivity

- žák přečte hodnotu komponenty kitu TinyLab,
- žák vypíše tuto hodnotu na sériovém monitoru.

Metodika pro pedagoga

Tato úloha může být opět problematická z důvodu neznámých operací, které bude nutné vytvořit. Žáci jsou navedeni otázkami, které si mohou klást. Na všechny tyto otázky již mají z předchozích aktivit znát odpověď.

Vhodné otázky:

Jakou konstantu bude mít pin potenciometru?

Budeme aktivovat pin potenciometru jako vstupní nebo výstupní?

Pro diagnostiku správnosti postupu pedagog položí žákům otázku na nejvyšší a nejnižší dosaženou hodnotu vypsanou na monitoru. Nejnižší hodnota je nulová, maximální je 1023.

Aktivita 3.3 Ovládání hlasitosti

Otáčením jezdcem potenciometru rozsvěcuj postupně LED diody. V případě, že bude jezdec za třemi čtvrtinami dráhy, rozblikej LED diody jako varovný signál.

Cíle aktivity

- žák navrhne podmínky,
- žák otestuje funkčnost podmínek v programu.

Metodika pro pedagoga

Motivací může být výklad o využití potenciometru jako ovladače hlasitosti. V případě, že dochází k velmi hlasitému poslechu hudby, pak může dojít k poranění sluchového ústrojí a hrozí trvalé poškození sluchu.

V rámci této úlohy musí žáci využít předchozí znalosti podmínek, jak je znají z předchozího programování. Pokud by žáci nevěděli, jak nastavit hodnoty pro rozsvěcování, pak je k dispozici zadání s nápovědou hodnot, které mohou využít.

Kritickým momentem v této aktivitě může být problém s podmínkami. V tomto případě lze žákům napovědět českým překladem požadovaných podmínek, tedy:

Když je hodnota potenciometru menší než 255, rozsvít diodu L1.

Když je hodnota potenciometru menší než 511, rozsvít diodu L2, jinak ji zhasni.

Nápověda by však měla v tomto bodu přestat, aby se žáci snažili vytvořit další podmínky sami.

Tímto bude program hotový a po zkompilování jej bude možno vizuálně otestovat.

Aktivita 3.4 Ověření znalostí

Program nefunguje. LED dioda neustále svítí, i když je podmínka dobře. Vyhledej chybu, odstraň jí.

```
const int LED = 13;
const int POT = 0;

void setup() {
  pinMode(POT, INPUT);
  pinMode(LED, INPUT);
}
void loop() {
  Serial.println(analogRead(POT));
  if (analogRead(POT) < 250) {
    digitalWrite(LED, HIGH);
  } else {
    digitalWrite(LED, LOW);
  }
}
```

Cíle aktivity

- žák odhalí chybu v programu,
- žák opraví chybu v programu,
- žák popíše fungování programu.

Metodika pro pedagogy

Tato diagnostická úloha je zaměřena na znalost rozdílu mezi vstupní a výstupní funkcí pinů. U příkazu `pinMode` jsou tyto funkce totožné, tedy vstupní. Pin LED diody musí však být nastaven jako výstupní.

Diagnostickou aktivitu lze i pro žáky méně zkušené. Úprava aktivity spočívá v omezení, kde se daná chyba nachází, to znamená, že omezíme zadání pouze na funkci `setup`.

Dále se soustředí i na názvosloví. V rámci první otázky se ptáme na funkci, to znamená, že správná odpověď by měla být „setup“. Poslední částí prověří celkovou orientaci v programu, kdy žák musí popsat fungování programu.

4.5 TLAČÍTKA

Výuka ovládání tlačítek v TinyLabu bude probíhat v rámci jedné vyučovací hodiny. Potřebné prekoncepty by již žáci měli mít díky předchozímu programování.

Cíle této hodiny jsou:

- žák popíše rozdíly mezi analogovým a digitálním vstupem,
- žák vytvoří program, kde se budou používat tlačítka.

Potřebné materiály ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita04/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Aktivita 4.1 Hodnoty pinů tlačítek

Zjisti, jaké hodnoty lze získat z pinů, kam jsou připojena tlačítka.

Cíle aktivity

- žák vypíše hodnotu analogového a digitálního signálu,
- žák otestuje funkčnost programu.

Metodika pro pedagoga

Vy počátku této aktivity si žáci musí zodpovědět na otázky:

- na jaké piny jsou připojena tlačítka,
- jsou tyto piny analogové nebo digitální,
- nastavím je jako piny vstupní nebo výstupní.

Žáci mají všechny potřebné prekoncepty pro úspěšné zvládnutí této úlohy. Těmito prekoncepty jsou:

- orientace na desce kitu TinyLab,
- rozlišení analogového/digitálního pinu,
- nastavení pinu.

V rámci této úlohy žáci porovnají hodnoty z analogových pinů mezi sebou. Cílem této aktivity je, aby žáci vysvětlili, proč se tyto hodnoty mohou drobně lišit. Je to z důvodu výroby a tolerance součástek.

V poslední části pracovního listu žáci odpovědí na otázku, zda se v případě digitálních hodnot jedná o pozitivní nebo negativní logiku. Jako nápověda může sloužit obrázek v pracovním listu.

Aktivita 4.2 Rozsvícení LED diod pomocí tlačítek

Rozsviť LED diody stiskem tlačítka, které se nachází pod danou LED diodou.

Cíle aktivity

- žák vytvoří podmínky pro správnou funkci programu,
- žák stiskem tlačítka otestuje funkčnost programu.

Metodika pro pedagoga

Tato aktivita slouží k fixaci znalosti podmínek, určení a správné nastavení pinů. Ve vyučovací pomůcce je zobrazení LED diod a tlačítek, na kterém může pedagog ukázat princip. Samotný program, díky znalosti všech prekonceptů, žáci vytvoří bez další nápovědy.

Žáci nejdříve nastaví piny D8 a D9 jako digitální a poté pin A5 jako analogový. Tyto piny budou sloužit jako vstupní. Piny LED diody budou nastaveny jako piny výstupní. V případě nastavení podmínek pro analogový vstup budou muset žáci zjistit hodnoty po stisku těchto tlačítek. Hodnotu tlačítek již získali v předcházející úloze (Aktivita 4.1).

Diagnostika správnosti bude provedena stiskem jednotlivých tlačítek a vizuální kontrola rozsvícení a následného zhasnutí LED diody nad tímto tlačítkem.

Aktivita 4.3 Tlačítko pro zapnutí a vypnutí

Využij příkazy ze zadání, vytvoř konstanty a proměnné. Na LED diodě simuluj funkci programu.

Cíle aktivity

- žák analyzuje části programu a seřadí je do správného pořadí,
- žák vytvoří dle dostupných informací konstanty a proměnné,
- žák reprezentuje chod programu pomocí LED diody,
- žák představí chod programu.

Metodika pro pedagoga

Tato aktivita může být skupinová, kdy skupina (maximálně 2 žáci) navrhne řešení tohoto zadání.

V této aktivitě budou mít žáci k dispozici části programu, které musí seřadit tak, aby tvořili jeden celek. Cílem bude, aby žáci analyzovali části programu a dokázali je seřadit do programu tak, aby byl funkční. Dále žáci navrhnou a realizují podmínku, která bude rozsvěcovat a zhasínat LED diodu.

Program po stisku tlačítka zapne diodu a po dalším stisku jí vypne.

Ověření funkčnosti programu žáci reprezentují pomocí stisku tlačítka, kdy LED dioda musí zůstat rozsvícená a po dalším stisku tlačítka musí LED dioda zhasnout.

Tato aktivita je náročnější z hlediska znalostí a orientace v programu, takže zároveň může sloužit jako diagnostická.

4.6 RELÉ, PIEZOELEKTRICKÝ MĚNIČ

Aktivita, která se zabývá vysvětlením funkce bzučáku a relé, je koncipována pro jednu vyučovací hodinu.

Cíle této hodiny jsou:

- žák ovládá spínání relé pomocí programu,
- žák rozezná bzučák.

Potřebné materiály ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita05/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Žáci se s pojmem relé setkali v první hodině. Pro připomenutí a jako aktivizační část hodiny lze relé přirovnat k vypínači, který je ale ovládán pomocí programu. Svorkovnice relé má označení COM, NC a NO. Při zapojení zařízení (např. motor, větrák, světlo) mezi svorky COM a NC dochází k spojení – relé je sepnuto a pomocí programu lze relé rozepnout. Při zapojení zařízení mezi svorky COM a NO dochází k opačné situaci. Funkčnost spínání relé lze ověřovat pouze sluchem, kdy dochází k poměrně hlasitému zvuku sepnutí.

Upozornění

Pin relé je nutno označovat celým názvem, to znamená, že nepostačuje pouze označení pin číslo 4, ale musí se použít název A4.

Aktivita 5.1 Spínání relé

Zjisti, při jaké nejmenší analogové hodnotě dojde k sepnutí relé.

Cíle aktivity

- žák ovládá různými hodnotami piny relé,
- žák vypíše nejmenší hodnotu, při které dojde k sepnutí relé,
- žák otestuje funkčnost programu.

Metodika pro pedagoga

Aktivita k ovládání relé bude pouze jediná. Relé je komponent, kterým se ovládají další periférie, které ale v setu TinyLab Maker kit absentují.

V rámci této aktivity budou muset žáci vyhledat nejmenší hodnotu, při které dojde k sepnutí relé. K výsledku se lze dostat 2 způsoby. První cestou je ruční změna hodnot v programu, která je zdlouhavější, ale může procvičit kódování. Po každém změně hodnoty bude docházet k rozhodovacímu procesu, zda se musí nastavit hodnota menší nebo větší. V simulovaném případě došlo k zjištění a potvrzení vhodné hodnoty až v desátém kroku (Obrázek 16), to znamená desetkrát upravit hodnotu v programu, zkompilovat, nahrát a ověřit.

<i>krok</i>	hodnota	sepnuto
1	512	1
2	256	1
3	128	1
4	64	0
5	96	0
6	112	0
7	120	0
8	124	0
9	126	0
10	127	0

Obrázek 16 - Tabulka výpočtu minimální hodnoty pro sepnutí relé, autor: Jiří Noska

Druhou možností je zvolit načtení hodnoty z jezdce potenciometru a touto hodnotou řídit spínání relé. Pouhým otáčením jezdce potenciometru a výpisem žáci brzy zjistí danou hodnotu.

V rámci této aktivity lze takto demonstrovat efektivnost některých postupů. Žáci sami rozhodnou, který postup je efektivní a který nikoliv.

Aktivita 5.2 Bzučák

Stiskem tlačítka zapni bzučák, stiskem druhého tlačítka ho vypni.

Cíle aktivity

- žák zapne bzučák jedním tlačítkem, druhým ho vypne,
- žák otestuje stiskem tlačítek funkčnost programu.

Metodika pro pedagoga

V rámci této aktivity žáci využijí bzučák k vydání zvukového znamení. Stiskem tlačítka ho bzučák zapnou, stiskem druhého tlačítka bzučák vypnou.

V tomto případě mohou žáci pouze použít příkaz `digitalWrite`, jelikož není stanovena frekvence daného tónu.

Funkčnost programu lze zkontrolovat auditivně. Při stisku jednoho z tlačítek uslyší pedagog tón, který po stisku druhého tlačítka ustane.

Aktivita 5.3 Zachraňte Titanic

Je 15. dubna 1912. Jsi radistou na palubě Titaniku. Po stisku tlačítka se vyše signál SOS.

Cíle aktivity

- žák vytvoří program pro signál SOS,
- žák po stisku tlačítka vyšle signál SOS.

Metodika pro pedagoga

Motivací je vyprávění o zkáze zaoceánského parníku Titanic. Na zkáze mělo vliv několik událostí. Jednou z nich, která mohla zachránit lidské životy, by bylo včasné vyslání signálu SOS.

Žáci musí v této aktivitě vytvořit signál, který se automaticky odešle po stisku tlačítka.

Kritickým momentem zde může být převod signálu SOS do programu. Musí se zde použít dvě odlišné doby zpoždění (příkaz *delay*) pro krátký a pro dlouhý tón. Na první pohled je program poměrně dlouhý a toho se žáci mohou zaleknout. Je vhodné zvukové signály postupně přidávat tak, aby se žáci v programu neztratili. Alternativou v případě nadanějších programátorů může být časový limit na vypracování programu.

Aktivita 5.4 Ověření znalostí

Vyplň pracovní list.

Cíle aktivity

- žák uvede správnou odpověď,
- žák analyzuje program,
- žák optimalizuje program.

Metodika pro pedagoga

Tato aktivita prověřuje dosavadní znalosti žáka. V pracovním listu jsou dvě výběrové otázky, které jsou zaměřeny na téma tlačítek a relé. Ve třetí otázce má žák za úkol popsat fungování programu. V tomto případě stisk tlačítka S1 zapne LED diodu a stisk tlačítka S2 LED diodu vypne. Poslední otázka se týká optimalizace programu. Žáci musí tento program analyzovat a navrhnout změny. V tomto programu jsou zcela zbytečně řádky 1, 4, 8, 10 a 11. U řádku 16 musí žáci změnit čtení z pinu 0 za proměnnou potenciometr nebo řádek 16 ponechají beze změny, ale v tom případě musí vyškrtnout řádek 15.

4.7 FOTOREZISTOR, TEPLOTNÍ ČIDLO

Výuka použití fotorezistoru je navržena na jednu vyučovací hodinu. Během první hodiny (Aktivita 1.1) žáci zjistili pozici fotorezistoru a teplotního čidla na desce a piny, na kterém jsou tyto senzory připojeny. Pozice senzorů bude v následujících aktivitách klíčová, jelikož budeme fotorezistor kvůli změně hodnot zakrývat či naopak osvětlovat a teplotní čidlo ovlivňovat dotykem.

Cíle této hodiny jsou:

- žák využije zapojení fotorezistoru v programu,
- žák vytvoří program pro světelné čidlo
- žák převede hodnotu z teplotního čidla na stupně Celsia, stupně Fahrenheita a Kelviny.

Žákům bude nutno vysvětlit funkci fotorezistoru. Jelikož ale již znají potenciometr, pak je vhodné ho přirovnat právě k této součástce.

Fotorezistor i teplotní čidlo je jako potenciometr. Místo ručního otáčení osou reaguje na světlo ze svého okolí nebo teplotu a tím mění svou hodnotu. Minimum i maximum této hodnoty je totožné jako u potenciometru.

V této vyučovací hodině není stanovena samostatná diagnostická úloha. Aktivity jsou náročnější a poskytují dobrou zpětnou vazbu o pochopení principů čtení analogových hodnot a další úkony s nimi.

Pomocný materiál ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita06/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Aktivita 6.1 Pracujeme s čísly

Vyplň pracovní list

Cíle aktivity

- žák přiřadí vhodné datové typy k proměnným,

- žák otestuje výstupy různých proměnných.

Metodika pro pedagoga

Žáci v pracovním listu získají znalosti číselných datových typů. V tabulce chybí však rozsah čísel datového typu *byte*. Ten musí sami vypočítat tak a ověřit výpisem tohoto čísla.

Dále žáci musí zjistit rozdíl mezi unsigned a signed číselnými datovými typy. Unsigned datové typy nemohou nabývat záporných hodnot.

V další úloze žáci zjišťují rozdíly mezi jednotlivými zápisy čísel a znalost prokazují odpovědí na otázky. V případě, že vydělíme celým číslem, pak je výsledek vždy celé číslo, takže v prvním zápisu by byl výsledek rovný 0. Další zápis zaokrouhlí díky zápisu číslo na dvě desetinná místa a poslední zápis vypíše číslo se zaokrouhlením na 4 desetinná místa.

Posledním úkolem je aplikace získaných znalostí ve formě opravy kódu. Žáci tento kód zadají do programu a opraví ho. Diagnostika opravy programu je vyhodnocena spuštěním programu a vypsáním čísla 0,0009765625.

Aktivita 6.2 Rozsvěcování LED diod v závislosti na osvětlení

Vytvoř z LED diod ukazatel stavu osvětlení. Při minimálním osvětlení budou LED diody zhasnuty, při vzrůstajícím osvětlení se budou rozsvěcovat a při maximálním osvětlení budou všechny LED diody svítit.

Cíle aktivity

- žák vytvoří program, který ovládá LED diody prostřednictvím fotorezistoru,
- žák ověří funkci fotorezistoru zakrýváním a osvětlováním.

Metodika pro pedagoga

V této aktivitě není pevně stanovena hranice, kdy se má jaká dioda rozsvítit. Pro žáka může být však obtížné si představit funkci tohoto programu. K tomu poslouží vyučovací pomůcka v úložišti.

Kritickým momentem v této aktivitě může být negativní hodnota fotorezistoru. Při úplném osvětlení je hodnota na pinu fotorezistoru rovna 0, při maximálním zakrytí je tato hodnota rovna 1023.

Prvním krokem bude určení konstant. Do nich vloží žáci piny fotorezistoru a všech diod. Následně ve funkci *setup* tyto piny aktivují. Pin fotorezistoru bude jako vstupní, piny LED diod jako výstupní. Funkce *loop* bude obsahovat podmínky, kde žáci provedou rozdělení podle svého uvážení.

Program po kompilaci a nahrání otestují přiložením prstu na fotorezistor (zakrytí), kdy by měly všechny diody zhasnout, a osvětlení zdrojem světla (například baterka telefonu), kdy by se měly všechny LED diody rozsvítit.

Aktivita 6.3 Výpis teploty ve stupních Celsia, Fahrenheita a Kelvina.

Proveď z hodnoty teplotního čidla přepočítání na stupně Celsia, na stupně Fahrenheita a na Kelvinů. Výsledky vypiš na sériový monitor. Tento výpis zformátuj, aby byl přehledný a dobře čitelný.

Cíle aktivity

- žák přečte hodnotu z teplotního čidla,
- žák provede přepočítání na stupně Celsia, stupně Fahrenheita a Kelvinů,
- žák provede formátování hodnot na sériovém monitoru,
- žák ověří funkci teplotního čidla stiskem čidla v prstech a sledováním změn teploty,
- žák ověří funkčnost převodu jednotek.

Metodika pro pedagoga

V pracovním listu je zobrazen výpočet teploty, ukázka datových typů a převody jednotek.

Žáci v první aktivitě této hodiny (Aktivita 6.1) v pracovním listu zjistili rozdíly mezi datovými typy *integer* a *float*. Zde vidí rozdíl mezi výsledky, které jsou schopni díky těmto datovým typům obdržet. Dále je zde informace o zaokrouhlování během výpočtu. V případě, že provedou žáci v programu výpočet, například 50/1024, pak výsledek vyjde nula, jelikož program zaokrouhlí výsledek na celá čísla. V případě 50/1024.00 dojde k zaokrouhlení na dvě desetinná místa. V případě, že za číslo napíšeme například 4, tak dojde k zaokrouhlení na 4 desetinná místa.

Převod na další jednotky musí žák nejdříve vyhledat. Toto vyhledávání prohlubuje žákovi digitální kompetence a kompetence k učení.

Je na žákově představitosti, jakým stylem provedou žáci výpis. Tato aktivita má za cíl zformátovat výsledek tak, aby byl přehledně zobrazen.

Žáci otestují funkčnost programu. Po spuštění programu a dotyku se musí měnit všechny tři hodnoty teplot.

4.8 ROTAČNÍ ENKODÉR

Aktivita týkající se rotačního enkodéru je navržena pro jednu vyučovací hodinu. S umístěním rotačního enkodéru jsou již žáci seznámeni, a to jak v rámci první hodiny (Aktivita 1.1).

Aktivizační část hodiny lze zaměřit položením otázky:

V čem se liší potenciometr od rotačního enkodéru?

Odpověď žáků se bude pravděpodobně týkat průběhu otáčení hřídele. Rotační enkodér má nekonečnou rotaci, zatímco potenciometr má jasně ohraničené krajní pozice.

Rotační enkodér slouží k snímání úhlu natočení, rychlosti, pozice a délky.

Pomocný materiál ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita07/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Aktivita 7.1 Princip rotačního enkodéru

Vypracuj pracovní list.

Cíle aktivity

- žák vyplní tabulku, která vysvětluje princip rotačního enkodéru,
- žák vyhodnotí průběh přepínání hodnot při otáčení hřídelí.

Metodika pro pedagoga

Princip rotačního enkodéru je poměrně složitý na pochopení. V pracovním listu žáci vyplní tabulku podle zadaných příkazů. Pro snazší pochopení doporučujeme vytisknout a sestavit pomůcku enkodéru (Obrázek 17 - Rotační enkodér, autor Jiří Noska), kde jsou při otáčení

vidět hodnoty a zároveň směr otáčení. Tato praktická pomůcka může velmi snadno představit princip fungování rotačního enkodéru.

Pro vyhodnocení směru otáčení rotačního enkodéru zjišťujeme logické hodnoty na pinech D6 a D7. Jeden z nich slouží k detekci změny pohybu (pin D6), druhý (pin D7) k určení jeho směru.

Nejdříve uložíme hodnotu pinu D6 do proměnné. Tuto hodnotu porovnáme s aktuální hodnotou pinu D6 a v případě změny uložíme novou hodnotu pinu D6 do proměnné. Dále testujeme, zda je hodnota pinu 6 rovna logické 1. Pokud ano, vyhodnocujeme směr pohybu pomocí logické hodnoty pinu D7. V případě, že je rovna logické 0, hřídel se otáčí po směru hodinových ručiček, jinak proti směru hodinových ručiček.

Protože si je fungování rotačního enkodéru náročné představit, je důležité mít sestavenou pomůcku a názorně si ukazovat jednotlivé kroky. Žáci na pomůcce nastaví výchozí stav, tady v okénkách budou vidět hodnoty 0. Poté začnou otáčet kolečkem hodnot a v okénkách se budou měnit hodnoty pinů. V tento moment budou žáci v tabulce (Tabulka 2 - Vyplněná tabulka určení směru pohybu rotačního enkodéru) vyhodnocovat jednotlivé stavy:

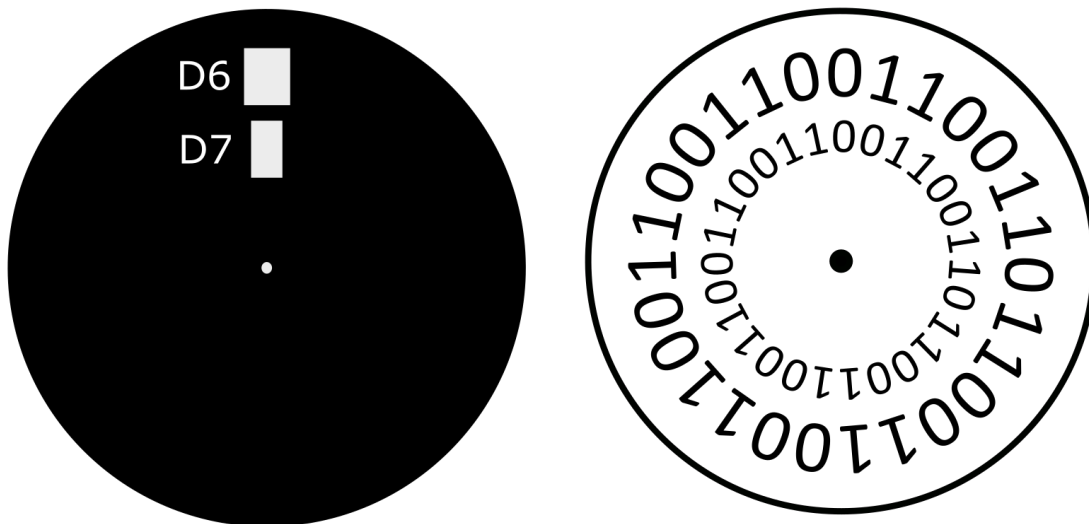
- stav změna – označí v případě změny hodnoty pinu 6 z log. 0 na log. 1 nebo naopak,
- stav test je log. 1 – označí v případě, že hodnota pinu 6 je rovna log. 1,
- stav hodnota směru – napíše logickou hodnotu pinu D7,
- stav směr – určí směr otáčení.

Tabulka 2 - Vyplněná tabulka určení směru pohybu rotačního enkodéru

<i>změna</i>		•		•		•		•		•		•		•		•		•
<i>test je log. 1</i>		•				•				•				•				•
<i>hodnota směru</i>		0				0				0				0				0
<i>směr</i>		→				→				→				→				→
<i>pin D6</i>	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
<i>pin D7</i>	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0

V dalším úkolu této aktivity musí žáci rozpoznat směr otáčení, jelikož v dosavadních tabulkách se jednalo o stálé otáčení jedním směrem a nyní bude směr otáčení nahodilý.

Poslední úloha je skupinová. Žáci vytvoří dvojice. Každý z této dvojice vypracuje tabulku s nahodilým otáčením rotačního enkodéru (využije pomůcku) a následně jí předají k vypracování spolužákovi. Po vypracování provedou vzájemnou kontrolu správnosti.



Obrázek 17 - Rotační enkodér, autor Jiří Noska

Aktivita 7.2 Rotační enkodér

Vytiskni na sériový monitor číslo, které se bude zvětšovat při otáčení hřídele rotačního enkodéru po směru hodinových ručiček a zmenšovat při otáčení hřídele proti směru hodinových ručiček.

Cíle aktivity

- žák prezentuje pochopení principu rotačního enkodéru,
- žák vytvoří program používající rotační enkodér,
- žák otestuje, zda program funguje dle zadání.

Metodika pro pedagoga

V případě tvoření prvního programu používajícího rotační enkodér je možné, aby pedagog asistoval při vytváření, to znamená, vytvářel podmínky a podával informace tak, aby postupně žáci dokázali splnit zadání. Informace však nesmí být doslovným návodem, ale pouze informační oporou.

Prvním krokem bude definování pinů, kterými jsou pin D6 a D7. Dále žáci vytvoří proměnnou, kam budou ukládat číslo, které se bude měnit při pohybu rotačním enkodérem

a nastaví mu výchozí hodnotu 0. Další proměnnou, kterou je nutné vytvořit je proměnná, kam se bude ukládat poslední stav pinu D6 a nastaví mu výchozí hodnotu pinu D6.

V dalším kroku ve funkci `setup` nastaví jednotlivé piny. U tohoto nastavení je odlišnost v podobě využívání pinu jako vstupních PULLUP pinů. Žáci tedy musí u nastavení použít atribut `INPUT_PULLUP`.

Ve funkci `loop` žáci vytvoří podmínku, která bude hlídat, zda došlo ke změně logické hodnoty na pinu D6 pomocí nerovnosti aktuální hodnoty pinu D6 a hodnoty uložené v proměnné. Tato nerovnost se zapisuje znakem vykřičník a rovnítkem. V případě, že se neshodují, pak se do proměnné uloží aktuální hodnota. Dále se zjišťuje, zda je aktuální hodnota pinu D6 logickou hodnotou 1. V kladném případě budou žáci pomocí podmínek zjišťovat logickou hodnotu pinu D7. V případě, že je zde hodnota 0, dojde k navýšení proměnné s číslem, v opačném případě dojde k ponížení této proměnné. Posledním krokem je samotný výpis této proměnné.

Pro zdatnější programátory pedagog upraví podmínky zadání, například:

číslo, které se má vypisovat musí být pouze kladné v rozmezí 0 až 50.

4.9 7 SEGMENTOVÝ DISPLEJ

Použití displejů je navrženo na jednu vyučovací hodinu. Prekonceptem je znalost umístění displejů na desce TinyLabu.

Cíle této hodiny jsou:

- žák připojí knihovnu k programu,
- žák využije displeje k zobrazení hodnot.

V aktivizační části hodiny žáci odpoví na otázku:

Kde jste se s těmito displeji setkali?

Ovládání těchto displejů by vyžadovalo použití neúměrného množství pinů. Díky knihovnám ale dojde ke značnému zjednodušení práce s těmito komponenty.

Pomocný materiál ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita08/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Aktivita 8.1 7segmentový displej

Vytvoř program z příkazu v zadání. Vypiš na 7 segmentový displej aktuální rok.

Cíle aktivity

- žák vytvoří program se zobrazením číslic na displeji,
- žák použije knihovnu.

Metodika pro pedagoga

Žáci v této aktivitě použijí předepsané příkazy, které poskládají tak, aby vytvořily funkční program. V případě chybějící knihovny vypíše program chybové hlášení. Zde pak pedagog nechá žáky vyhledat v programu knihovny, které jsou nainstalovány. Pro 7segmentový displej je nutné použít knihovnu LedControl.h. Pokud mají žáci složením příkazů program zkompileován a funkční, tak mohou přejít k dalšímu zadání. Na displeji zobrazí aktuální rok. Pedagog záměrně neuvádí, kde a co má žák změnit. Metodou explorační žák zkoumá, který příkaz vypisuje číslice na displej a jaké mu přiřadit hodnoty.

Aktivita 8.2 Výpis na 7segmentový displej

Vypisuj hodnotu potenciometru na 7 segmentový displej.

Cíle aktivity

- žák navrhne program pro výpis hodnoty na potenciometr,
- žák ověří správnost programu změnou hodnoty potenciometru.

Metodika pro pedagoga

Vyučovací pomůcka je k dispozici na úložišti. Tuto aktivitu lze začít otázkou:

Jak vypíšete číslo větší než 10?

Správná odpověď je díky využití funkce modulo. K vysvětlení pedagog využije vyučovací pomůcky uložené v úložišti. Znalost matematické funkce modulo je jedním z prekonceptů pro tuto hodinu.

Aktivita 8.3 Kuchyňská minutka

Vytvoř kuchyňskou minutku. Na 7segmentovém displeji zobraz odečet. Pomocí tlačítek nastav požadovaný čas – maximálně však 60 minut. Při posledních 5 sekundách dojde k zvukovému signálu.

Cíle aktivity

- žák využije předchozí znalosti k tvorbě komplexního programu,
- žák využije 7segmentový displej,
- žák otestuje funkčnost programu.

Metodika pro pedagoga

Tato aktivita lze splnit několika způsoby. Důležitý je konečný výsledek a argumentace, proč žák využil zrovna tento postup. Program, který je uložený v úložišti, nepoužívá pokročilejší programovací vztahy. Nejsou zde například vlastní funkce, které program a přehlednost programu zjednoduší.

Jako první musí žáci využít knihovny pro ovládání 7segmentového displeje. Jedná se o knihovny *LedControl*.

Dále musí definovat proměnné a konstanty. V prvotním nastavení, tedy funkci *setup*, provedou nastavení potřebných pinů. Poté provedou nastavení displeje pomocí příkazů *shutdown*, *setIntensity* a *clearDisplay*.

Funkce *loop* bude obsahovat kompletní program. V rámci tohoto programu mohou využít znalost principu rotačního enkodéru, který bude přidávat nebo ubírat zadaný čas. Žáci musí vytvořit podmínky pro nastavení maximální hodnoty, tedy 60 minut. Dále mohou využít znalost tlačítek a na jedno z nich umístit resetovací tlačítko. V případě startu minutky dojde k odpočítávání času a v posledních pěti sekundách i k aktivaci zvukového znamení.

Při vypisování na segment samozřejmě musí docházet k přepočtu času, který je datového typu *int* na nám známý zápis. Tento výpis může vypadat následovně:

```
lc.setDigit (0, 0, (cas / 60) % 100 / 10, false);
lc.setDigit (0, 1, (cas / 60) % 100 % 10, true);
lc.setDigit (0, 2, cas % 60 / 10, false);
lc.setDigit (0, 3, cas % 60 % 10, false);
```

Na první pozici se nachází desítky minut. Hodnotu proměnné *cas*, kam si ukládají čas zadany uživatelem, musí tedy vydělit 60, poté provedou modulo 100 a celé vydělí hodnotou 10. Takto postupují analogicky v rámci celého zobrazování proměnné času.

4.10 LCD DISPLEJ, FUNKCE, CYKLY

Jedná se o základní principy programování pro optimalizaci programů. Do této doby žáci pracovali s jednotlivými komponenty TinyLabu. Téma je navrženo na jednu vyučovací hodinu.

Cíle této hodiny jsou:

- žák vypíše hodnoty na LCD displeji,
- žák vytvoří funkce,
- žák navrhne program využívající cykly.

Pomocný materiál ke stažení:

<https://github.com/JiriNoska/TinyLab/tree/main/aktivita09/>

nebo rozcestník

<https://jirinoska.github.io/tinylab/>

Aktivita 9.1 LCD displej

Doplň program. Nastav, aby se vypsal na horní řádek tvé jméno a pohybovalo se doprava. Na spodním řádku bude tvé příjmení.

Cíle aktivity

- žák doplní potřebné údaje do programu,
- žák přepíše program dle zadání,
- žák vyhledá a opraví případné nedostatky v programu,
- žák otestuje funkčnost programu.

Metodika pro pedagoga

Žák má v pracovním listu zadany program, který není zcela úplný. Všechny potřebné prekoncepty by žák měl mít. Chybí nastavení LCD displeje. Hodnoty, které má vložit do

programu již žák zná z první aktivity (Aktivita 1.1). Dále v programu musí pozměnit text, jeden doplnit a změnit hodnoty podsvícení displeje.

Aktivita 9.2 Nefunkční program

Uprav program využívající funkce tak, aby fungoval. Vyplň pracovní list.

Cíle aktivity

- žák analyzuje funkce programu,
- žák opraví chyby,
- žák otestuje program.

Metodika pro pedagoga

Tento program má za úkol stisku tlačítka S1 vygenerovat náhodný příklad z malé násobilky a po pěti sekundách zobrazit výsledek.

V programu jsou vytvořeny záměrné chyby ve funkcích tak, aby byl žák nucen je k funkčnosti programu opravit. Tato aktivita má za cíl prověřit žákovu znalost funkcí, které jsou jedním z prekonceptů této a dalších hodin.

V programu jsou chyby na několika místech. Chybí zde deklarace proměnné `cinitell` datového typu `integer`, která je používána v dalších funkcích. Ve funkci `loop` chybí při volání funkce `soucinCisel` jedna z hodnot. Funkce `soucinCisel` rovněž nevrací hodnotu `vysledek`. Ve funkci `vypis` poté chybí jeden z parametrů a zároveň zde chybí uvedení datových typů. Po této opravě žáci kompilací a spuštěním programu otestují správnost jejich oprav.

V pracovním listu jsou položeny tři otázky prověřující základní znalosti k funkcím. Žák musí odpovědět, že pomocí příkazu `return` vrací návratovou hodnotu, že funkce `void` nemusí mít návratovou hodnotu. Poslední aktivita v pracovním listu je prověření názvosloví.

Aktivita 9.3 Malá násobilka II

Vypiš v sériovém monitoru pomocí cyklů malou násobilku od 1 do 10.

Cíle aktivity

- žák vytvoří cykly pro výpis malé násobilky,

- žák výpisem otestuje funkčnost programu.

Metodika pro pedagoga

Tento program bude obsahovat dva cykly, kdy jeden bude vnořený do druhého. Jsou zde dva možné kritické momenty. Prvním je, že žáci nevnoří tyto cykly do sebe nebo nevytvoří dvě rozdílné proměnné. Druhým kritickým momentem je rychlost načítání. V případě umístění cyklů do funkce `setup`, pak program splní vše v této funkci ještě předtím, než je aktivován sériový monitor. V tomto případě nedojde k výpisu. Toto lze eliminovat použitím příkazu `while`.

Toto zadání lze modifikovat pro horší žáky úpravou zadání, kdy dojde k výpisu pouze malé násobilky jednoho čísla. V tomto případě by byl eliminován problém s vnořením cyklů.

Aktivita 9.4 Faktoriál

Vytvoř program pro výpočet faktoriálu čísla od 1 do 12. Výsledek zobraz na displeji.

Cíle aktivity

- žák cyklus pro výpočet faktoriálu daného čísla,
- žáka správně přiřadí datové typy,
- žák otestuje funkčnost programu.

Metodika pro pedagoga

Tento program má za cíl, aby žák využil znalosti cyklů pro výpočet faktoriálu a znalost datových typů. Pro přehlednost je lepší využít funkce, ale není to podmínkou zadání.

Kritickým momentem v této hodině může být práce s cykly. V případě, že vstupuje proměnná do cyklu pro výpočet faktoriálu bez hodnoty, je jí automaticky přiřazena hodnota 0. Díky tomu pak vychází nulový výsledek. Je tedy důležité, aby byl počáteční vstup rovný hodnotě 1.

4.11 KOMPLEXNÍ ÚLOHY

Aktivita 10.1 Převodník z desítkové soustavy na binární a hexadecimální

Zadání:

Vytvoř převodník čísla z desítkové soustavy do dvojkové a šestnáctkové. Číslo zadávej pomocí tlačítek a zobrazuj na 7segmentovém displeji (maximální číslo bude 9999). Na LCD displeji zobraz dvojkovou a šestnáctkovou hodnotu zadaného čísla.

Cíl:

Žák využije předchozí znalosti a vytvoří komplexní úlohu.

Metodika pro pedagogy:

Tato úloha propojuje předchozí znalosti. Jedinou změnou a novým poznatkem je konverze čísla. Tato konverze probíhá však díky funkcím Arduina bez komplikací. Pro pokročilejší lze převod mezi desítkovou soustavou a dvojkovou nebo šestnáctkovou ručně vytvořit.

Samotným počátkem této úlohy je opět inicializace pomocí knihoven. Knihovnamy žáci ovládají 7segmentový displej a LCD displej. Ve funkci *setup*, což je prvotní nastavení, provedou inicializaci displejů a používaných pinů. V rámci této funkce volají funkci *vypis* s atributem *binPrevod* a *hexPrevod*. To jsou i proměnné typu *String*, které si vytvoří a kterým přiřadí hodnotu 0. Tato funkce vypisuje hodnoty na displeje. V rámci přehlednosti lze aplikovat i zarovnání vpravo, které vnoří do funkce *lcd.setCursor*. Tato funkce tedy bude vypadat následovně:

```
void vypis(String binPrevod, String hexPrevod) {
  lcd.clear();
  for (int i = 0; i < 4; i++) {
    lc.setDigit (0, i, number[i], false);
  }
  int delkaBin = binPrevod.length();
  int delkaHex = hexPrevod.length();
  lcd.setCursor(16 - delkaBin, 0);
  lcd.print (binPrevod);
  lcd.setCursor(16 - delkaHex, 1);
  lcd.print (hexPrevod);
}
```

Z funkce *loop* volají další funkci *stisk*. Také do proměnné *aktualniCas* typu *unsigned long* vloží hodnotu *millis*. Ta slouží k zamezení opakovaného nechtěného stisku tlačítka. Funkce *stisk* tedy zjišťuje, zda je stisknuto jedno z tlačítek. V případě stisku do proměnné *casStisku* vloží hodnotu *millis* a pokud je rozdíl mezi stiskem a aktuálním časem větší než 200 milisekund, pak se zjišťuje, které tlačítko je stisknuto. Při stisku se v proměnné *number*, která je polem, na dané pozici přičte hodnota 1 nebo v případě, že je větší než 9, pak se hodnota vynuluje. Při stisku je zároveň volána funkce *prevod*.

Funkce `prevod` převede hodnotu z pole `number` do proměnné `celeCislo`. Do proměnné `binPrevod` a `hexPrevod`, kterou použili již na začátku, uloží hodnotu `celeCislo` s daným převodem. Poté volají funkci `vypis`, která zajistí vypsání hodnoty na displejích. Celá funkce tedy může vypadat takto:

```
void prevod () {
    int celeCislo = number[0] * 1000 + number[1] * 100 + number[2] * 10 +
        number[3];
    String binPrevod = String (celeCislo, BIN);
    String hexPrevod = String (celeCislo, HEX);
    vypis (binPrevod, hexPrevod);
}
```

Aktivita 11.1 Morseova abeceda

Zadání:

Vytvoř vysílač Morseovy abecedy. Pomocí enkodéru zadávej jednotlivá písmena tvořící slovo. Toto bude následně po stisku jednoho tlačítka vysláno jako zvukový a obrazový signál (zvuk pomocí bzučáku, vizuální vysílání pomocí LED diody). Maximální délka slova bude 16 pozic (velikost řádku displeje).

Cíl:

Žák pro vytvoření úlohy využije předchozí znalosti. V rámci tohoto projektu je nutno pracovat s vícerozměrným polem.

Metodika pro pedagogy:

Tato úloha kombinuje předešlé úkoly. Žák se během tohoto úkolu naučí využívat pole, a to jak jednoduché, tak i vícerozměrné.

Obtížnost tohoto úkolu lze zvýšit začleněním čísel do Morseovy abecedy a odstraněním maximální délky slova.

V tomto programu využijí žáci pouze jednu knihovnu, která bude ovládat LCD displej. Ten si nastaví stejně, jako v předchozích úlohách. V rámci proměnných kromě standartních vytvoří proměnné typu `integer` definující délku čárky a tečky a také jednotlivých mezer. Samotná Morseova abeceda je pak vytvořena pomocí vícerozměrného pole, kdy řádky určují jednotlivá písmena a sloupce pak jednotlivé znaky. Velikost vícerozměrného pole tedy bude 26 řádek, což znázorňuje počet písmen v abecedě (bez diakritiky a bez písmene ch), a 4 sloupce. Počet sloupců je maximální počet znaků v písmenu Morseovy abecedy.

V případě ztížení úlohy a začlenění čísel bychom museli zvýšit počet sloupců na pět. Výše uvedené nastavení a vytvoření proměnných tedy může vypadat takto:

```
#include <LiquidTWI2.h>

int charNr = 97;
unsigned long aktualniCas;
unsigned long casStisku;
byte posledniPozice = digitalRead (6);
int text[16] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int pozice = 0;

int tecka = 200;
int carka = 500;
int mezera = 100;
int mezeraMeziSlovy = 1000;

int morse [26][4] = {
  {tecka, carka, 0, 0},           //a
  {carka, tecka, tecka, tecka}, //b
  {carka, tecka, carka, tecka}, //c
  {carka, tecka, tecka, 0},      //d
  {tecka, 0, 0, 0},             //e
  {tecka, tecka, carka, tecka}, //f
  {carka, carka, tecka, 0},      //g
  {tecka, tecka, tecka, tecka}, //h
  {tecka, tecka, 0, 0},         //i
  {tecka, carka, carka, carka}, //j
  {carka, tecka, carka, 0},     //k
  {tecka, carka, tecka, tecka}, //l
  {carka, carka, 0, 0},         //m
  {carka, tecka, 0, 0},         //n
  {carka, carka, carka, 0},     //o
  {tecka, carka, carka, tecka}, //p
  {carka, carka, tecka, carka}, //q
  {tecka, carka, tecka, 0},     //r
  {tecka, tecka, tecka, 0},     //s
  {carka, 0, 0, 0},            //t
  {tecka, tecka, carka, 0},     //u
  {tecka, tecka, tecka, carka}, //v
  {tecka, carka, carka, 0},     //w
  {carka, tecka, tecka, carka}, //x
  {carka, tecka, carka, carka}, //y
  {carka, carka, tecka, tecka}, //z
};

LiquidTWI2 lcd(0x20);
```

Ve funkci `setup` nastaví jednotlivé piny a také displej. Piny A5, D6, D7, D8 a D9 budou sloužit jako piny vstupní, pin D13 pak jako výstupní (zobrazení abecedy na LED diodě) a pin A1, které patří bzučáku, také jako pin výstupní.

Funkce `loop` může obsahovat pouze nastavení aktuálního času díky funkci `millis` a poté volání funkce `enkoder` a `stisk`.

Funkce `enkoder` slouží ke snímání pohybu rotačního enkodéru, kterým žáci zadávají jednotlivá písmena. Pro zadávání písmen použijí načítání proměnné typu `integer` od

hodnoty 97 do hodnoty 122. Tyto hodnoty v kombinaci s příkazem `char` umožní znázornění jednotlivých písmen. Z této funkce dále ještě volají funkci `vypisAktPismeno` a `vypis`.

```
void enkoder () {
    byte stav6 = digitalRead(6);
    if (stav6 != posledniPozice) {
        byte stav7 = digitalRead(7);
        posledniPozice = stav6;
        if (stav6 == HIGH) {
            if (stav7 == LOW) {
                if (charNr < 122) {
                    charNr++;
                } else {
                    charNr = 97;
                }
            }
        } else {
            if (charNr > 97) {
                charNr--;
            } else {
                charNr = 122;
            }
        }
    }
}
vypisAktPismeno ();
vypis ();
}
```

Funkce `stisk` hlídá stisknutí tlačítek. Tlačítko na pinu D9 slouží k resetu zadaného slova, tlačítko na pinu čD8 pak zahajuje zobrazení slova v Morseově abecedě. Tlačítko na hřídeli rotačního enkodéru, které je na pinu A5, potvrzuje zadané písmeno. Vstup je vždy ošetřen porovnáním aktuálního času a času posledního stisku, od kterého musí uplynout minimálně 200 milisekund. Při stisku hřídele enkodéru dojde k zapsání aktuálního písmene do pole `text` na danou pozici, která se po zapsání inkrementuje o hodnotu 1. Stisk tlačítka na pinu devět vyresetuje pole `text`, proměnnou `pozice` a vymaže i LCD displej. Tato část tedy může vypadat následovně:

```
void stisk() {
    if (aktualniCas - casStisku > 200) {
        if (analogRead(5) > 80 && analogRead(5) < 100) {
            text[pozice] = charNr;
            if (pozice < 15) {
                pozice++;
            } else {
                pozice = 0;
            }
        }
    }
    if (digitalRead(9) == LOW) {
        memset(text, "", sizeof(text));
        pozice = 0;
        lcd.clear();
    }
    if (digitalRead(8) == LOW) {
        poznejPismeno ();
    }
}
```

```

    casStisku = millis();
  }
}

```

Jednotlivé výpisy na displej je proveden díky funkcím *vypis* a *vypisAktPismeno*. Tyto funkce vždy v cyklu provedou výpis z pole text převedené do jednotlivých písmen pomocí funkce *char*. Dále se zde prověřuje velikost hodnoty, která musí být mezi hodnotou 97 a 122, což je mezi písmeny *a* a *z*. Tyto funkce tedy budou vypadat následovně:

```

void vypisAktPismeno() {
  lcd.setCursor (pozice, 0);
  lcd.print (char(charNr));
}

void vypis() {
  for (int i = 0; i < 16; i++) {
    lcd.setCursor (i, 0);
    if (text[i] > 96 && text[i] < 123 ) {
      lcd.print (char(text[i]));
    }
  }
}

```

Po stisku tlačítka na pinu 8 dochází k volání funkce *poznejPismeno*. Ta provede výpis pole *text* na displej a také volá funkci *ton* s hodnotou písmene, které má zaznít. To provedeme odečtením hodnoty 97 od proměnné v poli *text*. Takto tedy bude vypadat tato funkce:

```

void poznejPismeno () {
  lcd.clear();
  vypis();
  for (int i = 0; i < 16; i++) {
    if ((text[i]) > 96 && (text[i]) < 123 ) {
      lcd.setCursor (i, 1);
      lcd.print("*");
      ton (text[i] - 97);
    }
  }
  lcd.clear();
}

```

Posledním úkonem pak bude samotné přehrání zvuku a zobrazení na LED diodě. Toto provádí funkce *ton*. Ta bude obsahovat cyklus *for*, kdy bude načítat z vícerozměrného pole *morse* hodnoty, jestli se jedná o tečku nebo čárku. Jednotlivý druh je prezentován hodnotou integer, která znamená dobu tónu.

ZÁVĚR

V rámci mé diplomové práce jsem se zaměřil na využití výukového kitu TinyLab při vzdělávání budoucích pedagogů. Klasické rozdělení práce na dva bloky, teoretický a praktický napomáhá orientaci v tématu. V rámci teoretické části jsem ve zkratce představil jednotlivé komponenty a jejich funkci. Tato teoretická část nemá být teoretickým základem, ale pouze opěrným materiálem pro pochopení principu daných komponent. Právě proto jsem se snažil vybírat pouze podstatné informace tak, aby byly snadno pochopitelné a zapamatovatelné.

Hlavním cílem této práce bylo předložení komplexní sady úloh, které by pomohly budoucím pedagogům pochopit základ programování v jazyce wiring.

V praktické části jsem předložil několik úloh, které v časové dotaci kopírují výuku programování na Katedře výpočetní a didaktické techniky při Západočeské univerzitě v Plzni. Tyto úlohy mají v rámci složitosti vzestupnou tendenci a navazují na sebe tak, aby i méně zkušený pedagog byl v závěru výuky schopen naprogramovat složitější a komplexnější program. V praktické části jsem nepoužíval metody frontální výuky, ale snažil jsem se v rámci úloh podporovat explorační, která má za následek hlubší pochopení struktur. Poslední čtyři hodiny jsou věnovány komplexnějším úlohám, které mohou sloužit i jako zpětná vazba pro pedagoga, které výukové cíle nebyly naplněny.

Mezi dílčí cíle patřila ukázka možnosti využívat TinyLab v rámci výuky programování, a to nejen v přípravě budoucích učitelů, ale díky zvolené obtížnosti použít úlohy i pro výuku programování ve vyšších ročnících na základních školách.

RESUMÉ

Diplomová práce je zaměřena na využití výukového kitu TinyLab v přípravě budoucích učitelů. Práce je rozdělena do čtyř kapitol, kdy prvních tři jsou teoretického zaměření a poslední již předkládá sadu úloh, které demonstrují využití TinyLabu.

V teoretické části diplomové práce je zkráceně představena historie výukového kitu TinyLab, jeho varianty. Ve druhé, nejobsáhlejší teoretické kapitole, jsou představeny jednotlivé komponenty výukového kitu TinyLab Maker Kit, což je nezákladnější verze TinyLabu. Třetí teoretická část obsahuje představení základních dostupných nástrojů pro programování TinyLabu.

Praktická část představuje vytvořený kurz v platformě Moodle. Obsahuje 11 úloh, které jsou koncipovány pro 13 vyučovacích hodin. Tyto úlohy nabývají postupně na složitosti, od základního představení TinyLabu po komplexní úlohy.

Klíčová slova

TinyLab

Arduino

wiring

Programování

Arduino IDE

RESUMÉ

The thesis focuses on the use of the TinyLab educational kit in the preparation of future teachers. The thesis is divided into four chapters, where the first three are theoretical and the last one presents a set of tasks that demonstrate the use of TinyLab.

The theoretical part of the thesis briefly presents the history of the TinyLab teaching kit, its variants. In the second, the most comprehensive theoretical chapter, the individual components of the TinyLab Maker Kit, which is a more basic version of TinyLab, are introduced. The third theoretical section introduces the basic tools available for programming TinyLab.

The practical part presents the course created in the Moodle platform. It contains 11 exercises designed for 13 lessons. These assignments increase in complexity, from a basic introduction to TinyLab to complex assignments.

Keywords

TinyLab

Arduino

wiring

programming

Arduino IDE

SEZNAM LITERATURY

- Analog devices. 2021.** TMP36. *Analog devices*. [Online] 2021. <http://analog.com/en/products/tmp36.html>.
- Arduino. 2022.** Arduino Cloud. *Arduino Cloud*. [Online] Arduino, 2022. store.arduino.cc/digital/create.
- Arduino. 2021.** Arduino Leonardo. *Arduino*. [Online] 2021. http://arduino.cc/en/Main/Arduino_BoardLeonardo.
- BURGER, Ivan a HUDEC, Lubomír. 1989.** *Elektronické prvky*. Bratislava : Alfa, 1989. 80-05-00120-7.
- DOLEČEK, Jaroslav. 2007.** *Moderní učebnice elektroniky 1. díl*. Praha : Nakladatelství BEN - technická literatura, 2007. 80-7300-146-2.
- FAHLE, Eberhard. 2015.** The Arduino library code. *LedControl*. [Online] 2015. <http://wayoda.github.io/LedControl/pages/software>.
- KESL, Jan. 2003.** *Elektronika I*. Praha : BEN - technická literatura, 2003. 80-7300-074-1.
- MALINA, Václav. 2001.** *Poznáváme elektroniku VI. .* České Budějovice : Kopp, 2001. 80-7232-146-3.
- MANĚNA, Václav. 2015.** *Moderně s Moodle: Jak využít e-learning ve svůj prospěch*. Praha : CZ.NIC, 2015. 978-80-905802-7-5.
- MAXFIELD, Max. 2020.** Speeding Prototyping and Development with the Industrial Pi (I-Pi) SMARC Kit. *Embedded Computing Design*. [Online] 02. 20 2020. [Citace: 25. 01 2022.] <https://www.embeddedcomputing.com/technology/open-source/developmnet-kits/speeding-prototyping-and-development-with-the-industrial-pi-i-pi-smarc-kit>.
- Microchip Technology Inc. 2021.** ATmega32U4. *Microchip*. [Online] 2021. <http://microchip.com/en-us/product/ATmega32U4>.
- SELECKÝ, Matuš. 2016.** *Arduino: Uživatelská příručka*. [překl.] Martin HERODEK. Brno : Computer Press, 2016. 978-80-251-4849-5.
- SCHMACHTL CZ. 2005.** Rotační enkodéry ELTRA. *ELEKTRO: časopis pro elektrotechniku*. [Online] FCC Public, 2005. <http://www.odbornecasopisy.cz/elektro/casopis/tema/rotacni-enkodery-eltra--13671>.
- Sixfab Inc. 2015.** TinyLab: Prototype easier than ever. *IndieGoGo*. [Online] IndieGoGo, 2015. <https://indiegogo.com/projects/tinylab-prototype-easier-than-ever#/>.
- Texas Instruments. 2017.** Texas Instruments. *LM35 Precision Centigrade Temperature Sensors*. [Online] Texas Instruments, 2017. <http://ti.com/lit/ds/symlink/lm35.pdf>.
- TinyLab. 2022.** tinylab. *tinylab*. [Online] sixfab, 2022. <http://tinylab.cc>.
- TKOTZ, Klaus. 2001.** *Příručka pro elektrotechniku*. [překl.] Jiří HANDLÍŘ. Praha : Europa-Sabotáles, 2001. 80-86706-00-1.
- VODA, Zbyšek a tým HW Kitchen. 2018.** *Průvodce světem Arduina*. Bučovice : Nakladatelství Martin Stříž, 2018.
- VODA, Zbyšek. 2017.** TinyLab: piezo bzučák. *Bastlírna HW Kitchen*. [Online] HW Kitchen, 2017. <https://bastlirna.hwkitchen.cz/tinylab-piezo-bzucak/>.

VODA, Zbyšek. 2018. TinyLab: LCD displej. *Bastlírna HW Kitchen*. [Online] 05. 12 2018. <https://bastlirna.hwkitchen.cz/tinylab-lcd-displej/>.

VODA, Zbyšek. 2018. TinyLab: relé. *Bastlírna HW Kitchen*. [Online] HW Kitchen, 2018. <https://bastlirna.hwkitchen.cz/tinylab-rele/>.

VODA, Zbyšek. 2018. TinyLab: sedmisegmentový displej. *Bastlírna HW Kitchen*. [Online] HW Kitchen, 2018. <https://bastlirna.hwkitchen.cz/tinylab-sedmisegmentovy-displej/>.

VOJÁČEK, Antonín. 2017. Rotační enkodéry pro řízení pohonů a pohybu. *automatizace.hw.cz*. [Online] 30. 04 2017. <https://automatizace.hw.cz/rotacni-enkodery-pro-rizeni-pohonu-a-pohybu.html>.

SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ

Obrázek 1 - TinyLab Maker Kit, zdroj: https://www.tinylab.cc/	9
Obrázek 2 - TinyLab IoT Kit, zdroj: https://www.tinylab.cc/	9
Obrázek 3 - TinyLab Exclusive Kit, zdroj: https://www.tinylab.cc/	10
Obrázek 4 - Zapojení jednotlivých komponentů, zdroj: https://www.tinylab.cc/wp-content/uploads/2016/05/tinylab-pinout-diagram2.png	11
Obrázek 5 - Schéma zapojení relé, zdroj: https://ncd.io/relay-logic/	14
Obrázek 6 - schéma zapojení tlačítek S3, S4, zdroj: https://www.tinylab.cc/wp-content/uploads/2016/05/tinylab-pinout-diagram2.png	15
Obrázek 7 - Střída, autor: Jiří Noska	16
Obrázek 8 - ukázka principu snímání pohybu, autor: Jiří Noska	17
Obrázek 9 - zapojení 7segmentového displeje se čtyřmi členy, zdroj: https://z9q5c2a5.stackpathcdn.com/wp-content/uploads/2018/08/SH5461AS.pdf	17
Obrázek 10 - Arduino, verze 1.8.16	20
Obrázek 11 - Arduino IDE. verze 2.0.0-rc3	20
Obrázek 12 - Ukázka z mBlock - blokového programování Arduina, zdroj: https://ide.mblock.cc/	21
Obrázek 13 - Ukázka práce v Tinkercad, zdroj: https://www.tinkercad.com/things/3wm1LExCrK7-funky-jaban-albar/editel?tenant=circuits	22
Obrázek 14 - Základní deska kitu TinyLab, autor: Jiří Noska	27
Obrázek 15 - Semafor u přejezdu – pomůcka, autor: Jiří Noska	30
Obrázek 16 - Tabulka výpočtu minimální hodnoty pro sepnutí relé, autor: Jiří Noska	40
Obrázek 17 - Rotační enkodér, autor Jiří Noska	47

PŘÍLOHY

Tabulka 3 - Tabulka aktivit

hodina	výuková hodina	využití komponent							využití příkazy							nové poznatky										
		potenciometr	piezoelektrický měnič	fotorezistor	teplotní čidlo	relé	tlačítka	led diody	rotační enkodér	7segmentový displej	LCD displej	analogRead	analogWrite	delay	digitalRead	digitalWrite	map	pinMode	Serial	cykly	datové typy	funkce	konstanty	logické hodnoty	podmínky	proměnné
1	Seznámení se s TinyLabem						•								•		•			•		•				
2	LED						•						•		•		•			•		•	•	•	•	•
3	Potenciometr	•					•				•						•	•		•	•	•	•	•	•	•
4	Tlačítka					•	•				•		•				•	•		•	•	•	•	•	•	
5	Relé, piezoelektrický měnič	•	•		•	•						•	•	•	•		•	•		•	•	•	•	•	•	•
6	Fotorezistor, teplotní čidlo			•	•	•					•						•	•		•	•	•	•	•	•	•
7	Rotační enkodér						•						•				•	•		•	•	•	•	•	•	•
8	7 segmentový displej		•			•	•	•			•			•	•		•	•		•	•	•	•	•	•	•
9	LCD displej, cykly, funkce	•							•		•					•	•	•	•	•	•	•	•	•	•	•
10	Převodník z desítkové soustavy					•	•	•	•							•	•	•		•	•	•	•	•	•	•
11	Morseova abeceda		•			•	•	•	•	•			•	•	•		•	•		•	•	•	•	•	•	•