

PROJEKT

STEROWNIKI ROBOTÓW

---

Założenia projektowe

Stacja meteorologiczna

SM

---

*Skład grupy (X):*

Robert WINKLER, 226 474

*Termin: wtTN17*

*Prowadzący:*

mgr inż. Wojciech DOMSKI

12 czerwca 2018

# Spis treści

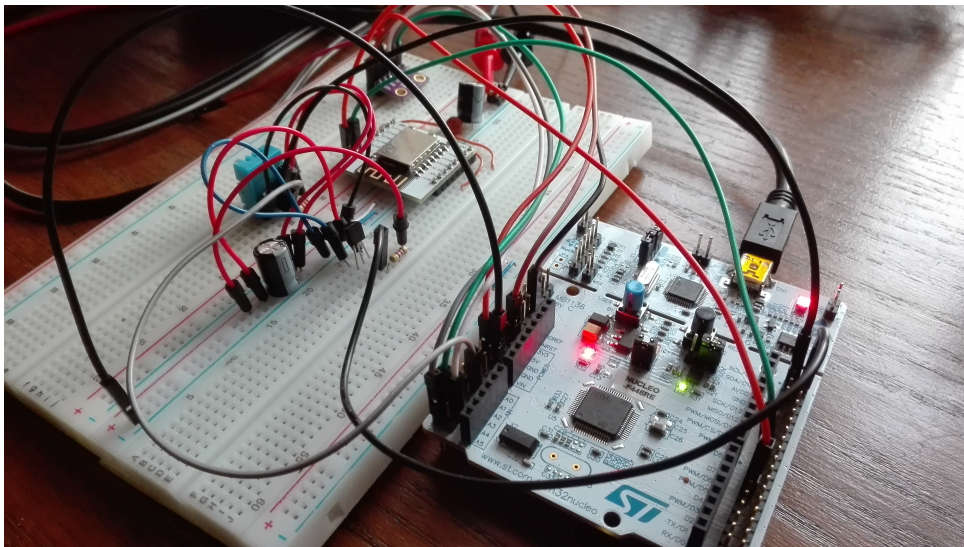
<b>1</b>	<b>Opis projektu</b>	<b>2</b>
<b>2</b>	<b>Konfiguracja mikrokontrolera</b>	<b>3</b>
2.1	Konfiguracja pinów . . . . .	3
2.2	Konfiguracja zegara . . . . .	4
<b>3</b>	<b>Konfiguracja peryferiów mikrokontrolera</b>	<b>4</b>
3.1	SPI2 . . . . .	5
3.2	SYS . . . . .	5
3.3	TIM2 . . . . .	5
3.4	TIM3 . . . . .	6
3.5	TIM4 . . . . .	6
3.6	USART1 . . . . .	6
3.7	USART2 . . . . .	7
<b>4</b>	<b>Opis funkcji pinów</b>	<b>7</b>
<b>5</b>	<b>Opis wykorzystanych układów zewnętrznych</b>	<b>8</b>
5.1	BMP280 - cyfrowy barometr, czujnik ciśnienia . . . . .	8
5.2	Opis układu . . . . .	8
5.3	Dane techniczne . . . . .	8
5.4	Modyfikowane rejestry . . . . .	8
5.5	DS18B20 - cyfrowy czujnik temperatury . . . . .	9
5.5.1	Opis czujnika . . . . .	9
5.5.2	Dane techniczne . . . . .	9
5.5.3	Wysyłane komendy . . . . .	9
5.6	DHT11 - Czujnik temperatury i wilgotności . . . . .	10
5.6.1	Opis czujnika . . . . .	10
5.6.2	Dane techniczne . . . . .	10
5.7	ESP12F - Moduł WiFi . . . . .	11
5.7.1	Opis modułu . . . . .	11
5.7.2	Dane techniczne . . . . .	11
5.7.3	Wykorzystywane komendy AT . . . . .	11
<b>6</b>	<b>Opis działania programu</b>	<b>11</b>
6.1	Komunikacja z DHT11 . . . . .	12
6.2	Komunikacja z DS18B20 . . . . .	12
<b>7</b>	<b>Podsumowanie</b>	<b>12</b>

# 1 Opis projektu

Celem projektu było wykonanie prototypu stacji meteorologicznej opartej na cyfrowych czujnikach wilgotności, temperatury i ciśnienia, komunikującej się przez bezprzewodową sieć WiFi z serwerem, na którym będą przechowywane dane pomiarowe. Pomiarów są prowadzone co 15 minut i wysyłane na serwer od razu po ich zakończeniu.

Urządzenie wykorzystuje czujniki DS18B20 (temperatura), DHT11 (wilgotność), BMP280 (ciśnienie). Zostało wykonane w oparciu o mikrokontroler z rodziny STM32.

Serwer nie jest częścią projektu. Został zrealizowany na RaspberryPi, na którym zainstalowany jest serwer Apache, obsługa PHP i baza danych MariaDB.



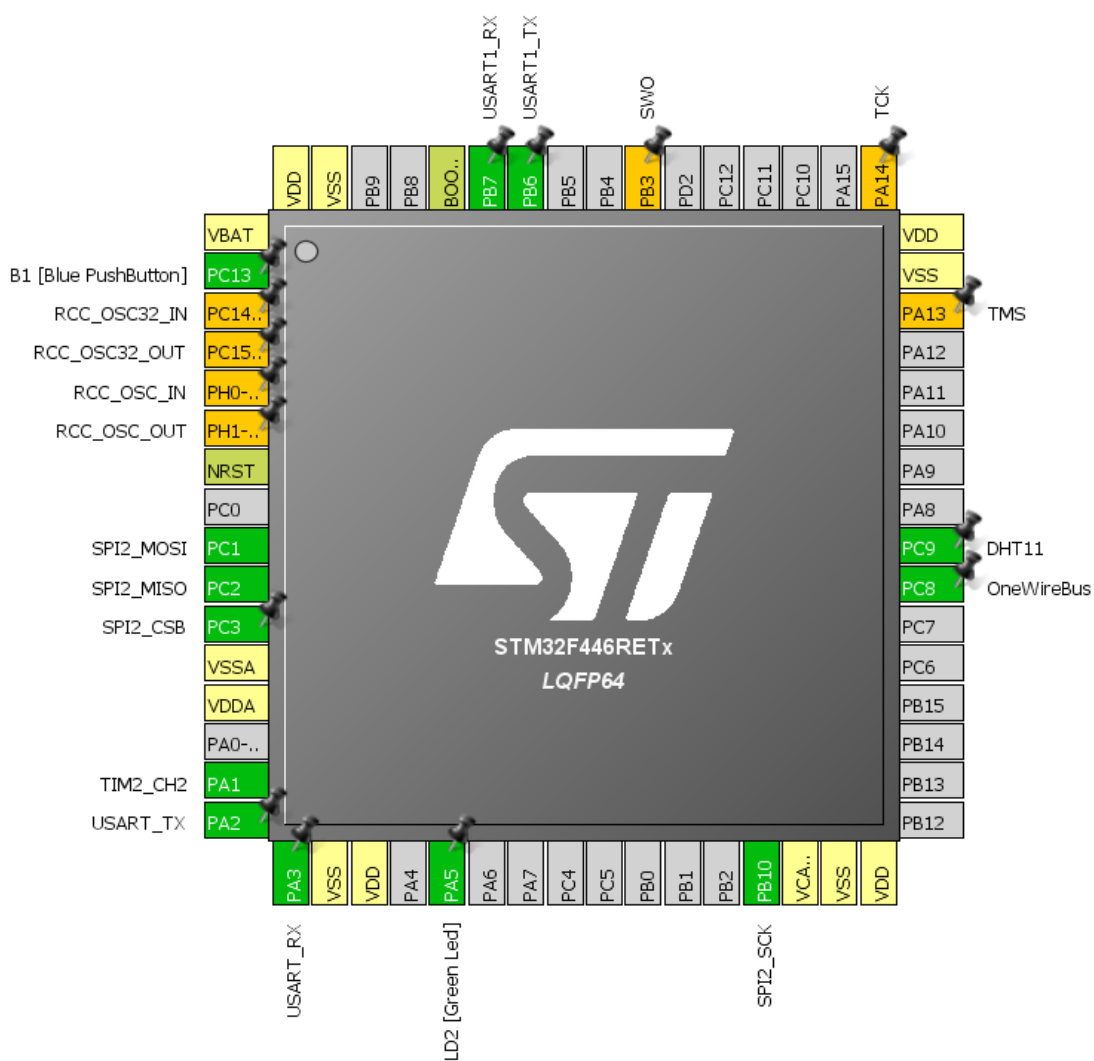
Rysunek 1: Zdjęcie urządzenia

## Zbierane dane pomiarowe:

- 3x Temperatura ze wszystkich czujników.
- Ciśnienie atmosferyczne
- Wilgotność względna

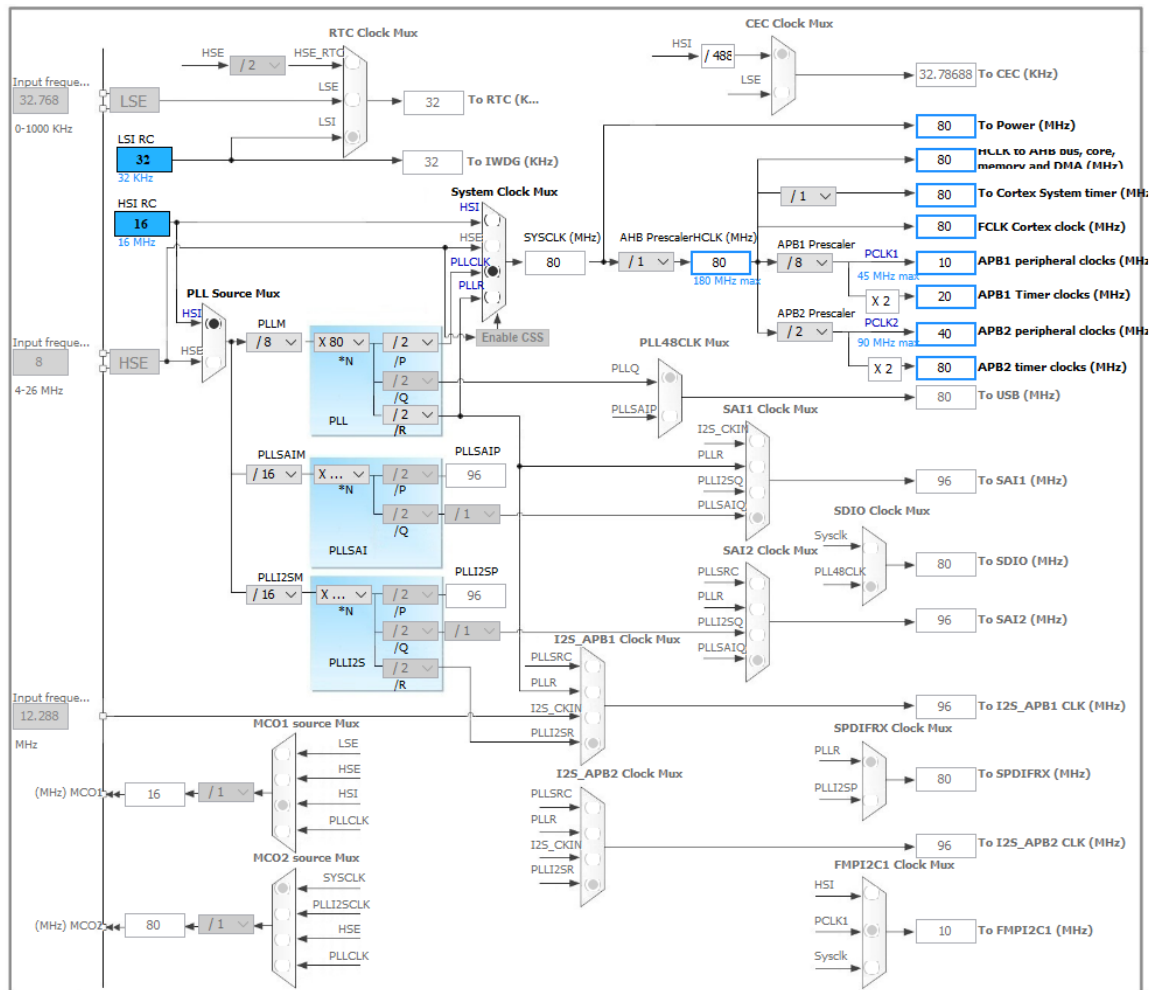
## 2 Konfiguracja mikrokontrolera

### 2.1 Konfiguracja pinów



Rysunek 2: Konfiguracja pinów mikrokontrolera w programie STM32CubeMX

## 2.2 Konfiguracja zegara



Rysunek 3: Konfiguracja zegara mikrokontrolera w programie STM32CubeMX

## 3 Konfiguracja peryferiów mikrokontrolera

Konfiguracja peryferiów mikrokontrolera została zamieszczona w formie tabel.

### 3.1 SPI2

Mode	Full-Duplex Master
<b>Basic Parameters</b>	
Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First
<b>Clock Parameters</b>	
Prescaler (for Baud Rate)	2
Baud Rate	<b>5.0 MBits/s*</b>
Clock Polarity	(CPOL) Low
Clock Phase	(CPHA) 1 Edge
<b>Advanced Parameters</b>	
CRC Calculation	Disabled
NSS Signal Type	Software

Tabela 1: Konfiguracja SPI2

### 3.2 SYS

Timebase Source	SysTick

Tabela 2: Konfiguracja SYS

### 3.3 TIM2

Clock Source	Internal Clock
Channel1	Output Compare No Output
Channel2	Input Capture direct mode
<b>Counter Settings</b>	
Prescaler (PSC - 16 bits value)	<b>1 *</b>
Counter Mode	Up
Counter Period (AutoReload Register - 32 bits value)	<b>4294967295 *</b>
Internal Clock Division (CKD)	No Division
<b>Trigger Output (TRGO) Parameters</b>	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	(Reset UG bit from TIMx_EGR)
<b>Output Compare No Output Channel 1</b>	
Mode	Frozen (used for Timing base)
Pulse (32 bits value)	0
CH Polarity	High
<b>Input Capture Channel 2</b>	
Polarity Selection	<b>Both Edges *</b>
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter (4 bits value)	0

Tabela 3: Konfiguracja TIM2

### 3.4 TIM3

Clock Source	Internal Clock
Channel1	Output Compare No Output
<b>Counter Settings</b>	
Prescaler (PSC - 16 bits value)	1 *
Counter Mode	Up
Counter Period (AutoReload Register - 32 bits value )	65535 *
Internal Clock Division (CKD)	No Division
<b>Trigger Output (TRGO) Parameters</b>	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	(Reset UG bit from TIMx_EGR)
<b>Output Compare No Output Channel 1</b>	
Mode	Frozen (used for Timing base)
Pulse (16 bits value)	10 000 *
CH Polarity	High

Tabela 4: Konfiguracja TIM3

### 3.5 TIM4

Mode	Clock Source
<b>Counter Settings</b>	
Prescaler (PSC - 16 bits value)	49999 *
Counter Mode	Up
Counter Period (AutoReload Register - 32 bits value )	359999 *
Internal Clock Division (CKD)	No Division
<b>Trigger Output (TRGO) Parameters</b>	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	(Reset UG bit from TIMx_EGR)

Tabela 5: Konfiguracja TIM4

### 3.6 USART1

Mode	Asynchronous
<b>Basic Parameters</b>	
Baud Rate	115200
Word Length	8 Bits (Including Parity)
Parity	None
Stop Bits	1
<b>Advanced Parameters</b>	
Data Direction	Recieve and Transmit
Over Sampling	16 Samples

Tabela 6: Konfiguracja USART1

### 3.7 USART2

<b>Mode</b>	<b>Asynchronous</b>
<b>Basic Parameters</b>	
Baud Rate	115200
Word Length	8 Bits (Including Parity)
Parity	None
Stop Bits	1
<b>Advanced Parameters</b>	
Data Direction	Recieve and Transmit
Over Sampling	16 Samples

Tabela 7: Konfiguracja USART2

## 4 Opis funkcji pinów

Nr pinu	Pin	Funkcja	Nazwa
2	PC13	GPIO_EXTI13	B1 [Blue PushButton]
3	PC14-OSC32_IN*	RCC_OSC32_IN	
4	PC15-OSC32_OUT*	RCC_OSC32_OUT	
5	PH0-OSC_IN*	RCC_OSC_IN	
6	PH1-OSC_OUT*	RCC_OSC_OUT	
9	PC1	SPI2_MOSI	
10	PC2	SPI2_MISO	
11	PC3	GPIO_Output	SPI2_CSB
15	PA1	TIM2_CH2	
16	PA2	USART2_TX	USART_TX
17	PA3	USART2_RX	USART_RX
21	PA5	GPIO_Output	LD2 [Green Led]
29	PB10	SPI2_SCK	
39	PC8	GPIO_Output	OneWireBus
40	PC9	GPIO_Output	DHT11
46	PA13*	SYS_JTMS-SWDIO	TMS
49	PA14*	SYS_JTCK-SWCLK	TCK
51	PC10	GPIO_Output	Test
55	PB3*	SYS_JTDO-SWO	SWO
58	PB6	USART1_TX	
59	PB7	USART1_RX	

Tabela 8: Opis funkcji pinów mikrokontrolera

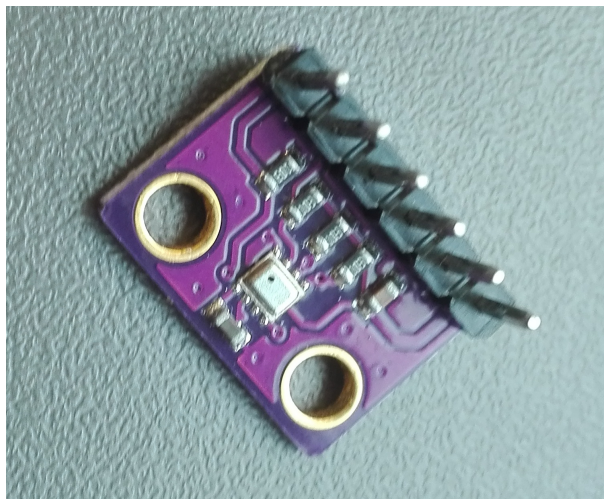


## 5 Opis wykorzystanych układów zewnętrznych

### 5.1 BMP280 - cyfrowy barometr, czujnik ciśnienia

#### 5.2 Opis układu

BMP280 to cyfrowy barometr zaprojektowany z myślą o zastosowaniach mobilnych. Znajduje się w małej obudowie. Małe wymiary i niski pobór energii pozwalają na zastosowanie go w zasilanych bateryjnie urządzeniach jak telefony komórkowe, moduły GPI i zegarki.



Rysunek 4: Czujnik BMP280

#### 5.3 Dane techniczne

Operation range (full accuracy)	Pressure: 300...1100 hPa
Operation range (full accuracy)	Temperature: -40... 85°C
Absolute accuracy (Temp. @ 0...+65°C)	±1 hPa
Relative accuracy (Temp. @ +25...+40°C)	± 0.12 hPa (typical)
Average current consumption (1 Hz data refresh rate)	2.74 uA, typical
Supply voltage VDD	1.71 ... 3.6 V
Resolution of data Pressure	0.01 hPa ( < 10 cm)
Resolution of data Temperature	0.01 C
Temperature coefficient offset (+25°...+40°C @900hPa)	± 0.12 hPa (typical)
Interface	I <sup>2</sup> C and SPI

Tabela 9: Parametry techniczne czujnik BMP280

#### 5.4 Modyfikowane rejestry

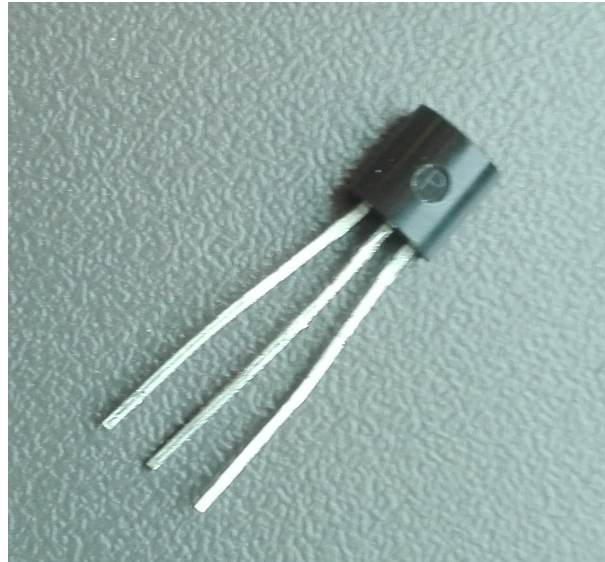
Rejestr	Adres	Wartość	Znaczenie
config	0xF5	0b000 <b>111</b> 00	Współczynnik filtru IIR 16
ctrl_meas	0xF4	0b <b>010</b> 11101	Oversampling temperatury x2
ctrl_meas	0xF4	0b010 <b>111</b> 01	Oversampling ciśnienia x16
ctrl_meas	0xF4	0b010111 <b>01</b>	Wymuszenie pojedynczego pomiaru

Tabela 10: Modyfikowane rejestry czujnika BMP280

## 5.5 DS18B20 - cyfrowy czujnik temperatury

### 5.5.1 Opis czujnika

DS18B20 to cyfrowy termometr zapewniający pomiar z rozdzielczością od 9 do 12 bitów. Posiada funkcję alarmu wyzwalanego po przekroczeniu lub obniżeniu wartości względem zadanego progu. Komunikuje się za pomocą interfejsu 1-Wire. Pracuje w zakresie temperatur od  $-55^{\circ}\text{C}$  do  $125^{\circ}\text{C}$



Rysunek 5: Czujnik DS18B20

### 5.5.2 Dane techniczne

- Komunikacja za pomocą jednego przewodu
- Każde urządzenie posiada unikatowy 64 bitowy kod
- Możliwość pracy w jednym z dwóch trybów zasilania
- Zakres napięć zasilania od 3.0V do 5.5V
- Mierzy temperaturę od  $-55^{\circ}\text{C}$  do  $125^{\circ}\text{C}$
- Posiada dokładność  $\pm 0.5^{\circ}\text{C}$  w zakresie  $10^{\circ}\text{C}$  -  $85^{\circ}\text{C}$

### 5.5.3 Wysyłane komendy

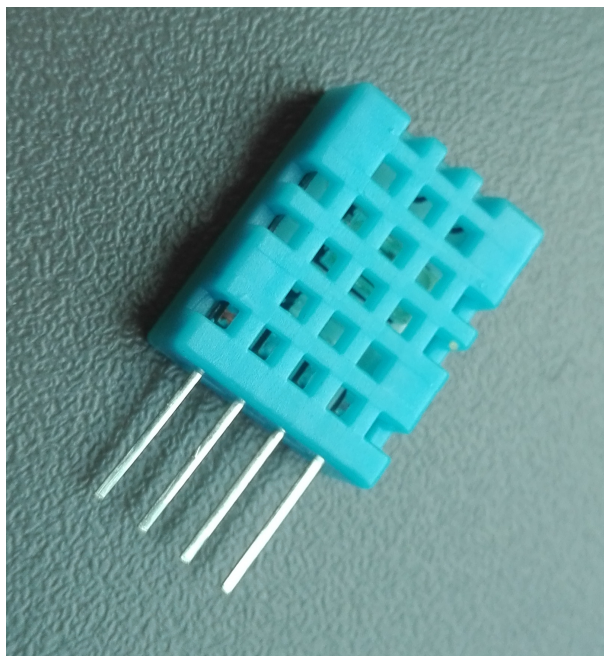
Komenda	Wartość	Działanie
Skip ROM	0xCC	Pozwala pominąć fazę dopasowania adresu do konkretnego urządzenia na linii.
Convert T	0x44	Wymusza pomiar temperatury
Read Scratchpad	0xBE	Odczytuje pamięć użytkową urządzenia ( w tym pomiary)

Tabela 11: Komendy wysyłane do czujnika DS18B20

## 5.6 DHT11 - Czujnik temperatury i wilgotności

### 5.6.1 Opis czujnika

DHT11 to cyfrowy czujnik temperatury i wilgotności. Oparty jest na sensorach zmieniających swoją rezystancję pod wpływem wilgotności oraz temperatury (termistor NTC)



Rysunek 6: Czujnik DHT11

### 5.6.2 Dane techniczne

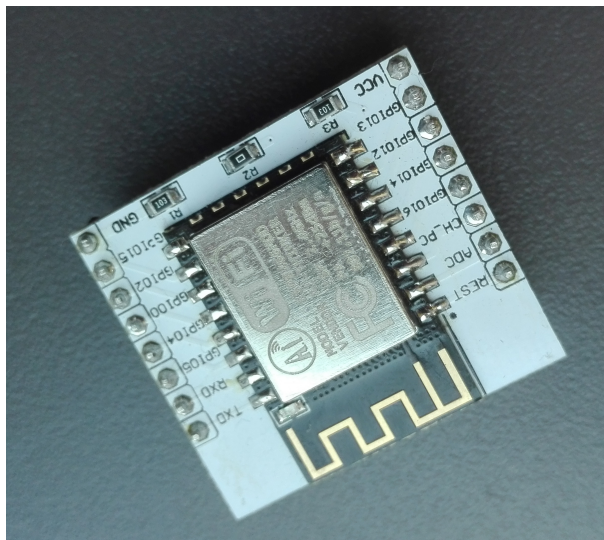
Parameters	Conditions	Minimum	Typical	Maximum
<b>Humidity</b>				
Resolution		1%RH	1%RH 8 Bit	1%RH
Repeatability			± 1%RH	
Accuracy	25°C		± 4%RH	
	0-50°C			± 5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
<b>Temperature</b>				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			± 1°C	
Accuracy		± 1°C		± 2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

Rysunek 7: Tabela z danymi technicznymi czujnika DHT11

## 5.7 ESP12F - Moduł WiFi

### 5.7.1 Opis modułu

Moduł WiFi oparty na chipie ESP8826. Do komunikacji wykorzystuje komendy AT.



Rysunek 8: Moduł ESP-12F

### 5.7.2 Dane techniczne

Napięcie zasilania	3.0 - 3.6 V
Średni pobierany prąd	80mA
Zakres temperatury pracy	od 40°C do 125°C

Tabela 12: Parametry techniczne modułu ESP-12F

### 5.7.3 Wykorzystywane komendy AT

Komenda	Działanie
AT	Sprawdzenie połączenia z modułem
AT+CWMODE_CUR=X	Ustawienie trybu pracy jako klient
AT+CWJAP_CUR=X	Połączenie ze wskazanym WiFi o danym hasle
AT+CIPSEND=X	Wysyłanie znaków do serwera

Tabela 13: Używane komendy AT

## 6 Opis działania programu

Działanie programu jest bardzo proste. Najpierw następuje inicjalizacja struktur przechowujących dane o czujnikach. Następnie nawiązywane jest połączenie z modułem WiFi i uruchamiany timer, który generuje przerwanie co 15 minut. W przerwaniu timera ustawiana jest flaga oznaczająca potrzebę przeprowadzenia pomiarów. Po jej odczytaniu w pętli głównej programu następuje wysyłanie zapytań do czujników, a zebrane dane pomiarowe są zapisywane do zmiennych globalnych. Po odczytaniu wszystkich danych wysyłane są one przez zapytanie GET do serwera, na którym znajduje się skrypt przetwarzający zapytanie i wpisujący dane do bazy danych.

Struktura zapytania kierowanego do serwera ma postać:

```
http://192.168.1.20/pages/receive/receive.php?CzujnikID=<id_czujnika>&
CzujnikHaslo=<haslo>&<kod_wartosci>=<wartosc>
```

gdzie:

- `< id_czujnika >` - ID Czujnika nadane przy rejestracji
- `< haslo >` - Hasło czujnika ustawione przez użytkownika
- `< kod_wartosci >` - Kod mierzonej wartości z tabeli kodów
- `< wartosc >` - Wartość zmierzona

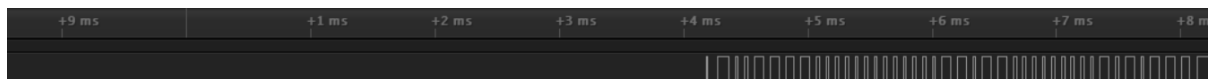
## 6.1 Komunikacja z DHT11

Komunikacja z czujnikiem DHT11 została zrealizowana w całości na przerwaniach w ramach maszyny stanów. Poszczególne etapy komunikacji to kolejne stany. Po poprawnym zakończeniu stanu następuje przejście do stanu następnego.

Poszczególne etapy komunikacji polegają na:

- Odczekanie 18ms po ustawieniu linii w stanie niskim za pomocą trybu Output Compare timera
- Podniesienie potencjału linii do poziomu zasilania
- Odczytanie odpowiedzi czujnika za pomocą trybu Input Capture timera

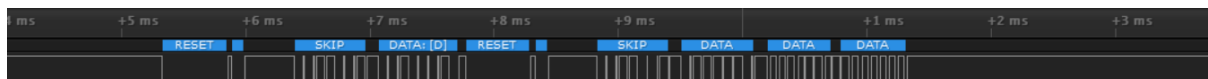
Na końcu komunikacji generowany jest callback informujący o sczytaniu danych.



Rysunek 9: Kształt sygnału na linii podczas komunikacji z DHT11

## 6.2 Komunikacja z DS18B20

Komunikacja z czujnikiem DS18B20 została zrealizowana w sposób analogiczny do tej z DHT11, z tą różnicą że każdy stan dzieli się na podstany. Z jednego stanu można przejść do innych, a wybór dokonywany jest w zależności od wywołanej funkcji, która zapisuje sekwencję stanów w specjalnym buforze. Przejście wszystkich podstanów jest obowiązkowe. Dane wysyłane w poszczególnych stanach są zapisywane w specjalnym buforze cyklicznym.



Rysunek 10: Kształt sygnału na linii podczas komunikacji z DS18B20

## 7 Podsumowanie

Projekt został zrealizowany w 99%. Zabrakło czasu na odnalezienie instrukcji usypiającej mikrokontroler. Program jest tak napisany, że gdyby to dopisać, urządzenie wybudziłoby się na czas dokonywania pomiarów.