

University of Montenegro  
Faculty of Electrical  
Engineering

Course: Industrial Electronics

Theme: Creating fluxmeter using light sensor and Arduino

Mentor :

Prof. dr Radovan Stojanović

Students:

Tijana Radojičić

Ognjen Vujičić

Isidora Šćepanović

Date and place: 10/03/2020

Podgorica

## Contents

Summary.....	3
Problem description .....	3
Software solution description .....	3
Hardware realisation .....	7
Verification on board.....	8
Literature.....	8

## Summary

This is a lab project, made the University of Montenegro, with mentoring of prof. dr Radovan Stojanovic. In further text, the problem is explained in details and we added explanations of hardware and software. Our code was written in C using Arduino Integrated Development Environment. In this project we want to demonstrate how to make fluxmeter using simple light sensor and standard Arduino board.

## Problem description

Our job is to project the fluxmeter using Arduino. It is supposed to measure the intensity of light detected by the photoresistor, and to show results in lux ( $lx$ ) on Arduino's Serial monitor. For this project we needed the help of MATLAB and android app for measuring intensity of light using sensor on mobile phone called "Lux Light Meter".

## Software solution description

- **General explanation of software**

For the realisation of this project we used the light sensor and connected it to Arduino WAVEGAT UNO R3 board (Figure 4). First, we wrote a piece of code for sensor reading using *analogRead()* function and verified it using Serial monitor of Arduino software.

When we were convinced that the readings are all right we then added timer interrupt into our code. Timer interrupt is used for performing task at specified time intervals regardless of other programme parts. Interrupts are useful in measuring signal at equal time intervals, calculating those intervals, sending a signal of a specific frequency etc.

Another thing we wanted to show in this project is use of *SerialEvent* function in Arduino. This function is called inside the *loop()* and it is used to collect serial data. In our case we used *SerialEvent* to change sampling frequency. Initially, we set our sampling frequency at 1Hz. Sending character 'a' in Serial monitor will increase frequency to 5Hz. If character 'b' is sent, sampling frequency is returned to 1Hz.

- **Calibration**

Our goal was to adjust software to show measuring results in lux similar to existing Android application called "Lux Light Meter", which uses mobile phone's light sensor to measure flux of light. So we had to compare our measurements with the measurements of this app for the same intensity of light. In other words, we needed to find mathematical dependency of intensity of light measured with app on value returned by Arduino. We wrote down two sets of thirty measurements done using Arduino and using the app. Those measurements (with photoresistor and with mobile phone) are done simultaneously, which means they are performed under same conditions in order to get as precise results as we can. Then we entered these results in math software MATLAB to calculate the equation.

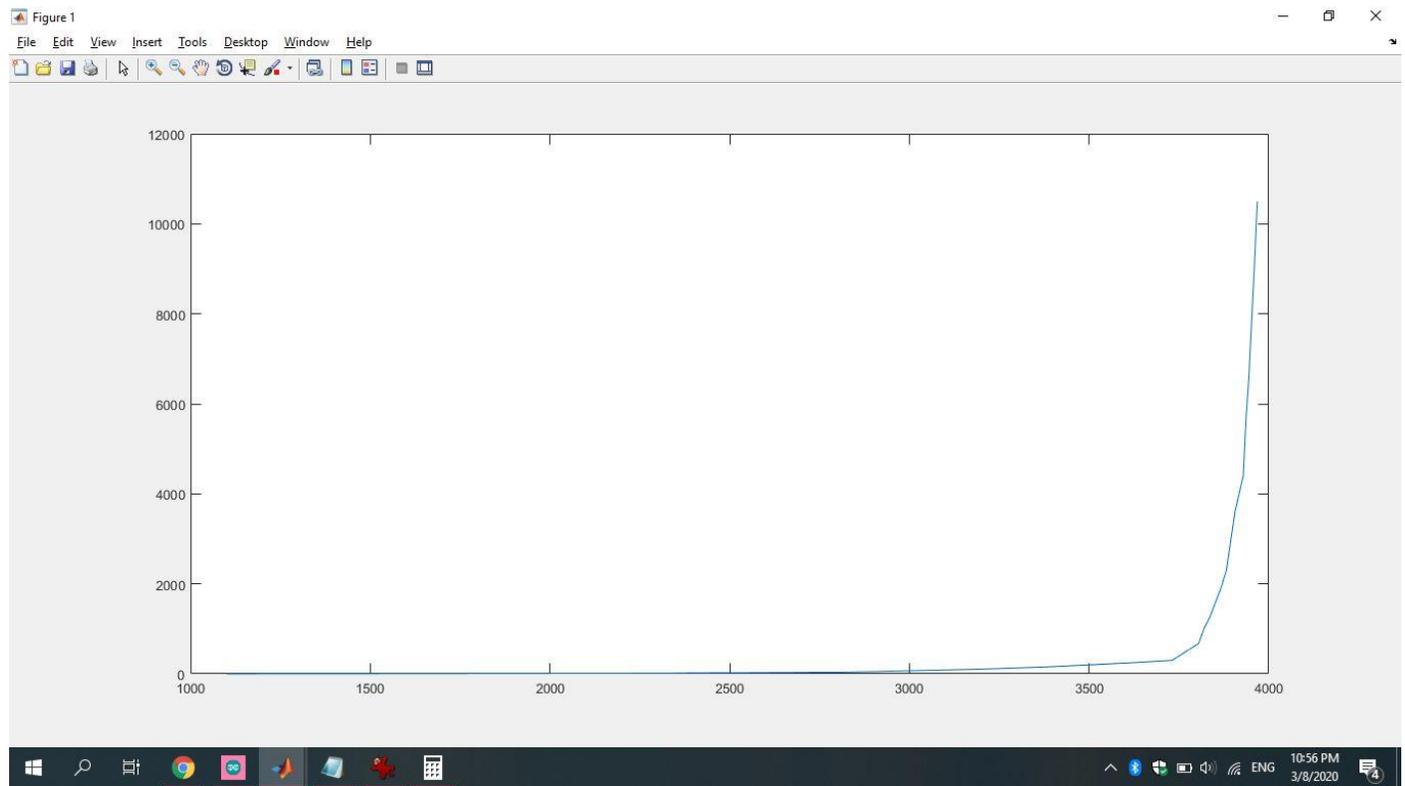


Figure 1. Curve

The curve from our measurements which recalls to the exponential curve is shown in Figure 1. As shown in the figure, for values returned by Arduino over approximately 3500, the values on y axis rapidly increases. Using single equation of this curve caused our program to return value of zero for the majority of measuring conditions.

Because of that we decided to split the range into two. So, instead of having one equation for the whole range of values on x axis, which would return zero for major part of values, we created two equations: one for values beneath 3700, and another for values over 3700. For both ranges we used about 10 to 15 pairs of values.

We used *Curve Fitting Tool* from MATLAB for approximation. Using *Curve Fitting Tool* involves entering two arrays: array x (measurements on Arduino) and array y (measurements on app). Then we choose the type of equation we want – in our case it was exponential equation  $f(x) = a \cdot \exp(b \cdot x)$ . MATLAB returns coefficients  $a$  and  $b$ . Results for both approximations are shown in Figures 2 and 3.

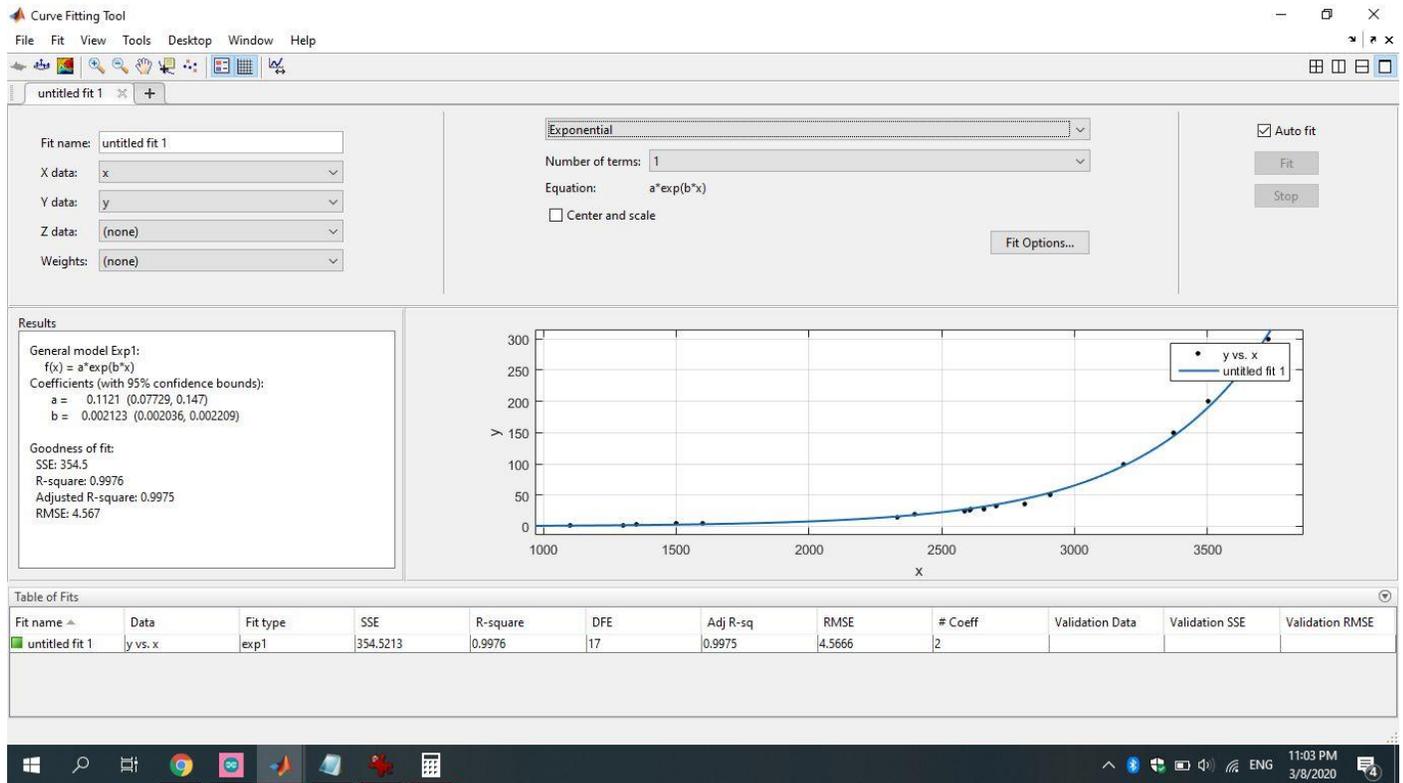


Figure 2. First equation

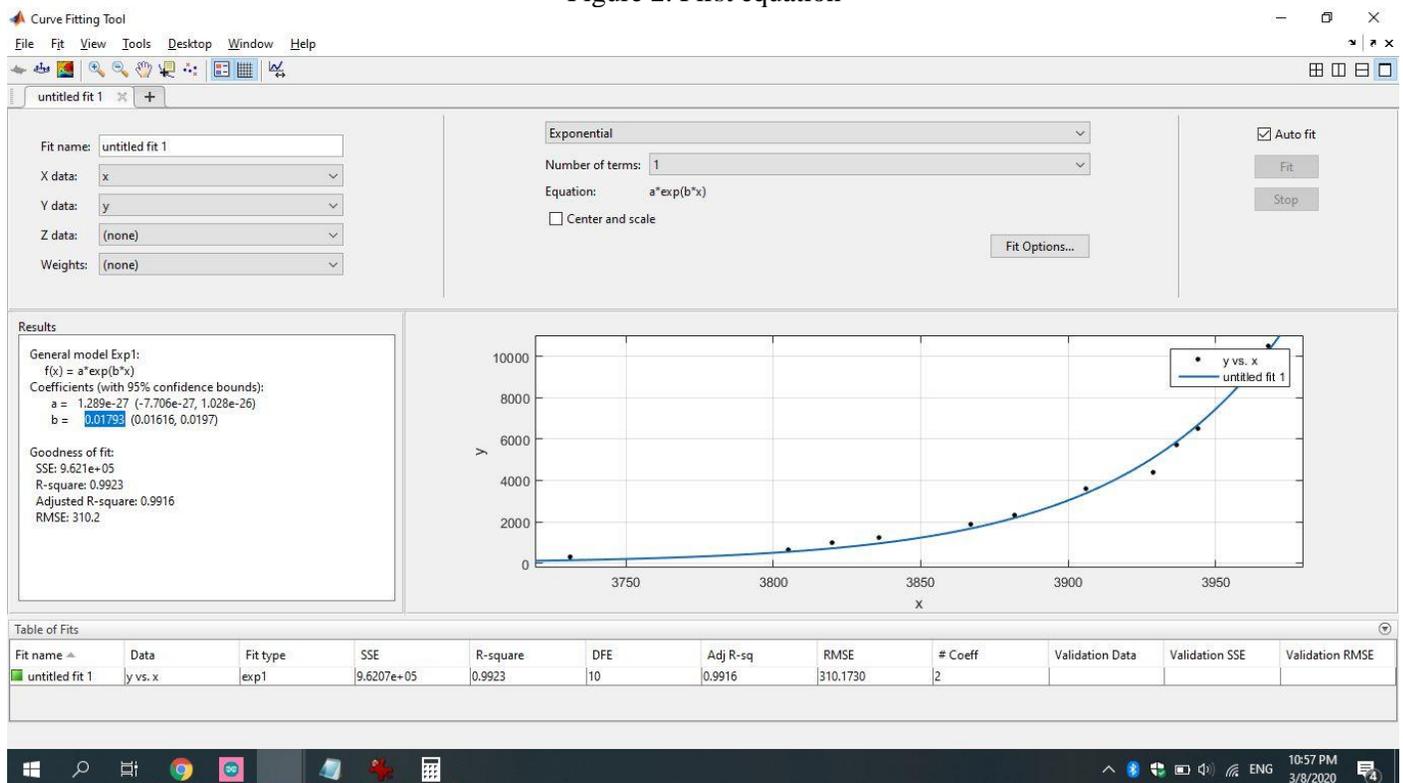


Figure 3. Second equation

Here is Arduino code (significant lines of code are explained in the comments) :

```
int pResistor = A0;
int value;
float result;
char inChar;

void setup() {
  pinMode(pResistor, INPUT);
  Serial.begin(9600);

  cli();//stop interrupts
  //set timer1 interrupt at 1Hz
  TCCR1A = 0; // set entire TCCR1A register to 0
  TCCR1B = 0; // same for TCCR1B
  TCNT1 = 0; // initialize counter value to 0
  // set compare match register for 1hz increments
  OCR1A = (16000000) / (1 * 1024) - 1;
  // turn on CTC mode
  TCCR1B |= (1 << WGM12);
  // Set CS10 and CS12 bits for 1024 prescaler
  TCCR1B |= (1 << CS12) | (1 << CS10);
  // enable timer compare interrupt
  TIMSK1 |= (1 << OCIE1A);
  sei();//allow interrupts
}
ISR(TIMER1_COMPA_vect) {
  value = analogRead(pResistor); // read the value of photoresistor
  if (value < 3700) {
    result = floor(0.1121 * exp(0.002123 * value)); //equation for values <3700
  }
  else {
    result = floor(1.289 * pow(10, -27) * exp(0.01793 * value)); // equation for values >3700
  }
  Serial.println(result); // printing results on Serial monitor
}
void loop() {
  if (inChar == 'a') {
    OCR1A = (16000000) / (5 * 1024) - 1; // change the sampling frequency
  }
  else if (inChar == 'b') {
    OCR1A = (16000000) / (1 * 1024) - 1;
  }
}
void serialEvent() {
  while (Serial.available()) {
    inChar = (char)Serial.read(); //get the entered character
  }
}
```

# Hardware realisation

As already mentioned in the first chapter, for hardware realisation Arduino WAVGAT UNO R3 board is utilized.

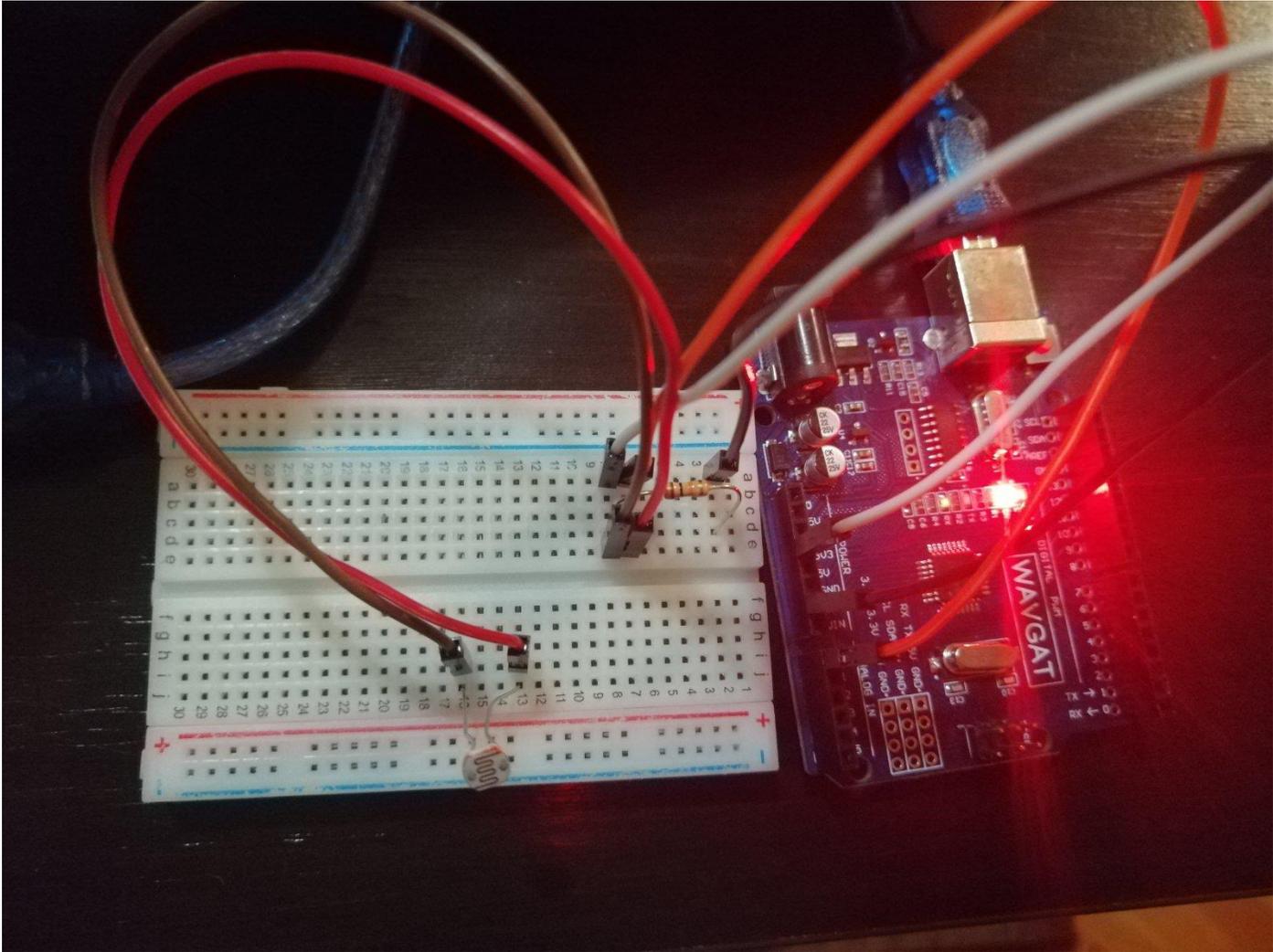


Figure 4. Realisation on Arduino board

## Verification on board

Verification was successfully done in a way described in previous paragraph.

## Literature

- [1] <https://www.instructables.com/id/Arduino-Timer-Interrupts/>
- [2] <https://play.google.com/store/apps/details?id=com.doggoapps.luxlight&hl=en>