

**Politechnika Śląska**  
**Wydział Automatyki, Elektroniki i Informatyki**



**Mikroprocesory - projekt**  
**Stacja pogodowa**

Studia niestacjonarne, sem. VII / NS1, EiT4/E1, rok akad. 2022/2023

Prowadzący: **dr hab. inż. Marcin Kubica**

Skład sekcji:

**Daniel Jędrzejczak**

**Marcin Szoltysek**

**Damian Śnieguła**

# **SPIS TREŚCI**

<b>1 CEL</b>	<b>3</b>
<b>2 OPIS TECHNICZNY</b>	<b>3</b>
<b>3 SCHEMAT IDEOWY</b>	<b>3</b>
<b>4 KOD PROGRAMU ATMEGA328P</b>	<b>4</b>
<b>5 WYKAZ ELEMENTÓW</b>	<b>6</b>
<b>6 PROTOTYP</b>	<b>6</b>

# 1 CEL

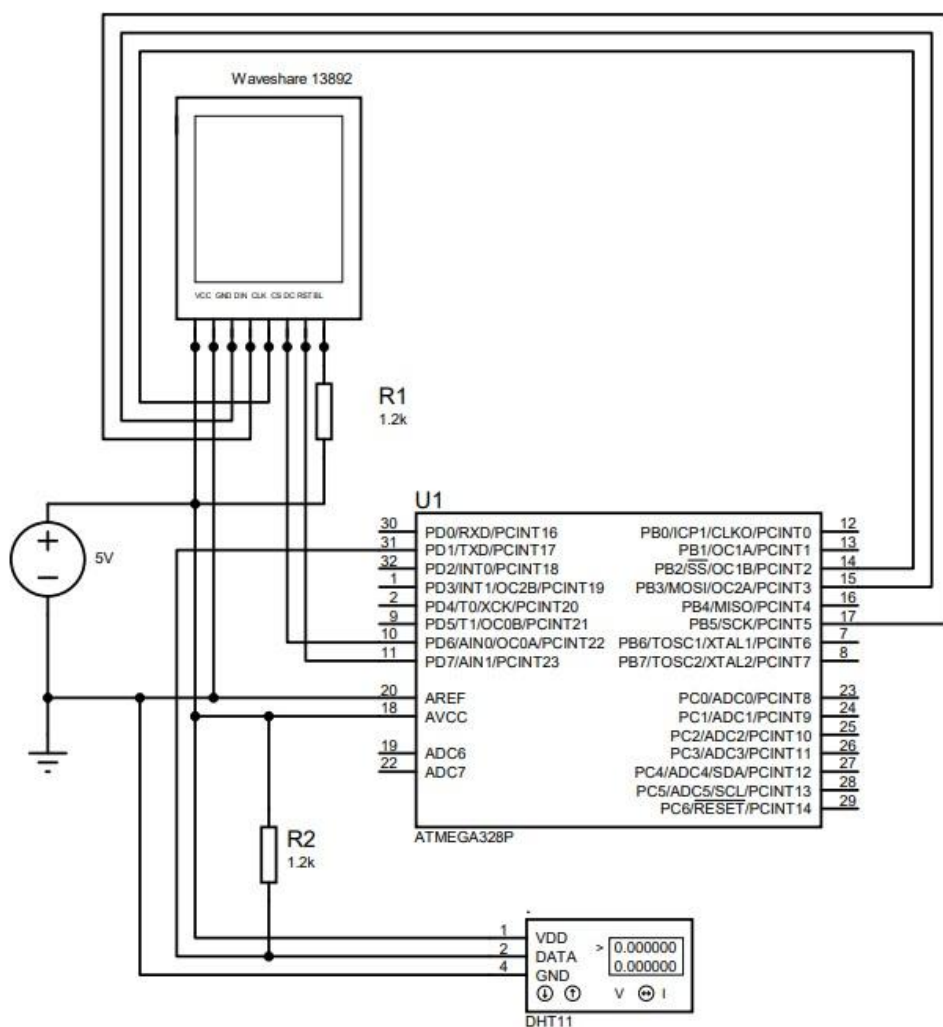
Celem projektu jest stworzenie stacji pogodowej, która ma dokonywać pomiaru temperatury i wilgotności. Wynik pomiaru ma być wyświetlany na wyświetlaczu LCD TFT. Pomiar ma być dokonywany co 20 sekund. Projekt zostanie zrealizowany na ATMEGA328P.

## 2 OPIS TECHNICZNY

Urządzeniem steruje ATMEGA328P, która pobiera informację z czujnika DHT11 o stanie temperatury i wilgotności. Mikroprocesor komunikuje się z czujnikiem poprzez interfejs cyfrowy (magistrale 1 Wire).

Czujnik posiada już na swojej płytce rezystor "podciągający" i kondensator filtrujący. Dane o aktualnych odczytach z czujnika są ekranizowane na wyświetlaczu TFT, który również komunikuje się z procesorem poprzez interfejs cyfrowy (magistrala SPI).

## 3 SCHEMAT IDEOWY



# 4 KOD PROGRAMU ATMEGA328P

```
#include<avr/io.h>
#include<stdlib.h>
#include<util/delay.h>
#include "DHT11/dht.h"
#include "include/spi.h"
#include "include/st7735.h"
#include "include/st7735_gfx.h"
#include "include/st7735_font.h"
#include "fonts/free_sans.h"

uint32_t i = 1;
int8_t temperatura = 0;           //zmienne dla dht
int8_t wilgotnosc = 0;
uint8_t temp_x = 0;
uint8_t hum_x = 0;
uint8_t temp_x_prev = 0;
uint8_t hum_x_prev = 0;

char temp[10];
char hum[10];

int map();
int main(void) {
    spi_init();
    st7735_init();

        _delay_ms(100);
        st7735_set_orientation(ST7735_PORTRAIT_INV);
        _delay_ms(100);
        st7735_fill_rect(0, 0, 130, 165, ST7735_COLOR_GREEN);
        st7735_draw_text(5, 18, "Temperature:", &FreeSans, 1, ST7735_COLOR_BLACK);
        st7735_draw_text(5, 58, "Humidity:", &FreeSans, 1, ST7735_COLOR_BLACK);

        // dodatkowe funkcje
        st7735_draw_line(0, 80, 128, 80, ST7735_COLOR_BLACK);
        st7735_draw_line(0, 107, 128, 107, ST7735_COLOR_BLACK);
        st7735_draw_line(0, 109, 128, 109, ST7735_COLOR_BLACK);
        st7735_draw_line(0, 158, 128, 158, ST7735_COLOR_BLACK);
        st7735_draw_line(0, 159, 128, 159, ST7735_COLOR_BLACK);

    while(1){

        if(dht_gettemperaturehumidity(&temperatura, &wilgotnosc) != -1) {

            //zamiana int na char
            itoa(temperatura, temp, 10);
            itoa(wilgotnosc, hum, 10);

            // zamazywanie poprzediej lczby kolorem tła ( kolejno: x, y, szerokość, wysokość, kolor)
            st7735_fill_rect(0, 25, 128, 20, ST7735_COLOR_GREEN);
            st7735_fill_rect(0, 65, 128, 15, ST7735_COLOR_GREEN);

            st7735_draw_text(5, 38, temp, &FreeSans, 1, ST7735_COLOR_BLACK);
            st7735_draw_text(50, 38, "st. C", &FreeSans, 1, ST7735_COLOR_BLACK);
            st7735_draw_text(5, 78, hum, &FreeSans, 1, ST7735_COLOR_BLACK);
            st7735_draw_text(50, 78, " % ", &FreeSans, 1, ST7735_COLOR_BLACK); //wyswietla

        }else{

            // zamazywanie poprzediej lczby kolorem tła ( kolejno: x, y, szerokość, wysokość, kolor)
            st7735_fill_rect(0, 25, 128, 20, ST7735_COLOR_GREEN);
            st7735_fill_rect(0, 65, 128, 20, ST7735_COLOR_GREEN);
```

```

st7735_draw_text(5, 40, "ERROR", &FreeSans, 1, ST7735_COLOR_RED);
st7735_draw_text(5, 80, "ERROR", &FreeSans, 1, ST7735_COLOR_BLUE);
}

// rysowanie wykresów
if(i <= 127){
    //Y = Y0 + (Y1-Y0)*(X-X0)/(X1-X0), yo = min_out, x0 = min_in, Y1 = max_out, x1 = max_in
    //mapowanie wilgotnosci na zakres wykresu
    uint8_t h_x = map(wilgotnosc, 30, 100, 0, 48);
    hum_x = 157 - h_x;

    //mapowanie temperatury na zakres wykresu
    uint8_t t_x = map(temperatura, 10, 35, 0, 26);
    temp_x = 106 - t_x;

    ///wyświetlanie punktami
    //st7735_draw_pixel(i, temp_x, ST7735_COLOR_RED );
    //st7735_draw_pixel(i, hum_x, ST7735_COLOR_BLUE );

    //wyświetlanie linią
    st7735_draw_line((i-1), temp_x_prev, i, temp_x, ST7735_COLOR_RED );
    st7735_draw_line((i-1), hum_x_prev, i, hum_x, ST7735_COLOR_BLUE );

    temp_x_prev = temp_x;
    hum_x_prev = hum_x;

}
else {
    i = 0;
    st7735_fill_rect(0, 81, 128, 26, ST7735_COLOR_GREEN);
    st7735_fill_rect(0, 110, 128, 48, ST7735_COLOR_GREEN);
}

i++;
_delay_ms(20000); //20sek
}

}

int map( int x, int in_min, int in_max, int out_min, int out_max ) {
    int a = (x-in_min)*(out_max-out_min);
    int b = (in_max-in_min);
    return (( a + (b/2)) / b + out_min);
}

```

## 5 WYKAZ ELEMENTÓW

Projekt wykonano z użyciem płytki stykowej oraz przewodów połączeniowych. Do wgrywania programu posłużono się konwerterem USB-UART FTDI.

Element	Ilość
Mikrokontroler AVR - ATmega328P-PU DIP	1
Czujnik temperatury i wilgotności DHT11	1
Wyświetlacz LCD TFT - kolorowy 1,8" 128x160px SPI	1
Rezystor węglowy 1/4W 1,2kk $\Omega$	2

## 6 PROTOTYP

