# Obsah

**Jan Špecián**

**Lukáš Souček**

**Martin Pecina**

**2017/18**

**Čítání průchodů**

Semestrální práce

**Operační systémy**

[Obsah - 2 -](#_Toc503219496)

[Zadání - 2 -](#_Toc503219497)

[Použité nástroje - 2 -](#_Toc503219498)

[Schéma zapojení - 2 -](#_Toc503219499)

[Připojení přes terminál - 3 -](#_Toc503219500)

[Nastavení uživatele - 3 -](#_Toc503219501)

[Nastavení autostartu našeho programu - 4 -](#_Toc503219502)

[Program - 4 -](#_Toc503219503)

[Metoda pro odesílání emailů - 5 -](#_Toc503219504)

[Inicializace - 5 -](#_Toc503219505)

[Hlavní smyčka - 6 -](#_Toc503219506)

[Závěr - 7 -](#_Toc503219507)

[Zdroje - 7 -](#_Toc503219508)

# Zadání

Pomocí Raspberry Pi a pohybového čidla čítat průchody kočky na kočičí záchod a po sedmi průchodech oznámit emailem potřebu uklidit záchod.

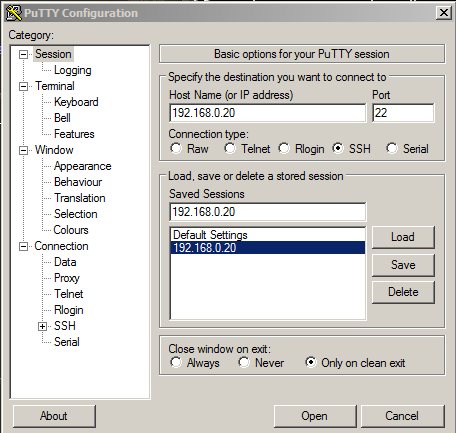
# Použité nástroje

* Raspberry pi 3 model B
* PIR modul SB00612A
* LED dioda s odporem pro 5V
* Operační systém Raspbian stretch lite
* Programovací jazyk Python
* Program Putty

# Schema.pngSchéma zapojení

# Připojení přes terminál

Router přidělil adresu 192.168.0.20. Zároveň připojil Raspberry pi do internetu.



# putty2.PNGNastavení uživatele

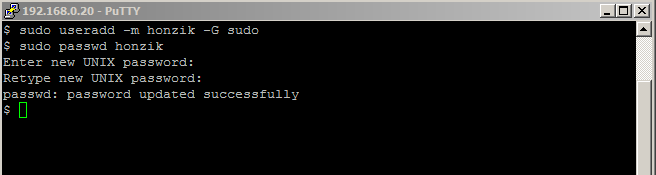
Vytvořili jsme nového uživatele uzivatel.

Vytvoří uživatele a vytvoří mu složku home, pokud ji nemá a vloží ho do skupiny práv sudo. Uživatele Pi jsme smazali příkazem:

sudo useradd -m uzivatel -G sudo

sudo passwd uzivatel

sudo deluser -remove-home pi



# Nastavení autostartu našeho programu

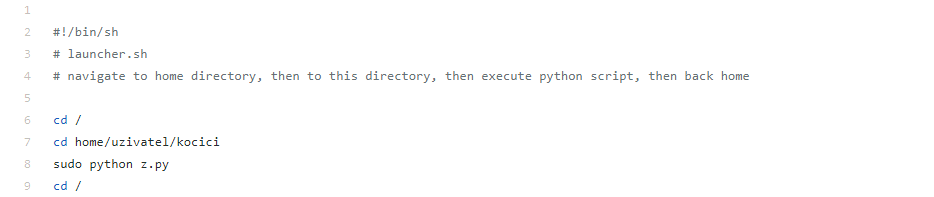
Chceme, aby se program spustil kdykoliv po zapnutí napájení, nebo po restartu systému.

nano launcher.sh

chmod 755 launcher.sh

home/mkdir logs

Skript pro spuštění našeho programu:



Crontab je proces na pozadí, který umožňuje spouštět skripty v určitých časech.

sudo crontab –e

Na konec připojím svojí direktivu pro akci po restartu:

@reboot sh /home/uzivatel/kocici/launcher.sh >/home/uzivatel/logs/cronlog 2>&1

Nyní po restartu poběží na Raspberry Pi náš program.

# Program

Program v jazyce Python používá knihovnu smtplib pro posílání emailů a knihovnu RPI.GPIO, pro přístup k vývodům z Raspberry.

|  |  |
| --- | --- |
|  | import random |
|  | from datetime import datetime |
|  | import smtplib  import time |
|  | import RPi.GPIO as GPIO |
|  |  |

# Metoda pro odesílání emailů

|  |
| --- |
|  |
|  | def send\_email(text):  to = 'jenda.specian@seznam.cz' |
|  | user = 'kocici.zachod@seznam.cz' |
|  | passwd = '<Kocicka>' |
|  | try: |
|  | smtpserver = smtplib.SMTP("smtp.seznam.cz",25) |
|  | smtpserver.ehlo() |
|  | smtpserver.starttls() |
|  | smtpserver.ehlo() # extra characters to permit edit |
|  | smtpserver.login(user, passwd) |
|  | header = 'To:' + to + '\n' + 'From: ' + user + '\n' + 'Subject:Upozorneni \n' |
|  | print(header) |
|  | msg = header + '\n ' + text +' \n\n' |
|  | try: |
|  | smtpserver.sendmail(user, to, msg) |
|  | print('Email poslan') |
|  | finally: |
|  | smtpserver.quit() |
|  | except Exception: |
|  | print("Nepovedlo se odeslat email: ") |
|  | print(text) |

# Inicializace

|  |
| --- |
|  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | LED = 18 # alias pro port |
|  | VSTUP = 23 # alias pro vstupni port - PIR cidlo |
|  | # privedenim logicke hodnity 1 na GPIO24 se ukonci program |
|  | #- nekonecna smycka do dalsiho restartu |
|  | UKONCENI = 24 # alias pro ukoncovaci pin |
|  |  |
|  | # konstanta, po kolika pruchodech posilat upozorneni |
|  | PO\_KOLIKA\_PRUCHODECH = 7 |
|  |  |
|  | GPIO.setmode(GPIO.BCM) |
|  | GPIO.setwarnings(False) |
|  |  |
|  | GPIO.setup(VSTUP,GPIO.IN) # vstupni port pro PIR |
|  | GPIO.setup(LED,GPIO.OUT) # vystupni port pro indikacni LED |
|  | GPIO.setup(UKONCENI,GPIO.IN) # vstupni port pro ukonceni programu |
|  |  |
|  |  |
|  |  |
|  | GPIO.output(LED,GPIO.LOW) # zhasnuti ledky na zacatku |
|  |  |
|  | pocet\_pruchodu = 0 # citani poctu aktivace PIR cidla |
|  |  |
|  | pracovat = True # navigacni promnna pro cyklus |

# Hlavní smyčka

|  |
| --- |
|  |
|  | while pracovat: |
|  |  |
|  | # ctu stav PIR cidla |
|  | state = GPIO.input(VSTUP) |
|  | if (state == True): |
|  |  |
|  | print("Pruchod") |
|  | print(pocet\_pruchodu) |
|  |  |
|  | # rozdvitim indikacni ledku |
|  | GPIO.output(LED, GPIO.HIGH) |
|  |  |
|  | #citam prichody |
|  | pocet\_pruchodu = pocet\_pruchodu + 1 |
|  | if(pocet\_pruchodu % PO\_KOLIKA\_PRUCHODECH == 0): |
|  | print("Vybrat zachod") |
|  | text = "Je potreba vybrat kocici zachod!\n " |
|  | with open("log.txt","r") as file: |
|  | text = text + file.read() |
|  | # vyprazdnim log soubor s casy pruchodu |
|  | open('log.txt', 'w').close() |
|  | #poslani emailu trva |
|  | #2minuty a Led dioda SVITI |
|  | send\_email(text) |
|  |  |
|  | #zapis do souboru |
|  | i = datetime.now() |
|  | # 2018/01/06 20:12:18 |
|  | # radek se zapiso do log souboru |
|  | radek\_pro\_zapis = i.strftime('%Y/%m/%d %H:%M:%S') |
|  | print(radek\_pro\_zapis) |
|  | with open("log.txt","a") as file: |
|  | file.write(radek\_pro\_zapis + "\n") |
|  |  |
|  | #pro testovani 3s |
|  | time.sleep(3.0) # cekani na dokonceni pouziti zachodu 5min, |
|  | else: |
|  | GPIO.output(LED, GPIO.LOW) |
|  |  |
|  | time.sleep(0.2) # perioda cteni z PIR cidla |
|  |  |
|  | # ctu stav ukoncivaciho vstupu - tlacitka |
|  | state2 = GPIO.input(UKONCENI) |
|  | if (state2 == True): |
|  | pracovat = False |
|  | print("Program skoncil") |
|  |  |
|  | if not pracovat: |
|  | GPIO.output(LED,GPIO.LOW) # zhasnuti indikacni ledky na konci |

# Závěr

* Poslání emailu trvá a zmrazí celé Raspberry na 40 sekund.
* Raspberry zapomíná kolik je hodin, pokud není připojené do internetu. Synchronizace času se nám podařila pouze manuálně.
* Nepájivé pole občas záhadnými zákmity aktivuje čidlo.
* Posílání emailu je překvapivě jednoduché pomocí seznam smtp serveru.
* Hlavní smyčku programu ukončujeme pomocí spínače, pro snadnější ukončování programu.

# Zdroje

* Raspberry Pi: Launch Python Script on Startup: 8 Steps. Instructables - How to make anything [online]. Copyright © 2017 Autodesk, Inc. [cit. 08.01.2018]. Dostupné z: <http://www.instructables.com/id/Raspberry-Pi-Launch-Python-script-on-startup/>
* Changing File Permissions « WordPress Codex. *Main Page « WordPress Codex* [online]. Dostupné z: <https://codex.wordpress.org/Changing_File_Permissions>
* SSH using Windows - Raspberry Pi Documentation. *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi* [online]. Dostupné z: <https://www.raspberrypi.org/documentation/remote-access/ssh/windows.md>
* How to Interface a PIR motion Sensor With Raspberry Pi GPIO . *Collection of the Best DIY Maker Projects | DIY Hacking*[online]. Copyright © EETech Media, LLC. All rights reserved. Unauthorized use and [cit. 09.01.2018]. Dostupné z: <https://diyhacking.com/raspberry-pi-gpio-control/>
* Stack Overflow - Where Developers Learn, Share, & Build Careers. *Stack Overflow - Where Developers Learn, Share, & Build Careers* [online]. Dostupné z: <https://stackoverflow.com/>