

Stacja pogodowa na Raspberry Pi

Fundacja Raspberry Pi wysłała do szkół na całym świecie ponad 10 tysięcy stacji pogodowych opartych na module HAT z zestawem potrzebnych sensorów. Teraz każdy w domu może skonstruować taką stację pogodową na bazie zaproponowanych modułów z sensorami lub po prostu samodzielnie łącząc układy scalone z komputerem jednopłytkowym Raspberry Pi.

Opisywany projekt stworzony został przez Fundację Raspberry Pi, jako prezentacja ciekawego wykorzystania tego komputera jednopłytkowego. Każdy może skonstruować zaprezentowaną stację pogodową lub też wykorzystać poszczególne elementy, aby zintegrować je ze stacją pogodową własnego projektu. Projekt pokazuje, jak podłączone i oprogramowane są poszczególne sensory, więc nadaje się także dla osób, które wcześniej nie miały zbyt wielkiego doświadczenia

z Raspberry Pi, programowaniem w Pythonie czy systemem operacyjnym Linux.

Co będzie potrzebne

Na rynku dostępnych jest wiele sensorów, które można wykorzystać do konstrukcji stacji meteorologicznej. Nie ma konieczności wykorzystywania dokładnie tych samych układów, jakie użyte są w tym projekcie, jednakże, jeżeli zmienimy wykorzystywane sensory, może wystąpić konieczność wprowadzenia zmian do bibliotek i programów napisanych w Pythonie, które zaprezentowane są poniżej.

O wyborze konkretnych sensorów w tego rodzaju projektach decyduje szereg cech, takich jak dostępność, koszty, wsparcie dla Linuksa i Pythona (które przekłada się na łatwość implementacji) oraz niezawodność i dokładność pomiaru. Nie oznacza to, że elementy wymienione poniżej są najtańsze albo najdokładniejsze – to kompromis

między tymi cechami, ze szczególnym uwzględnieniem prostoty implementacji i dobrego wsparcia dla Raspberry Pi. Jeżeli w naszym projekcie co innego jest najważniejsze, to możemy dobrać inne sensory.

W tym projekcie wykorzystano następujące sensory:

- Scalony sensor ciśnienia, temperatury i wilgotności BME280.
- Termometr elektroniczny DS18B20 (jako zewnętrzna sonda do pomiaru temperatury gruntu).
- Anemometr, anemoskop, pluwiometr z wyjściem RJ11 i pasujące gniazdzka.
- Przetwornik analogowo-cyfrowy MCP3008.

Oprócz wymienionych powyżej elementów przydatna będzie wodoszczelna obudowa – tę dobrać można we własnym zakresie, aby zmieścił się w niej moduł Raspberry Pi wraz z wszystkim modułami. Wszystko uzależnione jest od tego, jakie sensory wykorzystamy i w jaki sposób podłączymy je z używanym komputerem.

Oprogramowanie

Korzystać będziemy z części skryptów zawartych w pakiecie Oracle Raspberry Pi Weather Station. Nie musimy go w żaden sposób instalować, wystarczy sklonować repozytorium z GitHuba – w terminalu Linuksa na Raspberry Pi wpisujemy:

```
git clone https://github.com/RaspberryPiFoundation/weather-station
```

Następnie instalujemy Pythonowe biblioteki do obsługi układu BME280:

```
sudo pip3 install RPi.bme280
```

A także serwer bazy danych MariaDB, z którego korzysta stacja pogodowa:

```
sudo apt-get install -y mariadb-server mariadb-client libmariadbclient-dev
sudo pip3 install mysqlclient
```

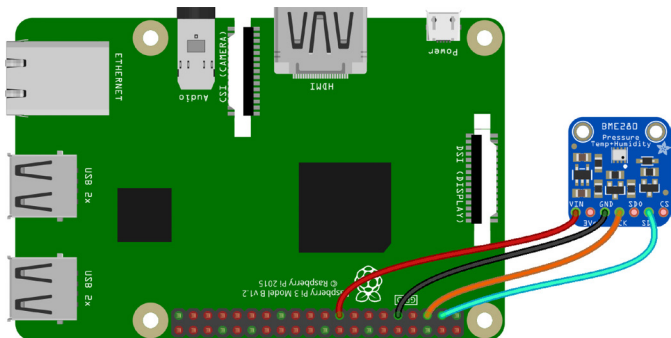
Pomiar wilgotności, temperatury i ciśnienia

Układ BME280 to przedstawiciel rodziny cyfrowych sensorów temperatury, wilgotności i ciśnienia atmosferycznego. Układ ten najprościej dołączyć do Raspberry Pi, wykorzystując gotowy moduł na płytce drukowanej. W tym projekcie wykorzystano moduł tego rodzaju, produkowany przez firmę Adafruit.

BME280 wyposażony jest w interfejsy I²C (adresy 0x77 i 0x77 konfigurowane pinem) oraz SPI do komunikacji z układem. W module Adafruit układ skonfigurowany jest do komunikacji poprzez I²C pod adresem 0x77.

Aby podłączyć układ do Raspberry Pi, połączyć musimy obie linie interfejsu I²C oraz zasilanie do modułu z sensorem. Na **rysunku 1** pokazano, jak podłączyć wejścia VIN, GND, SCK i SDA modułu do wyprowadzeń Raspberry Pi.

Po podłączeniu sensora i uruchomieniu komputera możemy zacząć tworzyć skrypt, który obsługiwać będzie ten sensor. W tym celu uruchamiamy dowolny edytor – nano, vim lub IDLE3 – i tworzymy plik `/home/pi/weather-station/bme280_sensor.py`.



Rysunek 1. Sposób podłączenia modułu Adafruit do Raspberry Pi

```
Listing 1.
import bme280 # biblioteka do obsługi sensora
import smbus2 # biblioteka do obsługi interfejsu I2C
from time import sleep # funkcja sleep do robienia pauzy w programie
port = 1
address = 0x77 # Adres BME280
bus = smbus2.SMBus(port) # Uruchomienie interfejsu I2C
bme280.load_calibration_params(bus, address) # Inicjalizacja sensora
while True: # nieskończona pętla
    bme280_data = bme280.sample(bus, address) # odczyt danych
    humidity = bme280_data.humidity
    pressure = bme280_data.pressure
    ambient_temperature = bme280_data.temperature
    print(humidity, pressure, ambient_temperature) # wydruk danych w terminalu
    sleep(1) # pauza - program czeka 1 sekundę
```

Skrypt z **listingu 1**, po uruchomieniu w Pythonie 3, powinien co sekundę w terminalu podawać aktualną wilgotność, ciśnienie i temperaturę. Aby sprawdzić, czy mierzone są one poprawnie, wystarczy np. chuchnąć na sensor, aby zobaczyć, czy prezentowane dane zmieniają się zgodnie z oczekiwaniem.

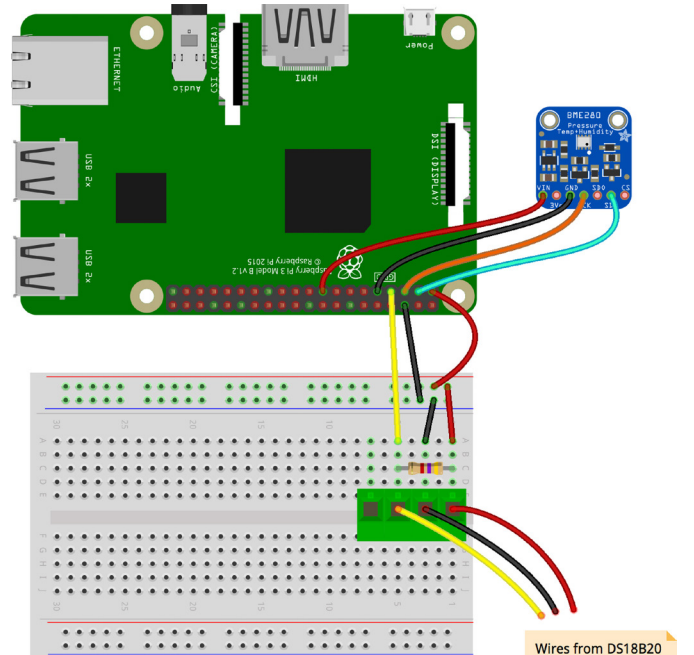
Teraz możemy zmienić `while True:` w skrypcie, na `def read_all():` czyli definicję funkcji `read_all()`, która odpowiedzialna będzie za odczyt z sensora. Aby funkcja zwracała dane, zamiast drukować je w terminalu `print(...)`, zamieniamy na `return humidity, pressure, ambient_temperature`. Usuwamy także pauzę z końca skryptu.

Temperatura gruntu

BME280 będzie podawać temperaturę powietrza, ale może być ona zupełnie inna niż temperatura gruntu. Powietrze może być np. znacznie cieplejsze niż grunt, szczególnie jeśli jest zmrożona. Sonda do pomiaru temperatury, wetknięta w glebę, jest przydatnym dodatkowym sensorem w stacji meteorologicznej i może być wykorzystana do wskazania np. obecności lodu/mrozu w zimie. Do pomiaru temperatury wykorzystano czujnik DS18B20. Występuje on w wielu wersjach, w tym w wersji z wodoodporną sondą termiczną. Taki właśnie czujnik zastosować możemy w budowanej stacji pogodowej.

Sensor DS18B20 ma trzy wyprowadzenia – masę, zasilanie i dane. Najprościej jest podłączyć go, wykorzystując płytkę stykową lub płytkę uniwersalną, tak jak pokazano na **rysunku 2**, czyli zasilając układ z napięcia 3,3 V i podciągając linię danych do zasilania. Linię danych oczywiście łączymy z Raspberry Pi.

Sensor temperatury wykorzystuje do komunikacji interfejs 1-wire. Jest to szeregowy interfejs komunikacyjny, wykorzystujący zaledwie jedną linię danych. Jest on obecny w Raspberry Pi, ale domyślnie nie jest włączony. Aby to zmienić, otwieramy z pomocą dowolnego



Rysunek 2. Sposób podłączenia termometru DS18B20 do Raspberry Pi

edytora plik konfiguracyjny `/boot/config.txt` i dodajemy na jego końcu, w nowej linii `dtoverlay=w1-gpio`. Następnie dodajemy w pliku `/etc/modules` moduły do obsługi interfejsu 1-wire, które mają być załadowane podczas startu systemu operacyjnego: `w1-gpio` i `w1-therm`. Możemy teraz zresetować komputer, aby wprowadzone zmiany weszły w użycie.

Teraz możemy skorzystać ze skryptu, który pobraliśmy wcześniej z GitHuba. Powinien on znajdować się tutaj: `/home/pi/weather-station/ds18b20_therm.py`. Uruchamiamy go z pomocą IDLE3 lub bezpośrednio (po wejściu do folderu z plikiem) wpisując w linii komend:

```
python3 ds18b20_therm.py
```

Powinniśmy zobaczyć w terminalu temperaturę odczytywaną za pomocą tego sensora. Możemy włożyć go do ciepłej lub zimnej wody, aby zobaczyć zmiany odczytywanej temperatury i potwierdzić działanie sensora.

Anemometr i anemoskop – mechaniczne sensory wiatru

Dotychczasowo wszystkie podłączone sensory, były pasywnymi, elektronicznymi sensorami – układem scalonym, który biernie tkwił w środowisku, którego parametry mierzył. Z urządzeniem do pomiaru prędkości i kierunku wiatru – anemometrem i anemoskopem – jest inaczej. Są to urządzenia mechaniczne, które poruszają się zgodnie z wiatrem; ich ruch jest monitorowany przez układ elektroniczny, który pozwala na monitorowanie stanu sensora przez Raspberry Pi.

W prezentowanym projekcie wykorzystano zestaw firmy Argent Data Systems numer 80422, który składa się z anemometru do pomiaru prędkości wiatru, anemoskopu do pomiaru jego kierunku i pluwiometru do pomiaru ilości opadów atmosferycznych. Urządzenia te wyposażone są we wtyczki RJ11; dlatego też, aby podłączyć je do Raspberry Pi, zastosować możemy przejściówki, takie jak pokazano na **fotografii 3** lub po prostu obciąć wtyki i podłączyć kable bezpośrednio do pinów komputera lub poprzez płytkę uniwersalną.

Anemometr, jaki wykorzystany jest w tym projekcie, jest niezwykle prosty. Ma on trzy ramiona z czaszami na końcu. Są one popychane przez wiatr, co powoduje, że cała konstrukcja obraca się wokół własnej osi. W obracającym się elemencie anemometru zainstalowany jest magnes stały, a w jego podstawie znajduje się kontaktron. Zwiera się on w momencie, gdy magnes zainstalowany w wiatraku, znajdzie się w odpowiedniej pozycji. Na jeden pełny obrót wiatraka przypadają dwie takie pozycje.

Z punktu widzenia elektrycznego, kontaktron zachowuje się tak samo, jak np. przycisk. Dlatego też sygnały z anemometru monitorować można jak naciśnięcie przycisku. Aby zmierzyć prędkość wiatru wystarczy zmierzyć ile razy na sekundę do wejścia GPIO w Raspberry docierają impulsy z kontaktronu. Anemometr wyprowadzony jest na pinach 3 i 4 złącza RJ11. Podłączamy jeden z nich do masy, a drugi do wejścia GPIO 5 w Raspberry Pi. Do odczytu stanu układu wykorzystywać będziemy pythonową bibliotekę `gpiozero`. Potrzebny jest nam prosty skrypt;

```
Listing 2.
from gpiozero import Button # biblioteka do obsługi przycisków
wind_speed_sensor = Button(5) # konfigurujemy przycisk na GPIO 5
wind_count = 0 # licznik obrotów anemometru
def spin(): # funkcja uruchamiana przyciskiem
    global wind_count
    wind_count = wind_count + 1 # inkrementacja licznika
def reset_wind(): # funkcja resetowania licznika
    global wind_count
    wind_count = 0
wind_speed_sensor.when_pressed = spin # przypisanie funkcji do przycisku
```

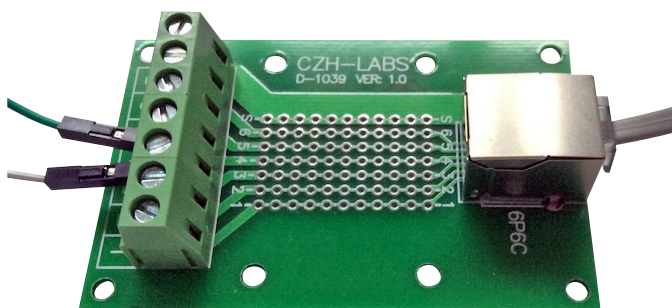
```
Listing 3.
import statistics
store_speeds = []
while True:
    start_time = time.time()
    while time.time() - start_time <= wind_interval:
        reset_wind()
        time.sleep(wind_interval)
        final_speed = calculate_speed(wind_interval)
        store_speeds.append(final_speed)
    wind_gust = max(store_speeds)
    wind_speed = statistics.mean(store_speeds)
    print(wind_speed, wind_gust)
```

```
Listing 4.
from gpiozero import MCP3008 # biblioteka do obsługi ADC
import time
adc = MCP3008(channel=0) # inicjalizacja ADC
while True:
    print(adc.value) # wydruk zmierzonej wartości
```

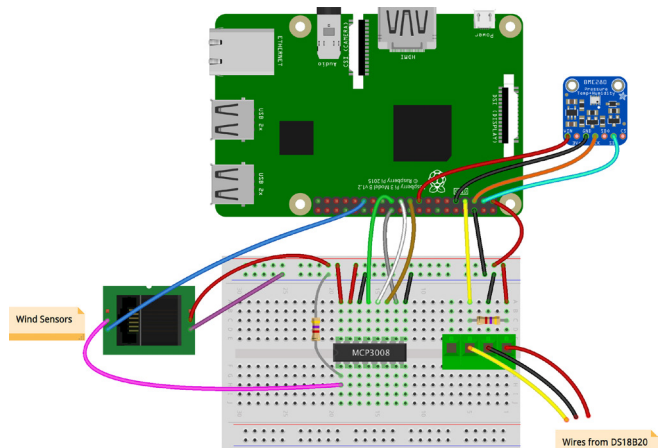
tworzymy plik w `/home/pi/weather-station/wind.py`, w którym wpisać możemy nasz program przedstawiony na **listingu 2**.

Teraz pozostało jedynie sprawdzać co ustalony czas wartość licznika i wyliczać na podstawie tego prędkość wiatru. Karta katalogowa anemometru podaje, że jeden impuls na sekundę oznacza wiatr wiejący z prędkością 2,4 km/h. Uśrednianie prędkości wiatru w dużych odstępach czasu nie jest jednak najlepszym pomysłem – zupełnie

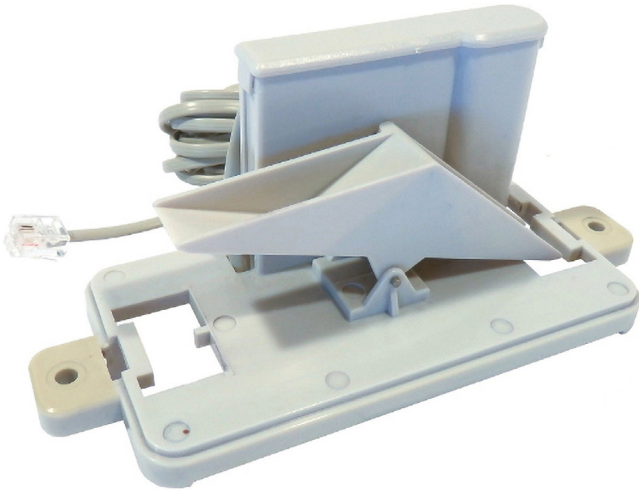
```
Listing 5.
from gpiozero import MCP3008 # biblioteka do obsługi ADC
import time
adc = MCP3008(channel=0) # inicjalizacja ADC
count = 0
volts = {0.4: 0.0, # słownik zawierający wszystkie napięcia i kąty
1.4: 22.5,
1.2: 45.0,
2.8: 67.5,
2.7: 90.0,
2.9: 112.5,
2.2: 135.0,
2.5: 157.5,
1.8: 180.0,
2.0: 202.5,
0.7: 225.0,
0.8: 247.5,
0.1: 270.0,
0.3: 292.5,
0.2: 315.0,
0.6: 337.5}
while True:
    wind = round(adc.value*3.3,1) # odczyt z ADC i konwersja do napięcia
    if not wind in volts:
        print('unknown value ' + str(wind) + ' ' + str(volts[wind]))
    else:
        print('found ' + str(wind) + ' ' + str(volts[wind]))
```



Fotografia 3. Przejściówka do podłączenia RJ11 do Raspberry Pi



Rysunek 4. Podłączenie anemoskopu i przetwornika MCP3008 do Raspberry Pi



Fotografia 5. Wnętrze wykorzystanego w projekcie pluwiometru

ignoruje to szybsze podmuchy wiatru – nie wieje on jednostajnie. Dlatego też dodać można skrypt obliczający nie tylko średnią, ale też maksymalną prędkość wiatru w danym okresie. Do skryptu z listingu 2 dopisujemy zawartość listingu 3.

Anemoskop jest bardziej skomplikowanym urządzeniem. Ruchomy element anemoskopu także wyposażony jest w magnesy, ale w podstawie znaleźć można aż osiem kontaktów, które przełączają oporniki. W zależności od kierunku wiatru inna będzie rezystancja układu.

Do pomiaru rezystancji przez Raspberry Pi wykorzystany zostanie dzielnik napięcia oraz przetwornik analogowo-cyfrowy MCP3008, którego zadaniem będzie digitalizacja napięcia wyjściowego z dzielnika. Dzielnik w tym przypadku zasilany jest napięciem 3,3 V i składa się z opornika 4,7 kΩ i anemoskopu. MCP3008 wyposażony jest w interfejs SPI, który podłączony jest do Raspberry Pi, jak pokazano na rysunku 4. Układ obsługiwany jest przez bibliotekę gpiozero, dzięki czemu korzystanie z niego jest bardzo proste.

Z pomocą prostego skryptu (listing 4) sprawdzamy, czy układ działa poprawnie.

Program ten będzie cały czas mierzył wartość napięcia. Powinniśmy być w stanie wykręcić 16 stabilnych wartości ADC – tyle jest możliwych pozycji anemoskopu i tyle jest przypadków rezystancji tego urządzenia. Możemy teraz stworzyć program, który będzie podawał nam, jaki jest dokładnie kąt ustawienia anemoskopu (listing 5).

```
Listing 6.
from gpiozero import Button
rain_sensor = Button(6)
BUCKET_SIZE = 0.2794
count = 0
def bucket_tipped():
    global count
    count = count + 1
    print(count * BUCKET_SIZE)
def reset_rainfall():
    global count
    count = 0
rain_sensor.when_pressed = bucket_tipped
```

Pluwiometr do pomiaru ilości opadów deszczu

Pluwiometr to urządzenie służące do pomiaru ilości opadu deszczu w milimetrach w danym obszarze, zazwyczaj na powierzchni jednego metra kwadratowego. Istnieją różne urządzenia tego typu, wykorzystujące rozmaite sensory do realizacji opisanego pomiaru. Zaprezentowany pluwiometr Argent Data Systems jest prostym mechanizmem, pozwalającym na szybkie pomiary ilości deszczu. Wykorzystuje on niewielki element z dwoma zagłębieniami działającymi jak kołyska. Po zapełnieniu się jednego z nich, element ten przechyla się, opróżniając pojemnik i naciskając prosty przycisk. W tej pozycji, wodą napełnia się drugie zagłębienie i cały cykl się powtarza. Na fotografii 5 pokazano, jak wygląda wnętrze urządzenia.

```
Listing 7.
from gpiozero import Button
import time
import math
import bme280_sensor
import wind_direction_byo
import statistics
import ds18b20_therm
wind_count = 0 # zlicza liczbę obrotów anemometru
radius_cm = 9.0 # średnica anemometru
wind_interval = 5 # co ile sekund mierzona jest prędkość wiatru
interval = 5 # co ile minut zapisywane są pomiary
CM_IN_A_KM = 100000.0
SECS_IN_AN_HOUR = 3600
ADJUSTMENT = 1.18 # współczynnik korekcyjny dla anemometru
BUCKET_SIZE = 0.2794 # wielkość pojemnika w pluwiometrze
rain_count = 0
gust = 0
store_speeds = []
store_directions = []
def spin():
    global wind_count
    wind_count = wind_count + 1
def calculate_speed(time_sec):
    global wind_count
    global gust
    circumference_cm = (2 * math.pi) * radius_cm
    rotations = wind_count / 2.0
    dist_km = (circumference_cm * rotations) / CM_IN_A_KM
    km_per_sec = dist_km / time_sec
    km_per_hour = km_per_sec * SECS_IN_AN_HOUR
    final_speed = km_per_hour * ADJUSTMENT
    return final_speed
def bucket_tipped():
    global rain_count
    rain_count = rain_count + 1
def reset_rainfall():
    global rain_count
    rain_count = 0
def reset_wind():
    global wind_count
    wind_count = 0
def reset_gust():
    global gust
    gust = 0

wind_speed_sensor = Button(5)
wind_speed_sensor.when_activated = spin
temp_probe = ds18b20_therm.DS18B20()
while True: # główna pętla programu
    start_time = time.time()
    while time.time() - start_time <= interval:
        wind_start_time = time.time()
        reset_wind()
        while time.time() - wind_start_time <= wind_interval:
            store_directions.append(wind_direction_byo.get_value())
            final_speed = calculate_speed(wind_interval)
            store_speeds.append(final_speed)
        wind_average = wind_direction_byo.get_average(store_directions)
        wind_gust = max(store_speeds)
        wind_speed = statistics.mean(store_speeds)
        rainfall = rain_count * BUCKET_SIZE
        reset_rainfall()
        store_speeds = []
        store_directions = []
        ground_temp = temp_probe.read_temp()
        humidity, pressure, ambient_temp = bme280_sensor.read_all()
        print(wind_average, wind_speed, wind_gust, rainfall, humidity, pressure, ambient_temp, ground_temp)
```


Do przechylenia zbiornika potrzebne jest, jak podaje karta katalogowa, 0,2794 mm deszczu. Tę wartość mnożymy razy liczbę impulsów, jaka została zliczona w jednostce czasu – podobnie jak realizowany był pomiar z anemometru. Wyjście pluwiometru podłączamy do GPIO 6 Raspberry Pi. Możemy teraz napisać prosty skrypt, który zapiszemy w `/home/pi/weather-station/rainfall.py` – listing 6.

Integracja wszystkich elementów programu

Finalny skrypt, obsługujący wszystkie elementy stacji pogodowej, przedstawiono na listingu 7.

Obsługa bazy danych

Ostatnim elementem systemu jest baza danych, do której zapisywane będą wyniki pomiarów stacji meteorologicznej. W powyższym skrypcie wyniki pomiarów są po prostu prezentowane w terminalu w postaci tekstowej. Aby móc zapisywać dane do bazy, konieczne jest jej skonfigurowanie. W tym celu w terminalu wpisujemy:

```
sudo mysql
```

Uruchomi to narzędzie konfiguracyjne baz danych. Najpierw musimy stworzyć użytkownika bazy oraz dodać mu wszystkie potrzebne uprawnienia:

```
create user pi IDENTIFIED by 'my54cr4t';
grant all privileges on *.* to 'pi' with
grant option;
```

gdzie „pi” to nazwa użytkownika a „my54cr4t” to hasło. Teraz możemy dodać bazę danych weather i ją uruchomić:

```
create database weather;
use weather;
```

Teraz możemy zdefiniować w ramach bazy danych weather tabelę WEATHER_MEASUREMENT. Musi ona zawierać wszystkie potrzebne pola, do przechowywania pomiarów naszej stacji pogodowej. Wprowadzamy zatem następującą definicję tabeli:

```
CREATE TABLE WEATHER_MEASUREMENT(
ID BIGINT NOT NULL AUTO_INCREMENT,
REMOTE_ID BIGINT,
AMBIENT_TEMPERATURE DECIMAL(6,2) NOT NULL,
GROUND_TEMPERATURE DECIMAL(6,2) NOT NULL,
AIR_PRESSURE DECIMAL(6,2) NOT NULL,
HUMIDITY DECIMAL(6,2) NOT NULL,
WIND_DIRECTION DECIMAL(6,2) NULL,
WIND_SPEED DECIMAL(6,2) NOT NULL,
WIND_GUST_SPEED DECIMAL(6,2) NOT NULL,
RAINFALL DECIMAL(6,2) NOT NULL,
```

```
CREATED TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( ID )
);
```

Teraz w głównym programie dodajemy import biblioteki do obsługi baz danych:

```
import database
```

Następnie musimy dostać się do bazy danych, jaką utworzyliśmy. Przed główną pętlą programu dodajemy:

```
db = database.weather_database()
```

Aby zapisać dane do bazy, wystarczy jedna linijka kodu, którą umieszczamy zamiast `print(...)`, który drukuje wyniki pomiarów:

```
db.insert(ambient_temp, ground_temp, 0, pressure,
humidity, wind_average, wind_speed, wind_gust,
rainfall)
```

Finalnie musimy dodać plik z danymi do logowania, który musi znaleźć się w tym samym folderze, co plik z programem – w naszym przypadku plik ten to `/home/pi/weather-station/credentials.mysql`. Zapisujemy w nim następujące dane; oczywiście, jeśli zmieniliśmy hasło, nazwę użytkownika albo nazwę bazy, musimy tutaj wpisać zmienioną nazwę:

```
{
„HOST”: „localhost”,
„USERNAME”: „pi”,
„PASSWORD”: „my54cr4t”,
„DATABASE”: „weather”
}
```

Dane z bazy pobierać można na wiele sposobów, poprzez interfejs webowy czy inne oprogramowanie. Do bazy dostać można się także poprzez sieć, co oznacza, że można dane te odczytywać zdalnie.

Podsumowanie

Opisany powyżej projekt pozwala na budowę stacji meteorologicznej wykorzystującej Raspberry Pi. Można wzbogacić ją o własne sensory, tylko pomysłowość konstruktora jest tutaj ograniczeniem. System uzupełnić można o czujnik nasłonecznienia (np. fotodiody lub fotoopornik), czujnik UV, monitor jakości powietrza etc.

Dane zapisywane przez system do bazy są łatwo dostępne, więc adaptować można dowolny system do prezentacji danych. Dowolny webowy front-end pozwoli na proste stworzenie strony, wyświetlającej zebrane przez nas dane.

Nikodem Czechowski

Źródło: <http://bit.ly/2JsLNLa>.

REKLAMA

Wstęp do Klubu AVT Elektronika

będziesz miał prawo do korzystania z szeregu przywilejów:

- do 50% zniżki w Sklepie AVT
- darmowe prenumeraty Wydawnictwa AVT
- do 50% zniżki w Ulubionym Kiosku
- Zapraszamy do zapoznania się z zasadami Klubu!

<http://bit.ly/2GaDwtQ>