

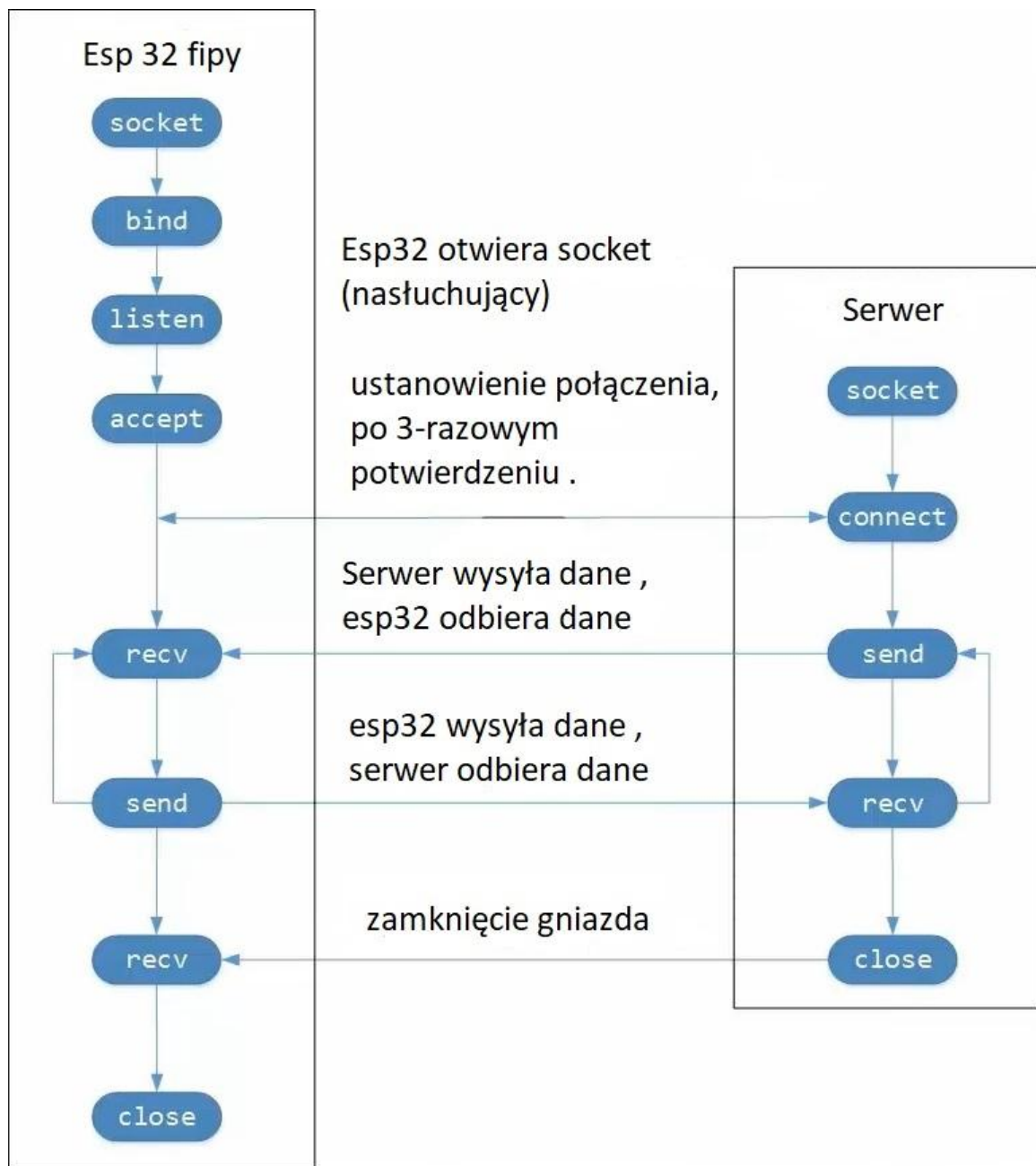
Tworzenie kanału komunikacyjnego - gniazda

Po utworzeniu pliku startowego `boot.py` wpisujemy do pliku `main.py` importowanie i używanie czujników wysokości, ciśnienia, temperatury, światła lub przyspieszenia oraz tworzenie gniazda.

Wymianę informacji za pomocą protokołów wyższego poziomu realizuje się najczęściej za pomocą tzw. gniazd (socket), które określa się jako abstrakcję programistyczną, oznaczającą punkt docelowy dwustronnej komunikacji dwóch procesów działających równolegle w sieci.

Gniazda - specjalne kanały komunikacyjne, działające według schematu:

- otwarcie kanału komunikacji (gniazda),
- zapis lub odczyt (przesłanie lub otrzymanie danych za pomocą gniazda),
- usunięcie kanału komunikacyjnego (zamknięcie gniazda).



Dzięki utworzeniu Gniazd z obu stron połączenia jest możliwa szybka transmisja danych pomiędzy urządzeniami. Gniazdo należy utworzyć na mikrokontrolerze oraz na serwerze.

Importowanie potrzebnych bibliotek zależnych od parametrów:

```
import pycom;
import time;
import machine;
import socket;
from pysense import Pysense;
from LIS2HH12 import LIS2HH12
from SI7006A20 import SI7006A20
from LTR329ALS01 import LTR329ALS01
from MPL3115A2 import MPL3115A2, ALTITUDE, PRESSURE
```

zadeklarować zmienne takie jak host i port:

```
host = '192.168.0.100'
```

host – adres, do którego przesyłane są dane (np. router)

```
port = 80
```

port – port, do którego dane mają trafić.

1. Tworzenie Gniazda na mikrokontrolerze odbierającego dane z pomiarów:

```
tcp_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM,
                          socket.IPPROTO_TCP) # dla połączenia przez Wi-fi WLAN
tcp_sock.bind(('0.0.0.0', 80)) # powiązanie adresu do gniazda
tcp_sock.listen(5) # nasłuchiwanie gniazda

while True: # jeżeli urządzenie chce ustanowić połączenie
    (sock, addr) = tcp_sock.accept() # akceptacja połączenia przychodzącego
    print('przychodzace dane od: {}'.format(addr)) # wyświetlenie adresu, z
                                                którego otrzymujemy dane
    data = sock.recv(1024) # odbieranie danych i zapisanie w buforze
                          (sugerowany rozmiar bufora 1024)
    while data: # jeżeli zmienna posiada dane
        dekodowanie = data.decode('utf-8') # dekodowanie danych na format utf-
                                           8
        print(dekodowanie)
        rozdzielenie = dekodowanie.split() # rozdzielenie elementów łańcucha
                                           (jeżeli jest więcej niż jeden)
        x = int(rozdzielenie[0]) # wybieranie elementów ciągu znaków, wybranie
        elementów oraz przypisanie parametrów do zmiennych
        y = int(rozdzielenie[1]) # "-"
        for i in range(0, x): # utworzenie pętli do wykonywania x pomiarów oraz
                               wysyłania
            py = Pysense()
            si = SI7006A20(py) # odczytywanie z bibliotek
            temp = si.temperature()
            wilgotn = si.humidity()
            mp = MPL3115A2(py, mode=ALTITUDE)
            wysok = mp.altitude()
            mpp = MPL3115A2(py, mode=PRESSURE)
            cisnienie = mpp.pressure()
            lt = LTR329ALS01(py)
            jasn = lt.light()
            li = LIS2HH12(py)
            przysp = li.acceleration()
            obrot = li.roll()
            pochyl = li.pitch()
            lancuch = '{"Wilgotnosc" : "%.2f" , "Cisnienie" : "%.2f" ,
                      ...}'%(wilgotn, cisnienie, ...) # wartości które zostaną
                      zapisane w bazie danych
```

```

postStr = 'POST /nasz_katalog/nasz_json.php HTTP/1.1\r\n' #
          nagłówek protokołu http w pliku nasz_json.php
hostStr = 'Host:%s:%s\r\n'%(str(host),str(port)) # adres ip oraz
          port
zawart = 'Content-Type:application/json\r\n' # rodzaj nagłówka
znaki = 'Content-Length:%s\r\n\r\n'%str(len(lancuch)) # ilość
        znaków
dane = postStr + hostStr + zawart + znaki + lancuch # złączenie
      wszystkich danych w całość
dan_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM,
      socket.IPPROTO_TCP ) # gniazdo wysyłające dane
dan_sock.connect((host,port)) # Połączenie z adresem ip hosta oraz
      portem
dan_sock.send(dane.encode()) # wysłanie danych z gniazda
odpowiedzi = dan_sock.recv(4096) # przygotowanie bufora do
      odpowiedzi serwera na wysłane dane
dan_sock.close () # zamknięcie gniazda wysyłającego dane
time.sleep(y)
break
tcp_sock.close () # zamknięcie gniazda odbierającego parametry

# tworzenie gniazda dla LoRa
lora_sock = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
# ustawienie prędkości transmisji LoRaWAN
lora_sock.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
# wysłanie przykładowych danych
lora_sock.send(bytes([0x01, 0x02, 0x03]))
...
lora_sock.bind(1) # powiązanie adresu do gniazda
...
dane = s.recv(64) # odbieranie danych i zapisanie w buforze
...
lora_sock.close () # zamknięcie gniazda odbierającego parametry

```

2. Tworzenie Gniazda po stronie serwera w pliku np. **abc.php**:

```

<html >
<head>
<meta http-equiv="Content-Language" content="pl">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title> ... </title>
<link rel="stylesheet" type="text/css" href="css/main.css" />
</head>
<body>
...
<?php
$host = "192.168.0.100";
$port = 80;
$output = $_POST['ile'] .' ' .$_POST['jak']; // 'ile', 'jak' parametry np. z
      formularza do ilości i częstości pomiarów
$socket1 = socket_create(AF_INET, SOCK_STREAM,0) or die("Could not create
      socket\n");
socket_connect($socket1,$host,$port ); //powiązanie gniazda z hostem i portem
socket_write($socket1, $output, strlen( $output)) or die("Could not write
      output\n"); // wpisywanie danych, które chcemy wysłać, do gniazda
socket_close($socket1); // zamknięcie gniazda
?>
...
...

```

3. Tworzenie pliku **nasz_json.php** w którym odbywa się odbiór i zapis danych do bazy danych:

```
<?php
// utworzenie połączenia z bazą danych
require_once "connect.php";
$polaczenie = @new mysqli($db_host, $db_user, $db_password, $db_name);
// sprawdzanie połączenia
if ($polaczenie->connect_error) {
    die("Connection failed: " . $polaczenie->connect_error);
}
// odebranie danych oraz zapisanie ich w buforze
$json = json_decode(file_get_contents('php://input'), true);
ob_start(); //
error_log( ob_get_clean() );
// instrukcja if w którym odbywa się odczytanie pożądaných wartości
if($json!=null){
    $Wilgotnosc = $json["Wilgotnosc"];
    $Cisnienie = $json["Cisnienie"];
    ...
    ...
    $sql=" INSERT INTO tabela (Wilgotnosc, Cisnienie, ..., ...)
VALUES ('$Wilgotnosc' , '$Cisnienie', ..., ...)";

    if(mysqli_query($polaczenie, $sql)){
        $response["code"] = "1";
        $response["message"] = "Powodzenie ";
        echo json_encode($response);
    }
    else{
        $response["code"] = "2";
        $response["message"] =mysql_error($polaczenie);
        echo json_encode($response);
    }
}
else{
    echo "Dane nie dotarły";
}
mysqli_close($polaczenie);
?>
```