THE ASSEMBLY OF A RESPONSIVE
ANALOG DATA READER USING ARDUINO

UNIVERSITY OF THE PHILIPPINES VISAYAS
COLLEGE OF ARTS AND SCIENCES
DIVISION OF PHYSICAL SCIENCES AND MATHEMATICS

CMSC 130 - LOGIC DESIGN AND DIGITAL COMPUTER CIRCUITS
2ND SEMESTER AY 2021-2022

ASSIGNMENT 5. ANALOG DATA IN ARDUINO

PREPARED BY:

DELA CRUZ , LLOYD WALLYS
LESCANO , RENMAR
ORAÑO , MYRTLLE GEM

MAY 23, 2022

Abstract

The objective of this laboratory project was to create an Arduino system that (a) read the analog input of the potentiometer which is read as voltage ranging from 0-5 volts, and (b) calculated the linearly dependent blinking interval of an LED. This dependency was modeled using a linear fuction obtained from the given parameters. The system was then built and programmed using an Arduino UNO and components from *Makerlab Electronics' Arduino Upgraded Learning Kit*. The output, printed in the Serial Monitor, were observed for the troubleshooting. Ultimately, no critical errors were identified and system, including both hardware and software, was successfully implemented.

Table of Contents

List of Figures

I. Design

A. Problem Statement:

The main objective of this project was to create a software for the LED (Light Emitting Diode) to blink when the potentiometer voltage ranges between 2-5 volts. The frequency of LED flashing was proportional to the potentiometer voltage. This assignment aimed to assess the student's understanding on (a) reading analog inputs and their conversion into digital signals, (b) the utilization of Arduino's digital write function to program a circuit, and (c) the management of the serial monitor to observe the circuit behavior.

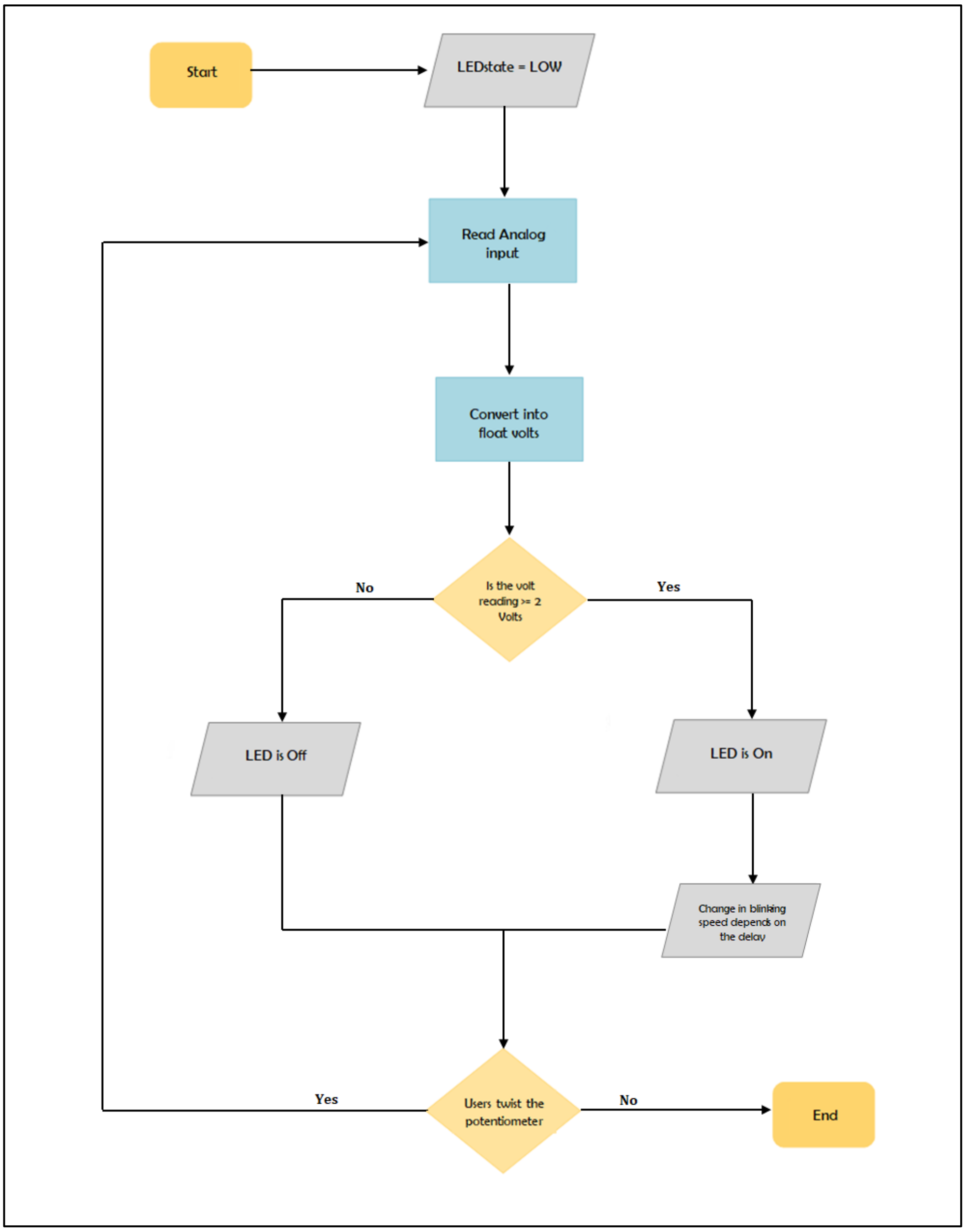


Fig. 1. The flowchart of the implementation of the Analog Data Reader

B. Overview:

For a systematic procedure, the following objectives were pursued:

- Learn the pin setup of a potentiometer
- Understand how input from the potentiometer was read
- Convert the sensor value into voltage
- Identify the required hardware and the circuit structure
- Develop the code to read and process the input from the potentiometer into the LED behavior
- Perform testing and debugging
- Record observations for the conclusion and recommendation

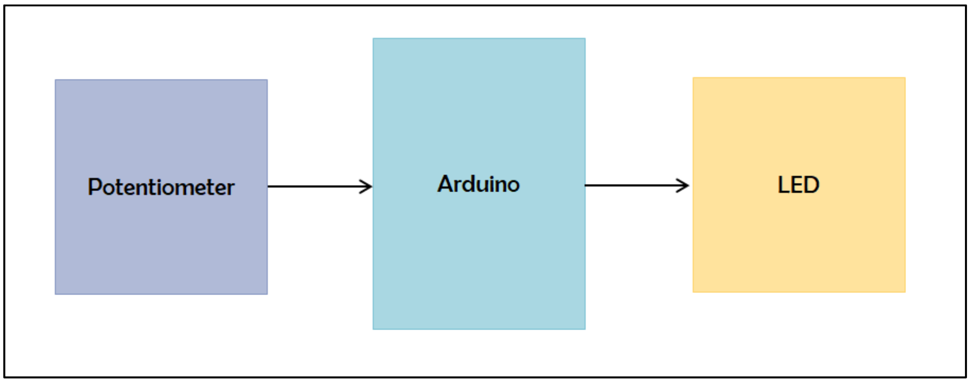


Fig. 2. The block diagram of the reader

C. Hardware:

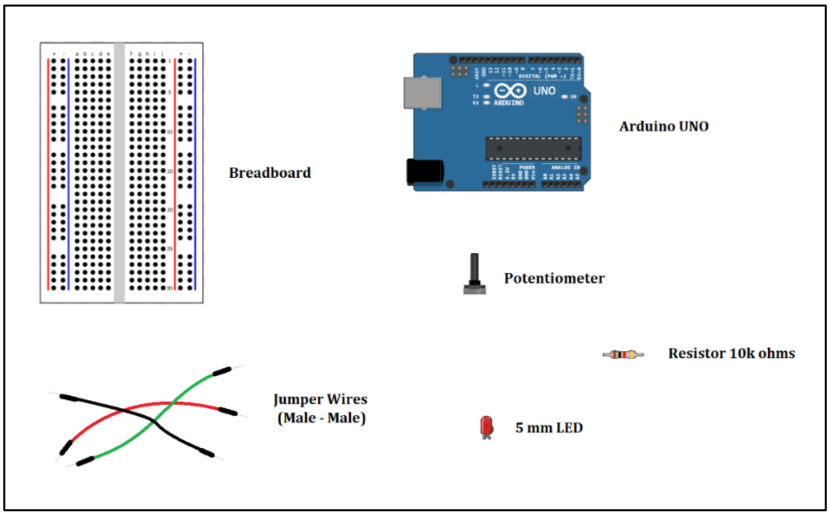


Fig. 3. The hardware required for the system

The following hardware components were used for the implementation:

- Potentiometer – produced changing resistance.
- LED – produced the program output.
- 10K Ω resistor – regulated the flow of electrical current and supplied voltage to the active device acting as a transistor.
- Arduino Uno – the platform to facilitate the interaction between software and hardware components
- Jumper Wires – connected the pins of each component
- Breadboard – facilitated the flow of electricity between the hardware components.

D. Circuit:

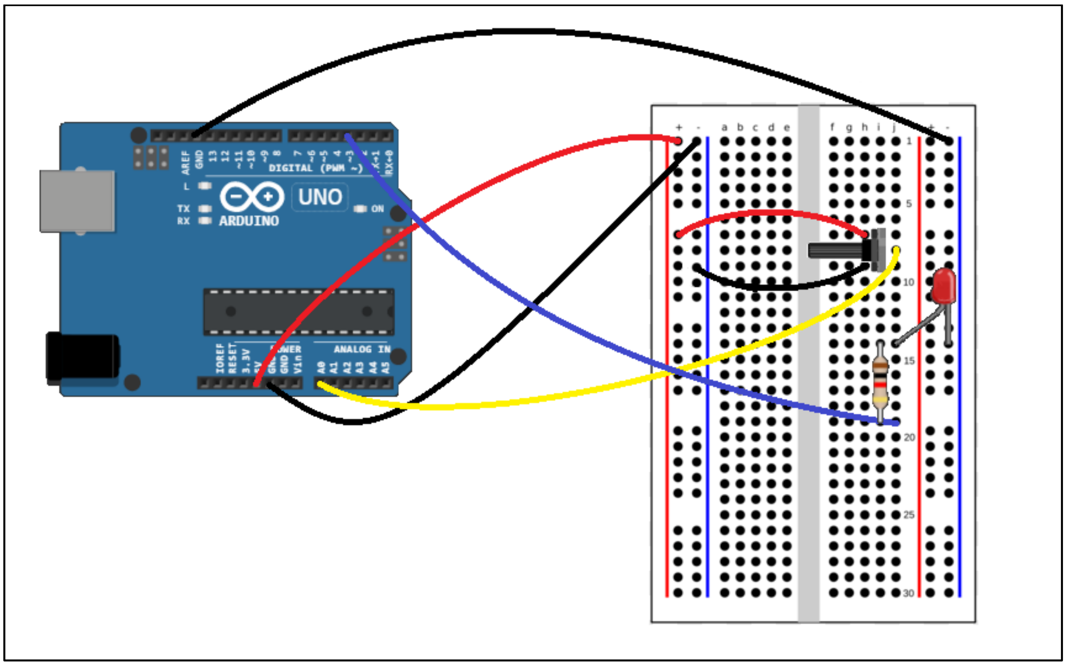


Fig. 4. The integration setup with an Arduino UNO

To build the circuit for the system, all three pins of the 10k ohm potentiometer were latched to the breadboard. The left pin is connected to the GND (ground) pin of the Arduino. Meanwhile, the right pin as the Vcc is connected to the 5V pin. Lastly, the signal terminal or the middle pin is connected to an analog pin A0

Upon turning its wiper, the potentiometer produces varying resistance. In order to measure the resistance value, voltage must be fed to the potentiometer from the 5V pin of the Arduino. An analog-to-digital converter (ADC) circuit on the Arduino board the reads the changing voltage and converts it into values between zero and 1023. When the wiper is turned counterclockwise, the input value closes in to zero. Conversely, the pin reads 5 volts when the wiper is entirely twisted clockwise (T.A Team, 2022).

After reading the voltage input, the behavior the LED is then determined. Given a basic LED circuit setup connected to ground and pin 3, digital signals are passed on writing either HIGH or LOW depending on the voltage reading. If the detected voltage is less than 2.00 volts, then the

LED will remain in LOW. Otherwise, it will blink at intervals linearly determined with the voltage input.

E. Schematic:

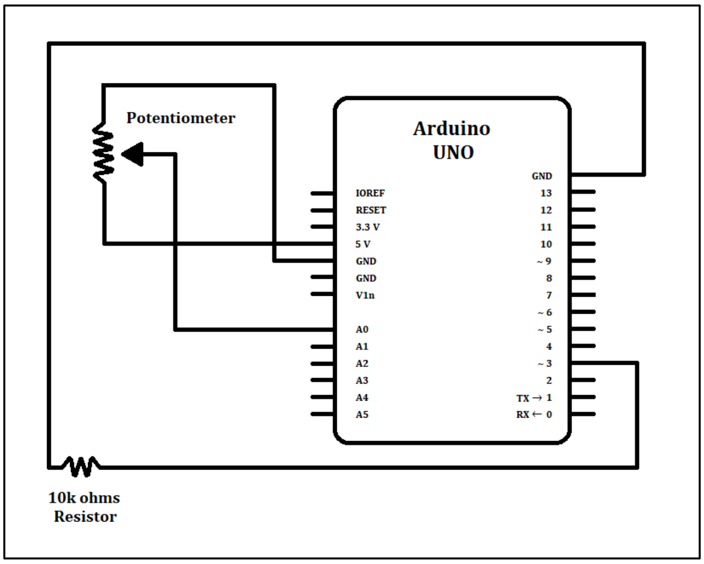


Fig. 5. The schematic for the integrated system

F. Preliminaries for the Program

Before the instructions for the circuit is developed, it is important to understand how the code variables can be modeled. Particularly, in this project, a function for the relationship between the output and input must first be obtained. The tables and figures below illustrate an approach on modeling the system.

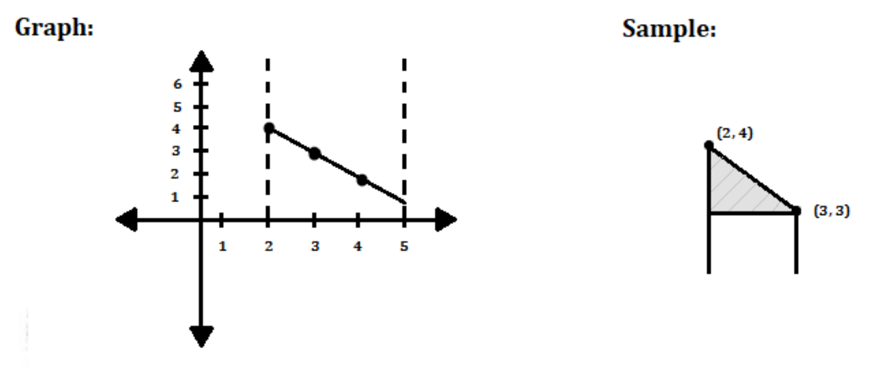
Given:

x = Potentiometer Voltage
delay = Interval in seconds

TABLE 1. THE FREQUENCY OF THE LED BLINKING was LINEARLY DEPENDENT ON THE POTENTIOMETER VOLTAGE

x (in volts)	delay (in sec)
2	4
3	3
4	2
5	1

This linear dependency from Table 1 can be represented in a 2-dimensional graph. When the inputs are plotted, there is an evident linear dependence of *y* with respect to *x*. Noticeably, as x increases, y decreases. Specifically, this illustrates a negative slope $m = -1$ (Fig. 3).



Slope:

$$m = \frac{rise}{run}$$

$$m = \frac{3-4}{3-2}$$

$$m = \frac{-1}{1} = -1$$

Fig. 6. The graph representation of the linear function

To obtain the function of this linear dependency, either the point-slope form or slope intercept form may be used. However, in using the slope-intercept form, an assumption must be made where the graph extends naturally to obtain its y-intercept. Purposely, the alternative formula is utilized for this experiment.

Given the formula $(y - y_0) = m(x - x_0)$, and sample points (2, 4) for (x_0, y_0), the following solution steps may be obtained:

$$(y - 4) = -1(x - 2)$$

$$(y - 4) = -x + 2$$

$$y = -x + 2 + 4$$

$$y = -x + 6$$

From this, the linear function is determined to be $y = -x + 6$. Note that y is expressed in milliseconds and is still not the value for the delay. Consequently, to obtain the expected values, the resulting function is multiplied by 1000; specifically, delay = 1000(y) in seconds. Finally, we can use this solution as a reference when constructing the code.

F. Code:

```
const int LED_PIN = 3;                                     // pin of LED
const int POT_PIN = A0;                                    // pin of potentiometer
const int b = 6;                                           // y-intercept
const int m = -1;                                          // slope
float d = 0;                                               // placeholder for delay value

void setup() {
  Serial.begin(9600);                                      // initialize serial at 9600 bits per second:
}

void loop() {
  int sensorValue = analogRead(A0);                        // read analog input from potentiometer

  // analog reading goes from 0 - 1023
  // convert toa voltage (0 - 5V)
  float voltage = sensorValue * (5.0 / 1023.0);

  // print voltage value in volts
  Serial.print("Input:\t");
  Serial.print(voltage);
  Serial.print(" volt/s");
  Serial.println();

  // if voltage meets threshold of 2 volts
  if (voltage >= 2){
    d = (m*voltage + b) * 1000;                            // solve for delay value (y = mx + b; y = delay, x = voltage input)
    Serial.print("Delay:\t");                              // print delay value in seconds
    Serial.print(d/1000);
    Serial.print(" second/s");
    Serial.println("\n");

    // blink LED given delay d
    digitalWrite(LED_PIN, HIGH);
    delay(d);
    digitalWrite(LED_PIN, LOW);
    delay(d);

  // if voltage does not meet threshold
  } else {
    digitalWrite(LED_PIN, LOW);                             // turn off pin
  }


}
```

Fig. 7. The setup and main loop of the program

II. Testing and Debugging

In the project's initial run, after building the system, the delay outputs were observed to be incorrect. Particularly, there was a mishandling of inputs where the results of the point-slope equation were directly applied to the delay value overlooking the parameter where the default time unit is in milliseconds. Accordingly, a factor of 1000 was multiplied to the respective values to generate the correct delay values in seconds.

```
00:05:03.027 ->
00:05:03.027 -> Input:  2.21 volt/s
00:05:03.074 -> Delay:  3.79 second/s
00:05:03.074 ->
00:05:03.074 -> Input:  2.21 volt/s
00:05:03.074 -> Delay:  3.79 second/s
00:05:03.120 ->
00:05:03.120 -> Input:  2.21 volt/s
00:05:03.120 -> Delay:  3.79 second/s
00:05:03.166 ->
00:05:03.166 -> Input:  2.21 volt/s
00:05:03.166 -> Delay:  3.79 second/s
00:05:03.213 ->
00:05:03.213 -> Input:  2.21 volt/s
00:05:03.213 -> Delay:  3.79 second/s
```

Fig. 8. The miscalculation for the delay values evident in the discrepancy between the output and timestamps

Subsequently, several tests were then performed to ensure the functionality of the circuit and to generate relevant findings. Specifically, the circuit was evaluated under the following situations:

a.) Turning the potentiometer wiper at varying frequencies and speed

To determine if the circuit can accurately and correctly detect the input—and accordingly produce the expected output—the wiper was shifted in irregular paces and intervals. While the outputs were correct, there was an observed delay in the detection of the voltage change especially when the wiper was turned in rapid succession. This, however, was expected as the program simultaneously attempts to finish the main loop first before recording new changes. For example, if the voltage was equal to 2 and the delay was 4 seconds, the main loop should take 8 seconds to finish (4 seconds HIGH, 4 seconds LOW). Therefore, if the wiper was turned after 6 seconds, there would be a delay of 2 seconds before the new input was detected.

```
22:08:23.236 -> Input:  2.82 volt/s
22:08:23.283 -> Delay:  3.18 second/s
22:08:23.283 ->
22:08:29.597 -> Input:  2.82 volt/s
22:08:29.643 -> Delay:  3.18 second/s
22:08:29.643 ->
22:08:35.985 -> Input:  5.00 volt/s
22:08:36.032 -> Delay:  1.00 second/s
22:08:36.032 ->
22:08:37.981 -> Input:  5.00 volt/s
22:08:38.028 -> Delay:  1.00 second/s
```

Fig. 9. The captured delay between the turning of the wiper and the detection of voltage change

b.) Observing the serial monitor without moving the circuit

To to verify if the circuit outputs consistent signals to the Arduino, tests were conducted where the potentiometer wiper was turned only once. It was observed that unless that voltage was maxed at 5 volts, the input would fluctuate by a margin between 0-0.03 volts. The software and connections were then checked to troubleshoot the issue. Unfortunately, these inconsistencies appear to be caused by hardware limitation. While these inconsistencies have minimal effect on the system, they were still noted for future references.

```
22:06:32.588 -> Input:  2.34 volt/s
22:06:32.588 -> Delay:  3.66 second/s
22:06:32.635 ->
22:06:39.910 -> Input:  2.34 volt/s
22:06:39.957 -> Delay:  3.66 second/s
22:06:39.957 ->
22:06:47.224 -> Input:  2.33 volt/s
22:06:47.271 -> Delay:  3.67 second/s
```

Fig. 10. The observed fluctuations in input despite the untouched circuit

c. Removing the resistor connected to the LED

The circuit used for this project, based on the sample described in *LED Blinking Control by Potentiometer* by techZeero, involves the use of a 10k ohm resistor connected to the LED. Due to the lack of proper understanding on circuits, this LED-to-resistor setup was assumed to be in rudimentary form. However, upon tinkering with the connections, it was accidentally discovered that the LED remains functional even without the resistor. Apparently, LED are expected to work with or without resistors (EE Power, 2020). Still, it is recommended to keep the ballast (limiting) resistor to avoid frying the LED from direct exposure to high voltages.

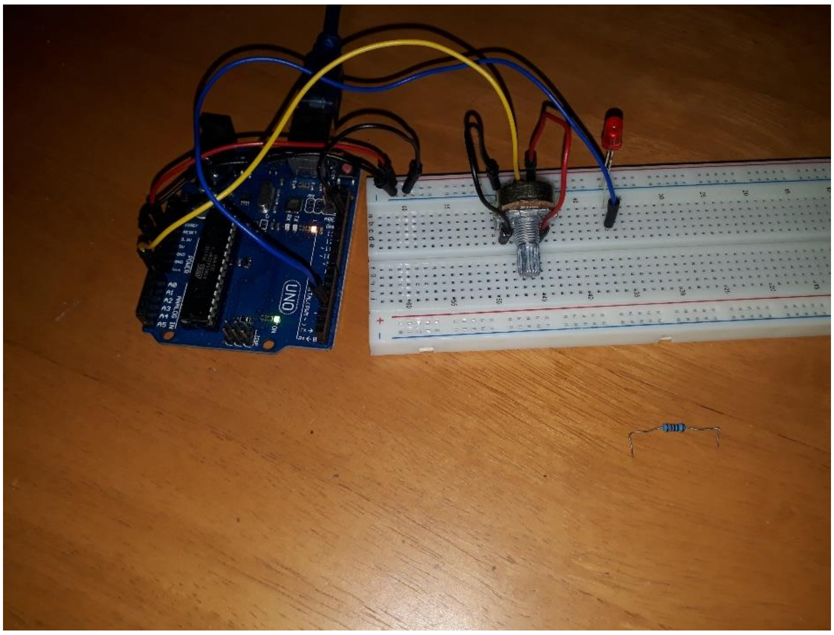


Fig. 11. An image of the LED blinking normally despite being disconnected from the resistor

Ultimately, all the insights gathered from the tests were recorded. Meanwhile, the persistent minor issues were distinguished as limitations in either software or hardware. Finally, it is established that no critical errors need to be addressed.

III. Conclusion

To summarize, the requirement for this project is to produce a circuit composed of an LED and a potentiometer integrated with an Arduino hardware. In addition, the LED must blink with a delay interval linearly dependent on the voltage generated through the potentiometer. Initially, the linear function was obtained by plotting the points and using algebra. Then, the circuit was modeled and built using basic components. Finally, the software was developed guided by the parameters of the circuit and the linear function for the delay interval. During testing, errors in calculations were observed but were immediately addressed. Moreover, inherent hardware and software limitations were discovered and taken into account when gathering the results.

Majority of the insights obtained sourced from the Arduino system including the flow of input and instructions as well as the behavior and limitations of the hardware. Consequently, it is recommended that a deeper understanding of the system is integral to isolate inconsistencies and adopt appropriate solutions.

Nevertheless, in conclusion, the implementation of the analog data reading system using Arduino was successful. It was able to properly detect the voltage input and program the LED to behave in the expected manner.

References

EETech Media, “Resistors for led circuits: Resistor applications: Resistor Guide,” *EEPower*, 2022. [Online]. Available: https://eepower.com/resistor-guide/resistor-applications/resistor-for-led/#. [Accessed: 23-May-2022].

T. A. Team, “Analog read serial: Arduino documentation,” *Arduino Documentation | Arduino Documentation*, 2022. [Online]. Available: https://docs.arduino.cc/built-in-examples/basics/AnalogReadSerial. [Accessed: 21-May-2022].

T. A. Team, “Read analog voltage,” *Arduino*, 2018. [Online]. Available: https://www.arduino.cc/en/Tutorial/BuiltInExamples/ReadAnalogVoltage. [Accessed: 21-May-2022].

T. Youngblood, “How to use Arduino's analog and digital input/output (I/O) - projects,” *All About Circuits*, 2015. [Online]. Available: https://www.allaboutcircuits.com/projects/using-the-arduinos-analog-io/#:~:text=as%20digital%20signals.-,The%20Arduino%20can%20input%20and%20output%20analog%20signals%20as%20well,%2Ddigital%20converter%20(ADC). [Accessed: 20-May-2022].

techZeero, “Led blinking control by potentiometer - circuit diagram, video tutorial,” *techZeero*, 31-May-2021. [Online]. Available: https://techzeero.com/arduino-tutorials/led-blinking-control-by-potentiometer/. [Accessed: 21-May-2022].