UCG
Univerzitet Crne Gore

UNIVERSITY OF MONTENEGRO
Faculty of Electrical Engineering

The measurement of temperature and humidity using Arduino with interrupt

Course: Industrial electronics

Authors

Radovan Rabrenović
Alen Šabotić
Mevludin Hot

Mentor

Prof. PhD Radovan Stojanović

March, 2020

Contents

Abstract

Reading speed of DHT11 sensor values was controlled using interrupt in this project. DHT11 is one of the multifunctional sensors that aside from temperature measuring sensor also has humidity measuring sensor. Because of its huge complexity and its software structure this project has big importance. Temperature and humidity measuring is done by using analogue pins on Arduino UNO platforms, while the user has a possibility of adjusting the reading speed of DHT11 sensors using potentiometer. DHT11 sensor has three pins, two for power supply and one for sending information. While potentiometer also has three pins and works the same way the sensor does.

2 Introduction

Our task was to measure temperature and humidity using Arduino UNO with its components and DHT11 sensor as well as potentiometer for adjusting the working frequency. Also, the goal was to find the a solution of the mentioned problem using timer interrupt.

3 Software realization

As said in the abstract of this research, the research is about controlling the reading speed of temperature as well as humidity, using DHT11. Knowing the fact that DHT11 sensor is multifunctional it is crucial to define libraries at the beginning. Command #include <dht.h> enables library for DHT11 sensor. On the other hand potentiometer functions the same way as the voltage divider, so there is no need to use particular library for it. in photo no.1 there is a scheme of connecting DHT11 sensor and potentiometer with arduino.

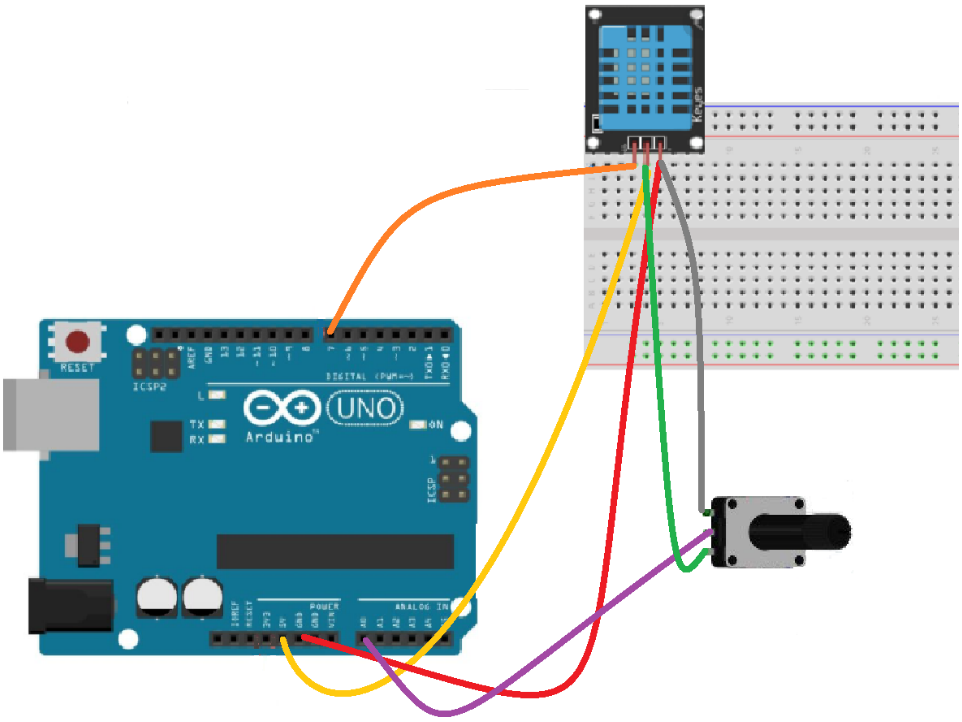

Photo 1. Scheme of connecting DHT11 sensor and potentiometer with arduino.

Communication between DHT11 sensor and potentiometer is done by using Arduino UNO platform. Therefore the interrupt was used, particularly "compare match A" on timer 1. Arduino has 3 timers (timer 0, timer 1 and timer 2), timer 0 and timer 2 being 10-bit (their resolution is 1024) and timer 1 is 16-bit (its resolution is 65536). Timer 1 was used for the sake of greater precision. For the interrupt to be successfully realized it is necessary to define values of all embedded registers. Registers TCCR1A and TCCR1B are set on 0 and then the certain bits were registered in them depending on their function. Considering that CTC mode was chosen

(Clear Timer on Compare Match Mode), it is necessary to put 1 in place of WGM12 bit in the TCCR1B register. It is done using the command: TCCR1B |= (1 << WGM12). This command is shifting 1 for the value of WGM12 bit position. Because the compare match A interrupt is used, it is necessary to define frequency for that to happen (f_{cmpA}). Defining that frequency is done by using formula provided by the manufacturer:

$$OCR1A = \frac{f_{clk}}{N f_{cmpA}} - 1$$

In the formula above f_{clk} is the frequency of the Arduino UNO platform and that is 16MHz, N is the prescaler that can have values of 8, 16, 256 and 1024, and OCR1A is the register which value has the number of cycles needed for achieving wanted frequency (f_{cmpA}). Because the prescaler with the value of 256 was chosen, therefore 1 is put in the CS12 bit of the TCCR1B register. Wanted frequency is 10Hz ($f_{cmpA} = 10$) and prescaler value is 256, according to the formula above, calculated number of cycles is 6249. So, 6249 cycles are needed to achieve frequency of 10Hz. Compare match A is going to happen when the value of TCNT1 register gets to the value of OCR1A register, it is necessary to set TCNT1 register on 0 at the beginning.
Compare match A interrupt is enabled with using OCIE1A bit in register TIMSK1. For Arduino to do interrupt it is necessary to also enable register that represents the global interrupt register. Register is enabled with the use of embedded function sei().
This research is based on the experimental circuit that is shown in photo 2.

Experimental circuit on which this research is based is shown in photo 2.

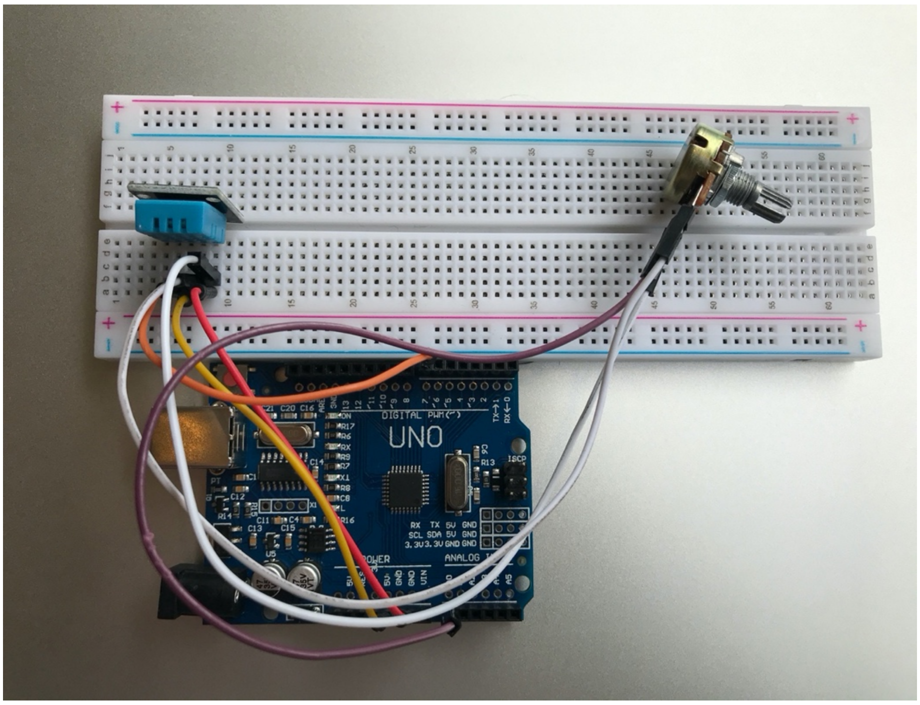

Photo 2.

The entire code is in the appendix bellow.

```
#include <dht.h>
dht DHT;

#define DHT11_PIN 7
#define potenciometar A0

int brojac = 0; // broji koliko je puta usao u interrupt
char duzina_trajanja;
int a = 1;
int vrijednost_potenciometra = 0;
int max_vrijednost_sekundi = 3; // pri maximalnoj otpornosti potenciometra arduino ce
ocitavati vrijednost sa
                                // senzora svakih max_vrijednost_sekundi+1 sekundi; u ovom slucaju je to
10s!!!
void setup(){
  cli(); // disable all interrupts

  TCNT1 = 0;
  TCCR1A = 0;
  TCCR1B = 0;
  TCCR1B |= (1 << WGM12); // CTC mode
  TCCR1B |= (1 << CS12); // preskaler
  OCR1A = 6249;
  TIMSK1 |= (1 << OCIE1A); // enable compare match A interrupt

  sei(); // enable all interrupts

  Serial.begin(9600);
}

void loop()
{
  if(brojac == 1) {
    citanje_senzora();
    vrijednost_potenciometra = analogRead(potenciometar);
  }


}

ISR(TIMER1_COMPA_vect) {
  brojac ++;
  if(brojac == a*10) {
    a = ((vrijednost_potenciometra*max_vrijednost_sekundi)/1023) + 1;
    brojac = 0;
    Serial.print("Vrijeme ocitavanja senzora je: ");
    Serial.println(a);
  }
}

void citanje_senzora() {
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperatura je: ");
  Serial.println(DHT.temperature);
  Serial.print("Vlaznost: ");
  Serial.println(DHT.humidity);
}
```

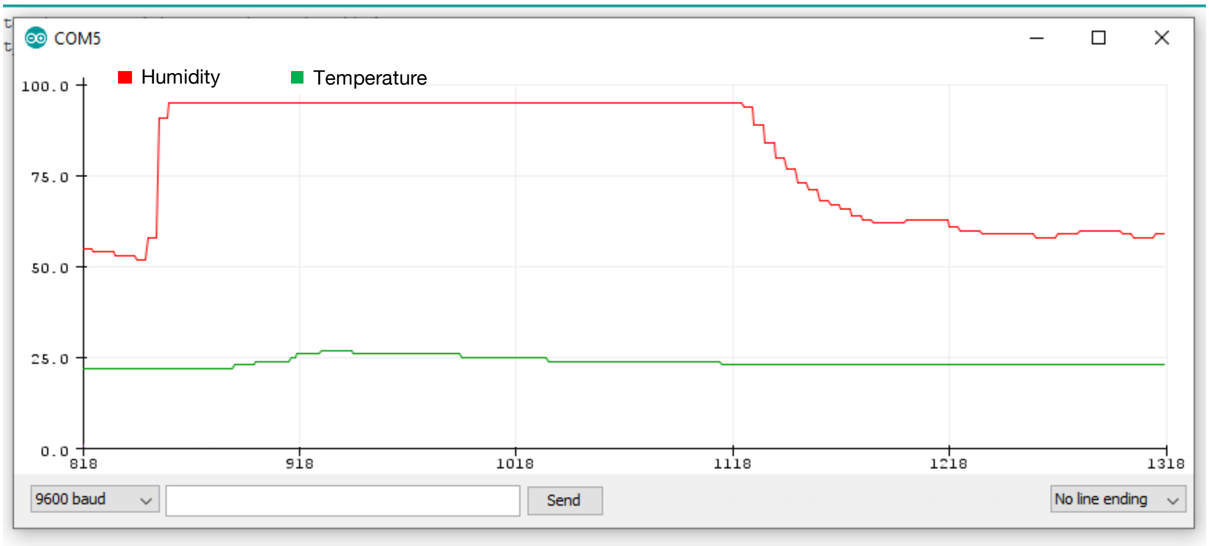

Photo 3. Temperature and humidity sensors readings

4 Link for video

https://youtu.be/fELoZ3Y2T54

5 Literature:

[1] http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
[2]https://create.arduino.cc/projecthub/techno_z/dht11-temperature-humidity-sensor-98b03b