

IOT na przykładzie esp8266

Arkadiusz Baraniecki
Michał Śmigielski
Artur Maćkowiak

allegro

Cześć!

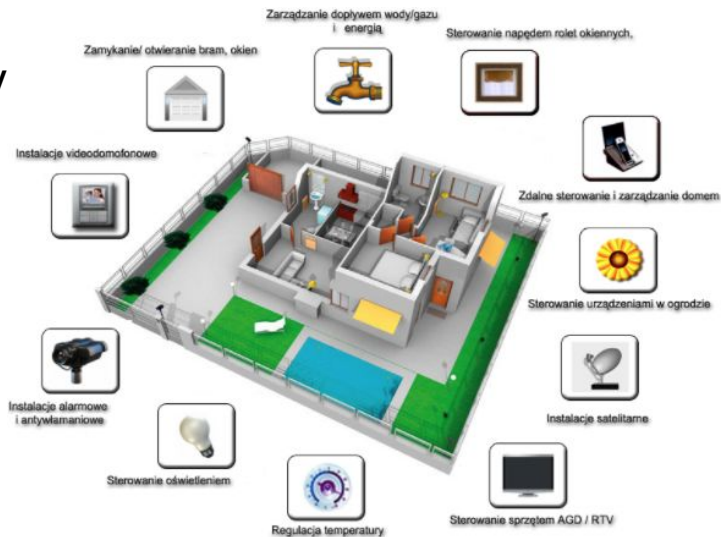


Spis treści

1. Wstęp
2. Projekt w Arduino - termometr po wifi
3. Projekt w Micropythonie - termometr
4. Termometr bez programowania
5. Wyznanie ;-) z przekaźnikiem
6. Podsumowanie

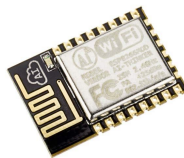
Dom inteligentny i IoT

- Dom inteligentny



- Warto samemu umieć zrobić czujkę czy sterownik
 - to uniwersalne
 - czasem nie ma dostępu do serwera i urządzenie musi być samodzielne

NodeMCU Esp8266 v3 - jak to?



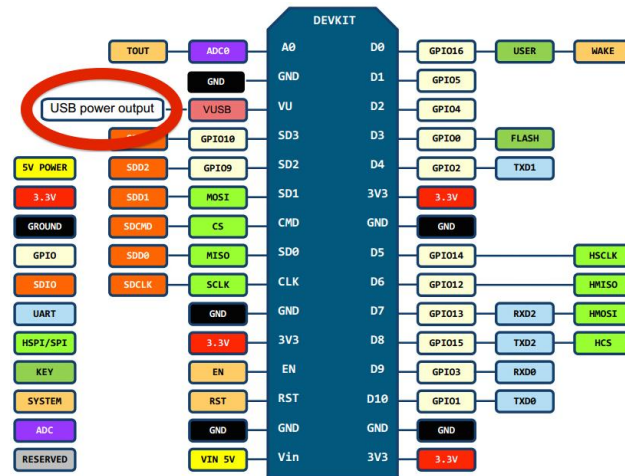
1. Moduł układu esp8266 NodeMCU:
 - ESP-12 + stabilizator napięcia + UART
 - Zgodny z Arduino,
 - 10 portów GPIO. Każdy z nich obsługuje PWM, I2C, SPI i 1-Wire
 - Wifi b/g/n/ z wbudowaną anteną
 - Obsługa stosu tcp/ip
 - Zasilanie 5V microUSB
 - Wbudowany konwerter usb-uart (układ CH340) - programowanie przy wykorzystaniu usb a tak naprawdę stary dobry port szeregowy pchnięty przez USB
 - Obsługa za pomocą Arduino IDE / LUA / microPython

2.. Oficjalnie 'nieoficjalnie'

Układ w wersji V3 nigdy nie został wydany przez NodeMCU. Układ stworzony przez chiński WEMOS.

Jest to wersja nieoficjalna. W stosunku do V2 różni się:

- Rozmiarem, sick!
- wykorzystaniem PINów zarezerwowanych w V2 zwłaszcza VU - wyjście 5V



Gdzie jesteśmy

1. Wstęp
2. Projekt w Arduino - termometr po wifi
3. Projekt w Micropythonie - termometr
4. Termometr bez programowania
5. Wyznanie ;-) z przekaźnikiem
6. Podsumowanie

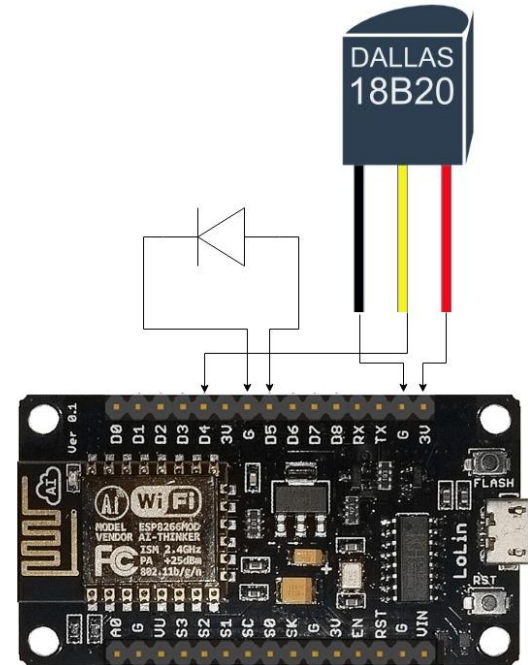
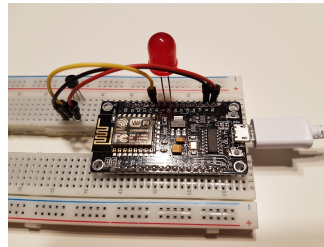
Organizacja pracy

- HipChat: _szkolenie_IOT
- WIFI: esp8266labs Hasło: esp8266labs
- Instalacja git-a
- Instalacja pythona 3.6
- Instalacja virtualenv-a
 - `(np. virtualenv -p python3.6 venv_allegro-tech-labs-iot)`
- `source venv_allegro-tech-labs-iot/bin/activate`
- PIP powinien być w wersji 9.0.3 (`pip -V i ew. pip install pip==9.0.3`)
- `git clone https://github.com/allegro/allegro-tech-labs-iot.git`
- `cd allegro-tech-labs-iot/`
- Instalacja narzędzia na warsztat: `pip install -e .`
- Instalacja Arduino i sterowników wg odpowiedniego pdf z podkatalogu “other”

Podłączanie esp8266 do laptopa

todo:

- budowa płytki prototypowej
- schemat:



Arduino IDE, struktura sketcha i...

Stage1 - diodomigacz

IDE

- kompilacja i wysłanie programu na urządzenie
- monitorowanie wyjścia

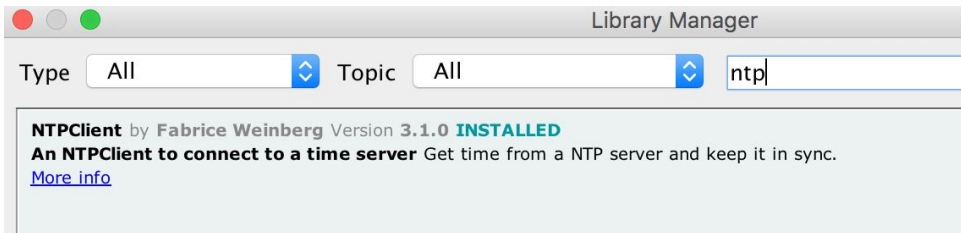
Sketch

- funkcje setup() i loop()
- inicjalizacja pinu
- inicjalizacja i komunikacja przez Serial



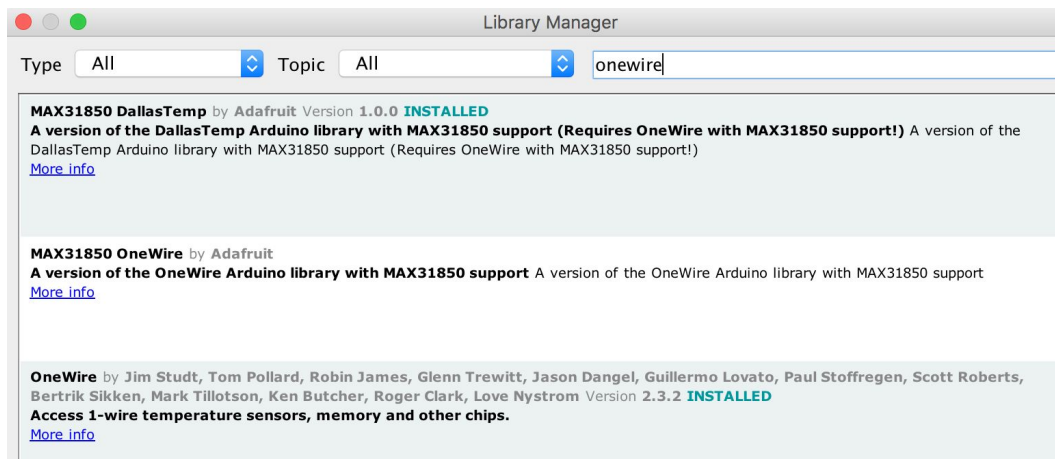
Stage2 - Podłączenie esp8266 do wifi

- podłączenie do wifi
- ustawienie czasu
-



Stage3 - Pomiar temperatury

- dtostrf... na razie nic wielkiego



Stage4 - Serwer http...

```
#include <ESP8266mDNS.h>

#include <ESP8266WebServer.h>

ESP8266WebServer server(80);

setup() {
    //----- Initialize MDNS responder
    //----- Initialize HTTP server
}

loop() {
    server.handleClient();
}
```

Gdzie jesteśmy

1. Wstęp
2. Projekt w Arduino - termometr po wifi
- 3. Projekt w Micropythonie - termometr**
4. Termometr bez programowania
5. Wyznanie ;-) z przekaźnikiem
6. Podsumowanie

Przygotowanie micropythona

- różnica pomiędzy pracą - nie flashujemy tylko wgrywamy skrypty
- Download: <https://micropython.org/download#esp8266> (wy już macie na dyskach w katalogu “other”)
- esptool (w virtualenv)

- `pip install esptool`
- `esptool.py --port /dev/cu.wchusbserial1420 erase_flash`
- `esptool.py --port /dev/cu.wchusbserial1420 --baud 115200 write_flash --flash_size=detect <-fm dio> 0 other/esp8266-20180511-v1.9.4.bin`

Użycie interpretera micropythona

- `screen /dev/cu.wchusbserial1420 115200` (żeby zobaczyć serial)
 - `help()`
 - `import machine`
 - `led = machine.Pin(14, machine.Pin.OUT)`
 - `led.on()`
 - `led.off()`

Stage5 - Wgrywanie programu (ampy)

- `pip install adafruit-ampy`
- `run, put, get, ls....`
- `ampy -p /dev/cu.wchusbserial11420 -b 115200 -d 2 put
micropython_05_basic/main.py`
- *reset urządzenia*
- Więcej o ampy: <https://learn.adafruit.com/.../install-ampy>

Stage6 - Pomiar temperature

- `ampy -p /dev/cu.wchusbserial11420 -b 115200 -d 2 put mds18x20.py`
- `ampy -p /dev/cu.wchusbserial11420 -b 115200 -d 2 put main.py`
- `reset...`

Stage7 - podłączenie do wifi

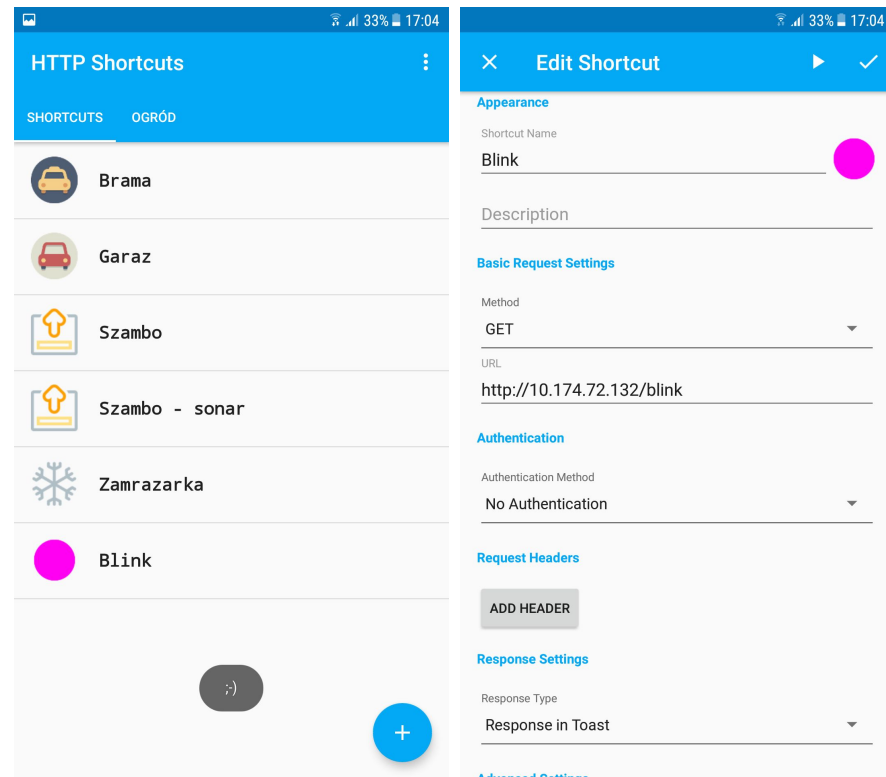
- `ampy ... put mwifi.py`
- `ampy ... put mntptime.py`
- `ampy ... put main.py`
- **soft reset vs hard reset**

Stage8 - udostępnienie temperatury po http

- Instalacja picoweb ([source](#)). With screen:
 - flash -> sieć...
 - `import gc`
 - `import upip`
 - `gc.collect()` *#czyści pamięć*
 - `upip.install('micropython-uasyncio')`
 - `gc.collect()`
 - `upip.install('picoweb/1.1')`
- Permanent till flashing

Stage8 - udostępnienie temperatury po http...

- `ampy ... put main.py`
- `reset...`
- `curl http://<ip-address>/blink`



Zalety i wady micropythona



- Moduły typowo pythonowe są niekompletne, np:
 - brak biblioteki smtp - jest po prostu pusty plik (i jest takich więcej)
 - urequests nie akceptuje “auth”, trzeba podawać “Authorization” w headerze
 - brak urllib.parse.urlencode(data) (koduje stringi na sztywno)
- Wygodnie się pisze logikę, ale trzeba sprawdzić czy są wszystkie potrzebne moduły, klasy i metody.

Gdzie jesteśmy

1. Wstęp
2. Projekt w Arduino - termometr po wifi
3. Projekt w Micropythonie - termometr
- 4. Termometr bez programowania**
5. Wyznanie ;-) z przekaźnikiem
6. Podsumowanie

Instalacja ESP easy - live demo

- Download

https://www.letscontrolit.com/wiki/index.php/ESPEasy#Loading_firmware

- `esptool.py --port /dev/cu.wchusbserial1420 --baud 115200
write_flash --flash_size=detect 0 ESPEasy_R120_4096.bin`
- Zaraz po wgraniu firmware'u, moduł ESP pracuje jako access point i rozgłasza sieć WiFi o nazwie **esp_0**. Hasło **configesp**
- łączymy się z tą siecią i podajmy sieć docelową
- łączymy się z esp w naszej sieci na adres `http://<ip>/setup`





ESP easy - sterowanie pinami

- `curl http://<IP-address>/control?cmd=GPIO,14,1`
- `curl http://<IP-address>/control?cmd=PULSE,14,1,500`

Więcej komend na

https://www.letscontrolit.com/wiki/index.php/ESPEasy_Command_Reference

ESP easy - pomiar temperatury

Task Settings	Value
Device:	Temperature - DS18b20  
Name:	temptest
Delay:	15
IDX / Var:	1
1st GPIO:	GPIO-2 
Device Nr:	1  ROM: 28-81-7-96-8-0-0-5f
Send Data:	<input type="checkbox"/>

Więcej na <http://grylewicz.pl/esp-easy-prosty-sposob-obsluge-czujnikow-wifi/>

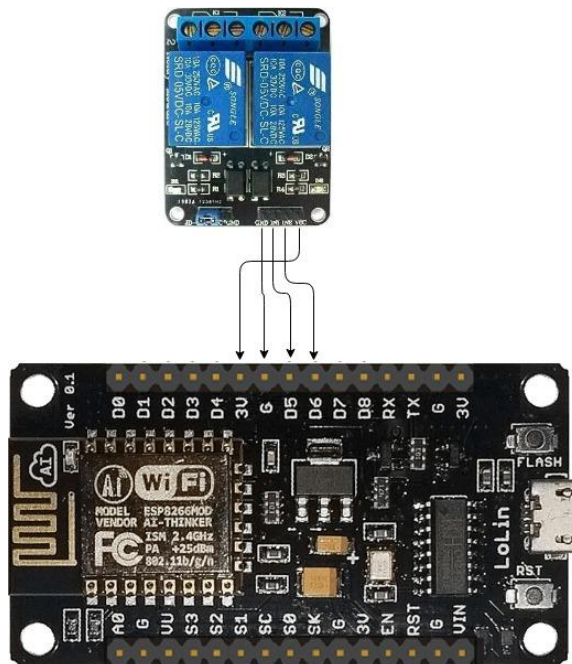
Gdzie jesteśmy

1. Wstęp
2. Projekt w Arduino - termometr po wifi
3. Projekt w Micropythonie - termometr
4. Termometr bez programowania
- 5. Wyznanie ;-)** z przekaźnikiem
6. Podsumowanie

Challenge

Schemat połączenia:

```
#static const uint8_t D0 = 16;  
#static const uint8_t D1 = 5;  
#static const uint8_t D2 = 4;  
#static const uint8_t D3 = 0;  
#static const uint8_t D4 = 2;  
#static const uint8_t D5 = 14;  
#static const uint8_t D6 = 12;  
#static const uint8_t D7 = 13;  
#static const uint8_t D8 = 15;  
#static const uint8_t D9 = 3;  
#static const uint8_t D10 = 1;
```



Gdzie jesteśmy

1. Wstęp
2. Projekt w Arduino - termometr po wifi
3. Projekt w Micropythonie - termometr
4. Termometr bez programowania
5. Wyznanie ;-) z przekaźnikiem
- 6. Podsumowanie**

Podsumowanie i linki

- różnice pomiędzy arduino, micropythonem a ESPEasy
- przydatne linki z projektami
 - <http://fritzing.org/home/>
 - <http://www.digikey.com/schemeit/project/>
 - <https://circuits.io/>
 - <https://create.arduino.cc/>
 - <https://www.hackster.io/>
 - <http://www.autodesk.com/products/eagle/overview>
 - https://github.com/spacehuhn/esp8266_deauther ;-)

Podsumowanie i linki

- linki - micropython
 - <https://micropython.org/download#esp8266>
 - <https://docs.micropython.org/en/latest/pyboard/library/index.html>
 - <https://github.com/micropython/micropython-lib>
- ESP easy
 - <https://www.letscontrolit.com/wiki/index.php/ESPEasy>
 - https://www.letscontrolit.com/wiki/index.php/ESPEasy_Command_Reference
 - <http://grylewicz.pl/esp-easy-prosty-sposob-obsluge-czujnikow-wifi/>
- Przykład zastosowania
 - <http://smart-house.adrian.czabanowski.com/przelacznik-sonoff/>
 - ale :-)) <https://techfreak.pl/sonoff-gotowy-uklad-sterowania-przekaznikiem-esp8266/>

Podsumowanie i linki

- esp i czujniki
 - <https://botland.com.pl/moduly-wifi/8241-modul-wifi-esp8266-nodemcu-v3.html>
 - allegro, alibaba/aliexpres, gearbest
 - do budowania custom firmware
 - <https://nodemcu-build.com/>
 - <http://nodemcu.readthedocs.io/en/master/>
- home automation soft
 - <https://home-assistant.io/>
 - <https://domoticz.com/>
 - <https://www.openhab.org/>
 - <https://www.supla.org/pl/>

Lista użytego sprzętu

- esp8266 nodemcu v3: <https://botland.com.pl/moduly-wifi/8241-modul-wifi-esp8266-nodemcu-v3.html>
- przekaźnik: <https://botland.com.pl/przekazniki/2043-modul-przekaznikow-2-kanaly-styki-7a240vac-cewka-5v.html>
- dioda: <http://allegro.pl/dioda-led-10mm-czerwona-dyfuzyjna-1500mcd-arduino-i6573430636.html>
- termometr
 - <https://botland.com.pl/czujniki-temperatury/165-czujnik-temperatury-ds18b20-cyfrowy-1-wire-tht.html>
 - lub
 - <https://botland.com.pl/czujniki-temperatury/4582-sonda-wodoodporna-z-czujnikiem-temperatury-ds18b20-3m.html>
- Płytką stykową: <https://botland.com.pl/plytki-stykowe/55-plytka-stykowa-830-otworow.html>
- Zestaw przewodów:
<https://botland.com.pl/przewody-polaczeniowe/1958-zestaw-przewodow-polaczeniowych-20cm-3-x-40-szt-m-m-z-z-m-z.html>
- Do tego zasilacz i kabel usb
- Linki pochodzą z botland, bo tam są dłużej. De facto sprzęt był kupiony na allegro.pl

FAQ

- specyfikacja układu esp8266: https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf
- Opis układu esp8266: <https://nettigo.pl/products/modul-wifi-esp8266-12f>
- jeśli po włączeniu prądu czasem urządzenie trzeba restartować, żeby działało
<https://github.com/esp8266/Arduino/blob/master/doc/boards.rst#boot-messages-and-modes>
- gotowe rozwiązanie: <http://blebox.eu/pl/>
-

Q&A

allegro