Středoškolská technika 2018

Setkání a prezentace prací středoškolských studentů na ČVUT

Meteostanice s Raspberry Pi

Jan Ruman

SPŠ a VOŠ Písek, Karla Čapka 402, 397 11 Písek

Anotace

Cílem práce bylo vytvořit malou domácí meteostanici, která by o počasí uživatele informovala jednoduchým a intuitivním způsobem. Jako zdroj informací by měla využívat internetový server.

Klíčová slova

Meteostanice; Raspberry Pi; Python; 3D tisk; OpenWeatherMap

Annotation

The goal of this work was to create small home weather station that would inform the user about weather situation in simple and intuitive fashion. It should use an internet server as a source of information.

Keywords

Weather station; Raspberry Pi; Python; 3D print; OpenWeatherMap

Obsah

1 ÚVOD

Inspirací k tomuto projektu pro mě byly hlavně klasické domácí meteostanice, které se na trhu pohybují již nějakou dobu. Ty většinou fungují na principu komunikace s měřícím zařízením, umístěným na vhodném místě venku. Takto získané informace bývají lokálně velmi přesné, což však není vždy náš primární požadavek. Pokud nám stačí informace z nejbližší meteorologické stanice, nepotřebujeme měřící zařízení vůbec.

Právě toto vedlo k nápadu vytvořit jednoduchou, graficky přívětivou domácí meteostanici. Místo externího modulu využívá dat dostupných na internetu, bez nutnosti konfigurace polohy a s jednoduchým připojením k internetu, ať už přes Wifi, nebo Ethernet.

2 POTŘEBY A VÝVOJOVÉ PROSTŘEDÍ

Potřeby a prostředí lze rozdělit na dvě skupiny: software a hardware. Do hardwaru patří samotný počítač a jeho příslušenství spolu s pouzdrem, software pak tvoří např. operační systém, který běží na Raspberry Pi.

2.1 Hardware

2.1.1 Raspberry Pi Model 2 B

Raspberry Pi je jednočipový počítač, který je srovnatelný se (slabším) stolním počítačem. Použitý mikroprocesor je z rodiny ARM, které lze nalézt i u běžného smartphonu. Počítač Raspberry Pi nabízí možnost instalace operačního systému Linuxu, RISC OS, nebo Microsoft Windows 10 IoT Core.

Ve srovnání s Arduinem má Raspberry Pi výhodu toho, že aplikace lze vyvíjet přímo na něm. Na desce je umístěno množství vstupů pro periferie, nabízí také možnost ovládání jiného zařízení pomocí GPIO kontaktů. Jako úložiště slouží SD karta. (1)

Obrázek 1: Raspberry Pi

2.1.2 3" LCD displej

K Raspberry Pi je připojen třípalcový TFT displej typu ILI9488 s rozlišením 400x240. Jeho velkou výhodou je nejen jeho pořizovací cena (250kč), ale i praktické zapojení přes GPIO piny, zajišťující pevné spojení s počítačem. Nevýhodou je nízká snímková frekvence (přibližně 5-10FPS) a obtížná instalace ovladačů.

Obrázek 2: LCD Displej

2.1.3 Síťová karta

Jednu z nejdůležitějších funkcí meteostanice – připojení k internetu – zajišťuje externí síťová karta, tzv. Wifi dongle, TL-WN722N V3. Operuje podle standardu IEEE 802.11bgn – až 150Mb/s, pásmo 2.4GHz. (2)

Obrázek 3: Síťová karta

2.1.4 Pouzdro

Kryt meteostanice je jedinou částí hardwaru, která nebyla zakoupena. Důvodem byla potřeba návrhu pouzdra, které by vyhovovalo všem požadavkům, především specifické velikosti displeje.

2.2 Software

2.2.1 Operační systém

Raspbian je výchozím operačním systémem Raspberry Pi založený na Debianu. Je dodáván s řadou předinstalovaných programů vhodných pro vzdělávání a programování, jako jsou např. Python, Scratch, Sonic Pi, Java, Mathematica a další. (3)

Obrázek 4: Raspbian

2.2.2 OpenWeatherMap API

Hlavní zdroj informací. Mapuje naměřené teploty k lokalitám a dává je k dispozici ve formě jak grafického rozhraní (webové stránky openweathermap.org), tak i JSON souboru na základě HTTP requestu.

Neplacená verze OpenWeatherMap API v současné době nabízí dva druhy informací, současný stav počasí a předpověď na pět dnů dopředu po 3 hodinách. Placená verze dále nabízí jak dlouhodobou předpověď a historická data, tak i možnost častějšího získávání informací. Tato práce využívá neplacené verze. (4)

2.2.3 Google Maps APIs

Společnost google nabízí zdarna několik rozhraní pro práci se zeměpisnou polohou. Zde budeme využívat Google Maps Geolocation API, sloužící k určení polohy a Google Maps Geocoding API, které použijeme pro získání názvu lokace na základě souřadnic. (5)

3 NÁSTROJE

3.1 Inventor Professional 2015

Autodesk Inventor je softwarová CAD aplikace firmy Autodesk. Slouží k tvorbě 3D modelů, výkresové dokumentace, prezentací a fotorealistické vizualizace a animace, ale lze využít i pro správu dokumentů a konstrukčních dat. (6)

Obrázek 5: Inventor logo

3.2 Python 2.7

Python je vysokoúrovňový skriptovací programovací jazyk. Nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmata, včetně objektově orientovaného, imperativního, procedurálního nebo funkcionálního. Operační systém Raspberry Pi podporuje Python bez potřeby instalace uživatelem. (7)

Obrázek 6: Python logo

3.3 Slic3r Prusa Edition

Slic3r slouží k převodu CAD modelů (formáty .stl a .obj) do formátu přijatelného pro 3D tiskárnu – G-code. Zároveň slouží k nastavení různých parametrů tisku, jako je např. šířka tisknuté vrstvy, použitý materiál nebo přítomnost podpěry. Upravená verze Prusa Edition nabízí lepší tisk díky vylepšené podpoře konkrétního druhu 3D tiskáren. (8)

Obrázek 7: Slic3r logo

3.4 Prusa i3 MK2S

Prusa i3 MK2S je 3D tiskárna určená pro tisk objektů o menším objemu. Tiskne nanášením vrstev rozehřátého filamentu, většinou z materiálů jako je ABS a PLA. Dodává se jako

Obrázek 8: 3D tiskárna

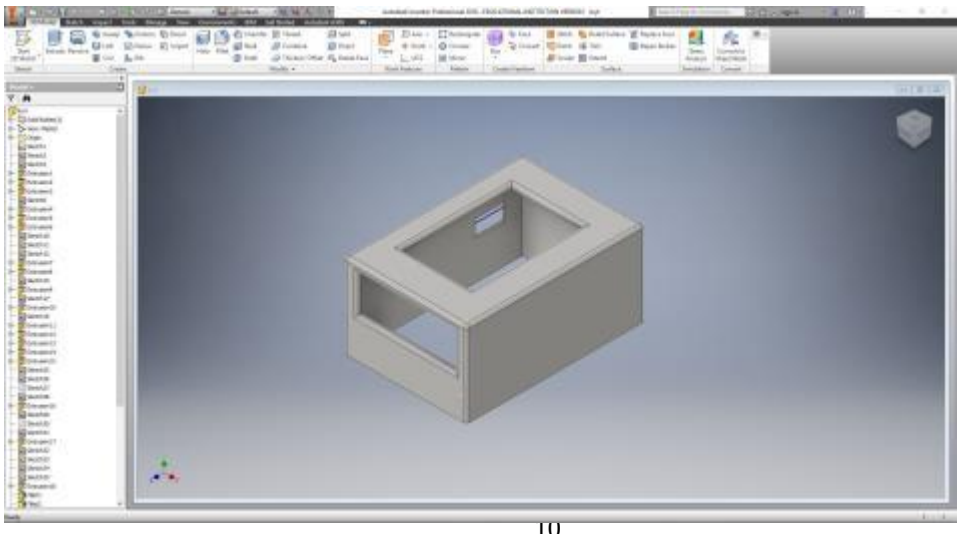
stavebnice. (9)

4 ČÁSTI METEOSTANICE

4.1 Pouzdro

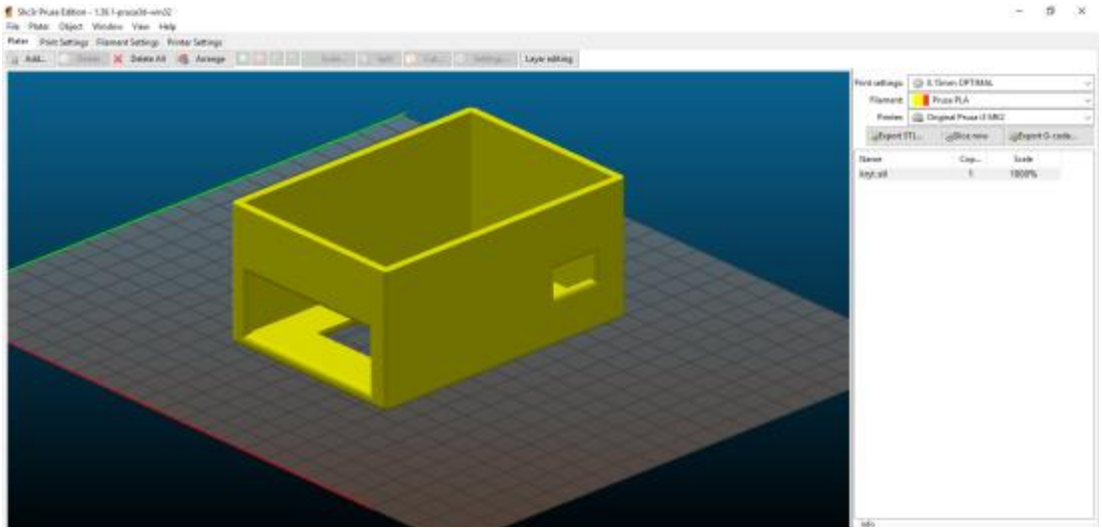
Pouzdro se skládá z předního a zadního krytu. Přední kryt plní většinu práce pouzdra. Je navržený tak, aby displej byl bez problémů vidět a zároveň nepřečnívali okraje displeje ven z meteostanice. Dále je uzpůsobený pro zapojení zdroje a síťové karty, popř. Ethernet kabelu. Zadní kryt slouží pouze k pevnému uzavření pouzdra.

K vytvoření 3D modelu jsem využil Inventor, především kvůli přehlednosti a možnosti pracovat s reálnými jednotkami vzdálenosti, což ne všechny podobné programy nabízejí.



Obrázek 9: Inventor model

Přípravu na tisk a převod do G-code jsem provedl v upravené verzi Slic3ru. Oba kryty jsem následně vytisknul na 3D tiskárně Prusa i3 MK2S.



Obrázek 10: Slic3r model

4.2 Program

Program, běžící na meteostanici, je rozdělen do několika souborů.

4.2.1 Hlavní soubor

Main.py byl vytvořen pro přehlednost, aby byl jasný soubor, kterým se aplikace spouští. Vytváří instanci třídy WeatherManager a spouští uživatelské rozhraní ze souboru gui.py.

```
import gui
import weather
from Tkinter import Tk

w = weather.WeatherManager()
root = Tk()
g = gui.GUI(root, w)
g.put_data()
root.mainloop()
```

4.2.2 Modul počasí

V souboru weather.py je uložena většina logiky programu. Obsahuje třídu WeatherManager. K získávání informací o počasí využívá knihovnu pyowm, umožňující jednoduchou

komunikaci se serverem openweathermap.org. K získání dat je potřeba OpenWeatherMap API klíč, požadovaná lokace a požadovaný časový úsek.

4.2.2.1 Inicializace, získání dat

Při inicializaci vytvoříme instanci OWM vložením API klíče a instanci modulu geolokace. Funkce observe zajišťuje získání souřadnic současné polohy a název místa, na kterém se meteostanice nalézá. Ty potom využije k získání dat o počasí v okolí. Je důležité upozornit Python interpreter na druh kódování textových řetězců – v našem případě UTF-8.

```
# coding=utf-8

import pyowm
import geo as g
import datetime
import dateutil.parser
import calendar

class WeatherManager:
    def __init__(self):
        self.owm = pyowm.OWM('bd494464200b3a800a37d4a20963fea2')
        self.place = g.Place()
        self.coords = None
        self.city = None
        self.position = None

    def observe(self):
        self.coords = self.place.getCoords()
        self.position = self.place.getPlace()
        parts1 = self.position.split(",")
        self.city = parts1[0]
        observation = self.owm.weather_at_coords(self.coords[0],
self.coords[1])
        return observation.get_weather()
```

4.2.2.2 Dnešní počasí

Předpověď pro dnešní den zajišťují funkce `getToday()` a `getTodayIcon()`. Obě využívají funkce `observe()`. `GetTodayIcon()` vrací cestu k obrázku zobrazujícím současný stav počasí. `GetToday()` vrací asociativní list údajů o současném počasí, datu a umístění.

```
    def getToday(self):
        w = self.observe()
        day =
calendar.day_name[dateutil.parser.parse(w.get_reference_time("iso"))
.weekday()]
        tmp = int(round(w.get_temperature('celsius')['temp']))
        sts = w.get_detailed_status()
        sts = sts[0].capitalize() + sts[1:]
        month = self.getMonthFromISO(w.get_reference_time('iso'))
        datum = self.getDayFromISO(w.get_reference_time('iso'))
        return {"temp":str(tmp) + "°", "sts":sts, "city":self.city,
"position":self.position, "icon":self.getTodayIcon(), "day":day,
"month":month, "datum":datum}

    def getTodayIcon(self):
        w = self.observe()
        icn = w.get_weather_icon_name()
        image = "icons/" + icn + ".png"
        return image
```

4.2.2.3 Předpověď

Předpověď je obsažena ve funkci `getForecast()`. Získaná data nejsou však ve vhodném formátu – předpověď začíná v nejbližším, ještě neuplynulém tříhodinovém úseku, což je však většinou ještě dnešní den. Funkce proto ignoruje data, vztahující se k dnešnímu dni a začíná pracovat až s těmi následujícími. Ty jsou vždy ve stejném formátu – 8 záznamů o stavu počasí po 3 hodinách. Pro naše potřeby je však vhodnější jeden údaj pro celý den, proto je teplota dne počítána jako průměr daných osmi hodnot. Stav se odvíjí od počasí v poledne daného dne.

```
    def getForecast(self):
        o = self.observe()
        w = self.owm.three_hours_forecast_at_coords(self.coords[0],
self.coords[1])
        forecast = w.get_forecast()
        forecasts = []
        mean = 0
        fs = 8
        icon = ""
        for f in forecast:
            if int(self.getDayFromISO(f.get_reference_time('iso')))
!= datetime.date.today().day:
                weekday =
calendar.day_name[dateutil.parser.parse(f.get_reference_time('iso'))
.weekday()]
                mean += f.get_temperature('celsius')['temp']
                if fs == 5:
                    icon = f.get_weather_icon_name()
                if fs > 1:
                    fs -= 1
                else:
                   forecasts.append({"temp":str(int(round(mean/8)))
+ "°", "day":weekday, "icon":"icons/" + icon + ".png"})
                    fs = 8
                    mean = 0
        return forecasts
```

4.2.2.4 Pomocné funkce

Funkce `getDayFromIso()` a `getMonthFromIso()` slouží pouze k získání dne a měsíce z textového řetězce ve formátu ISO 8601.

```
    def getDayFromISO(self, iso):
        parts1 = iso.split("-")
        parts2 = parts1[2].split(" ")
        return parts2[0]

    def getMonthFromISO(self, iso):
        parts1 = iso.split("-")
        return calendar.month_name[int(parts1[1])]
```

4.2.3 Modul grafického uživatelského rozhraní

Soubor gui.py se zabývá uživatelským rozhraním, k čemuž využívá grafickou knihovnu Tkinter. Sem se dostávají informace z modulu počasí, aby mohly být zobrazeny na displeji.

4.2.3.1 Inicializace

Při vytvoření instance GUI se získají data o počasí z příslušného argumentu. Dochází zde také k výběru barev a k nastavení vlastností samotného okna programu – název, kurzor, atd. Poté přichází rozdělení jednotlivých textových polí a obrázků do okna. Ty jsou nastaveny na pevné pozice, neměnné podle velikosti okna za pomocí Tkinter Place Geometry Manageru. Nebylo třeba řešit změnu velikosti okna, jelikož meteostanice má pevně dané rozlišení a program je určen pouze pro ni. Umístění a barva jednotlivých částí je otázkou designu.

```
# coding=utf-8

from PIL import Image, ImageTk
from Tkinter import Tk, Label, Button
import weather
import googlemaps
import pyowm

class GUI:
    def __init__(self, master, wet):
        self.master = master
        self.img = None
        self.tkImg = None
        self.wet = wet
        self.today_color = "#202020"
        self.other_day_color = "#353535"
        self.line_color = "#494949"
```

```
        self.text_color = "white"
        self.succ_request = False

        master.config(cursor="none")
        master.attributes("-fullscreen", True)
        # master.minsize(width=400, height=240)
        # master.maxsize(width=400, height=240)
        master.title("Weather")
        master.configure(background='white')

        self.bigImage = Label(master, bg=self.today_color)
        self.bigImage.place(width=160, height=155, x=0, y=0)

        self.bigTemp = Label(master, font=("Arial",48), padx=0,
pady=0, anchor="w", bg=self.today_color, fg="white")
        self.bigTemp.place(width=120, height=115, x=160, y=0)
...
```

Z praktických důvodů zde neuvádím celý kód. Zbytek inicializace je na principu posledních dvou odstavců (`self.bigImage = ...`).

4.2.3.2 Aktualizace okna

Funkce `put_data()` se stará o aktualizaci zobrazovaných informací, a také se stará o chyby spojené s připojením k internetu. Pokud se meteostanice zapne bez přístupu k internetu, bude na to upozorněno. V případě ztráty připojení k internetu zůstávají na displeji poslední získaná data.

```
    def put_data(self):
        print("updated")
        try:
            self.today = self.wet.getToday()
            self.forecast = self.wet.getForecast()
            self.succ_request = True
        except (googlemaps.exceptions.TransportError,
pyowm.exceptions.api_call_error.APICallError):
            if not self.succ_request:
                self.today = {"temp":"", "sts":"No internet
connection", "city":"", "position":"", "icon":"icons/error.png",
"day":"", "month":"", "datum":""}
                self.forecast = [{"temp":"", "day":"",
"icon":"icons/error.png"}]*4
```

Ikony stavu počasí nejsou ve formátu vhodném pro Tkinter, proto jsou formátovány pomocí knihovny PIL. Poté jsou spolu s dalšími informacemi vloženy do textových polí a polí pro obrázky.

```
        # Today
        self.todayImage = Image.open(self.today["icon"])
        self.todayImage = self.todayImage.resize((110,110),
Image.ANTIALIAS)
        self.todayImage = ImageTk.PhotoImage(self.todayImage)

        self.bigImage.config(image=self.todayImage)
        self.bigImage.img = self.todayImage
        self.bigTemp.config(text=self.today["temp"])
...
        # Day 3
        self.smallImage3 = Image.open(self.forecast[1]["icon"])
        self.smallImage3 = self.smallImage3.resize((30,30),
Image.ANTIALIAS)
        self.smallImage3 = ImageTk.PhotoImage(self.smallImage3)

        self.day3up.config(text=self.forecast[1]["day"])
        self.day3temp.config(text=self.forecast[1]["temp"])
        self.day3pic.config(image=self.smallImage3)
        self.day3pic.img = self.smallImage3
...
```

Důležitou částí aktualizace počasí je funkce `after()`, která se stará o neustálé opakování funkce `put_data()` v intervalu 5 minut.

```
        self.master.after(300000, self.put_data)
```

4.2.4 Modul geolokace

Soubor geo.py zprostředkovává údaje o poloze. Zjišťuje, na jakých souřadnicích se meteostanice nalézá a jak se daná lokace nazývá. Využívá k tomu Google Maps Geolocation API a Google Maps Geocoding API skrze knihovnu googlemaps. K určení místa nedochází pomocí modulu GPS připojenému přímo k Raspberry Pi. Při komunikaci s googlemaps dochází k předávání informací o geolokaci z bodu, který je co nejblíže zařízení a zároveň má k těmto datům přístup.

```
import googlemaps

class Place:
    def __init__(self):
        self.gmaps = None
        self.APIKey = 'AIzaSyBzhe9oVTn0YixpxnGDs4ZnkXar_ivldts'

    def getLoc(self):
        self.gmaps = googlemaps.Client(key=self.APIKey)
        return self.gmaps.geolocate()["location"]

    def getCoords(self):
        loc = self.getLoc()
        return [loc["lat"], loc["lng"]]

    def getPlace(self):
        loc = self.getLoc()
        full_address = self.gmaps.reverse_geocode((loc["lat"],
loc["lng"]))[0]["formatted_address"]
        parts = full_address.split(" ")
        place = parts[-2] + " " + parts[-1]
        return place.encode('utf-8')
```

5 SLOŽENÍ METEOSTANICE

5.1 Vývoj

5.1.1 Původní verze

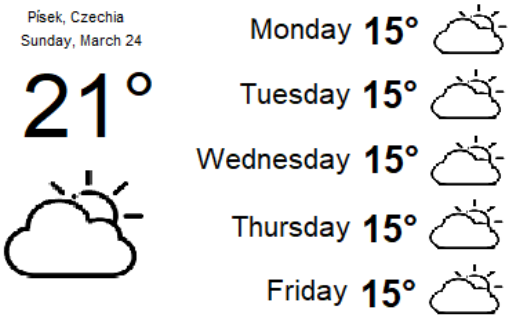
Jednotlivé verze programu se liší především druhem informací a grafickou podobou uživatelského rozhraní. Za zmínku stojí původní program, který zobrazoval pouze současnou

Obrázek 11: Původní verze

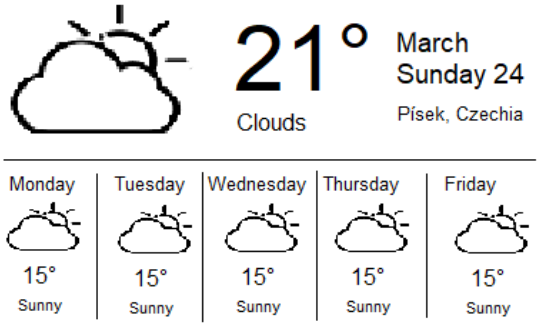
předpověď, zato detailnější. Ikony používal přímo ze sady OpenWeatherMap.

5.1.2 Grafické koncepty

Před vytvořením současného designu bylo potřeba zvážit i jiné možnosti vzhledu uživatelského rozhraní. Během vývoje jich vzniklo celkem 12. Pro časovou náročnost naprogramování jednotlivých konceptů jsem jako nástroj zvolil MS Paint, a to pro jeho



Obrázek 13: Koncept 1

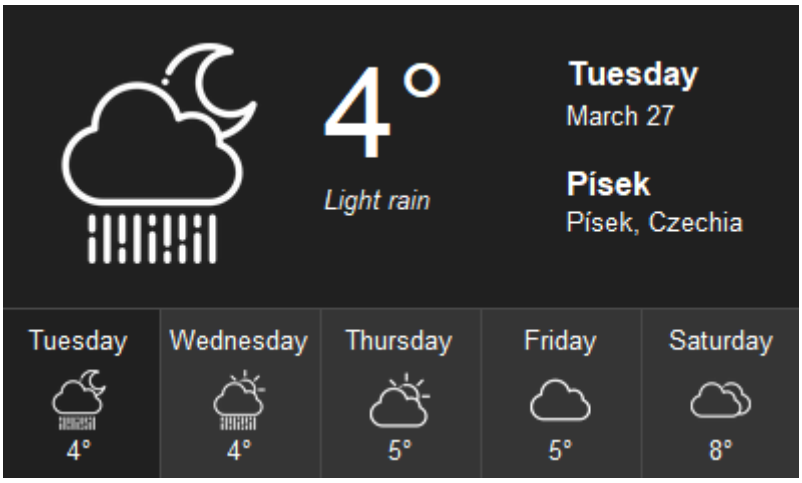


Obrázek 12: Koncept 2

jednoduchost.

5.2 Finální verze

Finální verze programu je rozebraná výše v kapitole Části meteostanice. Narozdíl od verze původní jsou zde využity jiné ikony počasí[1]. Spolu s nahráním programu do Raspberry Pi bylo potřeba také nainstalovat ovladače, změnit některá nastavení Raspbianu a smontovat pouzdro. Poté zbývalo pouze vše připojit a spustit. Na závěr jsem meteostanici umístil na podstavec ze stavebnice LEGO, který umožňuje přístup kabelu adaptéru do sítě. Program se spouští při přihlášení uživatele, to v našem případě probíhá automaticky, lze tedy říct, že se meteostanice dá zapnout pouhým zapojením do sítě.



Obrázek 15: Finální verze program

Obrázek 14: Finální verze meteostanice

[1] Použité ikonky jsou od autorů Freepik a Elias Bikbalutov z webu Flaticon, licence Flaticon Base License

6 ZÁVĚR

Práce byla zaměřená na tvorbu domácí meteostanice, využívající dat dostupných online. Toho jsem dosáhl spojením několika knihoven a vypracováním vlastních dílčích částí, které jsem potřeboval.

Do budoucna se nabízí možnost přenosu programu na platformu uzpůsobenou potřebám aplikace. Došlo by tím ke snížení nákladů a lepší účinnosti zařízení.

Kompletní zdrojový kód i se všemi potřebnými obrázky je dostupný na GitHubu.

SEZNAM OBRÁZKŮ

POUŽITÁ LITERATURA

1. Raspberry Pi. *Wikipedia.* [Online] 31. 3 2018. https://en.wikipedia.org/wiki/Raspberry_Pi.

2. Download for TL-WN722N V3 . *tp-link.* [Online] 31. 3 2018. https://static.tp-link.com/TL-WN722N(EU&US)_3.0.pdf.

3. Raspbian. *Raspberry Pi.* [Online] 31. 3 2018. https://www.raspberrypi.org/downloads/raspbian/.

4. Open Weather Map. *Open Weather Map.* [Online] 31. 3 2018. https://openweathermap.org/.

5. Google Maps APIs. *Google.* [Online] 31. 3 2018. https://developers.google.com/maps/.

6. Autodesk Inventor. *Wikipedia.* [Online] 31. 3 2018. https://cs.wikipedia.org/wiki/Autodesk_Inventor.

7. Python. *Wikipedia.* [Online] 31. 3 2018. https://cs.wikipedia.org/wiki/Python.

8. About. *Slic3r.* [Online] 31. 3 2018. http://slic3r.org/about.

9. 3D tiskárna stavebnice prusa i3. *Prusa3D.* [Online] 31. 3 2018. https://www.prusa3d.cz/3d-tiskarna-stavebnice-prusa-i3/.

10. Raspberry Pi. *Wikipedia.* [Online] https://en.wikipedia.org/wiki/Raspberry_Pi.

PŘÍLOHA 1: ZDROJOVÝ KÓD

Zdrojový kód: https://github.com/rumaak/rpi_weather.git

Součástí je i sada ikon.