|  |  |  |
| --- | --- | --- |
| Logo__SSPU_2016_Barva | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Meteostanice s databází** | | |
| Dominik Večerek | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2019/2020 | |

Poděkování

*Chtěl bych poděkovat především svému spolužáku Denisu Kurkovi za pomoc s jazykem python a kivy*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2019

*podpis autora práce*

Anotace

Předmětem práce je vytvoření meteostanice, která zaznamenává teplotu vlhkost a atmosférický tlak, tyto hodnoty následně vypíše na displeji a jednou za půl hodiny je taky uloží do databáze. Srdcem meteostanice je jednodeskový počítač Raspberry Pi 4, který za pomoci sběrnice I2C sbírá data o teplotě a tlaku z čidla bmp280 a vlhkost ze senzoru htu21d který rovněž může měřit teplotu. Offline databáze je spravována za pomoci knihovny SQLite. Zařízení je napájené ze sítě pomocí adaptéru 5V 3A s konektorem USB-C. Zdrojový kód v jazyce Python byl upravován ve vývojovém prostředí Visual Studio Code.

Klíčová slova

Raspberry Pi 4; BMP280; HTU21D; meteostanice

OBSAH

[Úvod 5](file:///C:\\Users\\dominik\\Desktop\\dokumentace.docx" \l "_Toc29912470)

[1 Měření Vlkosti 6](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912471)

[1.1 Relativní vlkost 6](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912472)

[1.2 Praktické využití 6](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912474)

[1.2.1 Zvlhčovače 6](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912481)

[1.2.2 Odvlhčovače 6](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912481)

[2 Využité technologie 7](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912479)

[2.1 Hardware 7](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912480)

[2.1.1 Seznam součástek 7](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912481)

[2.1.2 Raspberry Pi 4 7](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912482)

[2.1.3 Tlakový senzor BMP280 7](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912483)

[2.1.4 Vlhkostní senzor HTU21D 7](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912484)

[2.1.5 Raspberry Pi Touch Display 8](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912485)

[4.2 Software 8](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912486)

[4.2.1 Seznam použitého softwaru 8](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912487)

[4.2.2 Visual Studio Code 8](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912488)

[4.2.3 Python 8](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912489)

[4.2.4 Kivy 8](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912490)

[3 způsoby řešení a použité postupy 9](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912491)

[3.1 Zdrojový kód 9](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912492)

[3.1.1 Získání dat 9](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912493)

[3.1.2 Výpis na obrazovku 9](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912493)

[3.1.3 Operace s databází 10](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912493)

[4 výsledky řešení 11](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912501)

[Závěr 12](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912513)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 13](file:///C:\Users\dominik\Desktop\dokumentace.docx#_Toc29912514)

# Úvod

Již před výběrem konkrétního projektu jsem věděl, že chci pracovat na nějakém projektu založeném na využití mikrokontroleru, tuto myšlenku jsem nakonec povýšil a využil jsem jednodeskový počítač.

Projekt zaměřený na meteostanici jsem si vybral z důvodu

V této dokumentaci popisuji proces vývoje a zhotovení meteostanice. V první části dokumentace se soustředím na popis a vlastnosti součástek a softwaru, v druhé se více přiblížím k problematice vytváření projektu.

# Měření Vlhkosti

## Relativní vlhkost

Pokud se mluví o vlhkosti vzduchu, rozlišuje se mezi relativní a absolutní vlhkostí. Při rozdílných hodnotách teploty může vzduch přijmout různá množství vlhkosti. Přitom zásadně platí, že čím vyšší je teplota, tím více vlhkosti může přijmout. Navíc existuje při každé teplotě určitý bod, při kterém už vzduch nemůže přijímat vlhkost. Tento bod se nazývá bod nasycení nebo rosný bod. Od tohoto bodu se začíná tvořit kondenzát. Na rozdíl od relativní vlhkosti vzduchu popisuje absolutní vlhkost vzduchu přesné množství vlhkosti obsažené ve vzduchu v gramech vody na kilogram vzduchu (g/kg).

## Praktické využití

### Zvlhčovače

Optimální vlhkost je důležitá z hygienických důvodů a z hlediska vlastností hygroskopických materiálů. Správná vlhkost vzduchu je rozhodující i v technologických provozech a laboratořích, pro snižování elektrostatického náboje, zachování vlastností produktů i surovin a snižování prašnosti. Řízení vlhkosti je mimořádně důležité i pro konzervaci a uložení uměleckých předmětů, archiválií a historických dokumentů.

### Odvlhčovače

Snižování vlhkosti je klíčové při ochraně zdraví a staveb její před nadměrnou hodnotou. Nízká vlhkost vzduchu je důležitá i v některých technologických procesech a pro uchovávání hygroskopických materiálů. Kondenzační odvlhčování má největší účinnost při teplotách nad 20°C a pro vlhkosti do 40% r.v. Adsorbční odvlhčování je nezastupitelné pro nízké vlhkosti a funguje ideálně i při nižších teplotách pod 15°C, kde již běžné kondenzační odvlhčení ztrácí účinnost.

# Využité technologie

## Hardware

### Seznam součástek

* Raspberry Pi 4
* Senzor tlaku a teploty BMP280
* Senzor vlhkosti a teploty HTU21D
* Raspberry Pi Touch Display
* USB-C 5,1V=3A napájecí zdroj

### Raspberry Pi 4

Jednodeskový počítač Raspberry Pi 4 je postaven na 1.5GHz čtyřjádrový procesor ARM Cortex-A72 a dokáže rozběhat až 2 monitory při rozlišení 4K. Aktuální verze Pi 4 vlastním ve verzi s 4GB operační paměti RAM. Minimální napájení je 15W a na vstupně-výstupních 40 pinech se nachází jak 5V tak i 3.3V s počítačem jsem komunikoval přes vzdálený přístup (za pomoci programu VNC).

### Senzor tlaku a teploty BMP280

Senzor BMP280 vyvíjený firmou Bosch zasazen do tištěného spoje firmy Adafruit, který zajišťuje napájení jak 5V tak i 3,3V a komunikaci jak přes I2C tak přes SPI

### Senzor vlhkosti a teploty HTU21D

Senzor HTU21D zasazen do tištěného spoje firmy Adafruit, který zajišťuje napájení jak 5V tak i 3,3V a komunikaci přes I2C

### Raspberry Pi Touch Display

Tento sedmi palcový dotykový display pro Raspberry Pi umožnuje použití jako tablet nebo informační display a další použití. Display s rozlišením 800x480 je připojen na adaptér, který je následně připojen k Raspberry. Display také podporuje použití všech deseti prstů. A je plně funkční i bez klávesnice nebo myšky

## Software

### Seznam použitého softwaru

* Visual Studio Code
* Python
* Kivy
* SQLite

### Visual Studio Code

Visual Studio Code je opensource multiplatformní editor zdrojových kódů, který jsem využil k psaní kódu v jazyce python a kivy. Součástí editoru je také debugger a instalátor rozšíření.

### Python

Python je interpretovaný vyšší programovací jazyk, který nabízí dynamickou kontrolu datových typů. Je designovaný pro jednoduché čtení a použití a běží na systémech Mac, Windows a Unix. Je zvažován jako skriptovací jazyk a často se používá pro vývoj webových aplikací.

### Kivy

Kivy je opensource knihovna Pythonu pro vývoj uživatelského prostředí a jako python lze použít na všech platformách. Kivy je vhodné pro vývoj android i počítačových aplikací. Tato knihovna t umožnuje psát uživatelské prostředí přímo do kódu, nebo do zvláštního souboru s příponou ‘.kv‘.

# Způsoby řešení a použité postupy

## Zdrojový kód

### Získání dat

Nastavení adresy vstupu do proměnné

bme28\_i2c = busio.I2C(3,2)

htu21d\_i2c = busio.I2C(1,0)

Získání adresy senzoru a nastavení do proměnné

bme280 = adafruit\_bmp280.Adafruit\_BMP280\_I2C(bme28\_i2c,0x76)

htu21d = HTU21D(htu21d\_i2c)

Získání konečných dat do proměnných

teplota = bme280.temperature

vlhkost = htu21d.relative\_humidity

tlak = bme280.pressure

### Výpis na obrazovku

Nachystání UI za pomoci kivy

def \_\_init\_\_(self, \*\*kwargs):

super(MyGrid, self).\_\_init\_\_(\*\*kwargs)

Nastavení zobrazení přes celé okno

self.cols = 1

self.casZapisu = 0

Nachystání tlačítek a jejich akcí

self.temButton = Button(text="25 °C")

self.temButton.bind(on\_press=self.temButtonPressed)

self.humButton = Button(text="22 %")

self.humButton.bind(on\_press=self.humButtonPressed)

self.preButton = Button(text="995 hPa")

self.preButton.bind(on\_press=self.preButtonPressed)

Připravení plochy pro výpis a nastavení na dva sloupce

self.mainLayout = GridLayout(cols=2)

Výpis textu

self.mainLayout.add\_widget(Label(text="Teplota: "))

Výpis předem připraveného tlačítka

self.mainLayout.add\_widget(self.temButton)

self.mainLayout.add\_widget(Label(text="Vlhkost: "))

self.mainLayout.add\_widget(self.humButton)

self.mainLayout.add\_widget(Label(text="Tlak: "))

self.mainLayout.add\_widget(self.preButton)

self.add\_widget(self.mainLayout)

Následná funkce update probíhající každých 5 sekund

self.temButton.text = str(round(MyApp.bme280.temperature,1))+"C"

self.humButton.text = str(round(MyApp.htu21d.relative\_humidity,1))+"%"

self.preButton.text = str(round(MyApp.bme280.pressure,1))+"hPa"

### Operace s databází

Funkce pro Zápis

def zapis(self, cas, teplota, vlhkost, tlak):

Připojení databáze

conn = sqlite3.connect('my.db')

c = conn.cursor()

Připravení dotazu

sqlite\_insert\_with\_param = """INSERT INTO data (cas, teplota, vlhkost, tlak)

VALUES(?,?,?,?);"""

Sloučení proměnných do jednoho pole

data\_tuple = (cas, teplota, vlhkost, tlak)

Sloučení dotazu s polem proměnných

c.execute(sqlite\_insert\_with\_param,data\_tuple)

Vyvolání dotazu – uložení proměnných do databáze

conn.commit()

Odpojení od databáze

conn.close()

Funkce pro čtení

def cteni(self):

conn = sqlite3.connect('my.db')

c = conn.cursor()

c.execute("SELECT \* FROM data

ORDER BY id

DESC LIMIT 10")

kam = c.fetchall()

conn.commit()

conn.close()

return kam

# výsledky řešení

# **Závěr**

Seznam použitýCH INFORMAČNÍCH ZDROJů