



## **Středoškolská technika 2015**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **ROBOT S RASPBERRY**

**Miroslav Rössler**

Střední průmyslová škola elektrotechnická  
V Úžlabině 320, Praha 10

## **Poděkování**

Chtěl bych poděkovat panu Ing. Petru Hellebrandovi za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat. Mé poděkování patří též paní PhDr. Romaně Bukovské, za zprostředkování tématu k praktické maturitní zkoušce.

# **Anotace**

Cílem dlouhodobé maturitní práce spadající pod obor Řídící technika, bylo sestavit robota, který bude hýbat paží, reagovat na pohyb díky pohybovému čidlu a zobrazovat text pomocí maticového displeje. Mozkem robota je malý deskový počítač Raspberry pi B+.

# **Annotation**

The goal of long-graduation work under the Control engineering, was to build a robot that will move its arm to respond to movement thanks to a motion sensor and to display a text using a matrix display. The brain of the robot is a small board computer Raspberry Pi B +.

# Obsah

<b>1</b>	<b>Raspberry Pi .....</b>	<b>6</b>
1.1	Co to je?.....	6
1.2	Prostředí.....	7
1.3	GPIO.....	8
<b>2</b>	<b>Zapojení s LED diodou .....</b>	<b>9</b>
2.1	Zapojení LED diody .....	9
2.2	Semafor pomocí LED diod.....	10
<b>3</b>	<b>Maticový displej.....</b>	<b>13</b>
3.1	Zapojení.....	14
3.2	Programování .....	15
3.3	Tvorba obrázku v programu Gimp 2.....	15
<b>4</b>	<b>Servo motor .....</b>	<b>17</b>
4.1	Zapojení jednoho servo motoru.....	18
4.2	Zapojení dvou a více servo motorů .....	20
<b>5</b>	<b>Pohybové čidlo .....</b>	<b>22</b>
5.1	Zapojení s jednou LED diodou .....	23
5.2	Zapojení se servo motorem .....	24
<b>6</b>	<b>Konstrukce .....</b>	<b>26</b>
6.1	Celkové zapojení .....	26
6.2	3D tisk pohyblivé paže a hlavy .....	28
6.3	Úprava PC skříně.....	30
6.4	Modelování druhé paže .....	35

# Úvod

Jako dlouhodobou maturitní práci jsem si vybral robota s pohyblivou paží a maticovým displejem, který bude řízen pomocí malého počítače Raspberry pi. Téma mě zaujalo, protože jsem si chtěl vyzkoušet pracovat s počítačem Raspberry pi, ale nevyužívat ho jako klasický počítač, nýbrž jako řídicí jednotku. Jako první se budu muset seznámit s operačním systémem Raspbian. Poté pomocí programu Python vytvořit programy pro rozsvícení Led diod, abych se naučil pracovat s univerzálními vstupy a výstupy na desce počítače (GPIO). Další krok bude zobrazení textu na maticovém displeji. Dále připojím jedno a poté více servo motorů k počítači a pomocí programu python je rozpohybuji. Konstrukce robota bude tvořena ze skříně od počítače, která nahrazuje trup, pohyblivá paže a hlava budou vytisknuty pomocí 3D tiskárny. Robot by měl sloužit jako propagační materiál nebo vítací robot ve vestibulu školy.

# 1 Raspberry Pi

## 1.1 Co to je?

Raspberry Pi (RPi) je jednodeskový počítač o malé velikosti a hlavně nízké pořizovací ceně. Tento počítač je vyvíjen ve Velké Británii, kde ho i vyrábějí. Hlavním účelem tohoto malého počítače je podpořit výuku informatiky a programování ve školách. U nás se o tomto počítači moc nemluví a je zde využíván spíše pro domácí účely, jelikož RPi může sloužit jako stroj pro mnoho aplikací. Dá se například použít jako multimediální centrum k televizi. RPi zvládne přehrát film ve Full HD rozlišení při 30 snímcích za vteřinu.<sup>1</sup>



Obr. 1 – Raspberry pi B+, zdroj: [amazon.com]

Teprve roku 2012 vstoupil tento počítač do prodeje. Už za tři roky postoupil jeho vývoj tak rychle, že v roce 2015 se již prodává pátý model. Má práce je postavena na čtvrtém modelu, který nese název Raspberry Pi B+. Tento model má jednojádrový procesor taktovaný na 700Mhz a 512MB operační paměti. Jinak je totožný s modelem uvedeným dále.

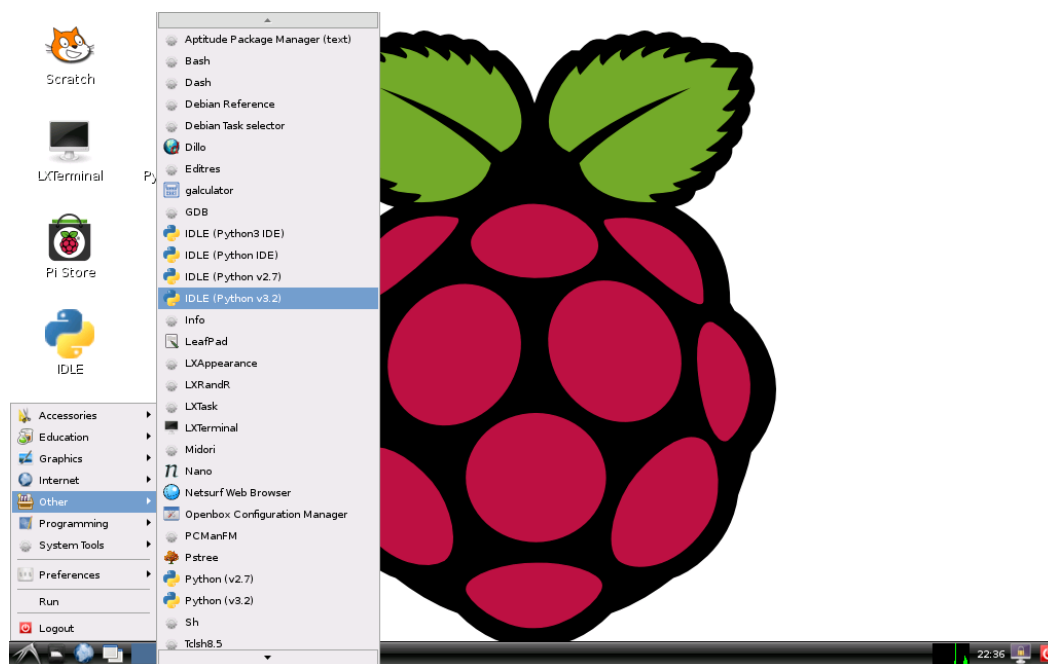
---

<sup>1</sup> FullHD – video soubor o vysokém rozlišení 1920x1080 pixelů

Hlavní výhodou, jak jsem již psal, je nízká cena. U každého nového modelu je cena stanovena v přepočtu kolem jednoho tisíce korun. Za tuto poměrně nízkou cenu obdržíme plně funkční počítač s názvem Raspberry pi 2 B, v dnešní době již se čtyřjádrovým procesorem, úctyhodnou 1GB operační pamětí, s konektorem RJ45 pro internetové připojení, čtyřmi porty USB, audio výstupem, HDMI portem pro připojení monitoru, se sloty na displej či kameru, se slotem na paměťovou kartu MicroSD, napájecím micro usb a 40 piny GPIO.<sup>2</sup>

## 1.2 Prostředí

Nejpoužívatelnějším operačním systémem je Linux. Pro RPi je vytvořeno mnoho distribucí (Raspbian, Openelec, Raspbmc, Pidora, Risc OS atd.). Ovšem nejpoužívanější distribuce je Raspbian. Je to operační systém založený na Debianu, který je optimalizován pro RPi. V této distribuci Linuxu jsou předinstalovány některé aplikace. Jsou tu aplikace pro účely programování jako je Python či Scratch. Dále jsou tu mediální programy pro přehrávání hudby, jednoduché hry typu Snake, které jsou vytvořeny v Pythonu, webový prohlížeč Midori, souborový manažer PCMan, který připomíná průzkumníka v operačním systému Windows a další. Obecně grafické prostředí pro tento model je výkonnostně velice náročné, hlavně pro procesor. Jen při pohybu oken souborového manažeru je vytíženost procesoru nad 50%. Samozřejmě se může počítač také ovládat v terminálu.

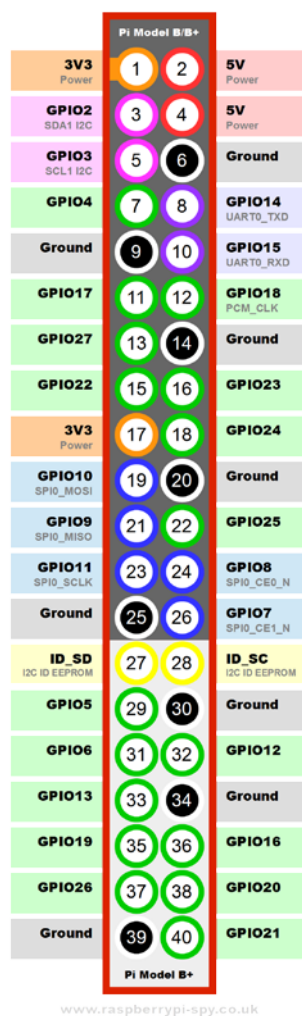


Obr. 2 – Raspbian, zdroj: [jasonrbriggs.com]

<sup>2</sup> GPIO – viz kapitola 1.3

## 1.3 GPIO

General Purpose Input/Output, pro mě nejdůležitější část na RPi, protože na nich je založena tato práce. U mého modelu B+ se jedná o 40 pinů vyvedených v rohu RPi místo původních 26 pinů, které mají starší modely. GPIO vývodů je zde 26. Jedná se o piny, které se dají programovat pomocí softwaru. Do těchto programovatelných pinů lze posílat elektrický signál nebo ho z pinů přijímat. Tím lze pracovat s připojeným hardwarem. Při používání pinů se musí dbát na to, že piny jsou připojeny přímo na čip a to bez žádné ochrany. Proto nesmíme na piny přivést napětí vyšší než je 3,3V, jinak se může celé RPi nenávratně poškodit. Jako další jsou zde dva piny I2C, dále jsou zde čtyři napájecí piny z toho dva na 3,3V a dva na 5V. Zbylé piny jsou země. Na piny existují také rozšiřující moduly. Například jde na piny připojit rozšiřující modul s osmi relé, dotykový displej, audio karta ke snímání a přehrávání HD zvuku, deska akcelerometru a mnoho dalších.



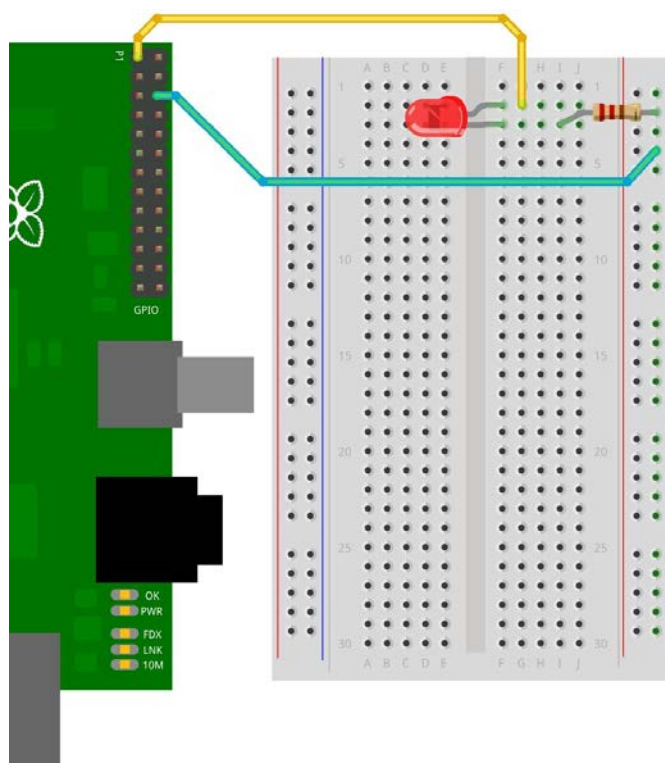
Obr. 3 – Popis GPIO, zdroj: [open-electronics.org]



## 2 Zapojení s LED diodou

### 2.1 Zapojení LED diody

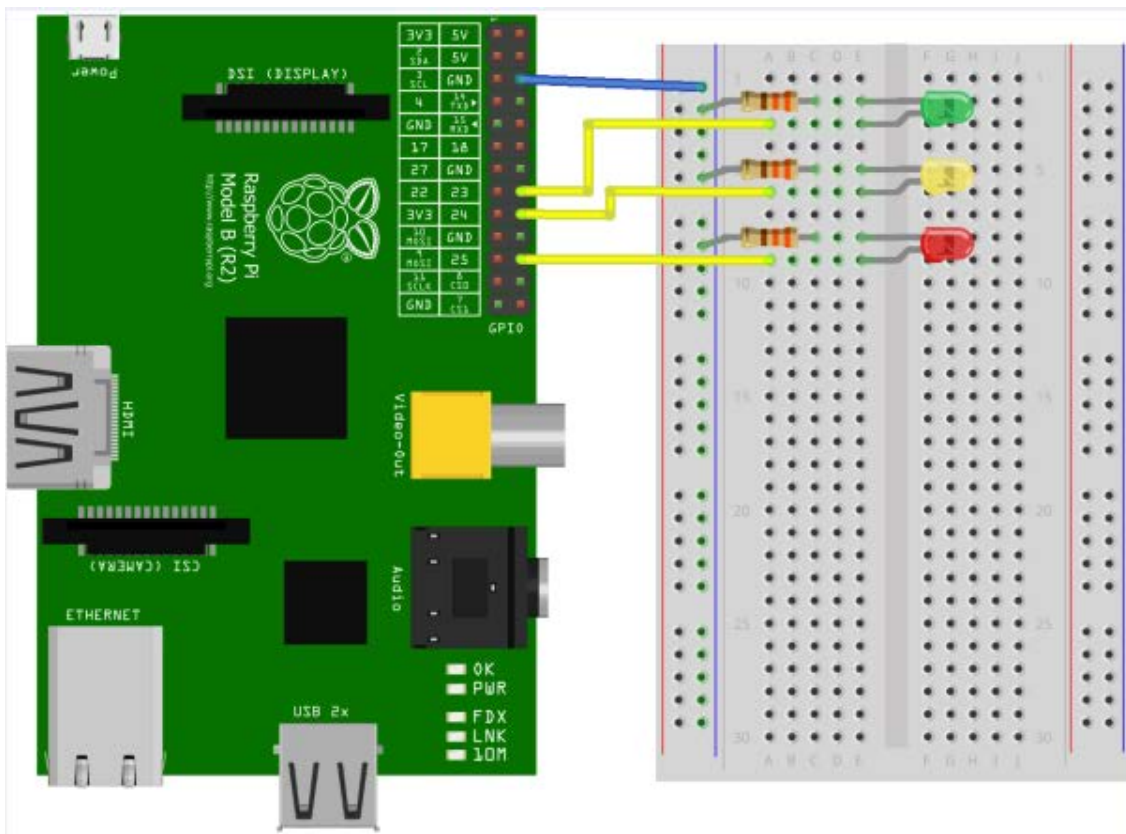
Jako první věc co je dobré udělat, je seznámit se s funkcí GPIO. Nejlehčí způsob je do nepájivého pole zapojit LED diodu a tu rozsvítit. Dioda se kamkoliv na nepájivém poli zapojí. Ke katodě LED diody je připojen rezistor o hodnotě 270 až 330  $\Omega$  (ohm), vyšší by tlumil LED diodu. Odpor je vyveden na nulu nepájivého pole. Nula nepájivého pole je propojovacím drátkem vyvedená na GPIO na pin země. Anoda LED diody se zapojí na první pin na GPIO, což je napájení o hodnotě 3,3 V. Díky tomuto zapojení se LED dioda rozsvítí. Pokud chceme LED diodu rozsvítit a poté zhasnout je zapotřebí anodu LED diody zapojit na programovatelný pin na GPIO. Pro ovládání GPIO pinů si musíme stáhnout knihovnu, která to umožňuje. Poté stačí vytvořit jednoduchý program v pythonu, kam se importuje knihovna na ovládání GPIO programovatelných pinů a pomocí příkazů se spouští nebo vypíná napájení na daném pinu, kde je připojena LED dioda.



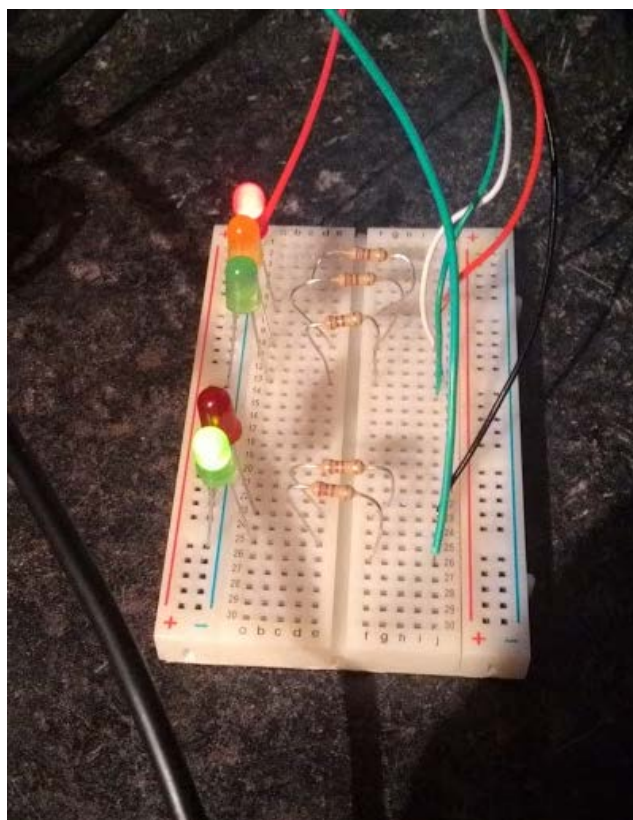
Obr. 4 – Zapojení jedné LED diody, zdroj: [projects.drogon.net]

## 2.2 Semafor pomocí LED diod

Dále si vytvoříme semafor skládající se z více LED diod. Konkrétněji to bude semafor pro auta a pro chodce. Vše sestavené na nepájivém poli. Srovnáme si pod sebou tři LED diody, červenou, žlutou a zelenou. O pár polí níže si zapojíme dvě LED diody, které budou představovat semafor pro chodce, červená a zelená. Ke každé katodě LED diody připojíme rezistor o hodnotě 270 až 330  $\Omega$  (ohm). Druhou nohu rezistorů připojíme na mínus nepájivého pole. Mínus nepájivého pole připojíme na 6 pin na RPi, což je zem. Každou anodu u LED diody připojíme na programovatelné piny na RPi. Například vrchní červenou LED diodu připojíme na 11 pin. Tento pin se nazývá GPIO17. Žlutou LED diodu připojíme na 12 pin, což je GPIO18. Zelenou LED diodu u semaforu pro auta připojíme na 13 pin, jehož název je GPIO27. U semaforu pro chodce zapojíme červenou LED diodu na pin číslo 15 s názvem GPIO22 a zelenou LED diodu připojíme například na 16 pin, což je GPIO23. Anodu můžeme zapojit podle sebe na jakýkoliv GPIO pin, podle toho bude poté vypadat program. Program si vytvoříme například v Pythonu. V programu si jako první krok musíme importovat knihovny. Importujeme si knihovnu `time`, kvůli postupnému rozsvícení, konkrétněji pozastavení cyklu LED diod. Dále si importujeme knihovnu pro ovládání GPIO pinů. Tuto knihovnu Python neobsahuje, tudíž si ji musíme stáhnout. K tomu nám postačí napsat pár řádků do terminálu. Jakmile máme importované knihovny, tak si definujeme piny, na kterých je připojena daná LED dioda. Po nadefinování pinů vytvoříme nekonečný cyklus `While true`. Do cyklu píšeme příkaz pro výstup, kde budeme nastavovat hodnotu `LOW` pro zhasnutí nebo `HIGH` pro rozsvícení LED diody. Jako první si rozsvítíme červenou LED diodu pro auta a zelenou pro chodce. A poté zadáme `time.sleep(2)`, aby nám LED diody zůstaly po dobu 2 vteřin rozsvícené. Následně zelenou LED diodu pro chodce zhasneme, rozsvítíme červenou a žlutou pro auta a opět na 2 vteřiny zastavíme. Jako další krok zhasneme oranžovou a červenou LED diodu pro auta a rozsvítíme zelenou. Zelenou necháme svítit například 5 vteřin. Dále zhasneme zelenou, rozsvítíme oranžovou a opět pozastavíme na 2 vteřiny. A jako poslední krok zhasneme oranžovou s červenou pro chodce a rozsvítíme chodcům zelenou. Takto se bude program opakovat díky cyklu `While true`.



Obr. 5 – Zapojení více LED diod na programovatelné piny, zdroj: [projects.drogon.net]



Příloha č. 6 – Zapojení semaforu

```
*TrafficLED1.py - D:\Dokumenty\ \DMP\TrafficLED1.py (2.7.9)*
File Edit Format Run Options Windows Help
import time # importuje knihovnu time pro použití příkazu sleep
import RPi.GPIO as GPIO # importuje knihovnu pro ovládání pinů

GPIO.setmode(GPIO.BOARD) # Definuje jaký typ značení pinů je použit
RED=11 # Zavádíme názvy barev místo číslo pinu, pro přehlednost
AMBER=13
GREEN=15
RED1=16
GREEN1=18

GPIO.setup(RED, GPIO.OUT) # Definujeme piny
GPIO.setup(AMBER, GPIO.OUT)
GPIO.setup(GREEN, GPIO.OUT)
GPIO.setup(RED1, GPIO.OUT)
GPIO.setup(GREEN1, GPIO.OUT)

while True: # nekonečný cyklus
    GPIO.output(RED, GPIO.HIGH) # rozsvítíme danou LED diodu
    GPIO.output(GREEN1, GPIO.HIGH)
    print "Red a Green" # vypíšeme si diody, které svítí

    time.sleep(2) # pozastavíme cyklus po dobu 2 vteřin
    GPIO.output(GREEN1, GPIO.LOW) # zhasneme danou LED diodu
    GPIO.output(AMBER, GPIO.HIGH)
    GPIO.output(RED1, GPIO.HIGH)
    print "Red and Amber"

    time.sleep(2)
    GPIO.output(AMBER, GPIO.LOW)
    GPIO.output(RED, GPIO.LOW)
    GPIO.output(GREEN, GPIO.HIGH)
    print "Green"

    time.sleep(5)
    GPIO.output(GREEN, GPIO.LOW)

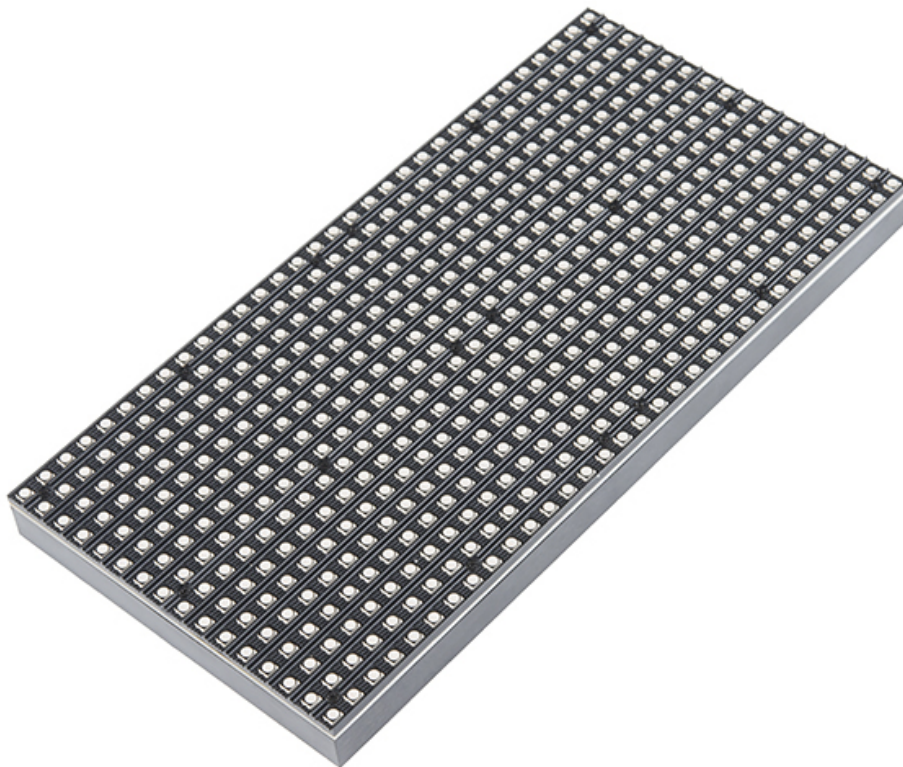
    GPIO.output(AMBER, GPIO.HIGH)
    time.sleep(2)
    GPIO.output(AMBER, GPIO.LOW)
    GPIO.output(RED1, GPIO.LOW)
    GPIO.output(GREEN1, GPIO.HIGH)
```

Ln: 22 Col: 70

Obr. 7 – Program pro semafor

### 3 Maticový displej

K zobrazování na tomto displeji se používají LED diody. U displeje pro tuto práci je použit RGB displej o velikost 16x32 LED diod. Tento typ displeje umožňuje zobrazovat jednotlivé body uspořádané do matice. Z bodů se poté vytváří obraz. LED diody mají spotřebu velice malou v desítkách miliampér. Ale i tak pro snížení spotřeby se LED diody napájí pulzně, to znamená, že blikají v krátkých intervalech. Tyto krátké intervaly lidské oko nezaznamená a tím pádem člověku přijde, že LED diody svítí v kuse bez blikání. Maticové displeje jsou všude kolem nás. Koukáme na ně a ani si neuvědomujeme, že to vlastně maticový displej je. Třeba v MHD, kde dopravní prostředky ukazují ne jen konečnou stanici. Displeje, kde čekáme na pořadové číslo, displej v kalkulačce, digitální hodinky a mnoho dalších věcí, kde se objevuje tento druh displeje.



Obr. 8 – Maticový displej, zdroj: [sparkfun.com]



## 3.1 Zapojení

Displej má na zadní straně tři konektory. Jeden konektor je pro externí napájení displeje. Další konektor je pro zapojení vstupu z RPi. Poslední konektor slouží jako propojovací konektor k propojení více těchto maticových displejů. K připojení displeje k RPi je použito 13 propojovacích drátků, které se z displeje vyvádí jen na programovatelné piny a na piny sloužící jako uzemnění. Piny na displeji jsou označeny písmenem, číslem a značkou uzemnění. Na napájecí piny na RPi se nepřipojuje nic, jelikož displej musí mít externí napájení na 5V při maximálním proudovém odběru 1A při rozsvícení všech LED diod na displeji. Jako napájení je zvolen zdroj ke stolnímu počítači. Aby mohl být zdroj použit bez základní desky je nutno propojit na hlavním napájecím konektoru zelený pin s černým pinem. V balení s displejem byl i napájecí kabel. Tento kabel je propojen s kabelem molex, který vychází z počítačového zdroje. Konkrétně je propojen drátem z napájecího kabelu od displeje na červený drát konektoru molex, který napájí 5V a na černý drát, což je zem. Zapojení si pro bezpečnost raději několikrát zkontrolujeme.



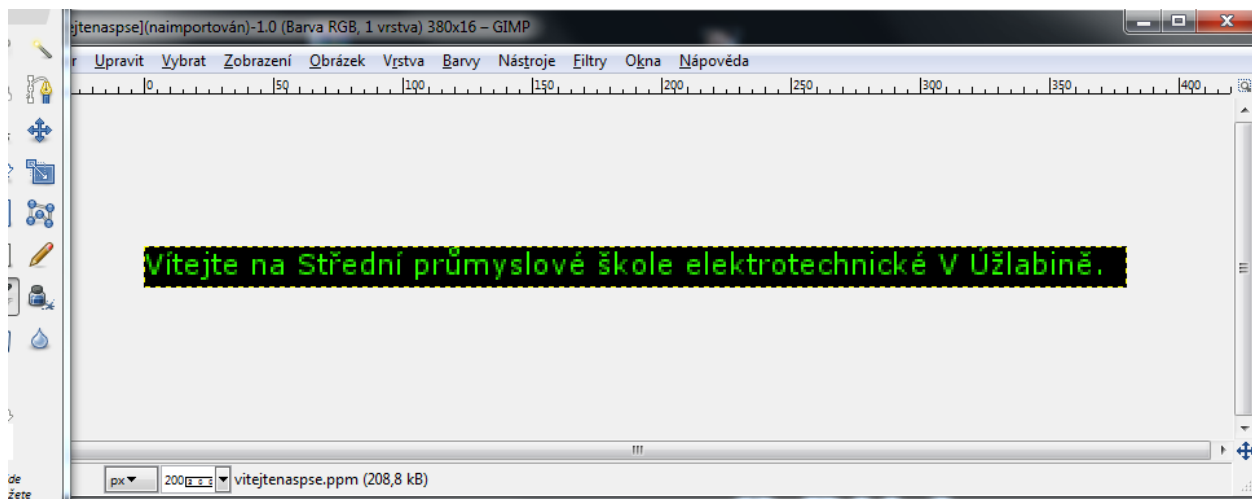
Obr. 9 – první zapojení displeje

## 3.2 Programování

Když jsme si zcela jisti zapojením, je třeba displej spustit. V terminálu se dostaneme do domovské složky, kde si vytvoříme složku s názvem displej16x32 pomocí příkazu `mkdir displej16x32`. Po vytvoření složky do ní vejdeme a to příkazem `cd displej 16x32`. Následně budeme potřebovat načíst kód z původního projektu, který je již vytvořen. Pomocí příkazu `git clone https:// github.com/hzeller/rpi-rgb-led-matrix/` si stáhneme požadovanou knihovnu. Tuto knihovnu si musíme zkompilovat, než pustíme program. V terminálu zadáme příkaz `cd rpi-rgb-led-matrix`, abychom se dostali do složky `rpi-rgb-led-matrix`. V této složce zadáme příkaz `make`, abychom provedli kompilaci. Jakmile se kompilace provede, můžeme si zkusit testovací, předdefinovaný program. Program se pustí pomocí příkazu `sudo ./ led-matrix`. Sudo proto, jelikož ke spuštění programu musíme mít práva správce. Pokud je vše v pořádku, tak se na displeji zobrazí pohybující se krychle. Krychle nebude celá, jelikož knihovna byla vytvořena pro displej o velikosti 32x32 LED diod. Proto, abychom si mohli na displeji zobrazit svůj vlastní text, je potřeba, abychom si například pomocí grafického programu Gimp vytvořili obrázek s textem. Když máme vytvořený text s příponou ppm, vložíme ho do složky `led.metrix`. Poté k zobrazení textu stačí zadat příkaz `sudo ./led-matrix 1 název_souboru.ppm` a text se nám zobrazí na displeji.

## 3.3 Tvorba obrázku v programu Gimp 2

Jelikož obrázek je zapotřebí ve formátu s názvem ppm, jehož zkratka znamená portable pixel format, musíme si opatřit grafický editor, který umí vytvořit obrázek v tomto formátu. Jako ideální program poslouží freewarový bitmapový grafický editor Gimp 2. V tomto programu si vytvoříme nový obrázek. Program se nás zeptá, jakou velikost obrázku chceme. Jelikož máme displej o velikosti 16x32 bodů, je třeba si zvolit výšku obrázku 16 pixelů. Šířka obrázku je volitelná. Může být i několika násobně větší než je displej, jelikož text se bude horizontálně posouvat. Jakmile máme vytvořený prázdný obrázek, tak si ho pomocí plechovky umístěné v panelu nástrojů vybarvíme načerno. Černou barvu pozadí vybereme kvůli tomu, aby byl text zřetelný. Díky černé barvě pozadí nesvítí všechny LED diody, ale pouze ty, kde je zobrazován text. Poté si vytvoříme novou vrstvu, do které vložíme text. Velikost textu přizpůsobíme výšce obrázku. Barvu textu si můžeme zvolit libovolnou, díky RGB displeji. Pokud máme vytvořený text, exportujeme si soubor tak, aby jeho přípona byla ppm.

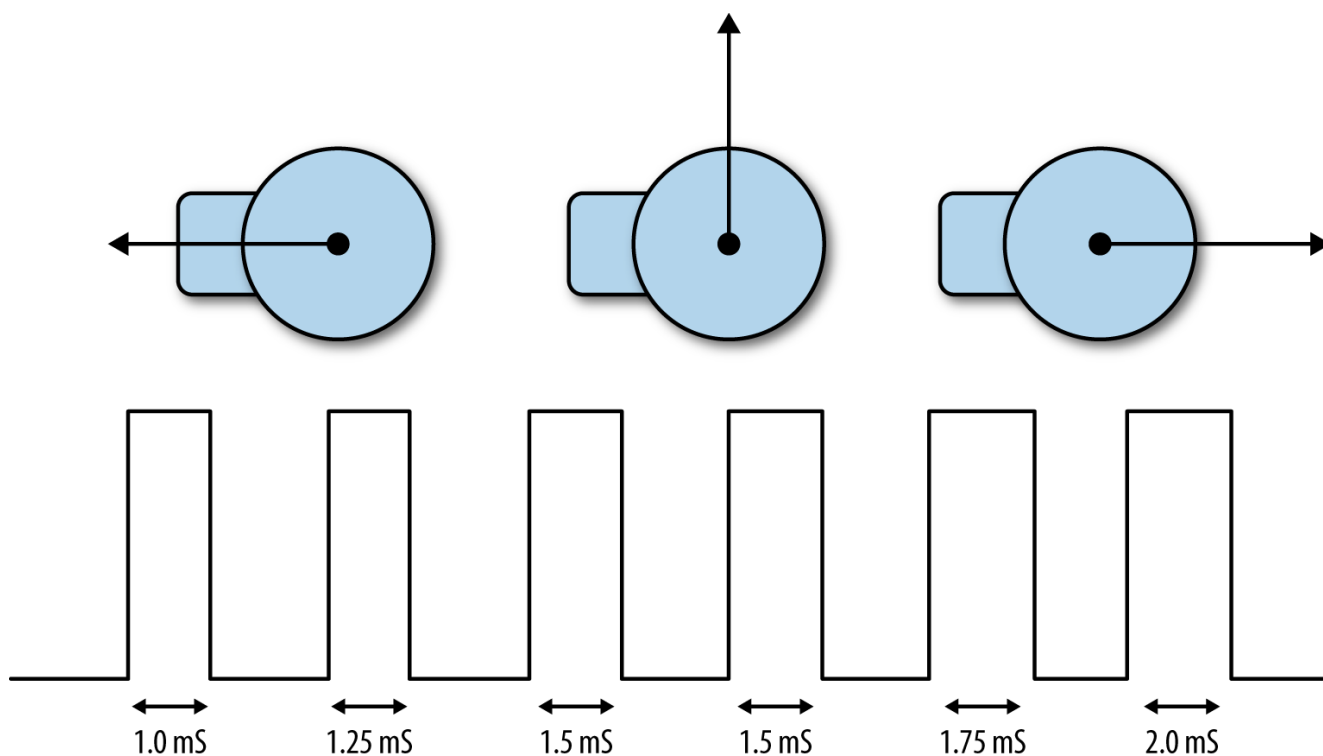


Obr. 10 – Tvorba textu



## 4 Servo motor

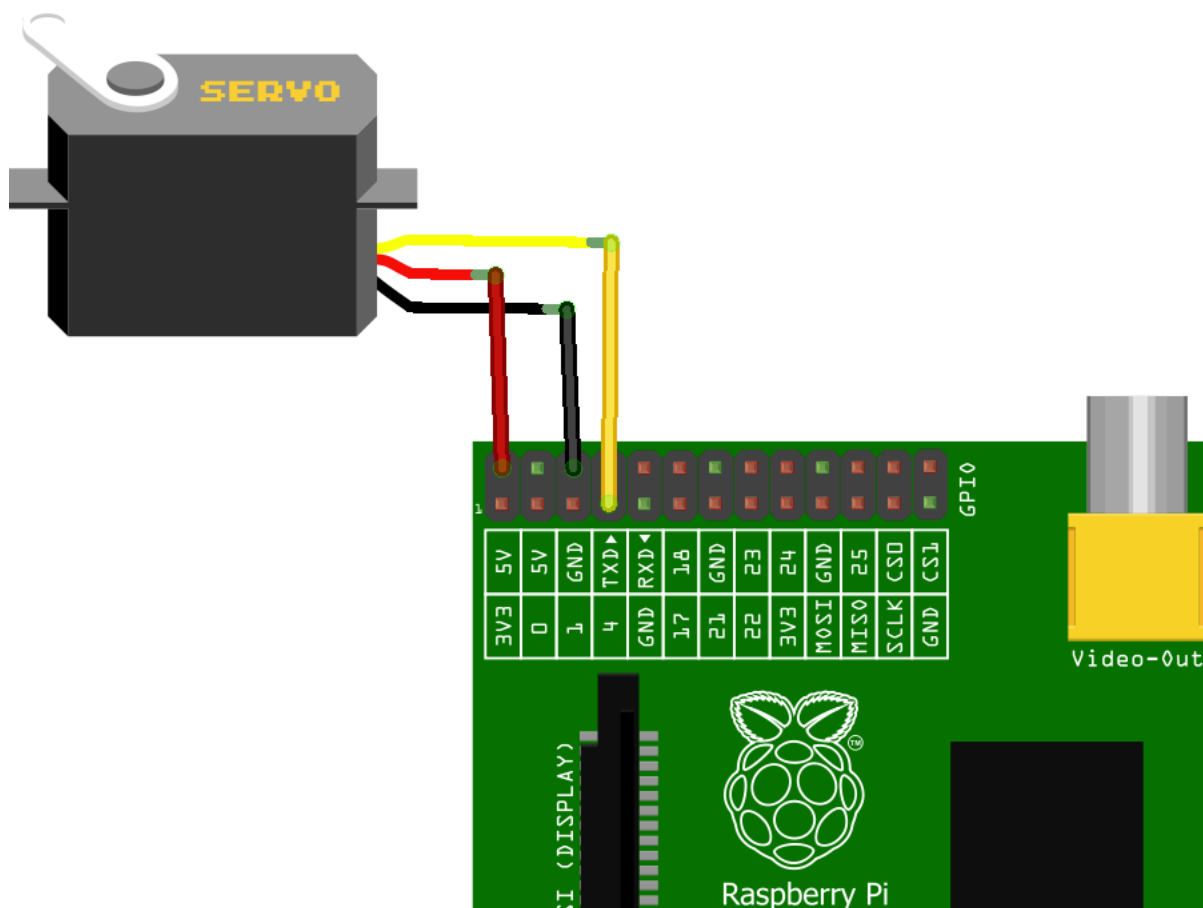
K pohybu paže bylo vybráno jako zdroj pohybu modelářské servo. Ze serva je přivedeno napájení a řídicí signál. Podle toho, jaký se řídicí signál do serva pošle, tak se servo otočí. Modelářské servo je složeno ze tří částí. První část je stejnosměrný motor, druhá část je převodovka a poslední část je řídicí elektronika. Převodovka převádí úhel natočení z motoru na zpětnovazební potenciometr. Servo se ovládá pomocí impulsů 1ms - 2ms. Impuls 1ms znamená úhel 0° natočení, 1.5ms 90° a 2ms 180°. Doporučená frekvence opakování pulzu je 50Hz. Natočení serva je plynulé u lepšího serva i o 1°. Ze serva vedou 3 dráty. Červený je kladné napájení, černý nebo hnědý je záporné napájení nebo-li zem a žlutý představuje řídicí impulsy. Většina modelářských servo motorů fungují na napájení o hodnotě 4.8V až 6V. Doporučuji si najít datasheet.



Obr. 11 – Princip servo motoru, zdroj: [razzpisampler.oreilly.com]

## 4.1 Zapojení jednoho servo motoru

V prvním kroku je dobré, jako u rozsvícení LED diody, zapojit jen jeden servo motor. Nejdříve jsem si zjistil, jak takové modelářské servo funguje. Jakmile jsem si nastudoval informace, mohlo se jít zapojovat. Červený drát ze serva je připojen například přímo na druhý pin na RPi, což je napájení o hodnotě 5V, které je pro modelářské servo ideální. Černý, respektive hnědý drát je vyveden na šestý pin, abych servo uzemnil. Třetí poslední žlutý drát musí být připojen na jeden z programovatelných pinů GPIO. Může to být například sedmý pin s názvem GPIO4. Dále se vytvoří program v Pythonu. Na začátek se importují dvě knihovny. První knihovnou je RPi.GPIO pro ovládání programovatelných pinů a druhou knihovnou je time, kvůli příkazu sleep pro pozastavení cyklu. Na další řádek se definuje číslování pinů. Buď se použije příkaz GPIO.setmode(GPIO.BCM), který znamená číslování pinů podle názvu pinu, nebo se může použít GPIO.setmode(GPIO.BOARD), to znamená číslování podle toho kolikátý pin je to v pořadí na RPi. Poté následuje příkaz, který definuje pin, kde je zapojen žlutý drát ze serva. V dalším příkazu se pojmenuje servo a zadá se opět číslo pinu s frekvencí opakování pulzu. Další příkaz určuje do jaké pozice se má servo otočit při spuštění. Dále následuje cyklus while. V cyklu se udává otáčení serva. Hodnoty jsou v rozmezí od 2,5 až do 12,5. Hodnota 2,5 představuje nulový úhel a hodnota 12,5 představuje úhel o velikosti 180°. Poté například hodnota 7,5 je střed, tudíž úhel otočení činí 90°.



Obr. 12 – Zapojení jednoho servo motoru, zdroj: [blogspot.tenettech.com]

```
*Untitled*
File Edit Format Run Options Windows Help
import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

GPIO.setup(7,GPIO.OUT)

p = GPIO.PWM(7,50)          #posílá 50 signálu za vteřinu z 21 pinu
p.start(7.5)                # zde se nastavuje výchozí pozice 7.5 = 90°

try:
    while True:            # nekonečná smička
        p.ChangeDutyCycle(12.5)    # pošle do serva signál na otočení na pozici 180°
        time.sleep(0.5)            # pozdržení smičky o půl vteřiny
        p.ChangeDutyCycle(7.5)     #pošle do serva signál na otočení na pozici 90°
        time.sleep(0.5)
        p.ChangeDutyCycle(12.5)    #pošle do serva signál na otočení na pozici 180°
        time.sleep(0.5)            |

```

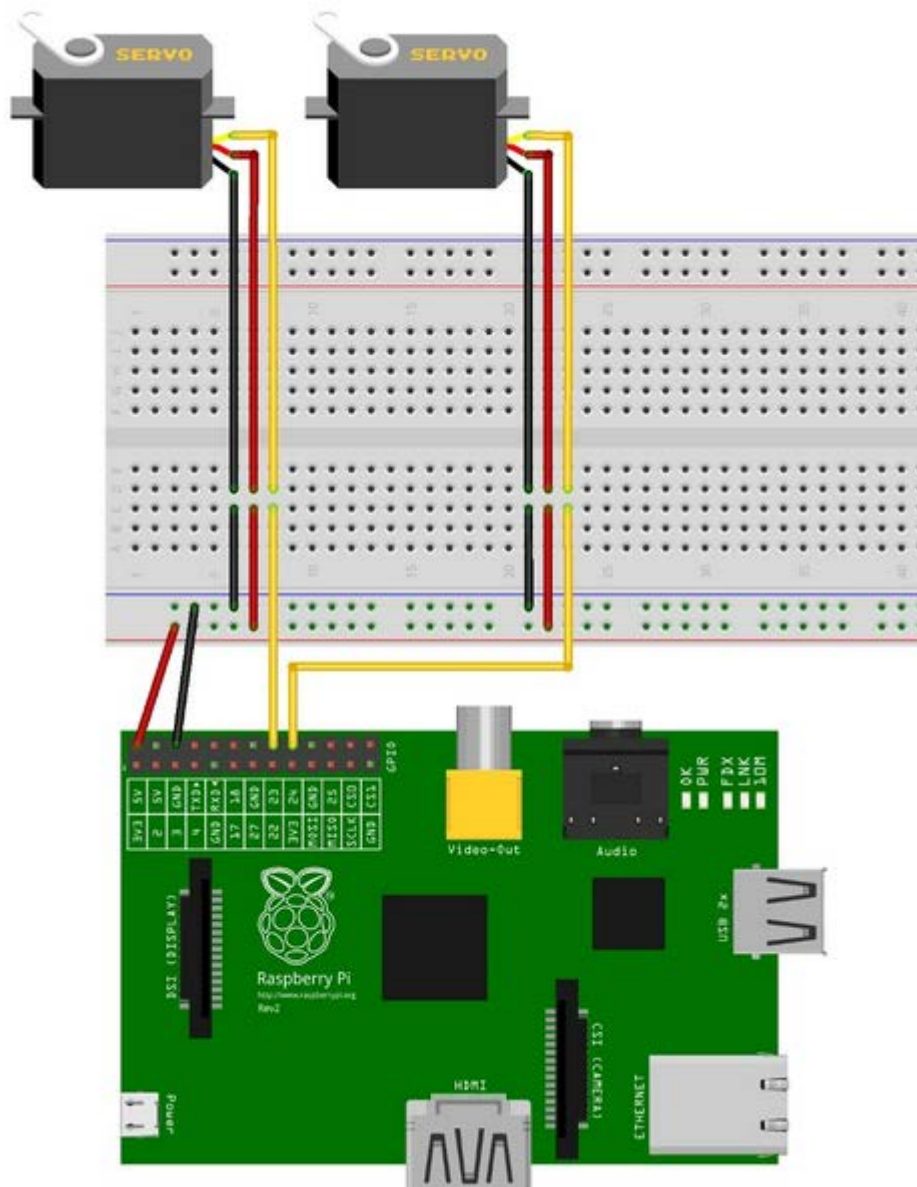
Ln: 27 Col: 42

Obr. 13 – Program pro pohyb servo motoru

## 4.2 Zapojení dvou a více servo motorů

Pro zapojení více modelářských servo motorů se postupuje podobně jako při připojení jednoho modelářského serva. Pokud se budou zapojovat dva servo motory bude stačit napájení z RPi, jelikož na Rpi jsou dva napájecí piny o 5V. Při zapojení více servo motorů, se musí použít externí napájení. K tomu se mi hodí trafo o napětí 5V a proudu okolo 2A. Jako u zapojení jednoho motoru se zapojí červený drát ze všech servo motorů na napájení a černý respektive hnědý drát na zem. Žluté drátky se musí zvlášť rozdělit na programovatelné piny

GPIO. Program pro více servo motorů se vytvoří podle programu pro jeden servo motor. Rozdíl je v počtu použitých programovatelných pinů a v počtu definovaných servo motorů. U každého servo motoru je nutné definovat na jakém pinu má přivedený žlutý drát pro řídicí impulzy a jaká se bude používat frekvence opakování pulzů v Hz. V cyklu programu se nachází určení polohy pro každý motor zvlášť. Může se udělat program pro servo motory, které se budou pohybovat současně nebo každé postupně. Vše je jen o rozdělení pohybů a zadávání příkazu sleep, pro pozastavení cyklu.



Obr. 14 – Zapojení dvou servo motorů, zdroj [blog.dawnrobotics.co.uk]

## 5 Pohybové čidlo

Aby paže mohly reagovat na pohyb, je třeba použít například pasivní infračervené čidlo, které se připojí k RPi. Takové čidlo funguje na principu pyroelektrického jevu, při kterém se pyroelektrické materiály deformují při změnách teploty. Obraz z okolí se promítá na povrch pyroelektrického materiálu. Jakmile v prostoru nastane změna teploty, kdy například projde člověk, tak se materiál v části povrchu teplotní změnou deformuje, a tím je možné zaznamenat indukovaný náboj na jeho povrchu. Pohybové čidlo se dá kalibrovat pomocí dvou trimrů. Prvním z trimrů se nastavuje citlivost čidla a druhým se nastavuje doba, za kterou se má spustit výstupní napětí poté, co čidlo zaznamená pohyb.

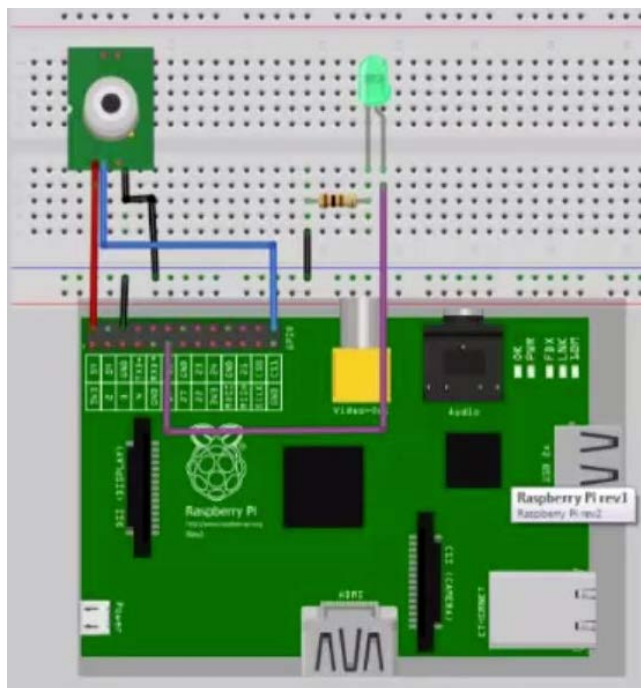


Obr. 15 – Pasivní infračervený sensor



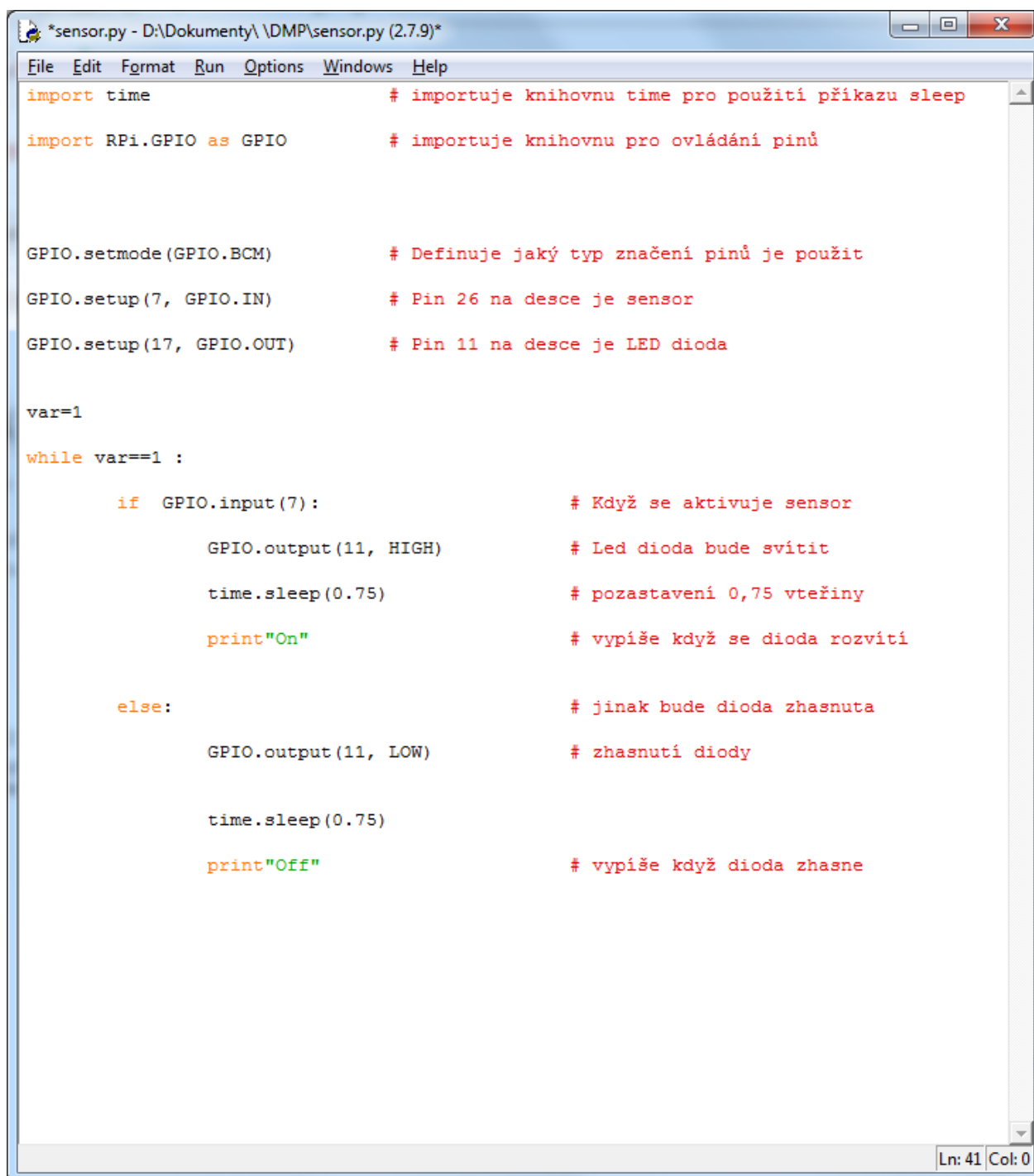
## 5.1 Zapojení s jednou LED diodou

Jako obvykle se pro jistotu zapojí jednoduché zapojení s LED diodou, aby se přešlo případným problémům. PIR čidlo má jako servo motor 3 vývody. Jeden vývod je napájení čidla. To se napájí 5V. Další vývod je výstup a třetí vývod je zem. Vše se zapojí do nepájivého pole, a z něj se dráty propojí na RPi. Červený drát z pohybového čidla je napájení a to se vyvede na druhý pin na RPi. Zem se vyvede poblíž napájení na šestý pin a výstupní drát se připojí na programovatelný pin, například na GPIO4. LED dioda se zastrčí kamkoliv do nepájivého pole. Anoda diody se připojí na programovatelný pin GPIO17, což je 11 pin na RPi. Katoda diody se opět musí uzemnit přes rezistor například o hodnotě 270  $\Omega$  (ohm). Druhá část odporu se připojí na zem, ta je například na šestém pinu na RPi. Jakmile je zapojení hotové, bylo na čase vytvořit program opět v Pythonu. Na začátek programu je nutné si importovat knihovnu na ovládání GPIO pinů a knihovnu na pozastavení sleep. Na další řádek se definuje, na jakém pinu je připojené čidlo a poté na dalším řádku se definuje, na jakém pinu je LED dioda. Vytvoří se cyklus, kde se v podmínce zadá, aby čidlo při detekování pohybu rozsvítilo LED diodu. Jinak ať je LED dioda zhasnutá. Za každou část podmínky je možno vypsát text pomocí příkazu print, aby bylo vidět, zda čidlo reaguje a dělá co má.



Obr. 16 – Zapojení PIR senzoru s LED diodou,

zdroj: [<https://www.youtube.com/watch?v=58-vInmHsXA>]

The image shows a screenshot of a Python script editor window titled '\*sensor.py - D:\Dokumenty\ \DMP\sensor.py (2.7.9)\*'. The window contains a Python script that configures an RPi.GPIO module to control an LED diode based on a PIR sensor's input. The code includes imports for the 'time' module and the 'GPIO' module from 'Rpi.GPIO'. It sets the GPIO mode to BCM, configures pin 7 as an input and pin 11 as an output. A loop with a variable 'var' set to 1 runs indefinitely. Inside the loop, it checks the state of pin 7. If it is high, it sets pin 11 to high, sleeps for 0.75 seconds, and prints 'On'. If it is low, it sets pin 11 to low, sleeps for 0.75 seconds, and prints 'Off'. The status bar at the bottom right shows 'Ln: 41 Col: 0'.

```
import time # importuje knihovnu time pro použití příkazu sleep
import RPi.GPIO as GPIO # importuje knihovnu pro ovládání pinů

GPIO.setmode(GPIO.BCM) # Definuje jaký typ značení pinů je použit
GPIO.setup(7, GPIO.IN) # Pin 26 na desce je sensor
GPIO.setup(11, GPIO.OUT) # Pin 11 na desce je LED dioda

var=1
while var==1 :
    if GPIO.input(7): # Když se aktivuje sensor
        GPIO.output(11, HIGH) # Led dioda bude svítit
        time.sleep(0.75) # pozastavení 0,75 vteřiny
        print"On" # vypíše když se dioda rozvítí
    else: # jinak bude dioda zhasnuta
        GPIO.output(11, LOW) # zhasnutí diody
        time.sleep(0.75)
        print"Off" # vypíše když dioda zhasne
```

Obr. 17 – Program pro zapojení PIR sensoru s LED diodou

## 5.2 Zapojení se servo motorem

Zapojení se servo motorem je obdobné zapojení jako s LED diodou. Opět se čidlo připojí na napájení, přímo na RPi, poté se zapojí zem a nakonec se zapojí výstup čidla na programovatelný pin. Červený drát u servo motoru se připojí na napájecí pin 5V. Černý nebo

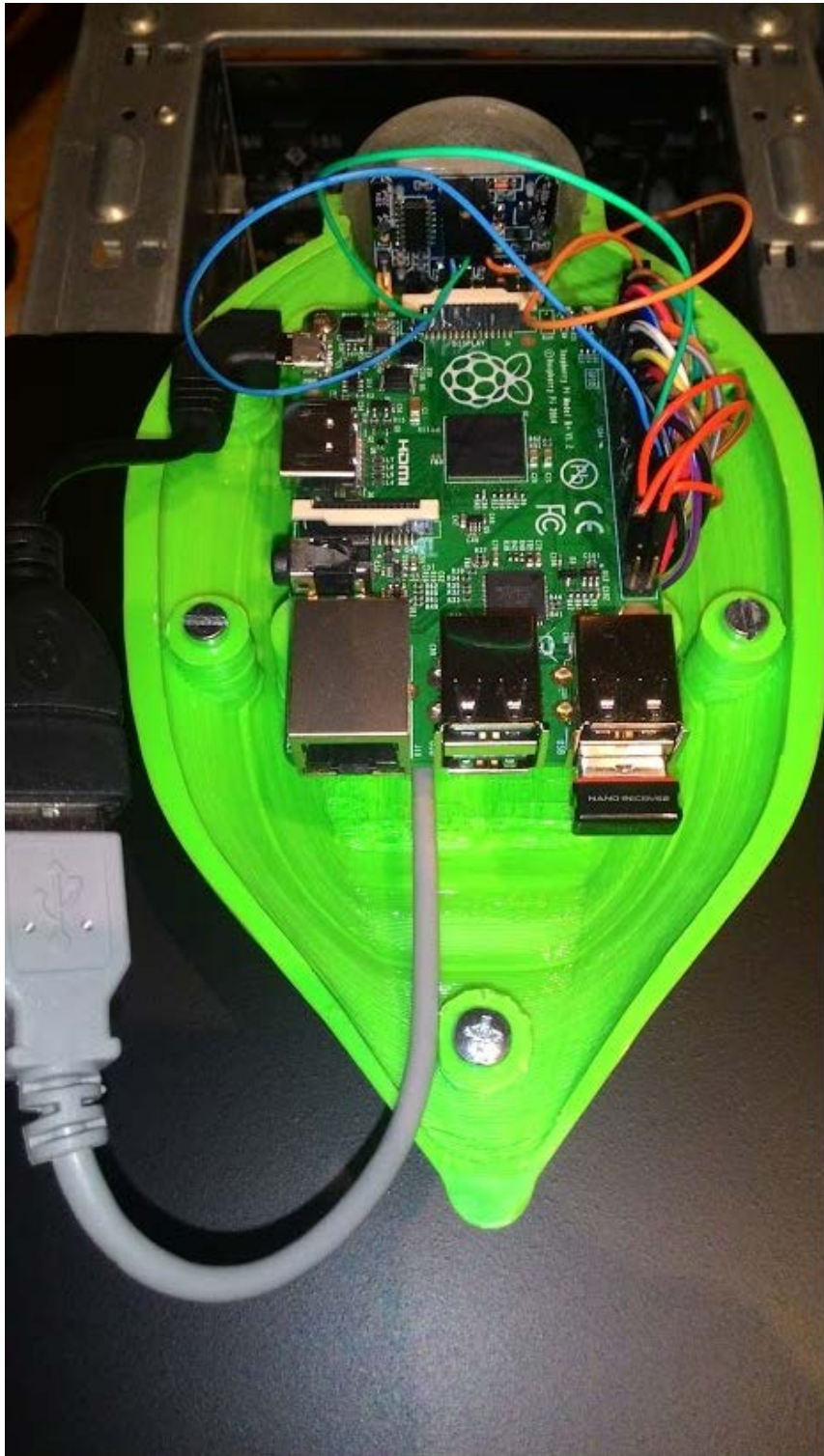


hnědý drát se připojí na zem. A poslední drát, oranžový nebo žlutý se připojí na programovatelný pin. Jakmile je zapojení připraveno, vytvoří se program. Program se opět bude psát v jazyce Python. Na první dva řádky napíšeme příkazy na import knihoven. Knihovnu na ovládání GPIO pinů a knihovnu pro pozastavení. Na další řádek se přidá příkaz znázorňující typ značení pinů. Jako další krok se vytvoří cyklus. V cyklu se zadají příkazy takové, aby se modelářské servo rozpohybovalo poté, co pohybové čidlo zaznamená pohyb. Část programu obsahující pohyb servo motoru se shoduje s programem pro pohyb servo motoru bez pohybového čidla. To znamená, že se bude opět určovat postupná poloha otáčení servo motoru. První zapojení servo motoru dopadlo přehřátím a následném uvaření modelářského servo motoru z neznámých důvodů. Důvod mi není znám, jelikož se stejným programem další servo fungovalo zcela bez problémů.

## 6 Konstrukce

### 6.1 Celkové zapojení

Zapojení displeje se shoduje se zapojením uvedeném v kapitole 3.1. Dále bylo nutné prodloužit délku drátků z modelářských servo motorů, tak aby dosáhly dovnitř skříně, kde je připojen napájecí a nulový drát do wago svorek. Z wago svorek jsou drátky přivedeny na konektor molex, který je připojen na počítačový zdroj. Jeden nulový drátek je prodloužen na pin RPi, aby byl uzemněn i přes RPi. Žluté, či oranžové drátky servo motorů jsou prodlouženy až do hlavy robota, kde jsou zapojeny na piny RPi. Pohybové čidlo je zapojeno přímo i s napájením na piny RPi. Napájení RPi je zařízeno tak, že je vytvořena redukce, která přivádí na USB napájení přímo ze zdroje, tak aby se při zapnutí zdroje zapnulo rovnou i Raspberry s celkovým programem. Celkový program se pustí hned po startu RPi automaticky. Maticový displej bude spuštěn nezávisle na pohybové čidlo pomocí scripta. To znamená, že bude stále v provozu. Mezitím se pustí program pro modelářské servo motory tak, aby se spouštěli v daném cyklu, který bude udávat pohyb, jakmile pohybové čidlo zaznamená pohyb ve svém dosahu.

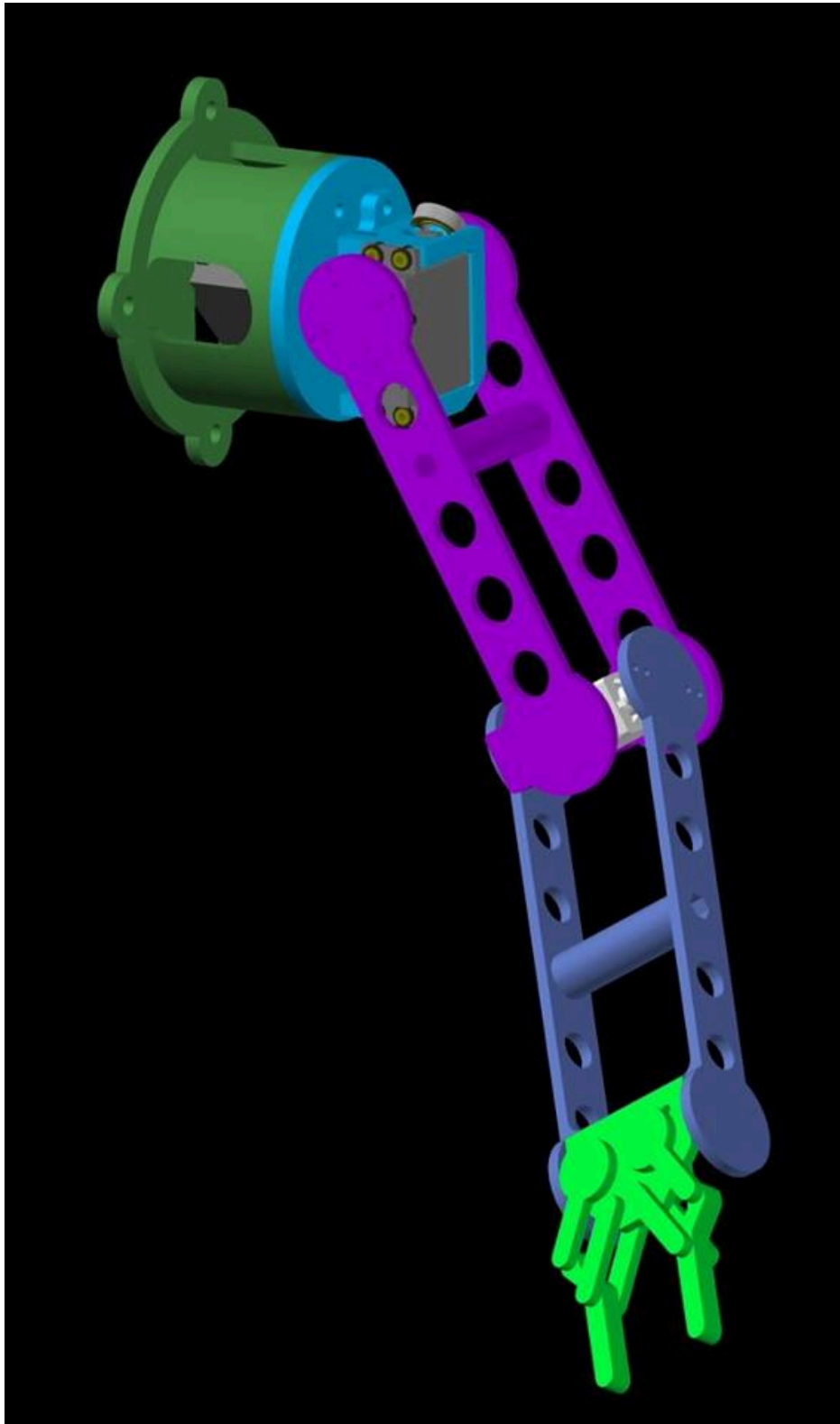


Obr. 18 – Detailní zapojení na piny RPi

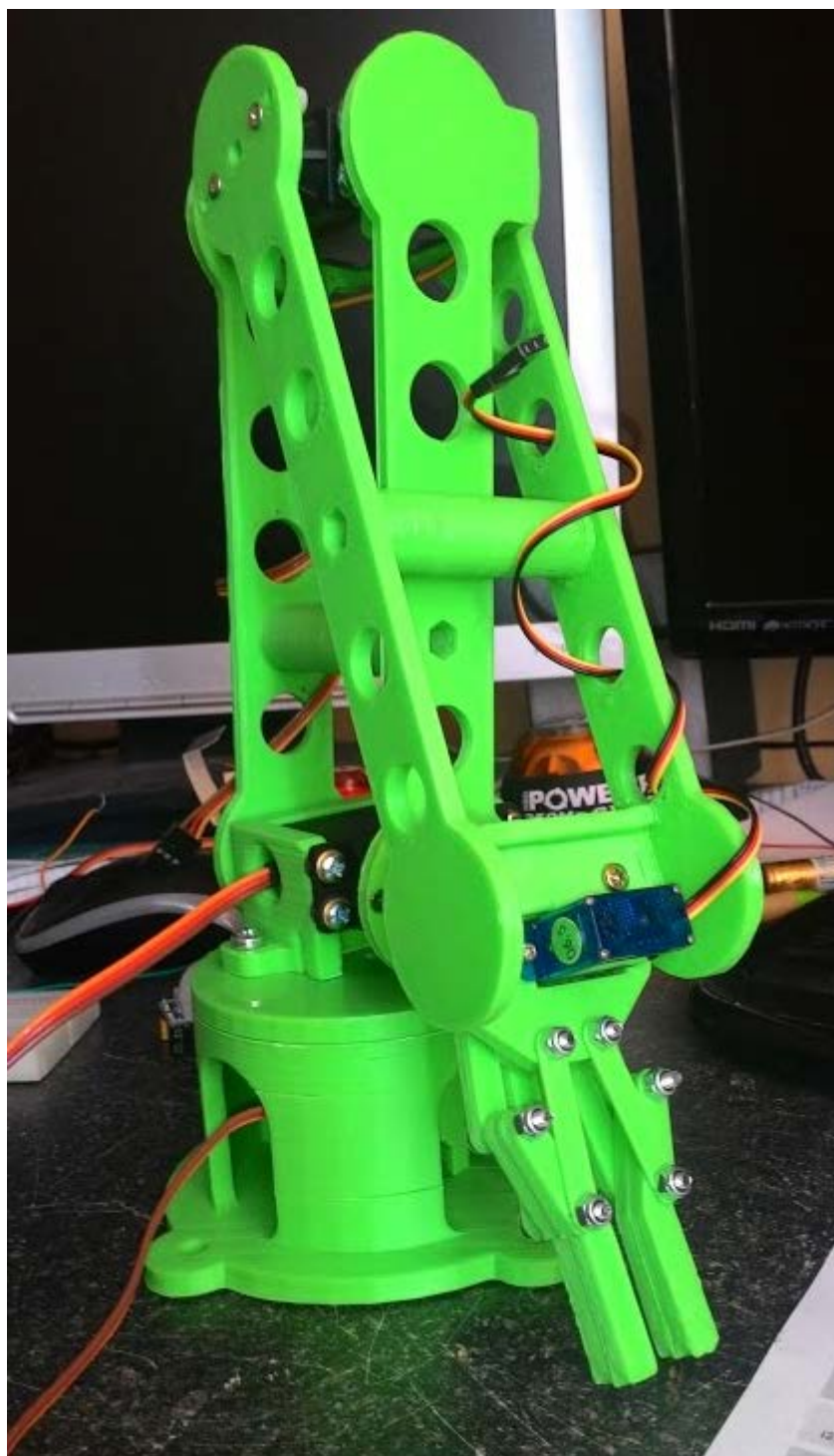
## 6.2 3D tisk pohyblivé paže a hlavy

Pro zhotovení pohyblivé paže se využilo školní 3D tiskárny. Nejdříve se musí v grafickém programu, například AutoCAD vytvořit detailně propracovaný návrh požadovaného výrobku. Program AutoCAD umožňuje ukládat 3D model ve formátu, který podporuje program Slicer. Program Slicer rozřeže danou část 3D modelu do vrstev, které 3D tiskárna postupně tiskne. Vše v návrhu musí do sebe perfektně zapadat. Aby vše do sebe přesně pasovalo, musíme zjistit přesné rozměry servo motorů, pohybového čidla a i malého počítače Raspberry pi B+, které si najdeme v datasheetu daného modelu. Nesmíme zapomenout na místa, kde se budou nacházet šroubky pro pospojování částí dohromady. Výrobek je vhodné si navrhnout po částech, které budou v různých vrstvách, aby se pak daly jednodušeji ukládat zvlášť, a poté bezproblémově otevírat v programu, který obsluhuje tiskárnu. Vytvořit 3D model v grafickém programu je časově velice náročné. Pozdější tisk je opět časově poněkud náročný a jedna část výrobku se může tisknout i několik hodin.

Hlava je přizpůsobena pro pohybové čidlo, které se nachází v čele hlavy. To je umístěno v transparentně vytisknutém kruhu, který je malými šroubky přichycen k pohybovému čidlu. Tento kruh přesně zapadá do drážek spodní a horní části hlavy. Dále je uvnitř hlavy vytvořen prostor pro umístění a uchycení řídicího počítače. V prostoru pro RPi se naskytla chyba, že konektor napájecího kabelu vyčníval z hlavy. Proto musel být zakoupen konektor, který je o 90° zahnutý a ještě musel být konektoru upilován roh, aby se do hlavy vešel. Hlava se skládá z několika částí, které jsou pospojovány pomocí šroubů.



Obr. 19 – 3D návrh pohyblivé paže



Obr. 20 – Vytisknutá a smontovaná pohyblivá paže

## 6.3 Úprava PC skříně

Jelikož na tělo robota by bylo potřeba na 3D tisk velké množství filamentu, tak se trup zhotoví z počítačové skříně. Skříň musí být dostatečně široká, aby do ní mohl být zapuštěn maticový displej, na kterém bude běhat text. Po dlouhém pátrání po široké skříně byla jedna

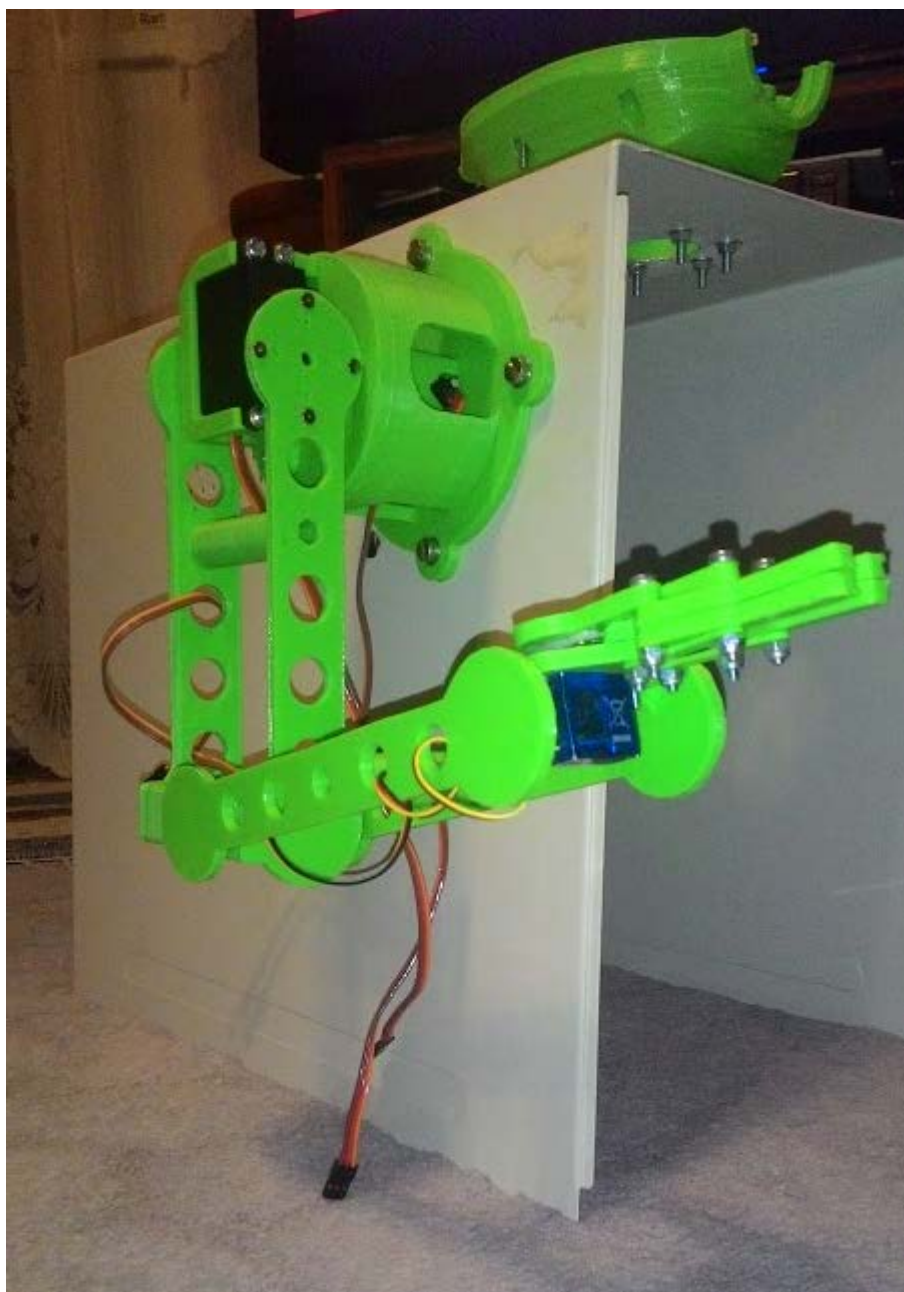
objevena ve školních dílnách. Bylo mi dovoleno si skříň vzít a využít jí pro účely praktické zkoušky. Přední panel skříně se musí upravit, jelikož displej je širší než mechanika na čele skříně. Toho se docílí odřezáním okrajů předního panelu. Vše se přesně vyměří, zkontroluje a odřeže, aby displej zapadl na své místo. Jelikož displej je nižší, než celý otvor pro mechaniku, je použita záslepka pro nevyužité místo na předním panelu. Záslepku bylo nutné snížit, aby zapadla přímo nad displej a zakryla otvor pro mechaniku. Pomocí pájky se záslepka zatavila k přednímu panelu. Do kovové konstrukce skříně jsou vyvrtány díry pro uchycení displeje. Přední panel byl bílý, až nažloutlý. Proto se ředidlem vyčistil a černým matným sprejem přestříkal. Černá matná barva byla vybrána, kvůli vhodnému kontrastu barev mezi trupem robota a jeho zbylých částí. Na přední straně skříně se nacházejí plastové doplňky, které byly také přestříkány, ale na zelenou barvu, aby ladily se zbytkem robota. Bočnice skříně a vršek skříně jsou z jedné části plechu. Že je to z jedné části jsem bral jako nevýhodu při sestavování, kvůli špatnému přístupu. Do vrchní části plechu je vyvrtáno šest otvorů pro šroubky, které uchycují hlavu robota. Ve středu uchycení robota je vyvrtán větší otvor o velikost zhruba 3cm pro protáhnutí drátů k počítači, který je umístěn v hlavě. Na pravé boční straně skříně jsou vyvrtány čtyři díry pro uchycení pohyblivé paže. Na druhé straně skříně je vyvrtána jedna díra pro druhou nepohyblivou paži. Celá tato část je také nastříkána černým matným sprejem. Vnitřek skříně je vyčištěn a uvnitř zůstává pouze uchycení pro mechaniku, které slouží jako výztuž pro celou skříň.





Obr. 21 – Vsazení displeje do skříně a přesříkaná skříně





Obr. 22 – Přidělení pohyblivé paže a hlavy



Obr. 23 – Nastříkaný zbytek skříně



Obr. 24 – Příprava kabelů na prodloužení vývodů ze servo motorů

## 6.4 Modelování druhé paže

Z důvodu omezeného počtu modelářských servo motorů a počtu programovatelných pinů na RPi není druhá paže poháněna automaticky. Podle mě dobrý nápad byl vymodelovat ruku z modelovací hmoty. Jako inspirace byla použita vytisknutá, pohyblivá paže. Z důvodu křehkosti hmoty, je ruka vytvořena užší. Při modelování jsem si navlhčil ruce, aby se mi modelovací hmota nelepila na prsty. Vyválel jsem si pomocí válečku plát, který jsem nařezal na úzké proužky, představující části ruky. V rameni a v loktu je vytvořen kloub. Na konci paže je nepohyblivé chapadlo, aby se ruka podobala vytisknuté předloze. Po ztvrdnutí hmoty jsem ruku nastříkal zeleným sprejem, aby byla více totožná s pohyblivou paží. Vymodelovaná paže slouží spíše jako dekorace, aby robot nebyl jednoruký.





Obr. 25 – finální podoba robota

# Závěr

Výsledkem dlouhodobé maturitní práce, je robot, který má pohyblivou paži díky modelářským servo motorům. Povedlo se zprovoznit vše, co se zprovoznit mělo. Nejtěžší bylo zapojit a zprovoznit maticový displej, hned po vyladění paže. Robot měl reagovat na pohyb, což se povedlo díky pohybovému senzoru. U konstrukce se vyskytly problémy v komunikaci o vytvoření návrhu pro 3D tisk. Nakonec se vše v řádném termínu stihlo rychle vytisknout a sestavit dohromady. Z celkové práce mám dobrý pocit, jelikož jsem si nikdy nepomyslel, že bych mohl sestavit robota a jsem rád, že mi tuto možnost škola nabídla a zprostředkovala materiál.

# Seznam použité literatury a zdrojů

MONK, Simon. learn.adafruit.com [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <https://learn.adafruit.com/connecting-a-16x32-rgb-led-matrix-panel-to-a-raspberry-pi?view=all>

HORÁČEK, Petr. Linuxsoft.cz [online]. [cit. 22.3.2015]. Dostupný na  
WWW: [http://www.linuxsoft.cz/article.php?id\\_article=1953](http://www.linuxsoft.cz/article.php?id_article=1953)

<http://razzpisampler.oreilly.com/> [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <http://razzpisampler.oreilly.com/ch05.html>

Dawn robotics [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <http://blog.dawnrobotics.co.uk/2013/10/using-the-dagu-pantilt-kit-with-the-raspberry-pi/>

FEJT, Radek. Senzory [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <http://senzory.aspone.cz/pir.htm>

Pick up a Pi [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <http://www.pickupapi.com/blog/pitronics-part-7-servo-motors/>

Raspberry Pi Spy [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <http://www.raspberrypi-spy.co.uk/2013/01/cheap-pir-sensors-and-the-raspberry-pi-part-1/>

HENDERSON, Gordon. Gordons Projects [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <https://projects.drogon.net/raspberry-pi/gpio-examples/tux-crossing/>

Youtube [online]. [cit. 22.3.2015]. Dostupný na  
WWW: [https://www.youtube.com/watch?v=OJn44OjGJ\\_w](https://www.youtube.com/watch?v=OJn44OjGJ_w)

PÍŠTĚK, Ondřej. Ondrův web [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <http://www.pistek.eu/blog/2013/08/15/arduino-6-modelarska-serva/>

Youtube [online]. [cit. 22.3.2015]. Dostupný na  
WWW: <https://www.youtube.com/watch?v=58-vInmHsXA>

SKOPKA, Jan. 3D tiskárna Prusa i3, Uživatelský manuál. Spše v Úžlabině, Praha 10: 2014