MEMORANDUM

TO:	Dr. Christopher Peters
FROM:	Yelnur Abilakim
DATE:	21 Oct 2021
SUBJECT:	ECE-303 Lab 3: Serial Communication

Summary

In this experiment, the same circuit schematic is used as in the previous one. The duty cycle is being incremented from 1% to 100% in 1% increments. For each iteration, points are taken and plotted on the graph to visualize the system behavior.

Introduction

In this experiment, the LED and photocell circuits from previous lab are kept intact. Unlike the previous experiment, the PWM approach can be replaced with analogWrite(). The goal of this lab is to implement automatic measuring and data acquisition, and to establish communication from the Arduino to MATLAB through the serial port. Either MATLAB or Python can be used for writing code. MATLAB was used in this instance and, as per lab protocol, the code was written without using the Arduino toolkit. Note that if Python was used instead of MATLAB, neither of the Arduino-focused toolkit/library/packages would have been permitted by the lab protocol as well.

Methods

Experimental apparatus used in this lab:

- Breadboard.
- Arduino MEGA 2560 connected as follows:
  - To laptop via USB cable.
  - To outlet via power cable.
- One 1kΩ resistor.
- One 10kΩ resistor.
- One photocell.
- Four jump wires.

The LED used in this experiment is a red LED in series with a 1kΩ resistor (either one can be connected to ground). On a different circuit, the photocell will be connected to the 5V pin, and in series with a 10kΩ resistor (resistor to ground).
The circuit used for the lab is shown in Figure 1.

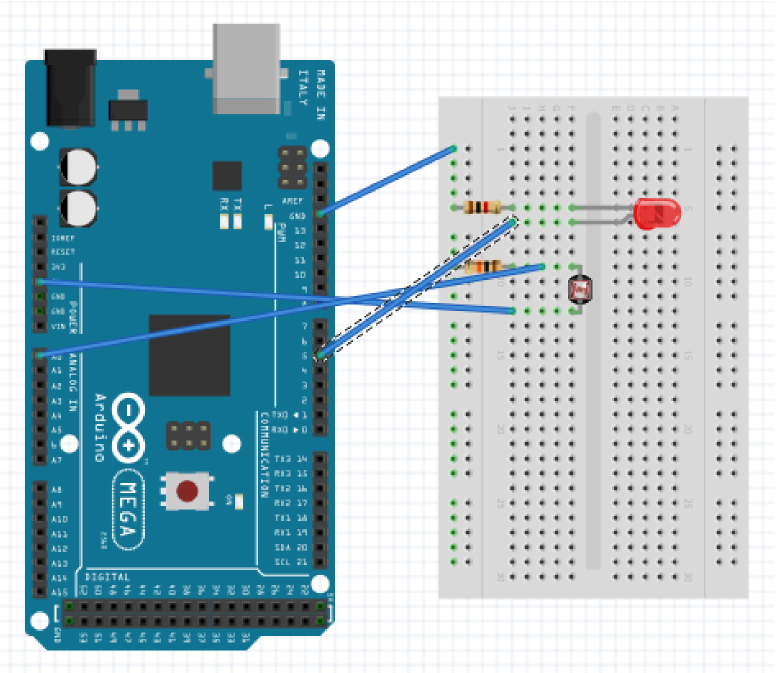


Figure 1. Circuit setup

Note that the LED should be placed near the photocell head, and photocell wiring needs to be bent such that the photocell head is in direct line with the LED. This minimizes energy losses, resulting in a more accurate measurement. In addition, the measurements should be performed in a dark environment so that external light sources do not interfere.

Results

The physical circuit used for the experiment, built according to Figure 1, is displayed in Figure 2.

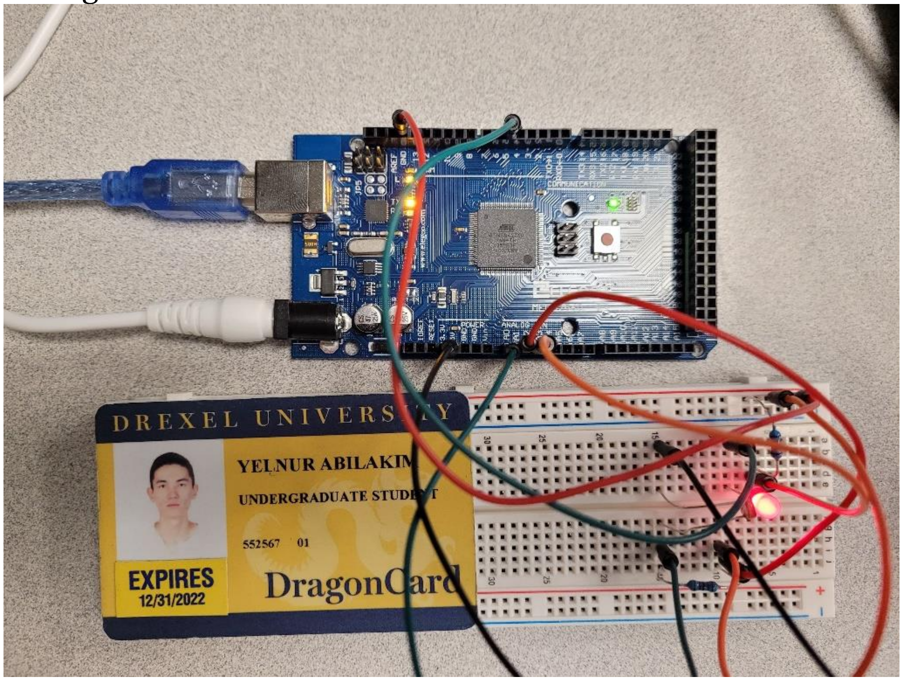
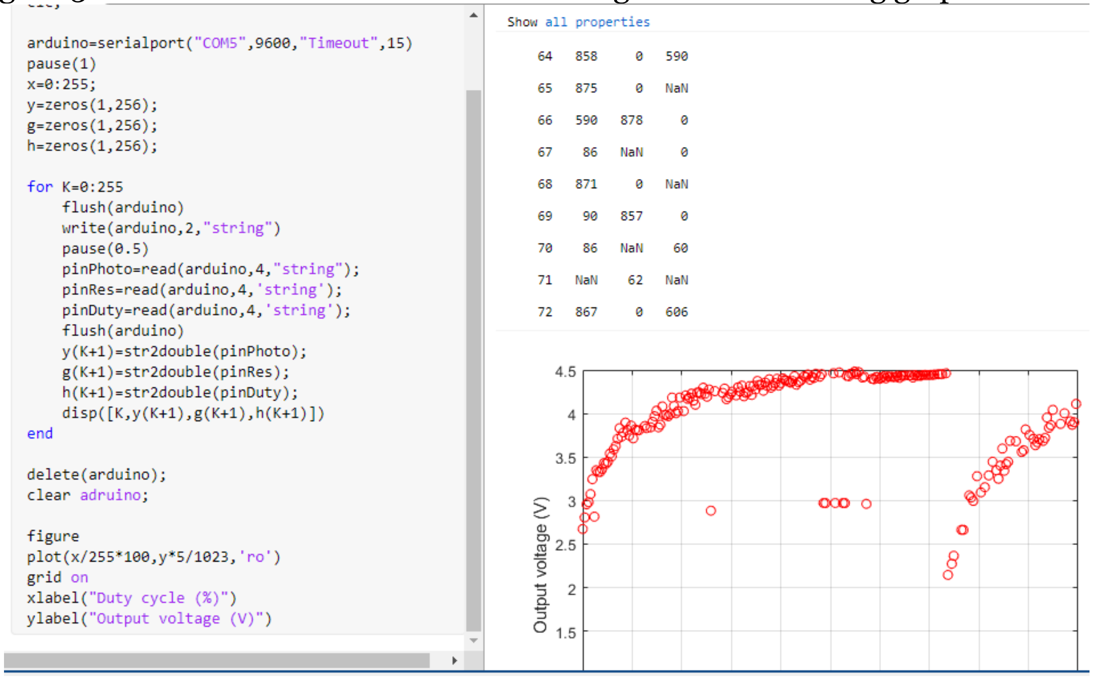


Figure 2. Physical circuit

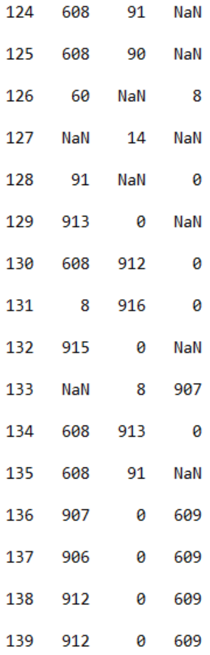
Due to a big number of outputs, only some parts of the outputs are displayed below. Figures 3-5 demonstrate Serial Monitor outputs through the MATLAB window at different ranges. In particular, Figure 3 shows part of the first half of the outputs, while Figures 4 and 5 indicate most of the second half of outputs. In addition, Figure 3 also includes the MATLAB code along with the resulting graph.



```
arduino=serialport("COM5",9600,"Timeout",15)
pause(1)
x=0:255;
y=zeros(1,256);
g=zeros(1,256);
h=zeros(1,256);

for K=0:255
    flush(arduino)
    write(arduino,2,"string")
    pause(0.5)
    pinPhoto=read(arduino,4,"string");
    pinRes=read(arduino,4,'string');
    pinDuty=read(arduino,4,'string');
    flush(arduino)
    y(K+1)=str2double(pinPhoto);
    g(K+1)=str2double(pinRes);
    h(K+1)=str2double(pinDuty);
    disp([K,y(K+1),g(K+1),h(K+1)])
end

delete(arduino);
clear adruino;

figure
plot(x/255*100,y*5/1023,'ro')
grid on
xlabel("Duty cycle (%)")
ylabel("Output voltage (V)")
```

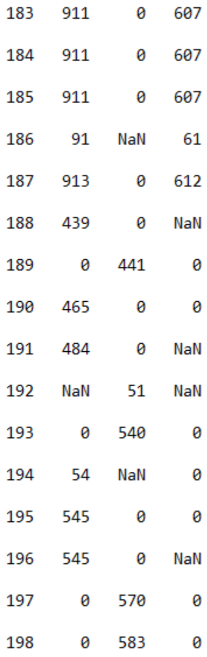
```
    64   858     0   590

    65   875     0   NaN

    66   590   878     0

    67    86   NaN     0

    68   871     0   NaN

    69    90   857     0

    70    86   NaN    60

    71   NaN    62   NaN

    72   867     0   606
```

Figure 3. Outputs (part 1), MATLAB code, and graph



```
   124   608    91   NaN

   125   608    90   NaN

   126    60   NaN     8

   127   NaN    14   NaN

   128    91   NaN     0

   129   913     0   NaN

   130   608   912     0

   131     8   916     0

   132   915     0   NaN

   133   NaN     8   907

   134   608   913     0

   135   608    91   NaN

   136   907     0   609

   137   906     0   609

   138   912     0   609

   139   912     0   609
```

Figure 4. Outputs (part 2)



```
183   911     0   607
184   911     0   607
185   911     0   607
186    91   NaN    61
187   913     0   612
188   439     0   NaN
189     0   441     0
190   465     0     0
191   484     0   NaN
192   NaN    51   NaN
193     0   540     0
194    54   NaN     0
195   545     0     0
196   545     0   NaN
197     0   570     0
198     0   583     0
```

Figure 5. Outputs (part 3)

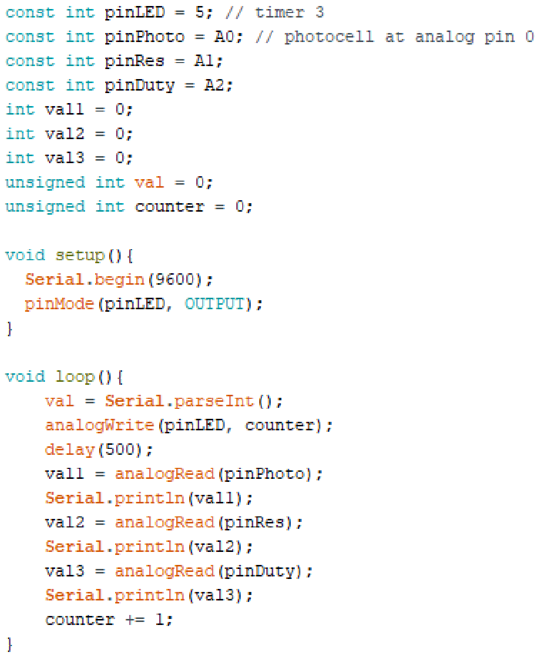
The Arduino code used for this experiment is appended at the end.

Conclusions/Recommendations

The experiment was successful. Automatic measuring and data acquisition were implemented, and communication from the Arduino to MATLAB through the serial port was established, thus achieving the goal of the lab. Taking advantage of analogWrite() function significantly reduces the complexity of the Arduino code, making the script shorter and easier to write. The Serial Monitor has displayed all required values as expected.

Appendices

Arduino code is shown in Figure 6.



```
const int pinLED = 5; // timer 3
const int pinPhoto = A0; // photocell at analog pin 0
const int pinRes = A1;
const int pinDuty = A2;
int val1 = 0;
int val2 = 0;
int val3 = 0;
unsigned int val = 0;
unsigned int counter = 0;

void setup(){
  Serial.begin(9600);
  pinMode(pinLED, OUTPUT);
}

void loop(){
    val = Serial.parseInt();
    analogWrite(pinLED, counter);
    delay(500);
    val1 = analogRead(pinPhoto);
    Serial.println(val1);
    val2 = analogRead(pinRes);
    Serial.println(val2);
    val3 = analogRead(pinDuty);
    Serial.println(val3);
    counter += 1;
}
```

Figure 6. Arduino code