

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Ústav fyziky

Možnosti komunikace mezi Raspberry Pi 3 a modulů Arduino – soubor laboratorních úloh

Autor: Tomáš Beran, DiS.
Vedoucí práce: Ing. Michal Šerý, PhD.

České Budějovice 2017

Beran, T., 2017: Možnosti komunikace mezi Raspberry Pi 3 a modulů Arduino – soubor laboratorních úloh. [Possibilities of communication between Raspberry Pi 3 and Arduino modules – laboratory tasks. Bc. Thesis, in Czech.] – 78 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Bakalářská práce pojednává především o možnostech připojení a komunikaci mezi zařízeními: Raspberry Pi 3 a Arduino Pro Mini. Bylo vypracováno celkem 11 pracovních úloh pro studenty vysoké školy. Základem úspěšného použití v praxi je pochopení a praktické poznání rozdílů mezi moduly Raspberry Pi a Arduino.

Klíčová slova

Arduino; Raspberry Pi; Python; Geany; Wiring; Processing; Linux; ATmega328; OneWire; SPI; I²C; USB; Komunikace;

Abstract

Title: Possibilities of communication between Raspberry Pi 3 and Arduino modules – laboratory tasks

Summary: Possibilities of communication between Raspberry Pi 3 and Arduino modules – laboratory tasks

Summary: Bachelor thesis deals primarily with connectivity and communication options between two devices: Raspberry Pi 3 and Arduino Pro Mini. There were elaborated eleven tasks for university students. Understanding and practical cognition of differences between Raspberry Pi and Arduino modules is elemental for its successful use in praxis.

Keywords:

Arduino; Raspberry Pi; Python; Geany; Wiring; Processing; Linux; ATmega328; OneWire; SPI; I²C; IoT; USB; UART; Communication;

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích, 11. prosince 2017

.....
Tomáš Beran

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Michalu Šerému, Ph.D. za pomoc při výběru tohoto zadání BP a hlavně za konzultace plné konstruktivních rad v průběhu samotného vypracování bakalářské práce.

Obsah

1	Úvod.....	7
	Typografická konvence.....	7
2	Cíl práce.....	8
3	Teoretická část	8
3.1	Arduino	8
3.2	Raspberry Pi	15
3.2.1	Možnosti SW řízení GPIO na Raspberry Pi	17
3.3	Typy přenášených informací – druhy sběrnic	17
UART.....	17
3.3.1	Ošetření spojení Raspberry Pi a Arduino	19
3.3.2	Úprava logických úrovní	19
3.4	Wiring	20
3.5	Arduino IDE.....	20
3.6	Fritzing	21
3.7	Raspbian + NOOBS	21
3.8	Python 3	21
3.9	Geany IDE.....	22
3.10	Markup.su	22
4	Praktická část	23
4.1	Použitý HW a SW	23
4.2	Postup řešení a teoretická východiska pro realizované úlohy.....	23
4.2.1	Oživení Raspberry Pi + první program.....	24
4.2.2	Oživení Arduino + první program	25

4.2.3	Vstupy – Raspberry Pi vs. Arduino	26
4.2.4	Výstupy Raspberry Pi + Arduino.....	29
4.2.5	Sériová linka spojení mezi Raspberry Pi a Arduino	31
4.2.6	I ² C sběrnice Raspberry Pi a Ard	34
4.2.7	SPI sběrnice Raspberry Pi a Ard.....	37
4.2.8	Raspberry Pi a Arduino – komunikace po ethernet	38
4.2.9	Komunikace Arduino → PC	42
4.2.10	Raspberry Pi → PC + Android	43
4.2.11	Praktická úloha – kombinace	44
5	Zhodnocení a závěr	46
6	Přílohy.....	47
6.1	Pracovní listy.....	48
6.1.1	Pracovní list č. 1.....	49
6.1.2	Pracovní list č. 2.....	50
6.1.3	Pracovní list č. 3.....	52
6.1.4	Pracovní list č. 4.....	55
6.1.5	Pracovní list č. 5.....	57
6.1.6	Pracovní list č. 6.....	61
6.1.7	Pracovní list č. 7.....	63
6.1.8	Pracovní list č. 8.....	64
6.1.9	Pracovní list č. 9.....	66
6.1.10	Pracovní list č. 10.....	67
6.1.11	Pracovní list č. 11.....	68
6.2	Seznam použitých zkratk a pojmů.....	69
6.3	Seznam tabulek	70

6.4	Seznam zdrojových kódů	71
6.5	Seznam schémat	72
6.6	Seznam obrázků	73
7	Zdroje.....	74

1 Úvod

V této bakalářské práci se primárně dozvíte o možnostech připojení a komunikace dvou rozdílných zařízení – Arduino mikropočítače postaveného na čipu ATmega328 a Raspberry Pi 3, nejnovější generací dnes již legendárního mikropočítače postaveného na platformě ARM.

Arduino je jednočipový programovatelný mikropočítač, který vykonává jednoduché úkoly při zachování velké rychlosti a stability. Je vhodný pro ovládání různých periférií nebo např. sběr dat z různorodých snímačů při velkých obnovovacích frekvencích až jednotek MHz. Výpočetní výkon Arduino je velice omezený. Pro následné zpracování dat je vhodné využít jiné zařízení s operačním systémem. Zajisté by bylo možné využít Android zařízení nebo stolní PC, ale každé z řešení má svá úskalí. Po konzultaci s vedoucím práce došlo k závěru využít mikropočítač Raspberry Pi. Toto spojení si bere dle mého názoru to nejlepší z obou dostupných světů – tedy rychlost, kontrolu nad časy, okamžité starty, nízkou spotřebu, jednoduchost a stabilitu ze strany Arduino a internetové, grafické a výpočetní možnosti Raspberry Pi. Důvody pro spojení těchto modulů jsou ve složitosti implementace sběru dat na Raspberry Pi a na druhé straně mj. komplikovaný (a cenově nevýhodný) přístup do sítě ethernet u Arduino. Samotný rozšiřující ethernet modul pro Arduino má stejnou hodnotu jako Raspberry Pi bez jakékoliv další přidané hodnoty. [1]

Typografická konvence

-  Menu tlačítko v Raspbian je pojmenováno jako Start

Použité písmo:

- položky menu grafického prostředí: Batang, velikost 10
- terminálové příkazy, názvy knihoven a souborů: **Consolas**, velikost 10, tučné
- kódy programovacích jazyků: `Consolas`, velikost 9
- popisky tabulek, obrázků, schémat a kódů: *Times New Roman*, velikost 9, kurzíva, tmavě modrá barva
- odkazy na webovské stránky: *Times New Roman*, velikost 10

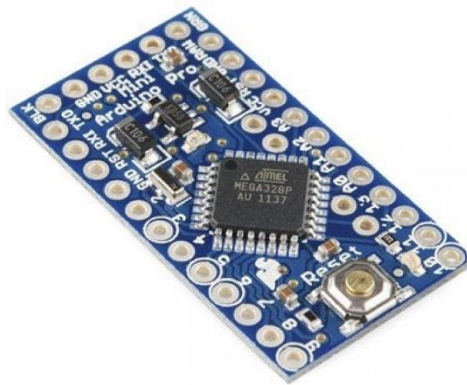
2 Cíl práce

Hlavním cílem práce bylo formou úloh seznámit zájemce s problematikou využití Raspberry Pi a Arduina pro konstrukce elektronických zařízení. Základem úspěšného použití je pochopení a praktické poznání rozdílů mezi Raspberry Pi a Arduinem.

3 Teoretická část

V této části se čtenář seznámí se SW a HW vlastnostmi použitých modulů

3.1 Arduino



Obrázek 1 - Arduino Pro Mini [2]

Arduino je open-source elektronická platforma založená na jednoduše použitelném hardware a software. Arduino mikropočítače umí číst vstupy (např. světlo na snímači, stisk tlačítka nebo například zprávu z komunikátoru Twitter) a přeposlat tento signál na výstup (např. aktivovat motor, rozsvítit LED diodu, zveřejnit cokoliv online). Můžete svému Arduino zařízení říci, co má dělat, posláním sady příkazů do mikrokontroléru. K tomu se používá speciální programovací Arduino jazyk založený na Wiring a speciální programovací prostředí Arduino IDE¹.

¹ existují i alternativní prostředí

Během let se Arduino objevilo v mnoha tisících projektech, od základních každodenních úkonů, až po komplexní vědecké nástroje. Celosvětová komunita tvůrců jako jsou studenti, domácí kutilové, umělci, programátoři i profesionálové, se shromáždili kolem této otevřené platformy. Jejich přínos přidal obrovské množství dostupných znalostí, které mohou být skvělou pomocí dalším zájemcům.

Arduino vzniklo v Ivrea Interaction Design Institute jako jednoduchý nástroj pro rychlé prototypy, primárně zaměřený pro studenty bez nutnosti znalosti elektroniky a programování. Postupem času se Arduino mikrokontroler začal adaptovat novým potřebám a výzvám, počínaje jednoduchými osmibitovými mikrokontroléry pro IoT aplikace, přes přenosná zařízení až po 3D tiskárny a další zažitá prostředí. Všechny Arduino mikrokontroléry jsou kompletně open-source, dovolují uživatelům vytvářet nezávislé projekty a adaptovat je na jejich konkrétní potřeby.

Výhody platformy Arduino:

- nízká cena – klon zařízení s totožnými vlastnostmi i HW stojí např. 60 - 200 Kč
- funguje skrz různé platformy – programovatelné na Windows, MacOS i Linux systémech
- jednoduché a čisté programovatelné prostředí – Arduino SW IDE je jednoduché pro začátečníky, zároveň však dostatečně flexibilní pro pokročilé uživatele
- rozšiřitelný SW, protože je Arduino uvolňováno jako open source (včetně veškeré dokumentace) je dostupný pro rozšíření zkušenými programátory. Programovací jazyk může být rozšířen C++ knihovnamí a lidé čekající na porozumění technických detailů mohou udělat snadný přechod k AVR-C programovacímu jazyku, na kterém je Arduino založeno. Můžete dokonce vkládat AVR-C kód přímo do Arduino programů.
- rozšiřitelný HW – schémata a popis knihoven Arduino mikropočítačů jsou zveřejněny např. pod licencí Creative Commons, zkušení designéři plošných spojů tak dokáží realizovat vlastní verze různorodých projektů [3]

Vzhledem k velkému množství různých Arduino mikropočítačů byla po konzultaci s vedoucím práce vybrána varianta Arduino Pro Mini, které je kompaktní, cenově dostupné, a přitom dostatečně výkonné pro všechny praktické úlohy.

Důvodů pro výběr platformy Arduino bylo hned několik:

- jednoduchost
- rozšířenost – velká komunita
- open source – dostupná dokumentace, schémata
- plná kontrola nad časováním
- vysoké frekvence pro sběr dat
- analogové vstupy/výstupy
- velké množství kompatibilních sběrnic pro komunikaci

Ukázka vybraných verzí Arduino a jejich stručný popis:

úroveň použití	typ arduino mikrokontroléru	stručný popis
základní úroveň	UNO	Arduino UNO je vhodný pro začátečníky v elektronice a programování
	LEONARDO	LEONARDO má integrovaný USB HID
	101	101 má integrovaný akcelerometr, gyroskop, bluetooth a dokáže rozpoznávat různá gesta
	NANO	stejná funkčnost s UNO, pouze menší
	MINI	menší verze než NANO – chybí USB konvertor
rozšířená funkcionalita	MEGA	MEGA 2560 má 54 digitálních I/O pinů a 16 analogových vstupů – vhodný pro větší projekty
	ZERO	jednoduché a výkonné rozšíření platformy na 32 bitů
	PRO MINI	vhodné pro malé prostory, nebo ke stálému umístění
internet věcí	YÚN	tato verze je perfektní k použití při navrhování nových zařízení a také do IoT projektů
	ETHERNET	arduino vybavené ethernet konektivitou (lze nahradit jinou verzí Arduina + "Ethernet shield")
	TIAN	silný třicetidvoubitový mikrokontrolér
přenosné	GEMMA	miniaturní přenosný mikro kontroler na bázi ATtiny85
	LILYPAD ARDUINO USB	vhodný na textilní výrobky, může být všit do látky

Tabulka 1 - ukázka různých verzí Arduino mikrokontrolérů [4]

Napájení

Arduino může být napájeno přímo FTDI konvertorem nebo připojením na napájecí vhodné piny. Díky přítomnému regulátoru dokáže fungovat až při napětí 12 V (musí být připojeno na RAW pin). Vcc – regulované napájení 3,3V (frekvence čipu 8 MHz) nebo 5 V (frekvence čipu 16 MHz) dle vybraného modelu. GND pin pro uzemnění [5]

Paměť

ATmega328 má 32 kB flash paměti pro uložení kódu (bootloader může použít až 2 kB z této paměti). Disponuje 2 kB SRAM a 1 kB EEPROM paměti, která může být čtena a zapisována za pomoci EEPROM knihovny. [5]

Vstupy/výstupy

Každý ze 14 digitálních pinů může být použit jako vstup i jako výstup. Pracují na VCC napětí. Maximální proud na jeden pin 40 mA, vnitřní pull-up odpor 20-50 kΩ. Některé z pinů mají speciální funkce:

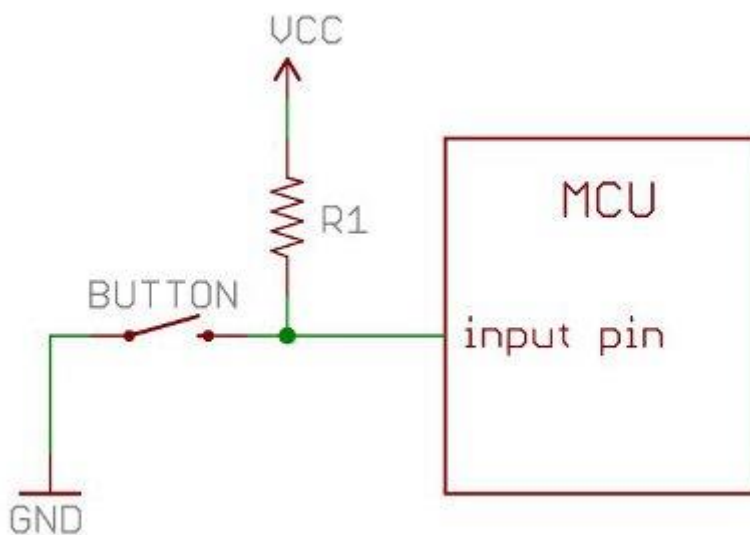
- sériové 0 (RX) a 1 (TX) piny – používány k přijetí (RX) a k odeslání (TX) TTL sériových dat
- vnější přerušení piny 2 a 3, mohou být nakonfigurovány jako spouštěče přerušení na nízkou hodnotu, vzestupnou nebo sestupnou náběhovou hranu, nebo na změnu hodnoty
- PWM – piny 3, 5, 6, 9, 10 a 11 – poskytují osmibitový PWM výstup s použitím `analogWrite` funkce
- SPI – piny 10(SS), 11(MOSI), 12(MISO), 13(SCK) – tyto piny podporují SPI komunikaci, která je poskytována skrytým HW – zatím není implementováno do Arduino jazyka
- LED – pin 13 v Arduino Pro Mini je integrována jedna LED dioda, která se spustí signálem HIGH a zhasne signálem LOW

Arduino Pro Mini má 8 analogových vstupů, každý z nich má desetibitové rozlišení (1024 hodnot). Analogové vstupy měří od 0 do VCC. Některé z pinů mají opět speciální funkci:

- I²C – piny A4 (SDA) a A5 (SCL) – podporují komunikaci I²C (TWI) při použití wire knihovny
- Reset pin – při dodání LOW signálu resetuje mikrokontrolér, užívá se pro přidání externího resetovacího tlačítka [5]

Použití pull-up a pull-down rezistorů

Použité vstupy musí být vždy jasně definované, tzn. připnuté ke GND nebo k VCC napětí 5 V. Pokud vstup nechám odpojený, pin má nedefinovaný stav (tzv. „plovoucí pin“), kde velice pravděpodobně dojde k chybě vyhodnocení. Načtená hodnota může náhodně kolísat mezi 1 a 0. Příčinou je především elektromagnetický šum okolí. Vstupy u Arduina Pro Mini mají integrované pull-up rezistory, které zajišťují jednoznačnost vstupního signálu při stisku tlačítka. Připojení pull-up rezistoru obrací logickou hodnotu, bez stisknutého tlačítka je na výstupu Vcc (logická 1), po stisku tlačítka se napětí uzemní a na vstupu zůstane zbytkové napětí blízké se GND (logická 0). Zde je jednoduché schéma pro pochopení:



Obrázek 2 - Zapojení pull-up rezistoru [6]

Druhá velice podobná možnost je zapojení jako pull-down rezistor. Ty nejsou integrovány ani do jednoho z použitých modulů. Princip je ale obdobný, oproti pull_up rezistoru neinvertuje logickou hodnotu, stisknuté tlačítko má tedy hodnotu 1, nestisknuté má hodnotu 0. Pull-down rezistor se připojuje na stejné místo viz Obrázek 2, jen se připojuje se zemí (GND).

Komunikace

Arduino Pro Mini má mnoho dostupných komunikačních prostředků s PC, jiným Arduinem nebo jinými mikrokontroléry. ATmega328 poskytuje UART TTL sériovou komunikaci, která je dostupná na pinech 0 a 1. Arduino SW zahrnuje sériový monitor, dovolující zaslat jednoduchá textová data z a do Arduino mikrokontroléru přes USB připojení.

SoftwareSerial knihovna umožňuje používat sériovou komunikaci na jakémkoliv digitálním pinu. ATmega328 zároveň podporuje I²C (TWI) a SPI komunikaci. Arduino SW obsahuje wire knihovnu pro jednodušší použití I²C sběrnice. [5]

Programování

ATmega328 obsahuje bootloader, který umožňuje nahrávat nové kódy bez použití externího HW programátoru. Tato komunikace používá protokol STK500. Je možné bootloader vynechat a nahrávat kód za pomoci externího programátoru. [5]

Možnosti připojení:

- I²C
- SPI
- OneWire
- UART
- Pomocí rozšiřujících modulů – Bluetooth, Ethernet, Seriový COM port, ...

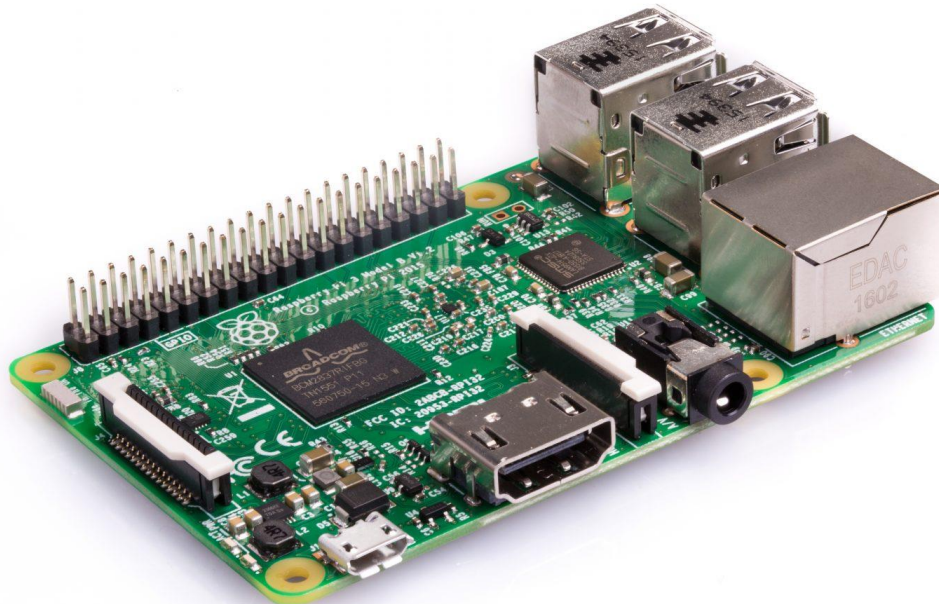
Programovací jazyky:

- ArduBlock
- C#
- Snap4Arduino
- Python
- Wiring [7]

Programovací editor:

- Arduino IDE
- Electron GUI
- Codebender
- Stino
- Eclipse
- Visual Studio
- Komodo Edit
- MariaMole
- Zeus
- Atmel Studio
- AVR-GCC
- CodeBlocks
- AduinoDroid
- PlatformIO
- Xcode [8]

3.2 Raspberry Pi



Obrázek 3 - Raspberry Pi 3 [9]

Pro bakalářskou práci bylo vybráno Raspberry Pi třetí generaci populárního mikropočítače. Přišlo na trh v roce 2016. S cenou 35\$ jde o velice levný a dostupný mikropočítač. Je oblíbený u domácích kutilů, škol či vědeckých pracovníků. Díky obrovské komunitě existuje nepřehledné množství různých aplikací a možností použití.

HW konfigurace Raspberry Pi 3:

- SoC: Broadcom BCM2837
- CPU: 4× ARM Cortex-A53, 1,2 GHz
- GPU: Broadcom VideoCore IV
- RAM: 1 GB LPDDR2 (900 MHz)
- síť: 10/100 Ethernet, 2,4 GHz 802.11n wireless
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- karty: microSD
- GPIO: 40pinový header
- porty: HDMI, 3,5mm analogový audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- kompatibilita se staršími verzemi Raspberry Pi 1 a 2 [10]

Operační systém použitý pro Raspberry Pi je ve většině Linux odlišných distribucí. Poskytují odlišné vlastnosti pro různé aplikace. Zde uvádím několik příkladů:

- Arch Linux ARM
- Fedberry – Fedora Remix
- OpenELEC + XBMC (dnes již Kodi)
- OpenSuseTumbleweed
- Raspbian
- Raspbmc (dnes již OSMC)
- RetroPie
- Ubuntu Mate

Pro zajímavost nelineuxový OS:

- Windows 10 IoT Core (verze Windows pro ARM architekturu) [11]

Modul komunikuje přes UART, I²C, SPI, USB, Ethernet port, Wi-fi, bluetooth,... [12]

Použit lze GPIO piny, USB konektor, Ethernet port nebo některou z bezdrátových přenosů – Wi-fi či bluetooth.

Programovací jazyky:

- Scratch
- Python
- HTML5
- JavaScript
- JQuery
- Java
- C
- C++
- Perl
- Erlang [11]

3.2.1 Možnosti SW řízení GPIO na Raspberry Pi

- připojení pull-up a rezistorů
- nastavení GPIO pinů jako vstupy, výstupy – digitálně
- HW PWM pouze pin 12 – pro více použití: SW PWM – riziko priorit operačního systému a následné nepřesnosti v aplikaci (např. u LED nevadí, u serva nebo OneWire komunikace může vadit)
- chybí A/D převodník – nelze SW nahradit – možno doplnit HW
- existují různé rozšiřující moduly (např. Pixie) [13]

3.3 Typy přenášených informací – druhy sběrnic

UART

Zkratka UART by se dala přeložit jako univerzální asynchronní přijímač/vysílač. Je to kousek hardwaru, který pomocí dvou pinů (většinou označovaných jako RX a TX) odesílá a přijímá data. Jelikož se jedná o asynchronní způsob komunikace, obsahuje přijímač i vysílač vlastní generátor hodinového signálu, kterým se UART řídí. A jelikož je UART univerzální, je také možné rychlost těchto hodin řídit (není stanovena), stejně jako velikost jednoho bajtu, počet stop bitů, paritní bit, atd.

Často můžeme také slyšet zkratku USART. Jedná se v podstatě o to samé, pouze s tím rozdílem, že je USART obsahuje i synchronní režim. [14]

Vlastnosti u mikrokontroléru ATmega32:

- Plný duplex (může současně přijímat i vysílat).
- Synchronní nebo asynchronní režim.
- Možnost nastavení přenosové rychlosti v rozsahu 1200 bps až 250 kbps.
- Podpora rámců délky 5 až 9 datových bitů a 1 až 2 stop bity.
- Generátor sudé/liché parity pro vysílač a hardwarové testování parity pro přijímač.
- Detekce ztráty znaku (Overrun).
- Detekce chyby rámce (Frame Error).
- Obsahuje digitální dolnofrekvenční propust pro detekci falešného start bitu a filtraci zákmitů datových bitů.
- Tři nezávislá přerušení (odvysílání znaku, vyprázdnění vysílacího registru a příjem znaku).
- Možnost zdvojnásobit přenosovou rychlost v asynchronním režimu. [15]

SPI

SPI komunikace se připojuje přes 4 vodiče:

- MISO – Master in, Slave out – Slave zařízení zasílá data Masteru
- MOSI – Master out, Slave in – Master odesílá data Slave zařízením
- SCK – serial clock – hodinový signál, slouží k synchronizaci komunikace, generovaný master zařízením
- SS – Slave select – umožní aktivovat/deaktivovat vybrané Slave zařízení

Sériová sběrnice SPI s rychlostí hodinového signálu až 125 MHz, umožňuje větší počet připojených zařízení – uzlů.

Jeden komunikující uzel je nastaven jako řadič sběrnice (Master) a obsahuje generátor hodinového signálu. Zbytek uzlů funguje v režimu Slave a je synchronizován přivedením hodinového signálu od Master uzlu. To umožňuje obousměrnou komunikaci dat.

Podpora SPI sběrnice je implementována v mnohých populárních mikrořadičích (AVR, PIC16xxx).

SPI sběrnice se jednoduše a snadno implementuje. Není nutné řešit přepínání mezi vysíláním a příjmem. Neumožňuje však jednoduché využití více Master zařízení (uzlů) najednou.

Je nutné konfigurovat registry pro volbu synchronizace hodinového signálu (může reagovat na náběžnou hranu, na sestupnou nebo na obě hrany). [15]

I²C

Oproti SPI je sběrnice I²C vybavena pouze jedním datovým vodičem SDA, z čehož vyplývá, že se data přenáší polo-duplexně. Také to znamená poněkud složitější interní strukturu všech připojených zařízení, protože příslušné piny musí být možné přepínat ze vstupního režimu na režim výstupní.

Každému uzlu je přiřazena jednoznačná adresa. K Master uzlu může být teoreticky připojeno až 128 zařízení při použití sedmibitové adresy. Pokud použijeme desetibitovou adresu, dostaneme až 1024 dostupných adres pro zařízení.

Přenosové rychlosti I²C sběrnice jsou od 10 kpbs, až po 3,4 Mbps. [16] [17]

OneWire

OneWire sběrnice je podobná I²C, má nižší datovou propustnost a delší dosah. Síť vytvořená mezi Master zařízením a OneWire zařízením se nazývá MicroLAN. Velice často se pro připojení používá RJ11 konektor. [18]

Při použití s Raspberry Pi je komunikace nestabilní. Tuto vlastnost můžeme přisuzovat operačnímu systému, ovládá priority jednotlivých procesů běžících v mikropočítači. Díky tomu je velice obtížné číst korektní data např. z teploměru DHT11.

3.3.1 Ošetření spojení Raspberry Pi a Arduino

3.3.2 Úprava logických úrovní

Při spojení zařízení pracujících na různých napěťových úrovních je nutné použít zařízení, které změní logickou úroveň z hodnoty pro jedno zařízení na hodnotu pro druhé zařízení a opačně. Např. lze použít tzv. převodník úrovní (anglicky „level converter“).

- 1) Je nebezpečné zapojit vyšší napětí na nižší napěťovou úroveň, pravděpodobně dojde k poškození zařízení operující s nižším napětím.
- 2) Pokud zařízení rozpoznává signál o určité velikosti jako logickou 1, zařízení s nižším napětím pak nemusí stabilně na vstupu zařízení s vyšší logickou úrovní kýženou hodnotu promítnout.

Existují samostatné proudové či napěťové konvertory a ochrany.

Dostupné jsou zároveň i převodníky mezi jednotlivými druhy sběrnic.

3.3.2.1 Arduino 5V – Raspberry Pi 3,3V

Arduino Pro Mini má v režimu plného výkonu vstupně/výstupní napětí 5V. Raspberry Pi 3 má na GPIO pinech 3,3V, tzn. při připojení výstupu z Arduina na Raspberry Pi GPIO piny by po zapnutí napájení mohlo dojít k trvalému poškození Raspberry Pi. Abychom se tomuto riziku vyhnuli, je nutné využít výše zmiňovaný převodník úrovní, ten nám zároveň zvýší hodnotu výstupu z Raspberry Pi z 3,3V na 5V, aby logickou hodnotu Arduino dokázalo korektně definovat stav vstupu z Raspberry Pi.

Popis zapojených pinů a signálů:

- RXI – vysoko napěťový vstup, signál bude zmenšen a zaslán na RXO pin
- RXO – nízko napěťový výstup, signál je zmenšen z RXI pinu
- TXI – nízko napěťový vstup/výstup, oboustranný převod napětí – umožňuje vstupní nízký signál zvýšit na hodnotu vyššího výstupu (TXO)
- TXO – vysoký napěťový vstup/výstup, oboustranný převod napětí [19]

Schéma zapojení v praktické části viz 4.2.5 UART.

3.3.2.2 Sériová linka

Spojení po sériové lince je realizováno přes COM port – dříve byl integrovaný do počítačů – dnes se využívají převodníky USB. Pracuje na napětích +/-12V. Pro spojení s mikročipy a mikro kontroléry lze využít tzv. RS232 Shifter Board. Jedná se o převodník mezi RS232 a TTL standardem, funguje jako jednoduchý posuvný registr. [20] [21]

3.4 Wiring

Wiring je pravděpodobně nejrozšířenějším programovacím jazykem pro programování v prostředí Arduino IDE. Pokud by byl vybrán jiný programovací jazyk, v pozdější části BP by mohlo dojít ke zbytečným komplikacím při diagnostikování případných chyb v kódu. [22]

3.5 Arduino IDE

Pro programovací jazyk Wiring lze využít několik programovacích prostředí. Arduino IDE je založeno na jazyku Processing a využívám jej jako standardní varianty doporučené na oficiálních stránkách².

² www.arduino.cc

3.6 Fritzing

V praktické části bude nutné naznačit schémata zapojení. Na základě doporučení od vedoucího práce byla vybrána intuitivní aplikace Fritzing. Pro soukromé využití je licencovaná zdarma, v knihovnách má k dispozici většinu dostupných elektronických součástek včetně mikropočítačů a mikrokontroléru. Chybějící knihovny lze stáhnout z internetové databáze a doplnit tak aplikaci o veškeré potřebné komponenty.

3.7 Raspbian + NOOBS

Operační systém použitý v BP je Raspbian – velice rozšířená distribuce Linuxu pro Raspberry Pi na scéně. Na oficiálních stránkách existuje instalátor NOOBS, který jak název napovídá, navede krok po kroku i průměrného uživatele k úspěšnému konci instalace. Více informací k instalaci bude uvedeno v praktické části v úloze č. 1.

Ovládání Raspbianu probíhá za pomoci grafického prostředí, jehož součástí je i konzole, která je použita pro zadávání příkazů a zobrazování zpětné vazby od systému.

3.8 Python 3

Pro programování v Raspberry Pi 3 byl vybrán programovací jazyk Python v jeho třetí verzi. Tento jazyk je velice rozšířený, má výbornou dokumentaci a podporu. Začátečníci se jej poměrně snadno naučí. Z těchto důvodů byl Python 3 zvolen jako vhodnou variantu pro studenty VŠ, pro které budou vytvořeny úlohy v praktické části bakalářské práce.

Pro práci s HW perifériemi je nutné Python rozšířit o knihovny, které jsou načteny pomocí funkce `import` v hlavičce každého zdrojového kódu. Popis těchto knihoven přesahuje rámec BP.

3.9 Geany IDE

Geany patří mezi rychlé a jednoduché multiplatformní programovací editory. Dá se využít pro různé programovací jazyky. Např. C, Java, PHP, HTML, Python, Perl, Pascal a další.

Navíc je defaultně implementován v již zmiňované distribuci Linuxu Raspbian instalovanou přes instalátor NOOBS.

Lze jej používat na většině dostupných platformách, např. Linux, FreeBSD, OpenBSD, MacOS X, Windows... [23]

Poskytuje dostatečně komfortní a jednoduché prostředí pro programování včetně automatického zabarvování kódů dle použitých syntaxí – tato funkce pomůže s přehledností psaného kódu. Zároveň Geany uživateli během psaní kódu našeptává funkce, to urychlí psaní kódu a může omezit množství chyb, které uživatel způsobí při zadávání kódů.

3.10 Markup.su

Markup.su³ je internetová stránka pro formátování kódů.

Pro kódy z Arduino IDE je použit jazyk ActionScript (žádný bližší Wiringu nebyl k dispozici), styl Active4D.

Pro kódy psané jazykem Python – v aplikaci Geany je zvolen jazyk Python, styl Active4D.

³ www.markup.su

4 Praktická část

Navrhnutí, vytvoření a vyzkoušení modelových úloh ilustrujících vybrané způsoby komunikace.

4.1 Použitý HW a SW

Pro vypracování praktické části byl použit následující HW:

- Stolní PC s procesorem Intel i5-6500, základní deska Gigabyte GA-B150-HD3; 8 GB DDR4 RAM 2133 Mhz
- Klon mikropočítače – Arduino Pro Mini + FTDI konvertor pro zasílání dat z PC
- Breadboard deska
- Raspberry Pi 3 rev. B
- microSD karta Lexar o kapacitě 16 GB – Class 10 (minimální garantovaná rychlost čtení a zápisu 10 Mb/s)

A následující SW:

- Arduino IDE 1.8.1
- Fritzing 0.9.3
- NOOBS_lite_v2.3 (pro instalaci Raspbian)
- Raspbian GNU v8 JESSIE
- Geany v.1.24.1
- Python 3
- Windows 10 ve verzi 1607 (build 14393.693);

4.2 Postup řešení a teoretická východiska pro realizované úlohy

Následující uvedené postupy (modelové úlohy) řeší krok po kroku oživení systémů Raspberry Pi a Arduino do použitelného stavu. Pracovní listy lze použít dvěma způsoby:

- 1) jako pomůcka pro samostudium
- 2) jako samostatné dílčí úlohy

Na základě modelových úloh je možné snadno vytvořit jiné úlohy. Navrhované úlohy zdaleka nejsou vyčerpávající.

V textu je detailně vysvětlen postup, ze kterého mohou následně čerpat řešitelé při použití a případných nejasnostech pracovních listů.

Některé modelové úlohy vycházejí ze vzorových příkladů uváděných v oficiální internetové dokumentaci.

4.2.1 Oživení Raspberry Pi + první program

V první úloze je vysvětlen postup pro instalaci operačního systému Raspbian do Raspberry Pi 3, nainstalování a připojení k VNC a SSH. Dále je popsáno vytvoření prvního skriptu v programovacím jazyce Python:

- ze stránky projektu⁴ byla stažena online instalace NOOBS lite (pokud není k dispozici připojení k internetové síti, je nutno stáhnout offline verzi instalace)
- microSD paměťová karta 16 GB minimálně třídy 10 („class“ 10) byla ve Windows naformátována do formátu FAT32
- archiv byl rozbalen a zkopírován na připravenou microSD paměťovou kartu
- po připojení napájení došlo k provedení boot sekvence z paměťové karty
- během online instalace bylo Raspberry Pi 3 připojeno k internetu (přes Wi-fi nebo ethernet kabel)⁵
- dle postupu instalátoru byl nainstalován operační systém Raspbian.
- po instalaci proběhl automatický restart [24]
- pro Raspberry Pi je vhodné použít vzdálený přístup, byla zvolena technologie VNC, umožňující zobrazení grafického prostředí, v našem případě, na počítači s OS Windows
 - po připojení k VNC serveru byl spuštěn terminál
 - Start → Accessories → LXTerminal
- zadáním příkazu `sudo apt-get update` se nainstalovaly aktualizace aplikací a knihoven
- příkazem `sudo apt-get install realvnc-vnc-server` byl nainstalován RealVNC server
- aktivování funkce VNC bylo provedeno ve Start menu: Start → Raspberry Pi configuration → Interfaces → VNC → Enabled
- zařízení bylo restartováno příkazem `sudo reboot`
- na počítači byl nainstalován SW VNC viewer v6.0.2 ze stránky realvnc.com⁶
- Připojení ke sdílené ploše pomocí aplikace VNC Viewer
 - v administraci wi-fi routeru byla vyčtena IP adresa Raspberry Pi, tuto IP adresu zadáme do aplikace VNC viewer
 - přihlašovací údaje – uživatelské jméno: pi; uživatelské heslo: raspberry
- změna rozlišení vzdálené obrazovky v OS Raspbian: Start → Preferences → Raspberry Piconfiguration → set resolution → 1280*720px
- Python editor GeanyProgrammer's editor:
 - Start → Programming → GeanyProgrammer's Editor
- vytvoření nového programu File → New (with template) → main.py
- smazán kód kromě prvního řádku
- na další řádek vepsán kód `print „Zdravim Python!“`.
- stiskem klávesy F5 došlo ke zkompilování a spuštění skriptu.
- v konzoli je vypsán text `Zdravim Python!`.

⁴ <https://www.Raspberry.Pi.org/downloads/noobs>

⁵ neplatí pro offline instalaci

⁶ <https://www.realvnc.com/download/viewer/>

4.2.2 Oživení Arduino + první program

Ve druhé úloze je předvedeno, jak zapojit Arduino Pro Mini s konvertorem FTDI. Bude vytvořen první program ovládající pin 13 (s integrovanou LED diodou).

- Instalace SW Arduino IDE – ze stránky projektu Arduino⁷ byla stažena aplikace Arduino IDE (pro Windows).
- Arduino bylo připojeno do breadboard a dle schématu níže spojeno s FTDI konvertorem:

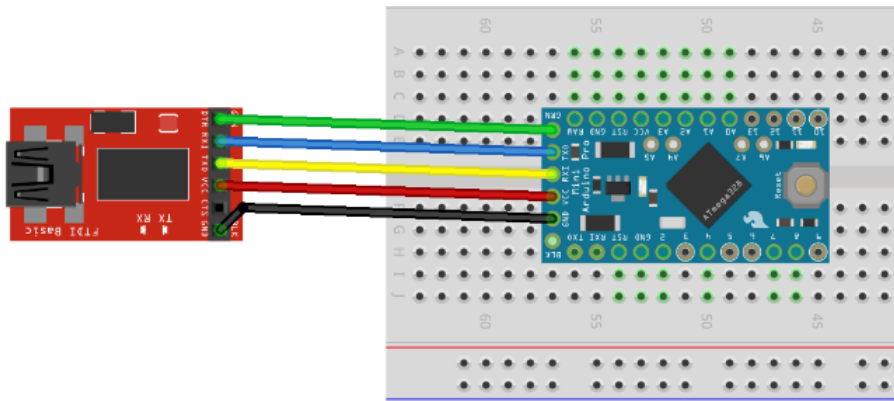


Schéma 1 – Arduino, FTDI konvertor

- Po spuštění Arduino IDE bylo nutné provést prvotní nastavení:
 - Nástroje → Vývojová deska → Arduino Pro or Pro Mini
 - Nástroje → Procesor → ATmega328 (5V, 16 MHz)
 - Nástroje → Port → COM4
 - Nástroje → Programátor → AVRISP mkII
- přes aplikaci Arduino IDE byl do mikrokontroléru nahrán následující kód:

```
// setup funkce pro nastaveni vstupu a vystupu
// probehne po resetu ci zapojeni napajeni pouze jednou
void setup() {
}
// tato smycka pobezi stale dokola
void loop() {
  Serial.print(„test“); // vypise „test“ do serioveho monitoru
  delay(1000)           // cekani 1000ms
}
```

Kód 1 – Arduino, úloha č. 1, první program, upraveno a převzato z [25]

Při zobrazení sériového monitoru: Nástroje → Sériový Monitor

Se každých 1000ms vypisuje hláška „test“.

⁷ <https://www.arduino.cc/en/Main/Software>

4.2.3 Vstupy – Raspberry Pi vs. Arduino

Jako demonstrační zapojení digitálního vstupu bylo použito tlačítko.

4.2.3.1 Arduino

V této úloze bude vysvětleno, jak zapojit tlačítko k Arduino Pro Mini. Jak nakonfigurovat pin č. 2 jako vstup a jak rozsvítit integrovanou LED diodu na pinu č. 13 po jeho stisknutí. Obvod byl zapojen dle následujícího schématu:

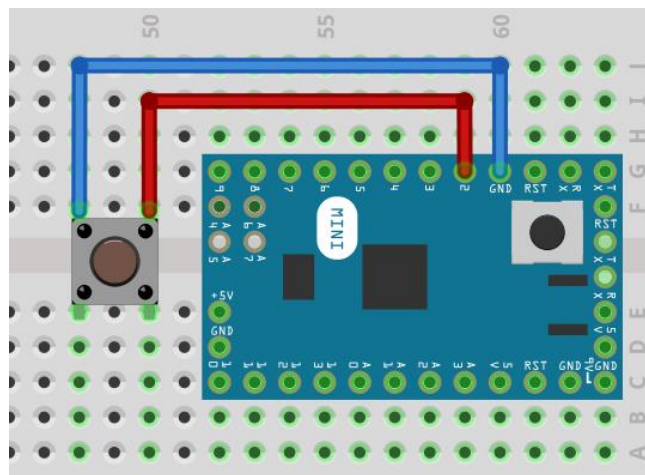


Schéma 2 – Arduino, tlačítko

Pull-up rezistor se v Arduino IDE aktivuje při deklarování pinu jako vstupu přes funkci `pinMode(2, input_pullup)`. Přidáním `_pullup` bylo zajištěno připojení pull-up rezistoru. Zapojení otáčí logickou hodnotu. LOW hodnota na vstupu je tedy logická 1/True, HIGH hodnota je logická 0/False. [26]

Do aplikace Arduino byl nahrán následující kód:

```
// nastavení neměnných konstant
const int tlacitkoPin = 2;    // číslo pinu s připojeným tlačítkem
const int ledPin = 13;       // číslo pinu s integrovanou led diodou
// proměnná, která se bude měnit
int tlacitkoStav = 0;        // proměnná pro sledování stavu tlačítka
void setup() {
    // nastavení ledPin jako výstup
    pinMode(ledPin, OUTPUT);
    // nastavení tlacitkoPin jako vstupu + připojení pullup resistoru
    pinMode(tlacitkoPin, INPUT_PULLUP);
}
void loop() {
    // načte stav tlačítka do proměnné tlacitkoStav
    tlacitkoStav = digitalRead(tlacitkoPin);
    // kontrola, zda je tlačítko stisknuté a pokud ano...
    if (tlacitkoStav == HIGH) {
        // ...rozsvítí led diodu
        digitalWrite(ledPin, LOW);
    } else {
        // jinak vypne led diodu
        digitalWrite(ledPin, HIGH);
    }
}
```

Kód 2 – Arduino, úloha č. 2, vstupy, upraveno a převzato z [25]

4.2.3.2 Raspberry Pi

V této úloze bude připojeno tlačítko k Raspberry Pi 3, bude nastaven GPIO pin a následně vyčtena hodnota ze vstupu.

- na breadboard bylo připojeno tlačítko na pin GPIO18 a GND dle následujícího schématu:

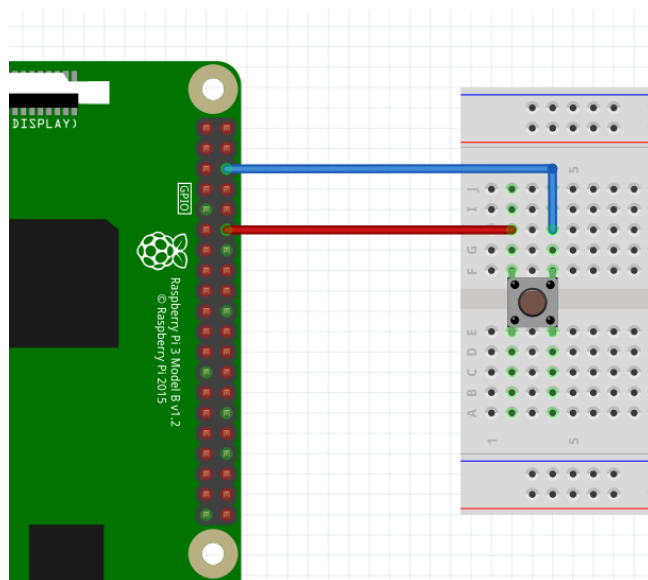


Schéma 3 – Raspberry Pi, tlačítko

- v Geany editoru byl vytvořen nový projekt
- do projektu byl vložen tento kód:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_UP)
while True:
    hodnota_vstupu = GPIO.input(18)
    if hodnota_vstupu == False:
        print('Tlacitko stisknuto')
        time.sleep(0.2)
```

Kód 3 – Raspberry Pi, úloha č. 3, vstupy

- po spuštění skriptu a stisku tlačítka bylo na konzoli vypsáno „Tlacitko stisknuto“
[27]

4.2.4 Výstupy Raspberry Pi + Arduino

V této úloze použijeme výstupy obou modulů k jednoduchému rozblikání LED diody. K oběma platformám bude z důvodu ochrany LED diody připojen odpor o hodnotě 330Ω.

4.2.4.1 Arduino

V této úloze se naučíme nastavit určitý pin jako výstup a jak rozblikat LED diodu s požadovanou frekvencí.

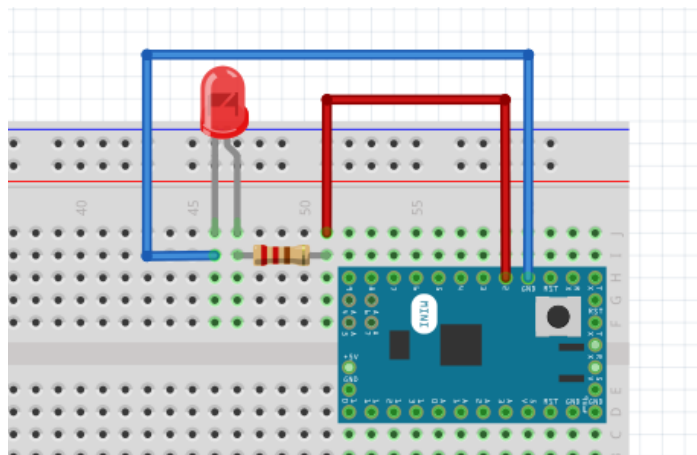


Schéma 4 – Arduino, LED dioda

Do Arduino Pro Mini byl nahrán následující kód:

```
const int ledPin = 2;
void setup() {
  // nastavi ledPin (pin 2) jako vystup
  pinMode(ledPin, OUTPUT);
}
void loop() {
  digitalWrite(ledPin, HIGH); // zapne vystup na pinu 2; rozsviti led diodu
  delay(1000);                // vterina cekani
  digitalWrite(ledPin, LOW);  // vypne vystup na pinu 2; zhasne led diodu
  delay(1000);                // vterina cekani
}
```

Kód 4 – Arduino, úloha č. 4, výstupy, upraveno a převzato z [25]

Po kontrole, kompilaci a spuštění kódu v Arduino Pro Mini se dioda rozblikala s frekvencí 0,5 Hz.

4.2.4.2 Raspberry Pi

Rozblikání LED diody u Raspberry Pi 3 je z pohledu zapojení velice podobné Arduino, po zapojení dle následujícího schématu:

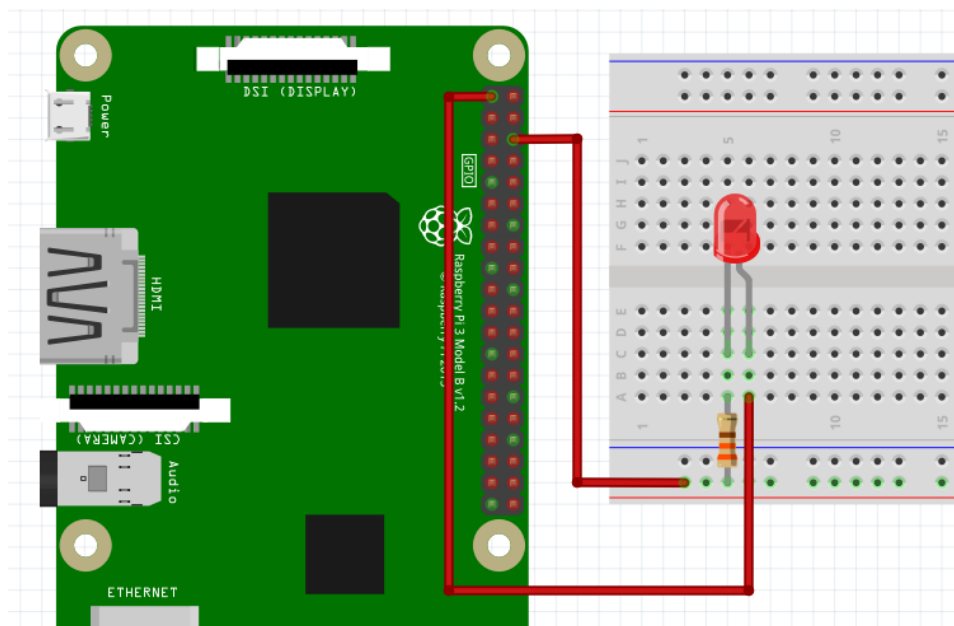


Schéma 5 – Raspberry Pi, LED dioda

Do aplikace Geany byl vložen následující kód:

```
import GPIO.RPi as GPIO ##import GPIO library
import time
GPIO.setmode(GPIO.BOARD) ##pouzit cislovani desky
GPIO.setup(11,GPIO.OUT) ##nastavit 11 pin jako vystup
while 1: ##nekonecna smycka
GPIO.output(11,1) ##Zapnout led diodu
time.sleep(1) ##pockat jednu sekundu
GPIO.output(11,0) ##Vypnout led diodu
time.sleep(1)
```

Kód 5 – Raspberry Pi, úloha č. 4, výstupy

Po spuštění python skriptu se připojená LED dioda rozbliká s nepřiliš přesným časováním.

U této úlohy je důležité si uvědomit, dříve zmiňovanou vlastnost Raspberry Pi: Mikropočítač nemá kvůli chytrému OS (Linux OS) naprostou kontrolu nad časováním. Procesor dle potřeby přerozděluje priority určitým procesům. Pokud tedy spustíme na stejně program rozblikání LED diody na Arduino i na Raspberry Pi ve stejný moment, můžeme sledovat, jak se Raspberry Pi bude postupně opožďovat. U jednoduché úlohy blikání LED diody nemusí tato vlastnost způsobovat problém, je však kritickým aspektem při používání některých synchronních metod komunikace – např. OneWire. Pro přesné odečítání časových pulsů existují RTC moduly (Real Time Clock - reálné hodiny času), využívají integrovaného oscilátoru. [28]

4.2.5 Sériová linka spojení mezi Raspberry Pi a Arduino

USB

V této úloze popíšeme, jak lze spojit Raspberry Pi 3 a Arduino Pro Mini pomocí kabelu USB. Raspberry Pi bude v úloze Master zařízením, Arduino bude v roli Slave.

- Připojení komponent dle schématu:

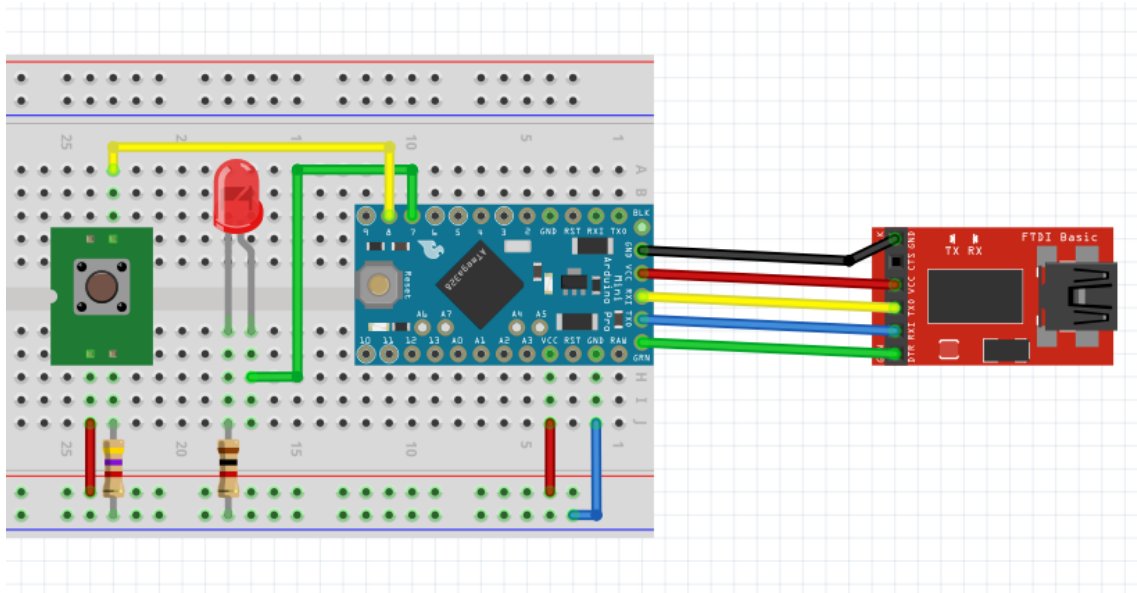


Schéma 6 – Arduino, LED dioda + tlačítko

- do terminálu byly vypsány následující příkazy:

```
sudo apt-get update  
sudo apt-get install arduino -y
```
- byl stažen Nanpy SW a FW:

```
git clone https://github.com/nanpy/nanpy  
git clone https://github.com/nanpy/nanpy-firmware
```
- v adresáři nanpy-firmware se příkazem `./configure.sh` vygeneroval soubor `cfg.h` do složky `nanpy`
- pro vytvoření složky v `nanpy-firmware` je potřeba spustit Arduino IDE, po spuštění aplikaci znovu uzavřít
- v terminálu: `cp -avr nanpy-firmware ~/sketchbook/libraries`
- v Arduino IDE byl vybrán a načten nanpy zdrojový kód: `file` → `sketchbook` → `nanpy-firmware` → `nanpy`

- Nastavení: tools → Board → Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega 328
- Serial Port → /dev/ttyUSB0
- do nového skriptu s názvem `button.py` byl vepsán následující kód:

```

from nanpy import (ArduinoApi, SerialManager)
from time import sleep
ledPin = 7
buttonPin = 8
ledState = False
buttonState = 0
try:
    connection = SerialManager()
    a = ArduinoApi(connection = connection)
except:
    print("Připojení se nepodařilo")
# pinMode se nastaví úplně stejně, jako kdybychom programovali přímo v Arduino IDE
# využije se vytvořený objekt a.
a.pinMode(ledPin, a.OUTPUT)
a.pinMode(buttonPin, a.INPUT)
try:
    while True:
        buttonState = a.digitalRead(buttonPin)
        print("Nase tlačítko je: {}".format(buttonState))
        if buttonState:
            if ledState:
                a.digitalWrite(ledPin, a.LOW)
                ledState = False
                print("LED OFF")
                sleep(1)
            else:
                a.digitalWrite(ledPin, a.HIGH)
                ledState = True
                print("LED ON")
                sleep(1)
except:
    a.digitalWrite(ledPin, a.LOW)

```

Kód 6 – Raspberry Pi, úloha č. 5, sériová komunikace přes USB

- v terminálu byl příkazem `Python3 buttonLED.py` spuštěn skript, monitoruje stisknutí tlačítka připojeného k Arduino Pro Mini a zároveň po stisku otáčí logickou hodnotu výstupu č. 7 opět přes Arduino. Tím rozsvěcuje a zhasíná připojenou LED diodu. [29]

Řešení problému:

Nefunkční příkazy pro instalaci SW. Objevila se chybová hláška: GPG error - clearsigned file isn't valid, got "NODATA". Řešení nalezeno na webové stránce⁸. Pro opravu byli zadány následující příkazy:

```
sudo apt-get clean  
sudo mv /var/lib/apt/lists /var/lib/apt/lists.broke  
sudo mkdir -p /var/lib/apt/lists/partial  
sudo apt-get update
```

Chyba byla ve skrytém proxy serveru od providera internetového připojení.

UART

Spojení přes UART sběrnici bude provedeno přes TX a RX piny v Arduino i Raspberry Pi. Přes tyto piny probíhá sériová komunikace na obou použitých modulech. Vzhledem k rozdílnému napětí je nutné využít převodník napětíových úrovní.

- Obvod byl zapojen dle následujícího schématu:

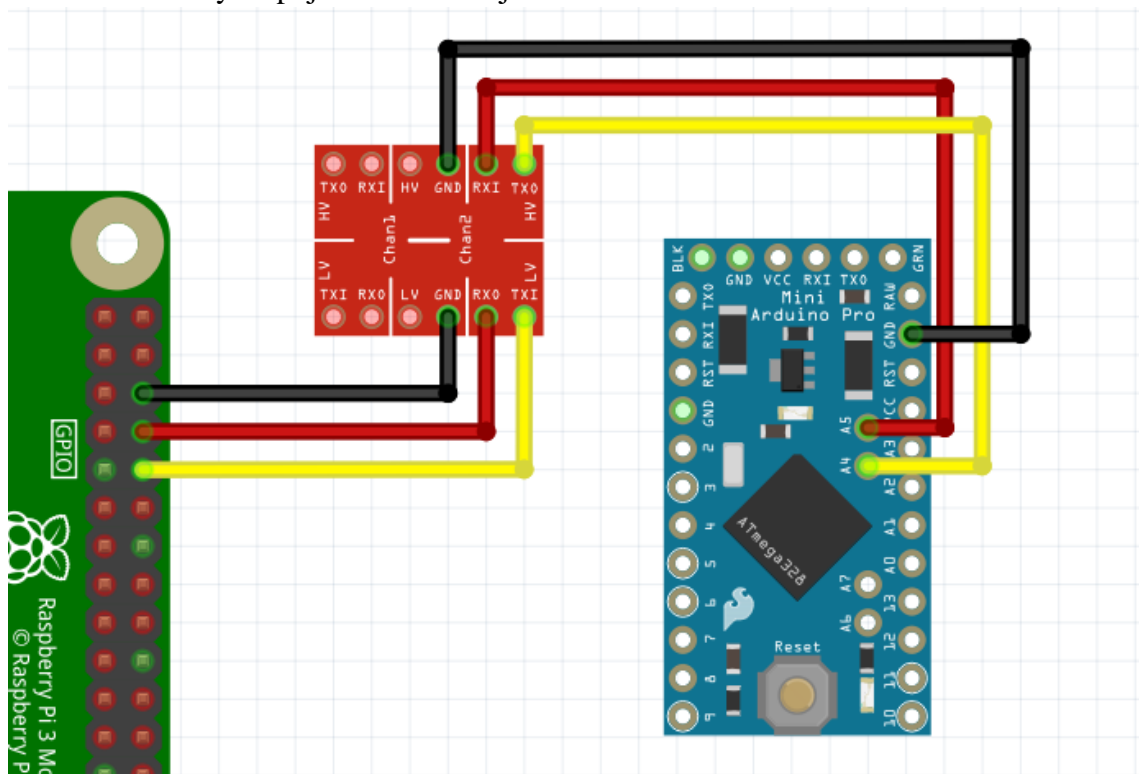


Schéma 7 – Arduino, Raspberry Pi, UART + převodník úrovní

⁸ <https://www.bigbrus.com/2017/04/09/gpg-error-clearsigned-file-isnt-valid-got-nodata>

- kód pro Arduino Pro Mini:

```

void setup() {
  Serial.begin(9600);    // zahajeni seriove komunikace
}
void loop() {
  if (Serial.available() > 0) {
    int incoming = Serial.read();
    Serial.print("znak obdrzen: ")
    Serial.print(incoming, DEC);
  }
}

```

Kód 7 – Arduino, úloha č. 5, UART

- pro přípravu SW komunikace na Raspberry Pi 3 bylo nutné v menu Raspberry Pi Configuration deaktivovat sériovou komunikaci, která by jinak zůstala blokována pro konzoli. Start → Volby → Raspberry Pi Configuration → Interfaces → Serial → Disable
- byl proveden restart Raspberry Pi příkazem `sudo reboot`
- instalace aplikace minicom, příkaz: `sudo apt-get install minicom`
- nyní je možné zahájit komunikaci příkazem `minicom -b 9600 -o -D /dev/ttyAMA0`
- po zadání textu do konzole Minicom se zasláná data zobrazí v sériovém monitoru Arduino IDE. [30]

4.2.6 I²C sběrnice Raspberry Pi a Ard

V této úloze předvedu připojení sběrnice I²C, Raspberry Pi bude připojeno jako Master a Arduino jako Slave.

- aktivace I²C v Raspberry Pi grafickém konfigurátoru: Start → Volby → Raspberry Pi Configuration → Interfaces → I2C → Enabled
- po změně bylo nutné restartovat Raspberry Pi
- v konzoli byl nainstalován `i2c-tools` příkazem `sudo apt-get install i2c-tools`
- příkazem `sudo adduser pi i2c` dojde k přidání oprávnění `i2c` uživateli `pi`.
- instalace `pPython-SMBus`, příkazem: `sudo apt-get install python-smbus`

- zdrojový kód pro Arduino Pro Mini:

```
#include <Wire.h>
#define SLAVE_ADDRESS 0x04 //nastaví adresu slave zařízení na 04
int number = 0;
int state = 0;
void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(9600); // start seriové linky pro výstup
  Wire.begin(SLAVE_ADDRESS); // nastavení slave modu
  Wire.onReceive(receiveData); // zpětné volání pro přijímání a odesílání dat
  Wire.onRequest(sendData);
  Serial.println("Připraven!");
}
void loop() {
  delay(100);
}
void receiveData(int byteCount){ // zpětné volání pro přijatá data
  while(Wire.available()) {
    number = Wire.read();
    Serial.print("přijata data: ");
    Serial.println(number);
    if (number == 1){
      if (state == 0){
        digitalWrite(13, HIGH); // Rozsvícení LED diody
        state = 1;
      }
      else{
        digitalWrite(13, LOW); // Zhasnutí LED diody
        state = 0;
      }
    }
  }
}
void sendData(){ // zpětné volání pro odeslaná data
  Wire.write(number);
}
```

Kód 8 – Arduino, úloha č. 6, I²C

Vytvoření skriptu pro Raspberry Pi:

```
import smbus
import time
bus = smbus.SMBus(1)
# Adresa, kterou urcujeme i2c zarizeni:
adresa = 0x04
def writeNumber(hodnota):
    bus.write_byte(adresa,hodnota)
    return -1
def readNumber():
    number = bus.read_byte(adresa)
    return number
while True:
    promenna = input ("Vloz cislo 1 az 9: ")
    if not promenna:
        continue
    writeNumber(promenna)
    print "Raspberry Pi: Ahoj Arduno, poslal jsem ti cislo ", promenna
    time.sleep(1)
    number = readNumber()
    print "Arduno: Ahoj Raspberry, obdrzel jsem od tebe cislo ", number
```

Kód 9 – Raspberry Pi, úloha č. 6, I²C

Zapojení dle schématu:

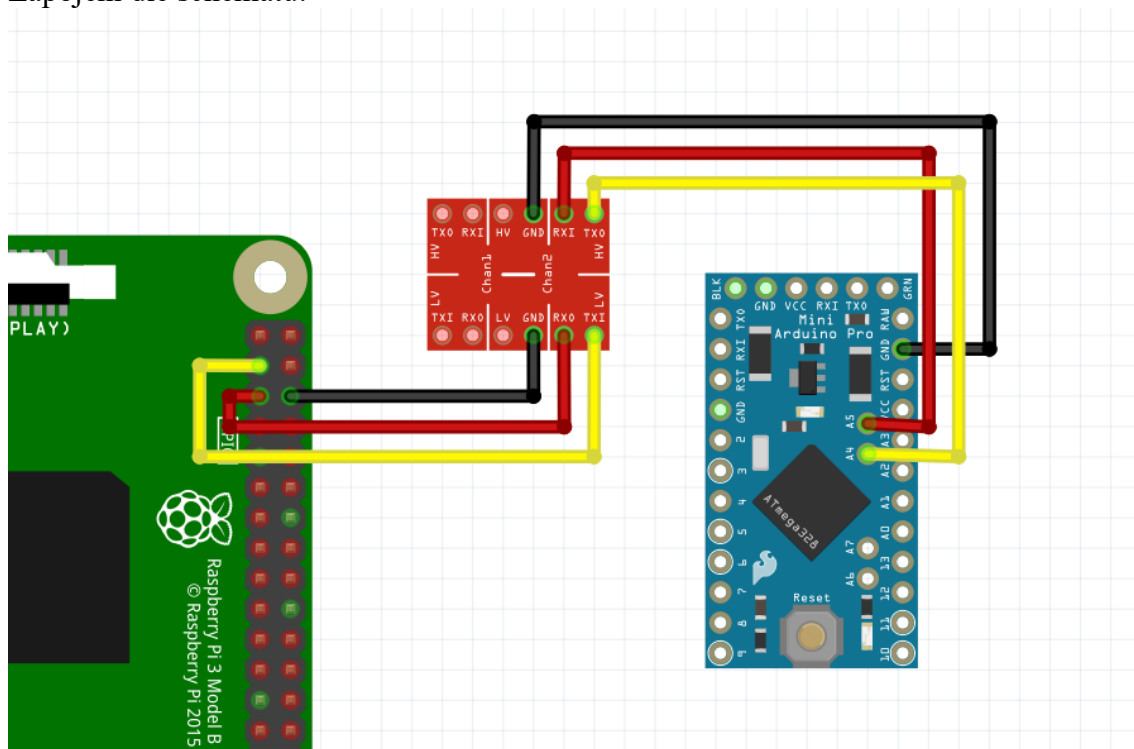


Schéma 8 – Arduino, Raspberry Pi, I²C + převodník úrovní

Po připojení bylo možné provést na Raspberry Pi kontrolu připojených zařízení příkazem do konzole `i2cdetect -y 1`, na adrese 0x04 je k dispozici Arduino Pro Mini. Po spuštění vytvořeného python skriptu vložíte číslo, které je po I²C sběrnici odesláno do Arduina, zde je přijato a opět odesláno zpět do Raspberry Pi. [31]

4.2.7 SPI sběrnice Raspberry Pi a Ard

Pro použití SPI sběrnice pro Raspberry Pi je nutné aktivovat SPI v Raspi-config. Start → Volby → Raspberry Pi Configuration → Interfaces → SPI → Enabled. Zároveň je potřeba doinstalovat knihovnu SPIDEV, popis této knihovny svým rozsahem přesahuje rozsah této bakalářské práce. Raspberry Pi je v této úloze použito jako Master zařízení. [32]

Pro použití SPI na Arduino Pro Mini:

Při komunikaci využíváme SPCR, SPSR a SPDR (SPI Control Register; SPI Status Register; SPI Data Register)

Detailnější popis je v komentáři přímo ve zdrojovém kódu:

```
// nastaví ledPin (pin 2) jako vystup
unsigned char hello[] = {'A','h','o','j',' ','R','p','i','\n'};

byte marker = 0;
void setup (void)
{
  pinMode(MISO, OUTPUT); //nastaveni MISO na vystup
  SPCR |= _BV(SPE); //nastavení SPI enable bitu
}
void loop (void)
{
  if((SPSR & (1 << SPIF)) != 0) //pokud byl prijat bit a zaroven doslo
                                k precteni ci zapisu registru SPDR
  {
    SPDR = hello[marker]; //nacteni hodnoty do data registru
    marker++;
    if(marker > sizeof(ahoj))
    {
      marker = 0;
    }
  }
}
```

Kód 10 – Arduino, úloha č. 7, SPI komunikace

Zapojení dle následujícího schématu:

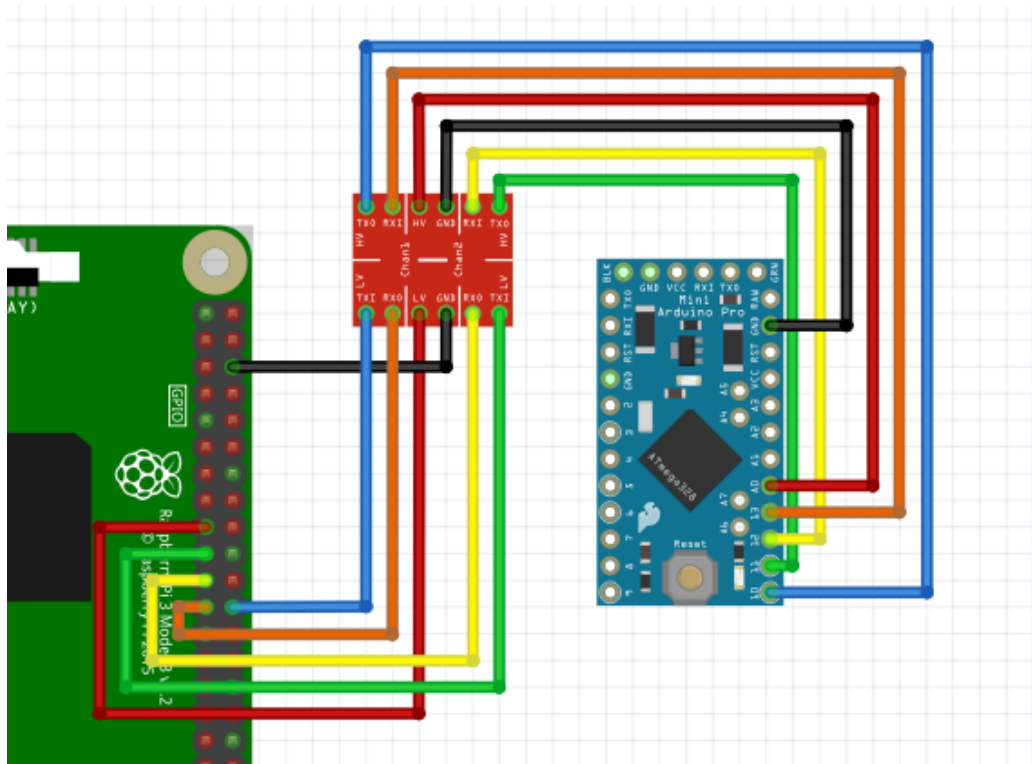


Schéma 9 – Arduino, Raspberry Pi, SPI + převodník úrovní

Po spuštění skriptu dojde k navázání komunikace pomocí SPI sběrnice. [33]

4.2.8 Raspberry Pi a Arduino – komunikace po ethernet

Komunikace po síti ethernet je pro praktické využití pro komunikaci mezi Arduino a Raspberry Pi nevýhodná, existují levnější a na implementaci jednodušší alternativy, jak obě zařízení spojit.

Arduino Pro Mini nedisponuje ethernet portem, je proto nutné samostatně dokoupit Ethernet Shield, jeho cena zvedá nákladnost použití Arduina na úroveň samotného Raspberry Pi 3.

V této úloze použijí oficiální Arduino Ethernet Shield operující s čipem W5500:



Obrázek 4 - Arduino Ethernet Shield [34]

Vzhledem k rozdílné velikosti Arduino Pro Mini je nutné připojit piny pomocí drátků. Ethernet Shield nepasuje přímo na Arduino Pro Mini.

Zde je schéma zapojení:

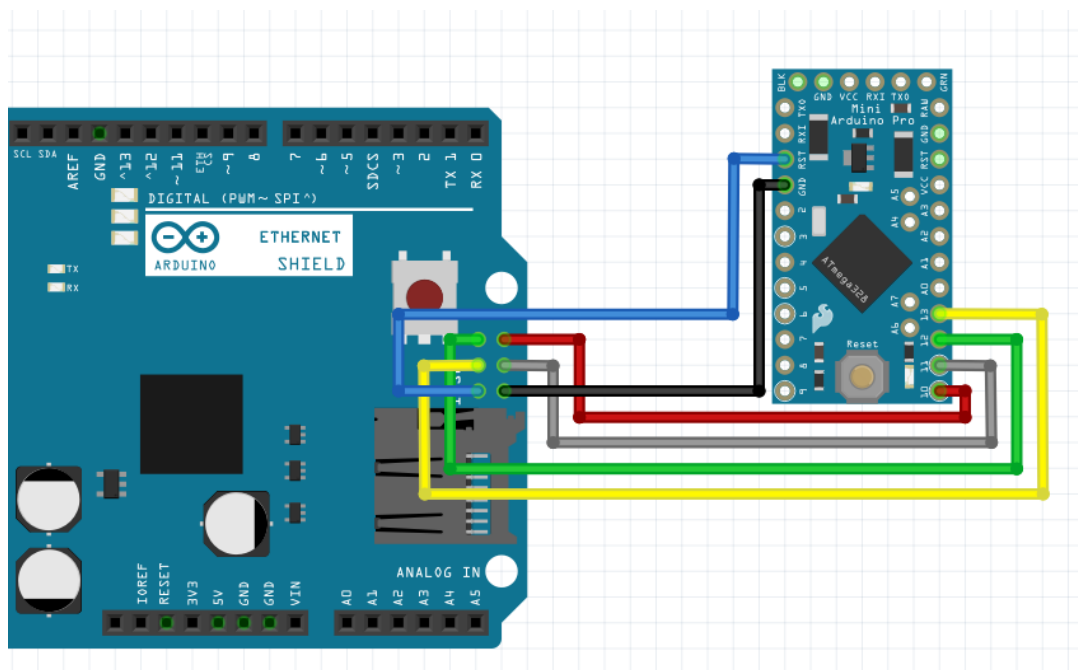


Schéma 10 – Arduino, ethernet shield

Kód zadaný do Arduino:

```
#include <SPI.h>
#include <Ethernet.h>
//mac adresu nize nastavit dle realne mac adresy HW:
//priklad zde: //90-a2-da-0f-25-E7 = 0x90, 0xA2, 0xDA, 0x0f, 0x25, 0xE7
byte mac[] = {0x90, 0xA2, 0xDA, 0x0f, 0x25, 0xE7};
//ip adresa for ethernet shield
byte ip[] = {192,168,1,113};
//pro telnet pouzije port 23
EthernetServer server = EthernetServer(23);
void setup() {
  Serial.begin(9600); //pro kontrolu seriove komunikace
  while(!Serial){
    ; //cekani na otevreni pripojeni
  }
  Ethernet.begin(mac,ip); // inicializace EthernetShield
  delay(1000);
  server.begin();
  if(server.available()){
    Serial.println("Klient dostupny");
  }
}
void loop() {
  // sem zapiste kod, který budete chtít opakovaně provádět
  EthernetClient client = server.available();
  message = client.read();
  server.write(message);
  server.write("Test!");
  Serial.println(message);
}
```

Kód 11 – Arduino, úloha č. 8, ethernet

Kód do python skriptu v Raspberry Pi:

```
data = None
timeout = 3 #casovy timeout v sekundach
msg = "Zprava z Raspberry Pi"
host = "192.168.1.113"
print ("Pripojuji se k " + host)
port = 23
s = socket(AF_INET, SOCK_STREAM)
print "Socket vytvoren"
ready = select.select([s], [], [], timeout)
s.connect((host, port))
print("Pripojeni vytvoreno")
if ready[0]: #cekani na data
    print("[INFO] Odesilam zpravu...")
    s.sendall(msg)
    print("[INFO] Zprava odeslana.")
    data = s.recv(4096)
    print("[INFO] Data obdrzeny.")
    print data
```

Kód 12 – Raspberry Pi, úloha č. 8, ethernet

Po spojení obou modulů ethernet kabelem a spuštění skriptu se přes protokol TCP/IP z Raspberry Pi přenesla zpráva: „Zprava z Raspberry Pi“, ta byla přijata Arduino zařízením. Arduino zpět na server pošle zprávu „Test!“ a vypíše na sériovém monitoru „Zprava z Raspberry Pi“. [35]

4.2.9 Komunikace Arduino → PC

- ze stránky⁹ byl stažen instalační soubor pro komunikaci mezi aplikací Excel a zařízením Arduino
- po instalaci nahrajeme kód ze souboru Arduino_Excel_21¹⁰

```
#include <rExcel.h>
long   idx = 0;           // index
int    outputTiming = 1000; // časování
float  a0;              // A0 pin reading
char   value[16];       // zapisovana nebo prectena hodnota
rExcel myExcel;        // trida pro vymenu dat s Excelem

void setup() {
  Serial.begin(115200);
  delay(1000);
  myExcel.clearInput();
  randomSeed(analogRead(5)); //pokud není připojen pin 5, dojde ke generování náhodných pulsu
}

void loop() {
  static unsigned long loopTime = 0;
  static unsigned long time1 = 0;
  int ret;
  loopTime = millis();
  if ((loopTime - time1) >= outputTiming) {
    time1 = loopTime;
    a0 = (float)analogRead(0) / 1023.0 * 5.0; // A0 pin čtení na hodnotu ve volttech
    rnd = (float)random(0, 1000) / 1023.0 * 5.0; // náhodná hodnota
    myExcel.write("Example", "B5", a0, 2); // psaní dat do bunek
    myExcel.write("Example", "B6", rnd, 2);
    myExcel.writeIndexed("Example", idx+1, 1, "%date%");
    myExcel.writeIndexed("Example", idx+1, 2, "%time%");
    myExcel.writeIndexed("Example", idx+1, 3, idx);
    myExcel.writeIndexed("Example", idx+1, 4, a0, 2);
    myExcel.writeIndexed("Example", idx+1, 5, rnd, 2); // konec psaní dat do bunek
    idx++;
    if (idx > 59) {
      myExcel.clear("Example", "A11:F70"); // pokud je překročeno 59 polozek v excelu, dojde ke smazání A11:F70 oblasti na "Example" listu
      idx = 0;
    }
  }
}
```

Kód 13 - Arduino, PC, komunikace s aplikací Excel

Spuštěním připraveného excel souboru Arduino_Excel_21.xls¹¹ byl uživatel vyzván o potvrzení maker.

V excel souboru je skryté menu pro nastavení COM portu a spuštění komunikace, vyvolá se stiskem CTRL + A.

Při zapnuté komunikaci je vidět, že Arduino úspěšně do Excelu posílá náhodné hodnoty. Pro konkrétní příklad připojení je možné editovat zdrojový kód pro kýžený výsledek. V rámci rozsahu bakalářské práce je tato úloha splněna. [36] [37]

⁹ <http://www.robertovalgolio.com/sistemi-programmi/arduino-excel>

¹⁰ Dokumenty\Arduino\Arduino_Excel_21\

¹¹ Dokumenty\Arduo_Excel\

4.2.10 Raspberry Pi → PC + Android

- příkazem `sudo apt-get install python-webpy` došlo k instalaci webpy serveru.
- vytvoření skriptu `server.php` v Raspberry Pi:

```
#!/usr/bin/python
import web
urls = (
    '/', 'index'
)
class index:
    def GET(self):
        return "Ahoj svete!"
if __name__ == "__main__":
    app = web.application(urls, globals())
    app.run()
```

Kód 14 – Raspberry Pi, úloha č.10, web server

- Oprávnění skriptu bylo nastaveno: `chmod +x server.py`
- po spuštění skriptu zadáme v PC nebo v Android zařízení do internetového prohlížeče IP adresu Raspberry Pi a číslo portu, např.: `192.168.0.10:8080`. Po spuštění stránky se na webovské stránce zobrazí text `Ahoj svete!`. [38]

Řešení problému:

- po spuštění skriptu se objevila chyba – `no module web`. Pro odstranění této chyby jsem zadal příkazy `sudo easy_install web.py`; `pip install lpthw.web`

4.2.11 Praktická úloha – kombinace: snímače → Arduino → Raspberry Pi → PC

Přes veškerou snahu se nepodařilo vyřešit problematiku ovládání Arduino zařízení přes Raspberry Pi z PHP stránky zobrazenou na PC. Testoval jsem různé postupy, komunikace mezi Arduino a Raspberry Pi fungovala bezchybně, PHP server Apache 2 nedokázal spustit python skript, který ovládal GPIO piny, ani neproběhla komunikace přes příkazy `i2cset` a `i2cget`, které jsou implementované v knihovně nainstalované v úloze 4.2.6 I2C sběrnice.

Byla zvolena alternativní úloha: Meteostanice s nízkou spotřebou:

Použité komponenty:

- Arduino Pro Mini
- Nokia 5110 LCD displej
- DHT22 snímač teploty
- BMP180 snímač tlaku
- fotorezistor

Zapojení dle schématu:

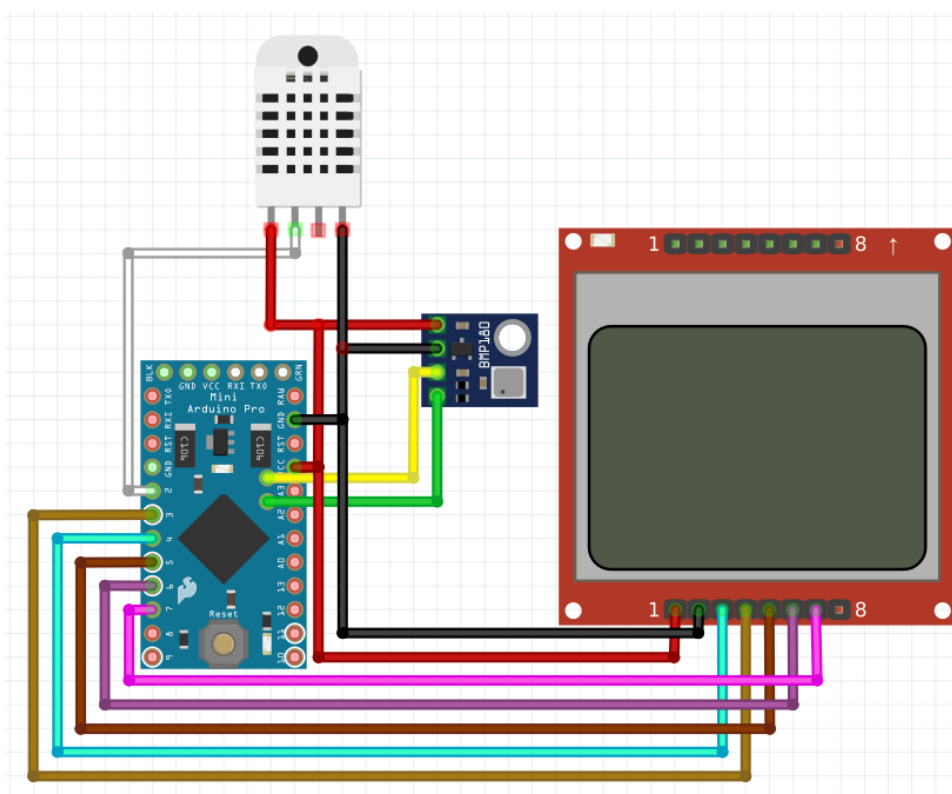


Schéma 22 – Arduino, BMP180 + Nokia 5110 LCD

Pro korektní fungování zdrojového kódu je nutné nahrát následující knihovny:

- LowPower¹²
- LCD5110¹³
- BMP180¹⁴

Kód pro Arduino IDE:

```
#include "DHT.h"
#include <SFE_BMP180.h>
#include <Wire.h>
#include <LCD5110_Graph.h>
#include "LowPower.h"
#define ALTITUDE 370.0 //nadmorská vyska umístění
#define DHTPIN 4 //pin pro připojení DHT22 snímače
#define DHTTYPE DHT22
LCD5110 lcd(5,6,7,9,8);
int LDRpin = 1;
int LDRpowerpin = 3;
SFE_BMP180 pressure;
extern unsigned char SmallFont[];
extern unsigned char TinyFont[];
float temperature;
int lightIntensity = 0;
DHT dht(DHTPIN, DHTTYPE);
void setup(void) {
  pinMode(LDRpowerpin, OUTPUT);
  lcd.InitLCD();
  lcd.setContrast(55);
  lcd.setFont(TinyFont);
  lcd.print("Weather Station v0.4",CENTER,0);
  lcd.update();
  dht.begin();
  pressure.begin();
}
void loop() {
  float humidity, pressure;
  digitalWrite(LDRpowerpin, HIGH);
  delay(100);
  lightIntensity = readLightIntensity();
  digitalWrite(LDRpowerpin, LOW);
  //vypnout/zapnout LCD
  if(lightIntensity<10)
  {
    lcd.enableSleep();
    sleepForTwoMinutes();
  }
  else {
    lcd.disableSleep();
    humidity = dht.readHumidity();
    pressure = readPressure();
    temperature = dht.readTemperature();
    delay(2000);
    lcd.clrScr();
    char tempF[6];
    char humF[6];
    char pressF[7];
    dtostrf(temperature, 5, 1, tempF);
    dtostrf(humidity, 5, 1, humF);
    dtostrf(pressure, 5, 1, pressF);

    lcd.setFont(TinyFont);
    lcd.print("TEMP",LEFT+17,0);
    lcd.print("HUM",LEFT+62,0);
    lcd.print("PRESSURE",CENTER,30);
    lcd.setFont(SmallFont);
    lcd.print(tempF,LEFT,10);
    lcd.print("-C",LEFT+30,10);
    lcd.print(humF,LEFT+45,10); //vypis vlhkosti
    lcd.print("%",RIGHT,10);
    lcd.print(pressF,12,40); //vypis tlaku
    lcd.print(" hPa",48,40);
    lcd.update();
    sleepForTwoMinutes();
  }
}
void sleepForTwoMinutes()
{
  for(int i=0;i<15;i++)
  LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
}
int readLightIntensity()
{
  int intensity = analogRead(LDRpin);
  intensity = map(intensity, 1050, 20, 0, 100);
  return intensity;
}
float readPressure()
{
  char status;
  double T,P,p0,a;
  status = pressure.startTemperature();
  if (status != 0)
  {
    delay(status);
    status = pressure.getTemperature(T);
    if (status != 0)
    {
      status = pressure.startPressure(3);
      if (status != 0)
      {
        delay(status);
        status = pressure.getPressure(P,T);
        if (status != 0)
        {
          p0 = pressure.seaLevel(P,ALTITUDE);
          return p0;
        }
      }
    }
  }
}
```

Kód 15- Arduino, úloha č. 11, převzato a upraveno z [39] s využitím Arduino IDE online

Po nahrání kódu a zapojení všech komponent dle schématu výše máme funkční obvod se spotřebou cca 4 mA. Tato aplikace je tedy vhodná pro napájení externí baterií. Např. se 4x AA bateriemi 1,5V vydrží fungovat minimálně 4 měsíce.

Úlohu lze modifikovat a upravit – např. přidat Raspberry Pi pro vzdálený přístup. Tato úprava však přesahuje rozsah této BP. [39]

¹² <https://github.com/rocketscream/Low-Power>

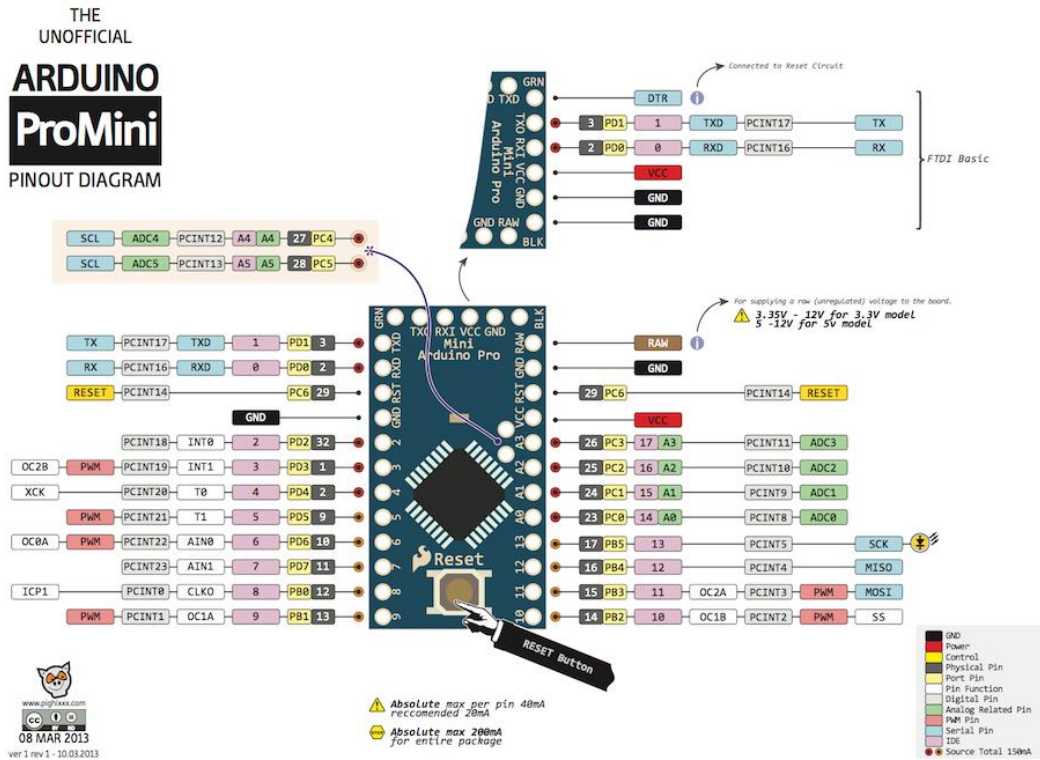
¹³ <http://www.rinkydinkelectronics.com/library.php?id=48>

¹⁴ https://github.com/LowPowerLab/SFE_BMP180

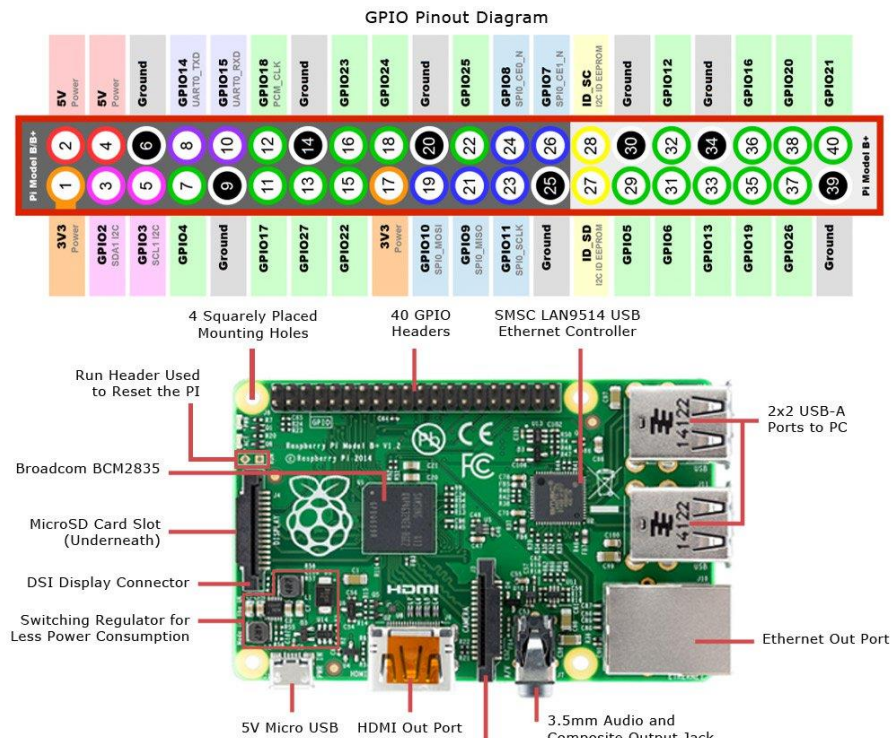
5 Zhodnocení a závěr

Cílem BP bylo formou úloh seznámit zájemce s problematikou využití Raspberry Pi a Arduina pro konstrukce elektronických zařízení. Čtenář po přečtení BP by měl chápat základní rozdíly mezi jednotlivými rozšiřujícími moduly. Zájemce o problematiku by měl dokázat určit jaký modul je vhodný pro zvolenou aplikaci. Měl by získat přehled o možnostech komunikace mezi moduly Raspberry Pi 3 a Arduino Pro Mini. V průběhu vypracování jsem se setkal s řadou drobných komplikací, které se po intenzivním hledání v dostupných zdrojích podařilo vyřešit. Drobné odchylky realizace od zadání byly způsobeny technickým omezením použitých zařízení. Popis SW vlastností je splněn v rámci dílčích kapitol.

6 Přílohy



Obrázek 5 – Arduino Pro Mini, popis pinů [47]



Obrázek 6 – Raspberry Pi 3, popis pinů [46]

6.1 Pracovní listy

Pracovní návody jsou vázány konfigurací HW a SW uvedenou v praktické části bakalářské práce, pro jiné varianty mohou nastat odlišnosti v nastavení či problémy s kompatibilitou. Pro totožnou interpretaci všech praktických úloh je tak nutné použít stejný HW i SW.

6.1.1 Pracovní list č. 1

Pracovní úkoly

- Nainstalovat do Raspberry Pi operační systém Raspbian
- V aplikaci Geany vytvořit první skript, který vypíše „Zdravim python“ na obrazovku.

Teorie

Odkaz na stažení NOOBS¹⁵. Raspberry Pi by mělo mít zajištěný přístup na internet (ethernet, nebo wifi). Pro instalaci systému použijte microSD kartu nejméně třídy 10, ideálně UHS 1.

příkaz	vysvětlení
sudo	provede příkaz jako správce
apt-get	získání aplikace
update	aktualizace systému
install	instalace aplikace
sudo apt-get update	celý příkaz pro provedení aktualizace - doporučuji provést před každou úlohou
sudo apt-get install ...	celý příkaz pro získání a instalaci nové aplikace, název doplňte místo ...
print	funkce pro výpis textu v konzoli
sudo reboot	restartuje zařízení

Tabulka 2 - seznam použitých příkazů

Použité přístroje a pomůcky

- Raspberry Pi 3
- PC

Pracovní postup

- stáhněte NOOBS balíček
- přehrajte obraz na microSD kartu
- nainstalujte Raspbian OS
- v aplikaci Geany vytvořte první skript, ve kterém funkcí print vypíšeme na obrazovku Zdravim Python!

¹⁵ <https://www.raspberrypi.org/downloads/noobs/>

6.1.2 Pracovní list č. 2

Pracovní úkoly

- Zapojte FTDI konvertor
- Nainstalujte a nastavte SW Arduino IDE
- Naprogramujte první program, který pravidelně po 1 sekundě bude vypisovat na sériový monitor test.

Teorie

Odkaz na stažení Arduino IDE¹⁶.

Nastavení Arduino IDE:

- Nástroje → Vývojová deska → Arduino Pro or Pro Mini
- Nástroje → Procesor → ATmega328 (5V, 16MHz)
- Nástroje → Port → COM4
- Nástroje → Programátor → AVRISP mkII

příkaz	vysvětlení
void setup()	funkce pro příkazy provedené po zapojení napájení pouze jednou
void loop()	funkce pro příkazy prováděné stále dokola, běží zpravidla po předchozí funkci
Seriál.print(„test“)	Vypíše test na sériový monitor
delay(XX)	Pozastavení programu na XX milisekund

Tabulka 3 – seznam použitých příkazů

Schéma zapojení

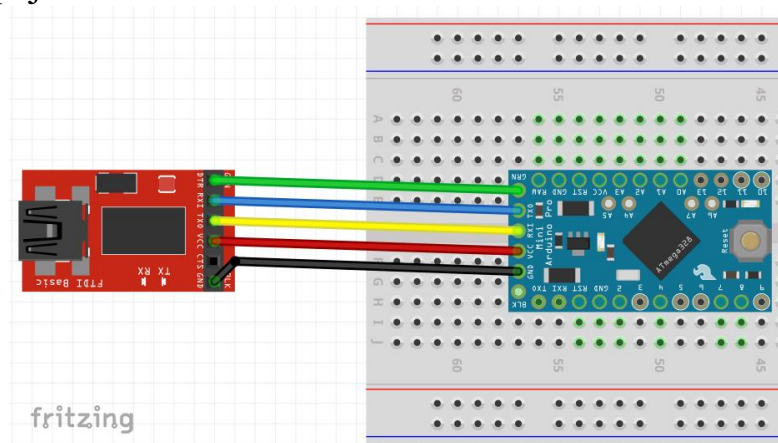


Schéma 11 - schéma zapojení vodičů mezi FTDI konvertorem a Arduino Pro Mini

¹⁶ <https://www.arduino.cc/en/Main/Software>

Použité přístroje a pomůcky
FTDI konvertor, Arduino Pro Mini

6.1.3 Pracovní list č. 3

Pracovní úkoly

Zapojte tlačítko a naprogramujte jeho funkci pro:

a) Arduino

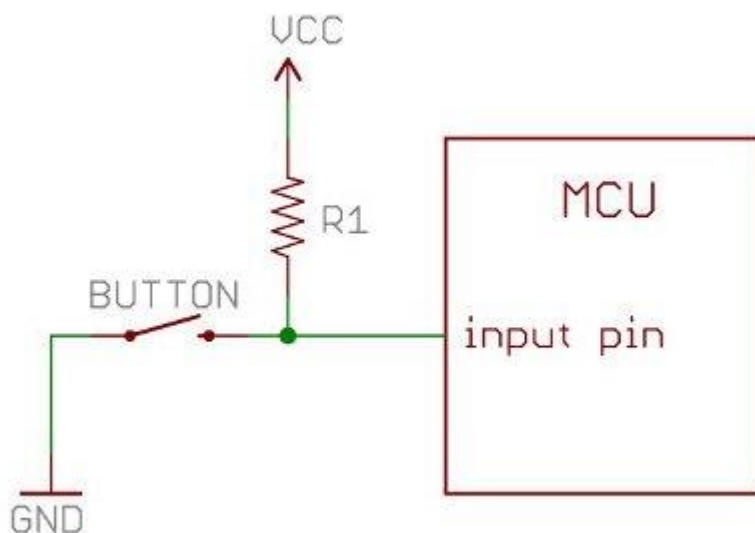
Po stisku tlačítka dojde ke změně stavu integrované LED diody na pinu č. 13.

b) Raspberry Pi

Po stisku tlačítka bude v příkazovém řádku vypsán text „Stisknuto tlačítko“.

Teorie

Pozor při nastavování vstupů na „plovoucí pin“, použijte následné zapojení s odporem PULL-UP.



Obrázek 7 - Zapojení pull-up rezistoru [6]

a) Arduino

příkaz	vysvětlení
pinMode(XX,YY)	nastavení pinu XX jako YY (např. INPUT_PULLUP)
digitalWrite()	nastavení hodnoty na výstupního pinu
const int	deklarace konstanty typu integer, její hodnota je číslo pinu
int	deklarace proměnné typu integer, její hodnota je nastavení proměnné
digitalRead()	načte hodnotu ze vstupního pinu
if .. else	funkce když (podmínka), jinak ..

Tabulka 4 – seznam použitých příkazů pro Arduino

b) Raspberry Pi

příkaz	vysvětlení
<code>import RPi.GPIO as GPIO</code>	načtení knihovny pro ovládání GPIO pinů
<code>import time</code>	načtení knihovny pro ovládání časovačů
<code>GPIO.setmode</code>	nastavení módu GPIO pinů
<code>GPIO.setup</code>	nastavení vstupů/výstupů na konkrétních pinech, definování připojení pull_up rezistorů
<code>while</code>	funkce smyčky, využívá se pro běh hlavního programu

Tabulka 5 - seznam použitých příkazů pro Raspberry Pi

Schéma zapojení

a) Arduino

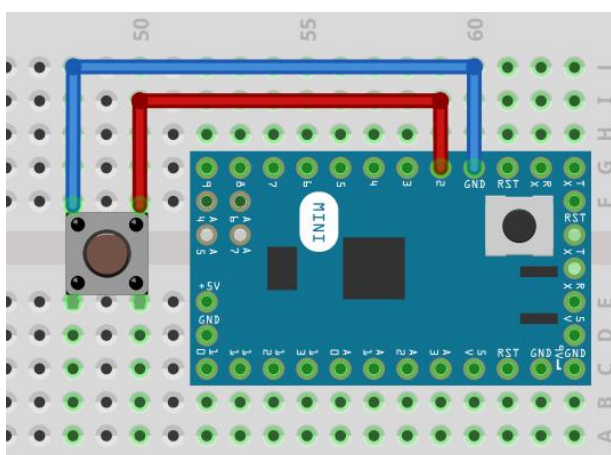


Schéma 12 – Arduino, tlačítko

b) Raspberry Pi

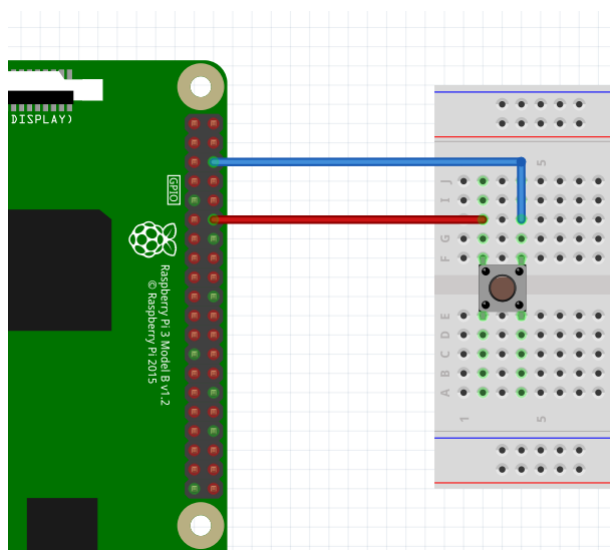


Schéma 13 – Raspberry Pi, tlačítko

Použité přístroje a pomůcky

- a) Arduino
 - Arduino Pro Mini
 - FTDI konvertor
 - Tlačítko
- b) Raspberry Pi
 - Raspberry Pi
 - Tlačítko

Pracovní postup

- a) Arduino
 - zapojte tlačítko k Arduino Pro Mini
 - naprogramujte program, který po stisknutí tlačítka změní stav LED diody
- b) Raspberry Pi
 - zapojte tlačítko k Raspberry Pi GPIO pinům
 - za pomoci znalostí z předchozích úloh a tabulky příkazů v teoretické části pracovního listu, který po stisknutí tlačítka vypíše na konzoli `Tlacitko stisknuto`.

6.1.4 Pracovní list č. 4

Pracovní úkoly

Zapojte obvod dle přiloženého schématu

Rozblikujte LED diodu frekvencí 1 Hz.

Teorie

a) Raspberry Pi

příkaz	vysvětlení
<code>import RPi.GPIO as GPIO</code>	načtení knihovny pro ovládání GPIO pinů
<code>import time</code>	načtení knihovny pro ovládání časovačů
<code>GPIO.setmode</code>	nastavení módu GPIO pinů
<code>GPIO.setup</code>	nastavení vstupů/výstupů na konkrétních pinech, definování připojení pull_up rezistorů
<code>while</code>	funkce smyčky, využívá se pro běh hlavního programu
<code>GPIO.output</code>	nastavení logické hodnoty na výstupním pinu

Tabulka 6 - seznam použitých příkazů pro Raspberry Pi

Schéma zapojení

a) Arduino

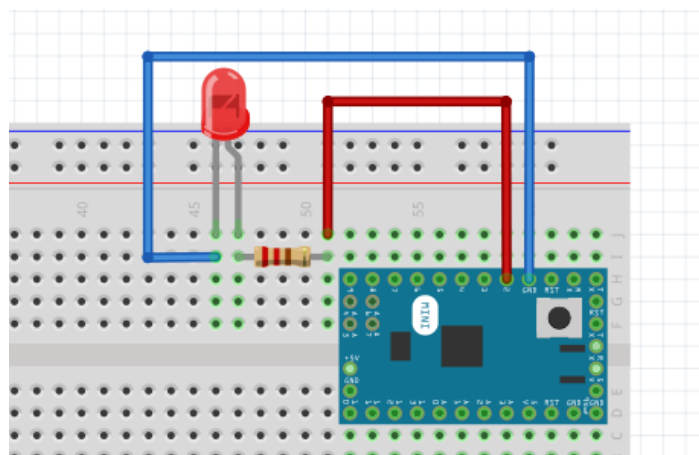


Schéma 14 – Arduino, LED dioda

b) Raspberry Pi

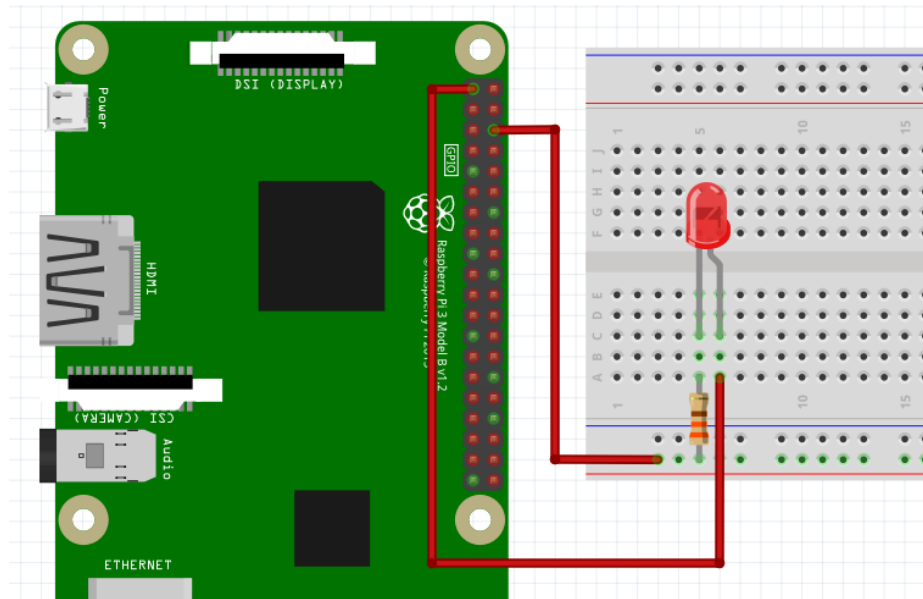


Schéma 15 – Raspberry Pi, LED dioda

Použité přístroje a pomůcky

- Raspberry Pi 3
- Arduino Pro Mini
- LED dioda
- Odpor 330 Ω

Pracovní postup

- Zapojte LED diodu dle vybraného schématu
- Nastavte pin jako výstup
- Naprogramujte výstup tak, aby se dioda rozblíkala frekvencí 1 Hz
- Porovnejte blikání obou modulů mezi sebou a diskutujte případné výsledky pozorování.

6.1.5 Pracovní list č. 5

Pracovní úkoly

V tomto úkolu zprovozníme sériové spojení mezi Raspberry Pi a Arduino Pro Mini.

a) USB

Pomocí kabelu USB -> miniUSB obstarajte sériovou komunikaci mezi Raspberry Pi a Arduino.

b) UART

Spojte oba moduly kabely na v TX, RX a GND pinech, kvůli rozdílným napětím využijte převodních logických úrovní.

Použité přístroje a pomůcky

a) USB

- LED dioda
- 330 Ω odpor
- tlačítko

b) UART

- převodník napěťových úrovní

Teorie

Převodník logických úrovní snižuje zajišťuje správnou napěťovou úroveň zaslano mezi oběma moduly.

a) USB

Arduino příkazy	vysvětlení
const int	deklarace konstanty typu integer, její hodnota je číslo pinu
int	deklarace proměnné typu integer, její hodnota je nastavení proměnné
digitalRead()	načte hodnotu ze vstupního pinu
if .. else	funkce když (podmínka), jinak ..
Try: .. Except: ..	funkce pro vyvolání chybových stavů
from nanpy import ()	import nanpy knihovny
from time import sleep	import sleep knihovny pro funkci spánku
Raspberry Pi příkazy	vysvětlení
GPIO.setup	nastavení vstupů/výstupů na konkrétních pinech, definování připojení pull_up rezistorů
while	funkce smyčky, využívá se pro běh hlavního programu
GPIO.output	nastavení logické hodnoty na výstupním pinu
git clone	zkopírování obsahu odkazu do Raspberry Pi
cp	kopírování souborů

Tabulka 7 – seznam použitých příkazů pro část USB

b) UART

Raspberry Pi příkazy	vysvětlení
Minicom -b 9600 -o -D /dev/ttyAMA0	Spuštění Minicom aplikace – nastavení parametrů komunikace
Arduino Pro Mini příkazy	vysvětlení
Serial.begin (9600)	zahájení sériové komunikace o určité rychlosti
Serial.available()	kontrola, zda je sériová komunikace k dispozici
Serial.print(“”“”)	vytiskne po sériové lince požadovaný řetězec
DEC	formát tisknutého textu

Tabulka 8 - seznam použitých příkazů pro část UART

Schéma zapojení

a) USB

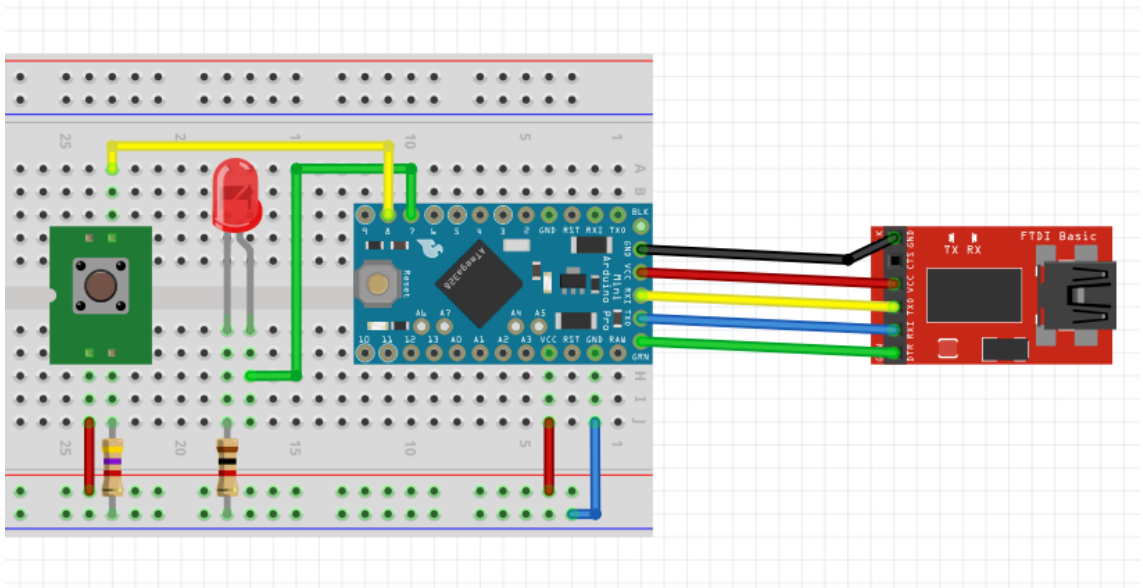


Schéma 16- zapojení tlačítka a diody pro komunikaci přes Raspberry Pi po USB

b) UART

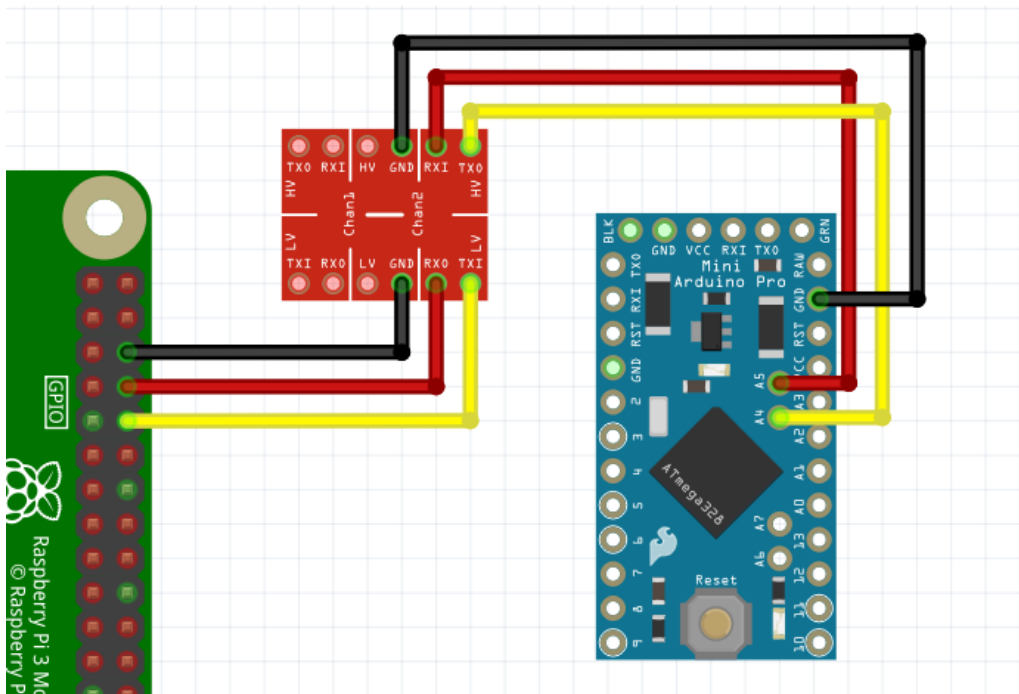


Schéma 17 – Arduino, Raspberry Pi, UART + převodník úrovní

Pracovní postup

a) USB

- nainstalujte Arduino IDE a zkopírujte z odkazů nanpy a nanpy-firmware
- spusťte Arduino IDE a otevřete stažený nanpy-firmware → nanpy
- nastavte aplikaci Arduino IDE (více info v teorii)
- naprogramujte komunikaci a spusťte python skript

b) UART

- deaktivujte sériovou komunikaci s konzolí v Raspbianu
- nainstalujte aplikaci minicom
- v Arduino Pro Mini vytvořte skript, který bude načítat sériovou komunikaci a vypisovat ji zpět přes sériovou linku
- proveďte test odesláním znaku v Minicom aplikaci v Raspberry Pi 3. Pokud Arduino pošle packet zpět, úloha je úspěšně splněná

6.1.6 Pracovní list č. 6

Pracovní úkoly

Vytvořte komunikaci mezi Raspberry Pi jako Mater a Arduino Pro Mini jako Slave použitím sběrnice I2C.

Raspberry Pi odešle po I2C sběrnici číslo, které Arduino načte na vstupu a odešle jej zpátky do Raspberry Pi.

Teorie

Aktivace I²C sběrnice: Start → Volby → Raspberry Pi Configuration → Interfaces → I2C → Enabled

a) Arduino

příkaz	vysvětlení
#include <Wire.h>	zahrnutí knihovny Wire
Serial.begin()	Zahájení seriové linky pro komunikaci
Wire.begin()	Nastavení slave módu a adresy slave zařízení (např. 0x04)
Wire.onReceive(receiveData)	nastavení manipulace s daty
Wire.onRequest(sendData)	nastavení manipulace s daty

Tabulka 9 – seznam použitých příkazů pro Arduino

b) Raspberry Pi

příkaz	vysvětlení
sudo apt-get install i2c-tools	přidání i2c tools
sudo adduser pi i2c	přidání oprávnění pro uživatele pi
sudo apt-get install python-smbus	přidání smbus knihovny do pythonu
import smbus	import knihovny smbus pro obsluhu sériové komunikace
bus.write_byte()	zápis dat po sběrnici
bus.read_byte()	čtení dat ze sběrnice
i2cdetect -y 1	skenování dostupných zařízení na I2C sběrnici

Tabulka 10 - seznam použitých příkazů pro Raspberry Pi

Schéma zapojení

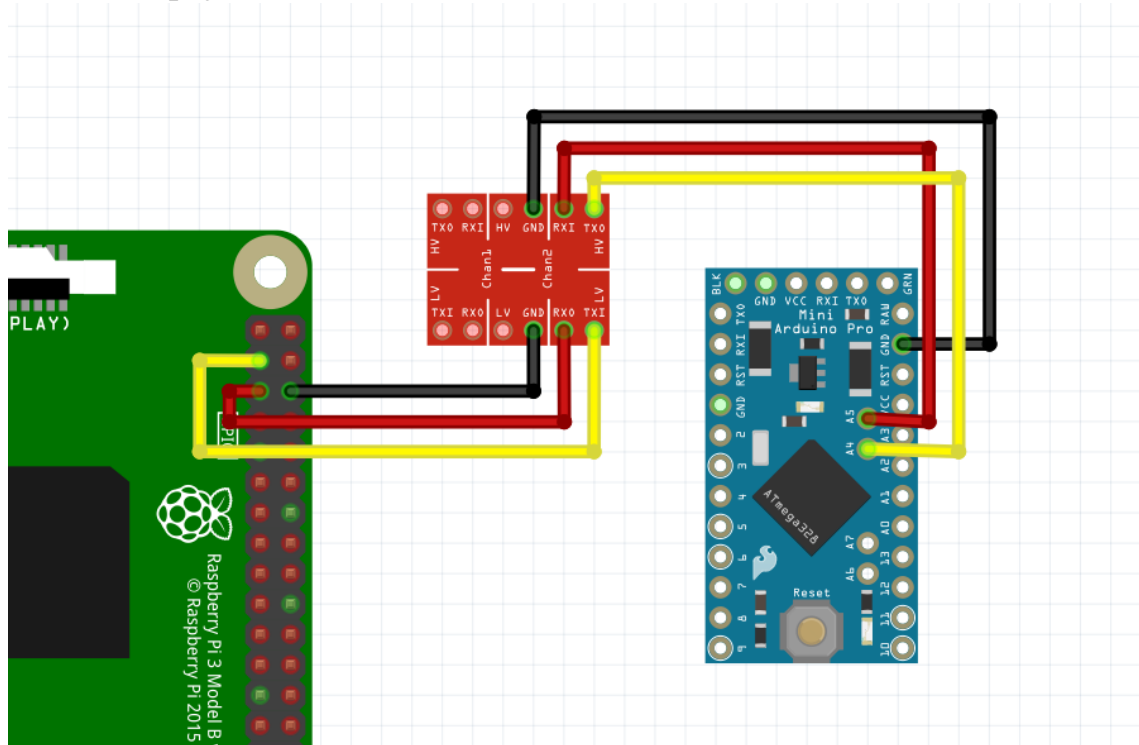


Schéma 18 – Arduino, Raspberry Pi, I²C + převodník úrovní

Použité přístroje a pomůcky

- Arduino Pro Mini
- Raspberry Pi 3
- převodník logických úrovní

Pracovní postup

- v Raspbianu aktivujte I2C komunikaci
- nainstalujte i2c-tools a python-smbus
- naprogramujte zdrojový kód k ovládní i2c komunikace na straně Raspberry Pi, aby komunikovalo jako Mater zařízení
- naprogramujte zdrojový kód pro Arduino Pro Mini, adresa zařízení bude 0x04.
- zapojte dle přiloženého schématu
- nahrajte program do Arduino Pro Mini a spusťte python skript na Raspberry Pi

6.1.7 Pracovní list č. 7

Pracovní úkoly

- za pomoci knihovny SPIDEV naprogramujte komunikaci na Raspberry Pi¹⁷
- naprogramujte komunikaci přes SPI na Arduino Pro Mini¹⁸
- spojte zařízení a potvrďte funkčnost komunikace mezi jednotlivými moduly

Schéma zapojení

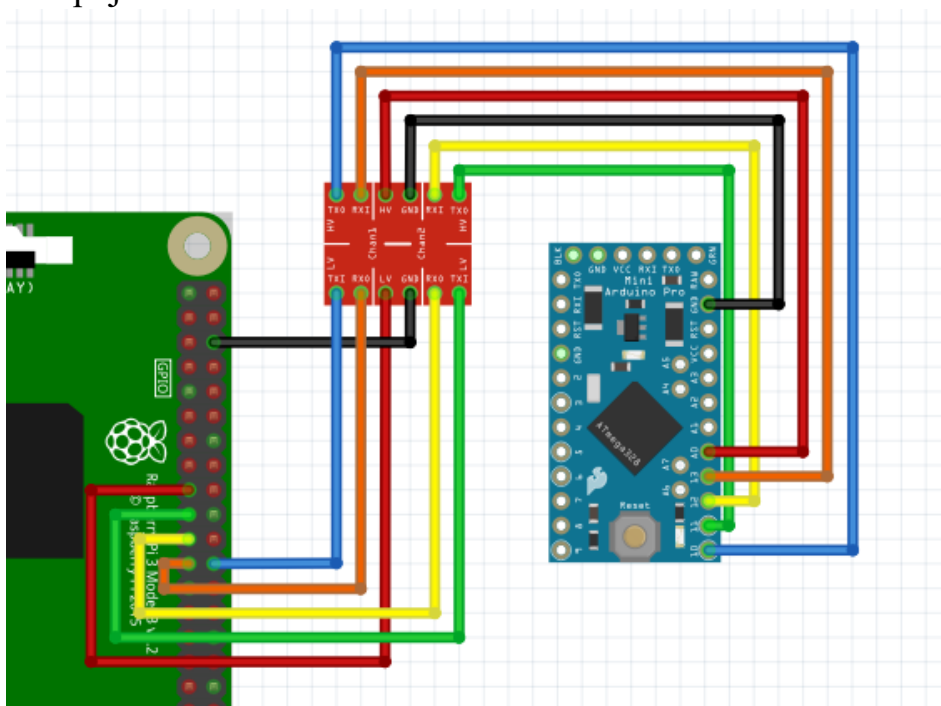


Schéma 19 – Arduino, Raspberry Pi, SPI + převodník úrovní

Použité přístroje a pomůcky

- Raspberry Pi 3
- Arduino Mini Pro
- Převodník logických úrovní

Pracovní postup

- Zapojte obvod dle přiloženého schématu
- Naprogramujte Raspberry Pi 3, použijte při tom knihovnu SPIDEV a programovací jazyk Python
- Naprogramujte Arduino Mini Pro

¹⁷ <https://github.com/doceme/py-spidev>

¹⁸ <http://robotics.hobbizine.com/raspiduino.html>

- Spustíte program a ověříte komunikaci

6.1.8 Pracovní list č. 8

Pracovní úkoly

Použijte Arduino Ethernet Shield pro připojení Arduino Pro Mini k Raspberry Pi pomocí Ethernet kabelu.

Naprogramujte komunikaci na obou modulech a ověříte funkčnost

Teorie

a) Arduino

- Použité knihovny: SPI.h; Ethernet.h

Příkaz	vysvětlení
EthernetServer	nastavení portu pro komunikaci
Ethernet.begin(mac, ip)	Inicializace Ethernet Shield

Tabulka 11 - seznam použitých příkazů pro Arduino Mini Pro

příkaz	vysvětlení
Socket(AF_INTE, SOCK_STREAM)	Vytvoření socketu pro komunikaci
connect((host,port))	Připojení pomocí ethernet portu
Select.select([s],[],[],timeout)	Parsování zasílaných informací

Tabulka 12 - seznam použitých příkazů pro Raspberry Pi

Schéma zapojení

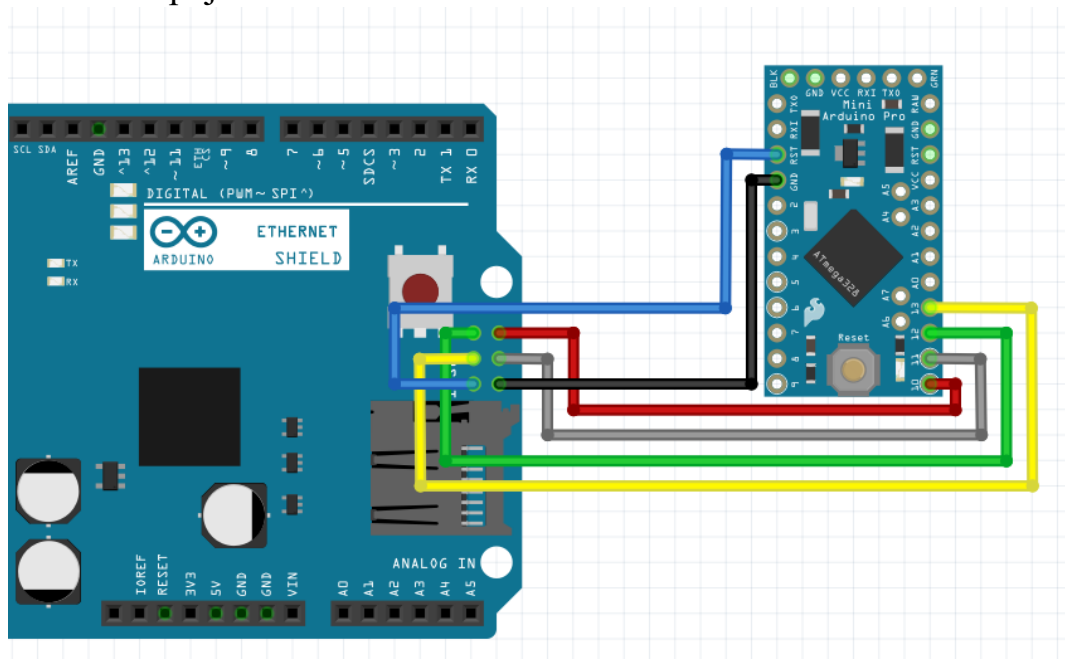


Schéma 20 - SPI zapojení přes převodník logických úrovní

Použité přístroje a pomůcky

- Raspberry Pi 3
- Arduino Mini Pro
- Arduino Ethernet Shield

Pracovní postup

- připojte Ethernet Shield k Arduino Pro Mini dle schématu
- spojte Raspberry Pi 3 a Arduino Pro Mini Ethernet kabelem
- zjistěte si potřebné IP a MAC adresy
- naprogramujte komunikaci na obou zařízeních

6.1.9 Pracovní list č. 9

Pracovní úkoly

Importujte do excel tabulky hodnotu zaslanou od Arduino Pro Mini.

Teorie

Jde o speciální příklad komunikace, nastudujte si kód dostupný v balíku na stránce¹⁹.

Použité přístroje a pomůcky

- Arduino Pro Mini
- PC
- MS Office 2013
- Balíček souborů pro komunikaci mezi Arduino a MS Excel

Pracovní postup

- stáhněte balík souborů pro komunikaci mezi MS Excel a Arduino
- upravte a importujte kód do Arduino Pro Mini
- spusťte a dle návodu nakonfigurujte Excel souboru Arduino_Excel_21.xls²⁰
- zkontrolujte funkčnost
- volitelné: připojte snímač a upravte komunikaci pro zasílání konkrétních hodnot

¹⁹ <http://www.robertovalgolio.com/sistemi-programmi/arduino-excel>

²⁰ Dokumenty\Arduino\Arduino_Excel_21\

6.1.10 Pracovní list č. 10

Pracovní úkoly

Nainstalujte web.py server, který bude možné zobrazit na Android zařízení a PC.

Vypište na novou webovskou stránku „Ahoj Svete!“

Teorie

Je nutné mít Raspberry Pi připojené přes Ethernet nebo WI-FI k síti, zjistit IP adresu zařízení – na tu se bude přistupovat pro zobrazení textu.

Použitá knihovna web.

příkaz	vysvětlení
class	vytvoření třídy objektů
def GET(self)	Definování funkce GET
app = web.applications(urls, globals())	Deklarace adres a oddělovačů
app.run()	Spuštění webového python serveru
chmod +x server.py	Nastaví oprávnění, aby byl skript spustitelný

Tabulka 13 - seznam použitých příkazů pro Raspberry Pi

Použité přístroje a pomůcky

- Raspberry Pi 3

Pracovní postup

- Nainstalujte webpy server
- Vytvořte skript pro zobrazení „Ahoj Svete!“ na webové stránce
- Upravte oprávnění skriptu
- Po spuštění skriptu zkontrolujte funkčnost na IP adrese Raspberry Pi 3 na portu 8080.

6.1.11 Pracovní list č. 11

Pracovní úkoly

- vytvořte jednoduchou meteostanici se snímáním barometrického tlaku, teploty a vlhkosti

Schéma zapojení

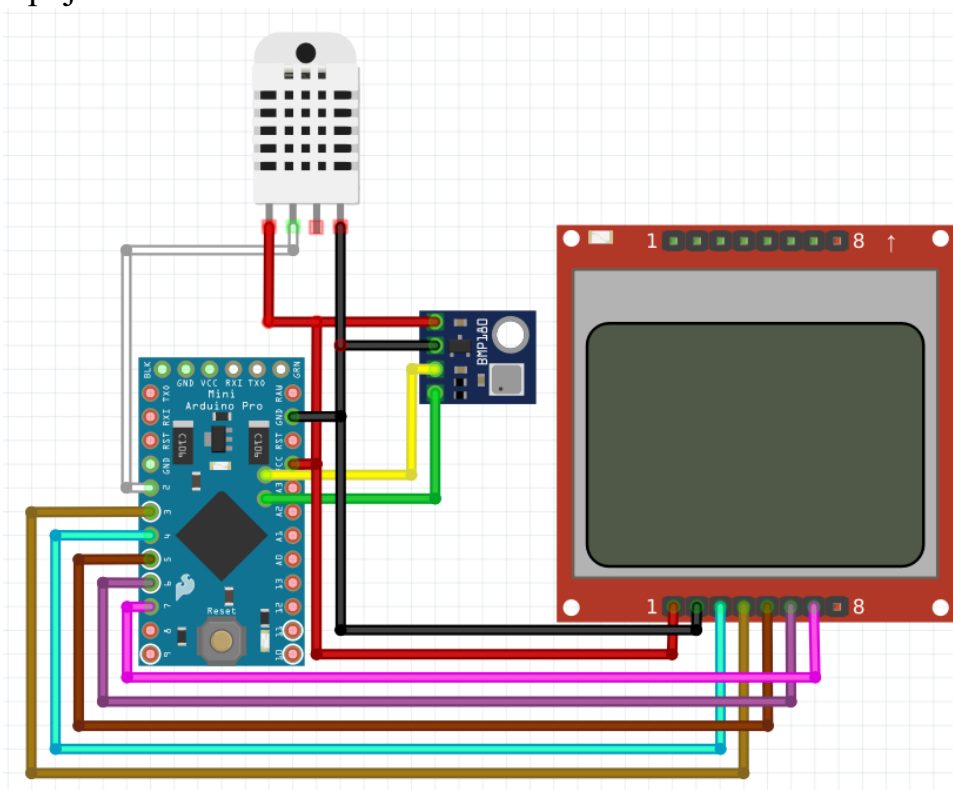


Schéma 21 – Arduino, BMP180 + Nokia 5110 LCD

Použité přístroje a pomůcky

- Raspberry Pi 3
- Arduino Mini Pro
- Nokia 5110 displej
- BMP180 – snímač barometrického tlaku
- DHT22 snímač teploty a vlhkosti

Pracovní postup

- vzhledem ke komplexnosti této úlohy prosím postupujte dle návodu na stránce²¹.

²¹ <http://educ8s.tv/arduino-weather-station-4/>

6.2 Seznam použitých zkratk a pojmů

SSH – secure shell – vzdálený přístup do konzole OS
VNC – virtual network computing – vzdálený přístup do grafického prostředí OS
OS – operační systém
SPI – serial peripheral interface – sériové periferní rozhraní
UART – universal asynchronous receiver – transmitter
USART – universal synchronous receiver – transmitter
SRAM – static access random memory – statická paměť
I²C – inter-integrated circuit – integrovaná sběrnice
HW – hardware
SW – software
FW – firmware
IoT – Internet of Things – internet věcí
ARM – advanced RISC machine – architektura procesorů
RISC – reduced instruction set computing – omezená instrukční sada
PC – personal computer - počítač
AVR-C – programovací jazyk pro AVR mikroprocesory
IDE – integrated development environment – programovací prostředí
USB – universal serial bus – univerzální sériová sběrnice
HID – human interface device – většinou zařízení, které obsluhuje člověk
FTDI – future technology devices international – konvertor RS232 a TTL komunikaci
TTL – transistor – transistor logic
RAW – surový napájecí pin (až 12 V vstup pro Arduino Pro Mini)
VCC – voltage common cathode – napětí společná katoda
GND – ground - zem
EEPROM – electrically erasable programmable read-only memory – elektronicky vymazatelná paměť pouze pro čtení
RX – received data – obdržená data
TX – transmitted data – odeslaná data
RXI – input RX – vstup RX
TXI – input TX – vstup TX
TXO – output RX – výstup RX
RXO – output TX – výstup TX
SS – serial slave
MOSI – Master out Slave in – Master zařízení ven, Slave zařízení dovnitř
MISO – Master in Slave out – Master zařízení dovnitř, Slave zařízení ven
SCK – synchronous clock – synchronní hodiny
SDA – synchronous data – synchronní data
SCL – synchronous clock – synchronní hodiny
TWI – two wire interface – dvoudrátové rozhraní
CPU – central processing unit – procesor
GPU – graphic processing unit – grafický procesor
RAM – random access memory – dynamická paměť, po vypnutí se maže
GPIO – general purpose input/output – vstupy/výstupy pro obecné použití
CSI – camera serial interface
DSI – display serial interface
microSD – micro secure digital paměťová karta
rev. – revize
min. – minimální/minimálně
RTC – Reálné hodiny času
SPCR – SPI Control Register
SPSR – SPI Status Register
SPDR – SPI Data Register [40]
BP – bakalářská práce
mA – miliampér

6.3 Seznam tabulek

Tabulka 1 - ukázka různých verzí Arduino mikrokontrolérů [4].....	10
Tabulka 2 - seznam použitých příkazů	49
Tabulka 3 – seznam použitých příkazů.....	50
Tabulka 4 – seznam použitých příkazů pro Arduino	52
Tabulka 5 - seznam použitých příkazů pro Raspberry Pi	53
Tabulka 6 - seznam použitých příkazů pro Raspberry Pi	55
Tabulka 7 – seznam použitých příkazů pro část USB	58
Tabulka 8 - seznam použitých příkazů pro část UART	58
Tabulka 9 – seznam použitých příkazů pro Arduino	61
Tabulka 10 - seznam použitých příkazů pro Raspberry Pi	61
Tabulka 11 - seznam použitých příkazů pro Arduino Mini Pro	64
Tabulka 12 - seznam použitých příkazů pro Raspberry Pi	64
Tabulka 13 - seznam použitých příkazů pro Raspberry Pi	67

6.4 Seznam zdrojových kódů

Kód 1 – Arduino, úloha č. 1, první program, upraveno a převzato z [25].....	25
Kód 2 – Arduino, úloha č. 2, vstupy, upraveno a převzato z [25]	27
Kód 3 – Raspberry Pi, úloha č. 3, vstupy	28
Kód 4 – Arduino, úloha č. 4, výstupy, upraveno a převzato z [25]	29
Kód 5 – Raspberry Pi, úloha č. 4, výstupy	30
Kód 6 – Raspberry Pi, úloha č. 5, sériová komunikace přes USB.....	32
Kód 7 – Arduino, úloha č. 5, UART.....	34
Kód 8 – Arduino, úloha č. 6, I ² C	35
Kód 9 – Raspberry Pi, úloha č. 6, I ² C.....	36
Kód 10 – Arduino, úloha č. 7, SPI komunikace	37
Kód 11 – Arduino, úloha č. 8, ethernet.....	40
Kód 12 – Raspberry Pi, úloha č. 8, ethernet	41
Kód 13 - Arduino, PC, komunikace s aplikací Excel	42
Kód 14 – Raspberry Pi, úloha č.10, web server.....	43
Kód 15- Arduino, úloha č. 11, převzato a upraveno z [39] s využitím Arduino IDE online.....	45

Není-li uvedeno jinak, pro vkládání zdrojového kódu byla použita webová stránka²².

²² <http://markup.su>

6.5 Seznam schémat

Schéma 1 – Arduino, FTDI konvertor	25
Schéma 2 – Arduino, tlačítko	26
Schéma 3 – Raspberry Pi, tlačítko	28
Schéma 4 – Arduino, LED dioda.....	29
Schéma 5 – Raspberry Pi, LED dioda	30
Schéma 6 – Arduino, LED dioda + tlačítko.....	31
Schéma 7 – Arduino, Raspberry Pi, UART + převodník úrovní.....	33
Schéma 8 – Arduino, Raspberry Pi, I ² C + převodník úrovní	36
Schéma 9 – Arduino, Raspberry Pi, SPI + převodník úrovní.....	38
Schéma 10 – Arduino, ethernet shield	39
Schéma 11 - schéma zapojení vodičů mezi FTDI konvertorem a Arduino Pro Mini	50
Schéma 12 – Arduino, tlačítko	53
Schéma 13 – Raspberry Pi, tlačítko	53
Schéma 14 – Arduino, LED dioda.....	55
Schéma 15 – Raspberry Pi, LED dioda	56
Schéma 16- zapojení tlačítka a diody pro komunikaci přes Raspberry Pi po USB	59
Schéma 17 – Arduino, Raspberry Pi, UART + převodník úrovní.....	59
Schéma 18 – Arduino, Raspberry Pi, I ² C + převodník úrovní	62
Schéma 19 – Arduino, Raspberry Pi, SPI + převodník úrovní.....	63
Schéma 20 - SPI zapojení přes převodník logických úrovní	65
Schéma 21 - Arduino, BMP180 + Nokia 5110 LCD.....	68
Schéma 22 - Arduino, BMP180 + Nokia 5110 LCD.....	684

Není-li uvedeno jinak, pro tvorbu schémat byla použita aplikace Fritzing, detailněji popsána v teoretické části BP.

6.6 Seznam obrázků

Obrázek 1 - Arduino Pro Mini [2]	8
Obrázek 2 - Zapojení pull-up rezistoru [6]	12
Obrázek 3 - Raspberry Pi 3 [9]	15
Obrázek 4 - Arduino Ethernet Shield [34]	39
Obrázek 5 – Arduino Pro Mini, popis pinů [47]	47
Obrázek 6 – Raspberry Pi 3, popis pinů [46]	47
Obrázek 7 - Zapojení pull-up rezistoru [6]	52

7 Zdroje

- [1] Co je to Arduino? | Arduino.cz. Arduino.cz - Webový magazín o Arduinu a elektronice [online]. Dostupné z: <https://arduino.cz/co-je-to-arduino/>.
- [2] Using the Arduino Pro Mini 3.3V - learn.sparkfun.com. Learn at SparkFun Electronics - learn.sparkfun.com [online]. Dostupné z: <https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v>.
- [3] Arduino - Introduction . Arduino - Home [online]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>.
- [4] Arduino - Products . Arduino - Home [online]. Dostupné z: <https://www.arduino.cc/en/Main/Products>.
- [5] Arduino Pro Mini. Arduino [online]. Dostupné z: <https://store.arduino.cc/arduino-pro-mini>.
- [6] Pull-up Resistors - learn.sparkfun.com. Learn at SparkFun Electronics - learn.sparkfun.com [online]. Dostupné z: <https://learn.sparkfun.com/tutorials/pull-up-resistors>.
- [7] Which Programming Languages Can You Use With Arduino?. MakeUseOf - Technology, Simplified [online]. Copyright © 2017 MakeUseOf. All Rights Reserved. [cit. 06.8.2017]. Dostupné z: <http://www.makeuseof.com/tag/programming-languages-can-use-arduino/>.
- [8] Alternatives to Standard Arduino IDE: Which One To Choose? | Into Robotics. Into Robotics | An effortless way to become a better roboticists [online]. Copyright © 2013 [cit. 13.7.2017]. Dostupné z: <https://www.intorobotics.com/alternatives-standard-ardui>.
- [9] Raspberry Pi 3 Model B - Raspberry Pi. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
- [10] Raspberry Pi 3 je tu, co nabídne? | Svět hardware. Svět hardware | homepage [online]. Copyright © 1998 [cit. 25.10.2017]. Dostupné z: <https://www.svethardware.cz/raspberry-pi-3-je-tu-co-nabidne/41920>.

- [11] Top 10 Programming Languages Ported to the Raspberry Pi - dummies. dummies - Learning Made Easy [online]. Dostupné z: <http://www.dummies.com/computers/raspberry-pi/top-10-programming-languages-ported-to-the-raspberry-pi/>.
- [12] RaspberryPi.cz – miniaturní levný počítač. RaspberryPi.cz – miniaturní levný počítač [online]. Copyright © Jiří Pešek [cit. 20.06.2017]. Dostupné z: <http://raspberrypi.cz/ops4/>.
- [13] Raspberry Pi a GPIO – co nefunguje - Zdroják. Zdroják - o tvorbě webových stránek a aplikací [online]. Dostupné z: <https://www.zdrojak.cz/clanky/raspberry-pi-a-gpio-co-nefunguje/>.
- [14] Arduino a sériová komunikace – uArt.cz. uArt.cz – Electronics, programming and stuff... [online]. Dostupné z: <https://uart.cz/139/arduino-a-seriova-komunikace/>.
- [15] UART (USART) – komunikujte sériově po dvou vodičích – ZAVAVOV. ZAVAVOV – Svět plný modelové železnice, elektrotechniky, bastlení, programování a zábavy [online]. Copyright © 2017 [cit. 12.11.2017]. Dostupné z: <http://zavavov.vzap.eu/cz/elektrotechnika/kom>.
- [16] Externí sériové sběrnice SPI a I²C - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 03.12.2017]. Dostupné z: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>.
- [17] Komunikace po sériové sběrnici I2C - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 18.10.2017]. Dostupné z: <https://www.root.cz/clanky/komunikace-po-seriove-sbernici-isup2supc/>.
- [18] 1-Wire – Wikipedie. [online]. Dostupné z: <https://cs.wikipedia.org/wiki/1-Wire>.
- [19] Bi-Directional Logic Level Converter Hookup Guide - learn.sparkfun.com. Learn at SparkFun Electronics - learn.sparkfun.com [online]. Dostupné z: <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide>.
- [20] A real serial console for your Raspberry Pi | Dave Cheney. Dave Cheney | The acme of foolishness [online]. Dostupné z: <https://dave.cheney.net/2014/01/05/a-real-serial-console-for-your-raspberry-pi>.

- [21] SparkFun RS232 Shifter Board Kit - PRT-00133 - SparkFun Electronics. SparkFun Electronics [online]. Dostupné z: <https://www.sparkfun.com/products/retired/133>.
- [22] Základní struktury jazyka Wiring | Arduino.cz. Arduino.cz - Webový magazín o Arduinu a elektronice [online]. Dostupné z: <https://arduino.cz/zakladni-struktury-jazyka-wiring/>.
- [23] Top 8 IDEs for Programmers, Coders and Beginners on the Raspberry Pi. Java PDF Blog | IDR Solutions [online]. Copyright © IDRSolutions Ltd 2017 [cit. 10.10.2017]. Dostupné z: <https://blog.idrsolutions.com/2014/12/top-8-ides-programmers-coders-beginners-ra>.
- [24] NOOBS - Raspberry Pi Documentation. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/noobs.md>.
- [25] Arduino Button Tutorial: 6 Steps (with Pictures). Instructables - How to make anything [online]. Copyright © 2017 Autodesk, Inc. [cit. 05.11.2017]. Dostupné z: <http://www.instructables.com/id/Arduino-Button-Tutorial/>.
- [26] Pull Up Resistors and Buttons | AddOhms #15 - YouTube. YouTube [online]. Dostupné z: <https://www.youtube.com/watch?v=wxjerCHCEMg>.
- [27] Arduino - BuiltInExamples . Arduino - Home [online]. Dostupné z: <https://www.arduino.cc/en/Tutorial/BuiltInExamples>.
- [28] Blinking a LED With Raspberry Pi: 3 Steps. Instructables - How to make anything [online]. Copyright © 2017 Autodesk, Inc. [cit. 27.11.2017]. Dostupné z: <http://www.instructables.com/id/Blinking-a-LED-with-raspberry-pi/>.
- [29] Raspberry Pi Tutorial 38 - Use an Arduino as a Slave with Python (Nanpy) - YouTube. YouTube [online]. Dostupné z: <https://www.youtube.com/watch?v=QumIhvYtRKQ>.
- [30] Raspberry Pi and Arduino Connected Over Serial GPIO - Oscar Liang. Oscar Liang - Sharing Knowledge and Ideas [online]. Dostupné z: <https://oscarliang.com/raspberry-pi-and-arduino-connected-serial-gpio/>.
- [31] Raspberry Pi and Arduino Connected Using I2C - Oscar Liang - Sharing Knowledge and Ideas [online]. Dostupné z: <https://oscarliang.com/raspberry-pi-arduino-connected-i2c/>.

- [32] GitHub - doceme/py-spidev. The world's leading software development platform · GitHub [online]. Copyright © 2017 [cit. 24.11.2017]. Dostupné z: <https://github.com/doceme/py-spidev>.
- [33] Raspberry Pi to Arduino SPI Communication. Design and Build Your Own Robot [online]. Dostupné z: <http://robotics.hobbizine.com/raspiduino.html>.
- [34] Arduino - ArduinoEthernetShield . Arduino - Home [online]. Dostupné z: <https://www.arduino.cc/en/Guide/ArduinoEthernetShield>.
- [35] python - Communicating between Raspberry Pi and Arduino over LAN - Stack Overflow. Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. Dostupné z: <https://stackoverflow.com/questions/31193686/communicating-between-raspberry-pi-and-ar>.
- [36] Roberto Valgolio. Roberto Valgolio [online]. Dostupné z: <http://www.robertovalgolio.com/>.
- [37] aquisição de dados para o Excel utilizando Arduino em 5 minutos! - YouTube. YouTube [online]. Dostupné z: <https://www.youtube.com/watch?v=BmYAGVmquo&t=124s>.
- [38] gpio - How do I send data from the Raspberry Pi to the PC - Raspberry Pi Stack Exchange. Raspberry Pi Stack Exchange [online]. Dostupné z: <https://raspberrypi.stackexchange.com/questions/12955/how-do-i-send-data-from-the-raspberry-pi-to-the-pc>.
- [39] Arduino Weather Station Project #4 - educ8s.tv - Watch Learn Build. educ8s.tv - Arduino, Raspberry Pi, DIY robots and more [cit. 11.08.2017]. Dostupné z: <http://educ8s.tv/arduino-weather-station-4/>.
- [40] www.google.com„[online]. Dostupné z: <http://google.com>“.
- [41] VÁŇA, Vladimír. Mikrokontroléry ATMEL AVR: programování v jazyce Bascom. Praha: BEN - technická literatura, 2004. ISBN 80-730-0115-2., 2004.
- [42] razzpisampler. Raspberry Pi Cookbook [online]. Copyright © 2013, O [cit. 15.09.2017]. Dostupné z: <http://razzpisampler.oreilly.com/ch07.html>.
- [43] MATOUŠEK, David. Práce s mikrokontroléry ATMEL. 2. vyd. Praha: BEN - technická literatura, 2006. μ C. ISBN 80-730-0209-4., 2006.

- [44] KOCOUREK, Petr a Jiří NOVÁK. Přenos informace. Praha: Vydavatelství ČVUT, 2004. ISBN isbn80-01-02892-5., 2004.
- [45] CHMELA, František. Automatizace a robotizace I: (Automatizace). Praha: České vysoké učení technické, 1992., 1992.
- [46] HRBÁČEK, Jiří. Komunikace mikrokontroléru s okolím. Praha: BEN - technická literatura, 2000. ISBN 80-860-5673-2., 2000.
- [47] HRBÁČEK, Jiří. Komunikace mikrokontroléru s okolím. Praha: BEN - technická literatura, 1999. ISBN 80-860-5642-2., 1999.
- [48] HAASZ, Vladimír, Jiří NOVÁK a Jaroslav ROZTOČIL. Číslíkové měřicí systémy. Vyd. 2. přeprac. Praha: České vysoké učení technické, 2000. ISBN 80-010-2219-6., 2000.
- [49] Jameco Electronics - Electronic Components Distributor [online]. Dostupné z: https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry_pi_circuit_note_fig2a.jpg.
- [50] „Arduino Pro Mini » mobilerobots.pl - all about mobile robots, sumo robots. mobilerobots.pl - all about mobile robots, sumo robots [online]. Dostupné z: http://www.mobilerobots.pl/index.php?p=1_95“.
- [51] „Generátor online citací - citace webových stránek podle ISO 960-2011. Generátor online citací - citace webových stránek podle ISO 960-2011 [online]. Copyright © Generování citací online 2015 [cit. 21.09.2017]. Dostupné z: <http://generator-citaci.cz/>“.