

Systemy Kontrolno-Pomiarowe laboratorium

Raport z projektu

Stacja pogodowa – pomiar temperatury oraz
wilgotności

Projekt wykonali:

Michał Woźniak

Jakub Pisarczyk

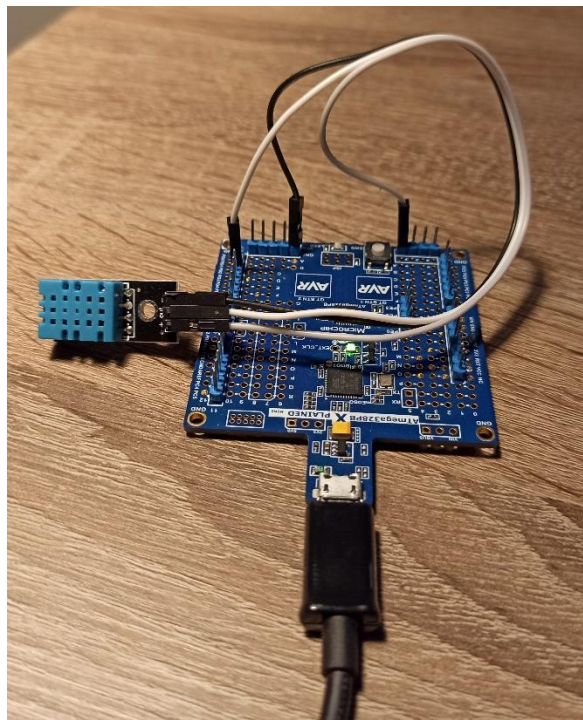
Prowadzący: dr inż. Jakub Sorocki

1. Założenia projektu

Pomiar temperatury oraz poziomu wilgotności przy użyciu czujnika DHT11, zaprogramowanie płytki ATmega328PB do komunikacji z czujnikiem oraz komputerem przy użyciu programu Microchip Studio, zaprojektowanie interfejsu graficznego do odczytu danych przy użyciu programu LabVIEW.

2. Udokumentowanie poszczególnych etapów

a) Połączenie fizyczne



Rys 1. Czujnik DHT11 podpięty do płytki ATmega328PB

Czujnik podpięty do zasilania 5V, GND oraz wyjście danych OUT do pinu PD2.

b) Kod obsługujący czujnik DHT11

```
#define F_CPU 16000000UL // Clock Speed ATmega328PB 16 MHz
#include <avr/io.h>
#include <stdlib.h>
#include <util/delay.h>
#include "usart.h"
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1
#define DHT11_PIN PORTD2 // PIN PD2
uint8_t c=0,I_RH,D_RH,I_Temp,D_Temp,Checksum;

void Request() /* MCU wysła impuls startowy */
{
    DDRD |= (1<<DHT11_PIN);
    PORTD &= ~(1<<DHT11_PIN); /* Ustawienie pinu na stan niski*/
    _delay_ms(18); /* czeka 18ms */
    PORTD |= (1<<DHT11_PIN); /* ustawia pin na stan wysoki */
}

void Response() /* gdy otrzyma odpowiedź z czujnika DHT11 */
{
    DDRD &= ~(1<<DHT11_PIN);
    while(PIND & (1<<DHT11_PIN));
    while((PIND & (1<<DHT11_PIN))==0);
    while(PIND & (1<<DHT11_PIN));
}

uint8_t Receive_data() /* odbiór danych */
{
    for (int q=0; q<8; q++)
    {
        while((PIND & (1<<DHT11_PIN)) == 0); /* sprawdzenie stanu odebranego bitu danych */
        _delay_us(30);
        if(PIND & (1<<DHT11_PIN))/* jeśli impuls służy niż 30us to stan logiczny „0” */
            c = (c<<1)|(0x01);
        else /* w innym wypadku stan logiczny „1” */
            c = (c<<1);
        while(PIND & (1<<DHT11_PIN));
    }
    return c;
}

int main(void)
{
    USART_Init(MYUBRR);
    char data[5];
    _delay_ms(1000); //stabilizacja zasilania

    while(1)
    {
        Request(); /* impuls startowy */
        Response(); /* otrzymanie odpowiedzi */
        I_RH=Receive_data(); /* zapisuje pierwsze 8 bitów w I_RH */
        D_RH=Receive_data(); /* zapisuje kolejne 8 bitów w D_RH */
        I_Temp=Receive_data(); /* zapisuje kolejne 8 bitów w I_Temp */
        D_Temp=Receive_data(); /* zapisuje kolejne 8 bitów w D_Temp */
        CheckSum=Receive_data();/* zapisuje kolejne 8 bitów w CheckSum */

        if ((I_RH + D_RH + I_Temp + D_Temp) != CheckSum){ // Sprawdzenie sumy kontrolnej
            USART_PutS("Error");
        }
        else
        {
            itoa(I_RH,data,10);
            USART_PutS(data);

            itoa(I_Temp,data,10);
            USART_PutS(data);
        }
        _delay_ms(1000);
    }
}
```

c) Kod definiujący funkcje połączenie USART

```
#include <avr/io.h>
#include <stdlib.h>
#include <util/delay.h>

#include "usart.h"

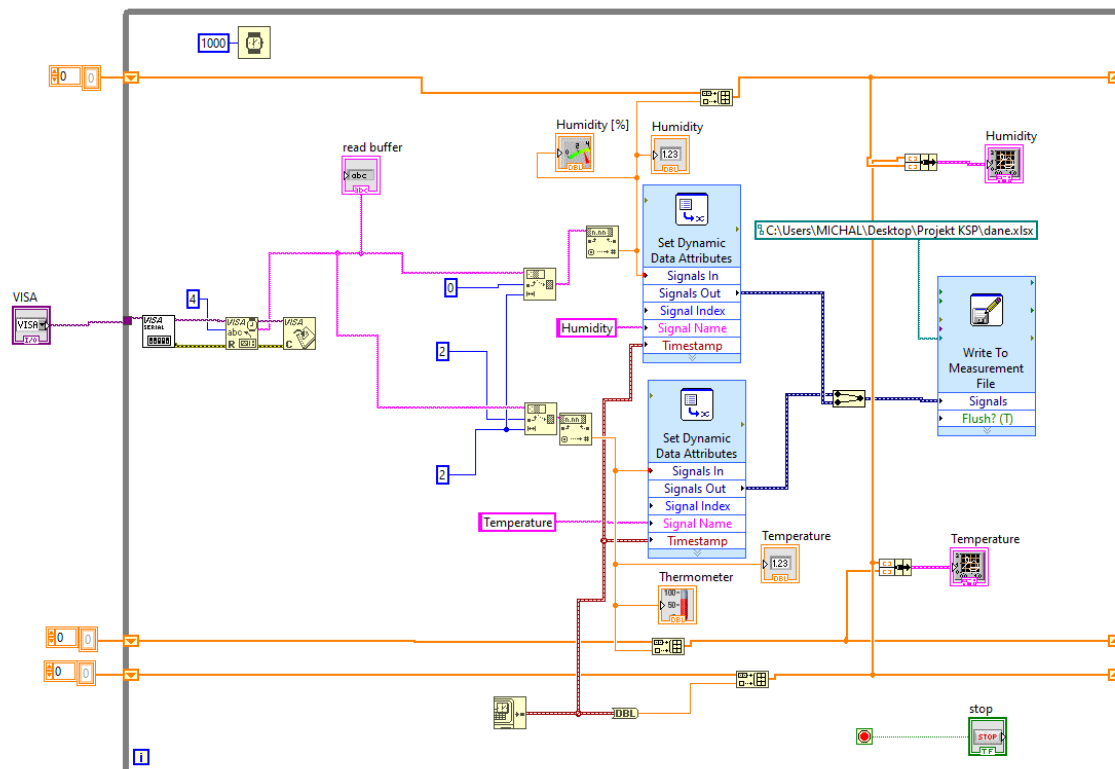
void USART_Init( unsigned int ubrr) // Inicjalizacja USART
{
    /* Ustawia bitrate */
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    /* odbieranie i wysyłanie */
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    /* format ramki 8 bitów danych i 1 bit stopu */
    UCSR0C = (3<<UCSZ00);
}

void USART_PutC( char data ) // ZNAK CHAR
{
    /* oczekiwanie na dane */
    while ( !( UCSR0A & (1<<UDRE0)) );
    /* wkłada dane do bufora */
    UDR0 = data;
}

void USART_PutS( char * s) // STRING - CIĄG CHAR'ÓW
{
    while( *s ) USART_PutC( *s++ );,
}

unsigned char USART_Receive( void )
{
    /* oczekiwanie na dane */
    while ( !(UCSR0A & (1<<RXC0)) );
    /* otrzymuje i zwraca dane z bufora*/
    return UDR0;
}
```

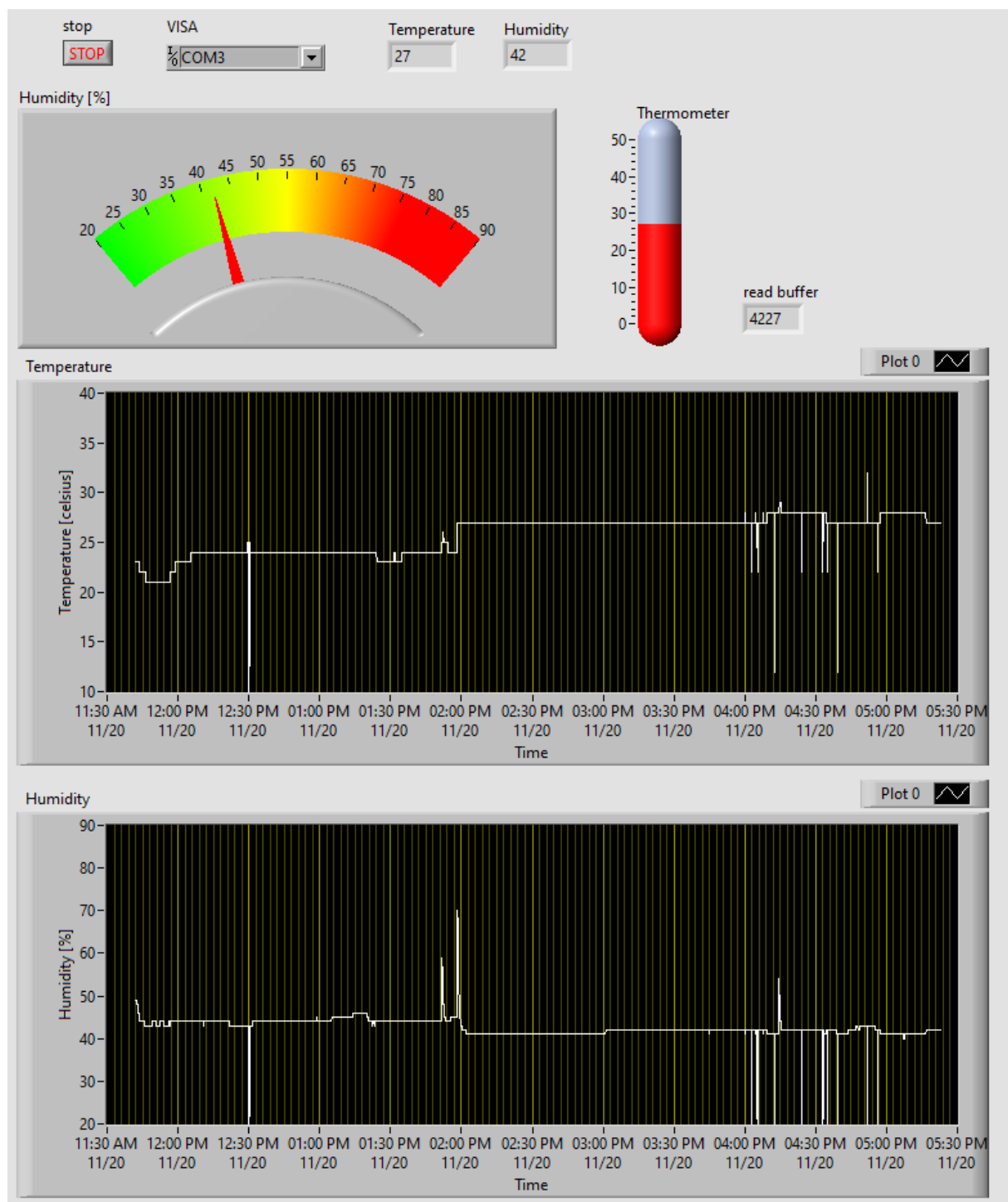
d) Diagram blokowy projektu stacji pogodowej w LabVIEW



Rys 2. Diagram blokowy

Układ otrzymuje dane z płytki na porcie COM3, zlicza 4 bity, wysyła dane w postaci „string” do kolejnych modułów. String dzielony jest po 2 znaki na część reprezentującą dane temperaturowe w postaci jednostki stopni Celsjusza, oraz procentowej wartości wilgotności w powietrzu. Następnie wartości te zamieniane są na zmienne liczbowe. Wartości w takiej postaci wyświetlane są na wykresie oraz w okienku w postaci cyfr. Dodatkowy etap to połączenie wyników w 2 wymiarowe tablice razem z wartością czasu rzeczywistego, oraz wyświetlenie na wykresach. Dane te są także zapisywane do pliku .xlsx który możemy otworzyć za pomocą np. Excela. Program pracuje w nieskończonej pętli while, konfigurowanej przez możliwość zmiany czasu ponownego wykonania pętli, podanego w ms.

e) Interfejs graficzny projektu stacji pogodowej w LabVIEW



Rys 3. Interfejs graficzny

(Podczas 6 godzinnego pomiaru, skoki pomiarowe wywołane przez ingerencje człowieka w celu zmiany właściwości środowiska pogodowego w jakim znajdował się czujnik)

f) Zapis odczytanych pomiarów do pliku .xlsx

	A	B	C
1	Time	Temperature	Humidity
2	20.11.2021 19:54:20,204	28	42
3	20.11.2021 19:54:21,441	28	42
4	20.11.2021 19:54:22,720	28	42
5	20.11.2021 19:54:23,992	28	42
6	20.11.2021 19:54:25,330	28	42
7	20.11.2021 19:54:26,604	28	42
8	20.11.2021 19:54:27,878	28	42
9	20.11.2021 19:54:29,156	28	44
10	20.11.2021 19:54:30,471	28	47
11	20.11.2021 19:54:31,831	28	50
12	20.11.2021 19:54:33,147	28	53
13	20.11.2021 19:54:34,447	28	57
14	20.11.2021 19:54:35,753	28	60
15	20.11.2021 19:54:37,036	28	63
16	20.11.2021 19:54:38,326	28	66
17	20.11.2021 19:54:39,618	28	68
18	20.11.2021 19:54:40,917	28	70
19	20.11.2021 19:54:42,250	29	72
20	20.11.2021 19:54:43,568	29	72
21	20.11.2021 19:54:44,858	29	75
22	20.11.2021 19:54:46,189	29	76
23	20.11.2021 19:54:47,485	30	76
24	20.11.2021 19:54:48,782	30	78
25	20.11.2021 19:54:50,071	30	79
26	20.11.2021 19:54:51,364	30	79
27	20.11.2021 19:54:52,649	30	81
28	20.11.2021 19:54:53,949	31	82
29	20.11.2021 19:54:55,258	31	82
30	20.11.2021 19:54:56,580	31	81

Rys 4. Zapis danych do pliku

Trzy kolumny zawierające aktualny czas wykonania pomiaru, temperaturę oraz poziom wilgotności.