

[SCIR] Raport 2

Bartłomiej Mastej

Warsaw University of Technology, Warsaw, Poland

1 Konfiguracja wezła

Wszelkie zmiany w projekcie są dostępne na GitHub'ie: <https://github.com/bartoszlomiej/weather-forecaster>. Jako, że platforma uruchomieniowa jest Raspberry Pi 4 model B, zatem konfiguracja przesyłu danych poprzez WiFi nie była skomplikowana. Zostało to wskazane w raporcie 1.

Podsumowując:

1. Komunikacja Raspberry z chmurą oraz z komputerem odbywa się poprzez wbudowany moduł WiFi.
2. Komunikacja z komputerem odbywa się poprzez protokół *ssh* do pełnej kontroli, oraz przez *scp* do przesyłu plików.
3. Komunikacja z chmurą wykorzystuje protokół MQTT.
4. Chmura użyta w projekcie to *Thingspeak*.

2 Wstępna komunikacja z chmurą

Do skonfigurowania połączenia z chmurą po stronie Raspberry Pi za pomocą protokołu MQTT została wykorzystana biblioteka *paho.mqtt*. Za pomocą gotowego API należy wpisać wymagane dane - id kanału, dane uwierzytelniające klienta, port, nazwę hosta, socket. Konfiguracja (bez danych uwierzytelniających - ze względów bezpieczeństwa) została zaprezentowana na Fig. 1.

Wysyłanie danych z przygotowanym payload'em okazuje się bardzo proste - wystarczy użyć funkcji *publish*.

Po stronie chmury należy udostępnić *id*. Na Fig. 2 zostało zaprezentowane prawidłowe odbieranie danych z raspberry pi. Jednakże, zamiast danych z czujników zostały wykorzystane parametry procesora (w celach testowych).

3 Parallelizm zbierania danych - multithreading

W celu wykonywania pomiarów z czujnika jednocześnie został wprowadzony multithreading z wykorzystaniem biblioteki *threading*. Za jego pomocą każdy z czujników zbiera dane jednocześnie. Przed ich ujednoliceniem następuje synchronizacja wątków.

W przyszłości ma to na celu jednoczesne przetwarzanie obrazu, zbierania danych z czujników jak i ich wysyłania jeśli chwilami będzie to konieczne. Jednakże, zostanie to rozważone ze względu na optymalizację poboru mocy.

```

import paho.mqtt.publish as publish
import string

class Connection:
    # The ThingSpeak Channel ID.
    # Replace <YOUR-CHANNEL-ID> with your channel ID.
    def __init__(self, client_ID, username, password):
        self.channel_ID = "1905459"#"<YOUR-CHANNEL-ID>"

        # The hostname of the ThingSpeak MQTT broker.
        self.mqtt_host = "mqtt3.thingspeak.com"

        # Your MQTT credentials for the device
        self.mqtt_client_ID = #"<YOUR-CLIENT-ID>"
        self.mqtt_username = #"<YOUR-USERNAME>"
        self.mqtt_password = #"<YOUR-MQTT-PASSWORD>"

        self.t_transport = "websockets"
        self.t_port = 80

        # Create the topic string.
        self.topic = "channels/" + channel_ID + "/publish"

```

Fig. 1: Test odbierania danych w chmurze.

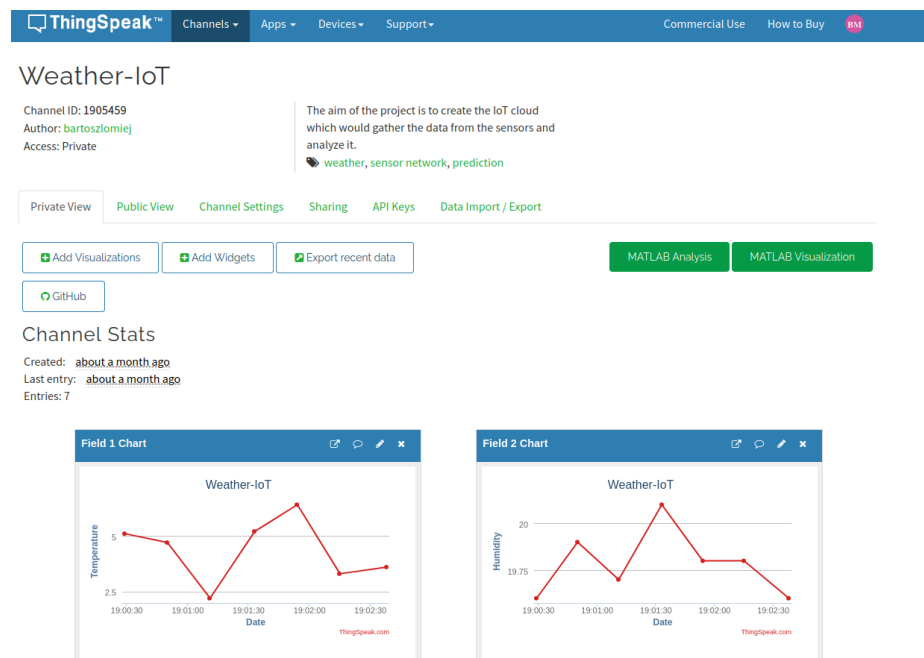


Fig. 2: Test odbierania danych w chmurze.