

## Przygotowanie urządzenia do komunikacji

W przypadku normalnego włączenia zasilania lub po naciśnięciu przycisku resetowania moduł Pycom uruchomi się w trybie standardowym; najpierw zostanie uruchomiony plik `boot.py`, a następnie `main.py`. Po dodaniu odpowiednich bibliotek trzeba umożliwić przesyłanie oraz odbieranie danych pomiędzy urządzeniem i serwerem.

W pliku startowym `boot.py` wpisujemy sposób połączenia układu z wybraną lokalizacją przy użyciu określonego sposobu komunikacji:

### 1. Wi-fi WLAN:

```
import pycom
import time
import machine
from network import WLAN
wlan = WLAN() # pobieranie bieżącego obiektu bez zmiany trybu
if machine.reset_cause() != machine.SOFT_RESET:
    wlan.init(mode=WLAN.STA) # konfiguracja jako stacja
    wlan.ifconfig(config=('192.168.0.100', '255.255.255.0',
        '192.168.0.1', '8.8.8.8')) # zgodnie z ustawieniami routera
    print('WLAN połączenie powiodło się!')
if not wlan.isconnected():
    wlan.connect('mojasiec', auth=(WLAN.WPA2, 'mojklucz'), timeout=5000)
    # tworzenie połączenia: identyfikator sieci, bezpieczeństwo i hasło,
    # maksymalny czas w milisekundach oczekiwania na połączenie
    while not wlan.isconnected():
        machine.idle()
```

(dodatkowo źródła: <https://docs.pycom.io/firmwareapi/pycom/network/wlan>  
<https://docs.pycom.io/tutorials/all/wlan>  
wykład z przedmiotu)

### 2. Wi-fi AP:

zmieniamy:

```
...
from network import WLAN
...
wlan.init(mode=WLAN.AP, ssid='mojasiec', auth=WLAN.WPA2, channel=7,
    antenna=WLAN.INT_ANT) # konfiguracja jako AP
...
```

(dodatkowo źródło: <https://docs.pycom.io/firmwareapi/pycom/network/wlan>  
wykład z przedmiotu)

### 3. Bluetooth:

pierwszy:

```
...
from network import Bluetooth
bth = Bluetooth() # tworzenie obiektu Bluetooth z nazwą bt
bth.start_scan(-1) # skanowanie w poszukiwaniu urządzeń BLE wysyłających
    ogłoszenia, bez ograniczenia czasowego (określony czas
    może być wyrażony w sekundach, np. 20)
```

```

bth.init(id=0, mode=Bluetooth.BLE, privacy=True, secure_connections=True,
         mtu=200) # jedno urządzenie, tryb BLE, ustawienie
                 prywatności adresu, bezpieczne połączenie, maksymalna
                 jednostka transmisji (MTU)
adv = bth.get_adv() # pobiera dane ze skanowania, które mają strukturę:
                   (mac, addr_type, adv_type, rssi, data)
if adv:
    conn = bth.connect(adv.mac) # tworzenie połączenie na podstawie
                                adresu MAC urządzenia, limit czasu określa czas w
                                milisekundach oczekiwania na zakończenie procesu połączenia
                                (przy braku dopóki połączenie nie powiedzie się lub upłynie
                                określony limit czasu z Bluetooth.timeout TimeoutError)

```

drugi:

```

...
from network import Bluetooth
...
bth2 = Bluetooth()
bth2.service('abc123', isprimary=True, nbr_chars=1, start=True) # tworzy
                       nową usługę: UUID usługi; czy usługa jest podstawowa;
                       liczbę cech, które będzie zawierać usługa; start - jeśli
                       True, usługa zostanie uruchomiona natychmiast.
bth2.set_advertisement(name='mojLoPy', service_uuid='1234567890123456') #
                               skonfiguruje dane, które mają być rozgłaszane, nazwa ciągu
                               wyświetlana w rozgłaszaniu, service_uuid identyfikator
                               usługi, która ma być rozgłaszana.

```

(dodatkowo źródło: <https://docs.pycom.io/firmwareapi/pycom/network/bluetooth>  
wykład z przedmiotu)

## 4. LoRaWAN:

```

....
from network import LoRa
import ubinascii
import struct
....
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)
# tworzenie parametrów uwierzytelniania ABP (Activation By Personalisation)
dev_adres = struct.unpack(">l", binascii.unhexlify('00000005'))[0]
nwk_klucz = ubinascii.unhexlify('2B7E151628AED2A6ABF7158809CF4F3C')
app_klucz = ubinascii.unhexlify('2B7E151628AED2A6ABF7158809CF4F3C')

# przyłączenie do sieci
lora.join(activation=LoRa.ABP, auth=(dev_adres, nwk_klucz, app_klucz))

```

Trzeba upewnić się, że do urządzenia podłączona jest antena przed wysyłaniem / odbieraniem wiadomości LoRa, ponieważ niewłaściwe użycie (np. bez anteny) może uszkodzić urządzenie.

(dodatkowo źródło: <https://docs.pycom.io/firmwareapi/pycom/network/lora>  
wykład z przedmiotu)