**Lawrence Technological University**

MRE 6183 – Mechatronic Systems II

Spring 2023



**Lab 2: Introduction to Computer Systems**

Ahmed Khan Mohammed Zia

Shanawaz Hussain Mohammed

*"I pledge that on all academic work that I submit, I will neither give nor receive unauthorized*

*aid, nor will I present another person's work as my own."*

**Lab Summary**

Lab 2 was an introduction to the Arduino UNO microcontroller and to program it through its official integrated development environment. We also experimentally observed the working of an NPN transistor (2n3904) as a switch by operating it through the Arduino UNO microcontroller.

**Prelab**

The code from Arduino tutorial for blinking the on-board LED included in Appendix A was referred for this Lab session. It was modified as per the Lab instructions:

1. LED on time = 0.5 seconds
2. LED off time = 1.5 seconds
3. Digital output pin for external LED = 12

**Arduino UNO**

The given Arduino UNO board has a 32-bit ARM Cortex M4 microcontroller (Renesas RA4M1) with 48MHz Clock speed. The board has 14 digital I/O pins of which 6 are PWM capable and 6 analog input pins. In terms of communication, it has an on-board USB circuit, a UART channel, a SPI channel, an I2C channel and a CAN bus. It also comes with an upgraded 14-bit DAC and upgraded memory features (256KB Flash, 32KB SRAM and 8KB EEPROM) [1]. Compared to its predecessor, UNO R3, the UNO R4 Minima shown in Figure 1 is a very advanced version.

A close-up of a circuit board

Description automatically generated

Figure 1: Arduino UNO R4 Minima

**Arduino IDE 2**

As recommended by Arduino, the Arduino IDE 2.3.2 [2] (shown in Figure 2) was used to program the UNO R4 Minima board. The Arduino language is a syntactical variant of C++; it has functions and variables common to multiple Arduino boards (UNO, Mega, DUE, nano, etc) and its own coding structure (setup and loop functions). It is simple and easy to learn.

A screenshot of a computer

Description automatically generated

Figure 2: Arduino IDE 2

**Transistors**

The diagram in Figure 3 shows an NPN transistor in normal, or forward bias, operation. When voltage is applied to the base to the transistor, current is free to flow from the collector to emitter. If an LED is connected as shown in Figure 3, the LED will light up when the digital output of Pin 2 goes HIGH (5 V). The 1KΩ and 150 Ω resistors are used to limit the current to the transistor base and LED, respectively. The Arduino UNO R4 digital output can supply 5V and the current limit is 8 mA [3]. Using Ohm's Law, 𝑉 = 𝐼𝑅, the 1KΩ resistor limits the actual current to 5 mA, which is below the board’s DIO current limit.

**Blinking an LED**

**3a.** The modified Arduino code for LED blinking in Appendix A was compiled and uploaded to Arduino UNO R4.

**3b.** Build the LED-transistor Circuit:

A circuit board with wires connected to it

Description automatically generated

Figure 3: Built LED Circuit with Arduino UNO

**3b.** Schematic:

A diagram of a circuit board

Description automatically generated

Figure 4: LED Circuit Schematic

**3c.** After compiling and uploading the code in Appendix A to Arduino UNO connected to the circuit, the LED started switching on and off. We connected Ch2 probe at D12 and checked the signal output as shown in Figure 5.

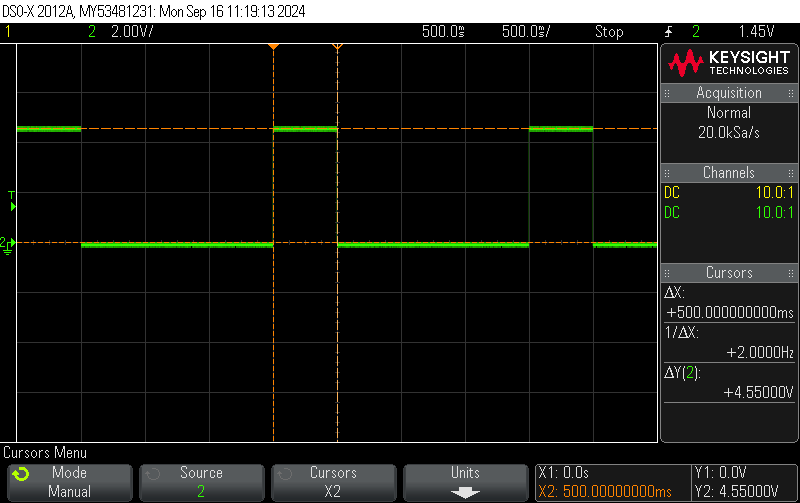


Figure 5: Output signal.

The LED turns on for 500 ms and turns off for 1500 ms. The total period is 2000 ms, Thus the frequency of LED is:

|  |  |  |
| --- | --- | --- |
|  |  | …(1) |

**3d.** We were able to increase the blinking rate by decreasing the on and off periods. At 20 ms of total period (5ms on and 15 ms off, or 25% duty cycle) we were able to make the LED appear as though it were constantly on. The rate for this period would be:

|  |  |  |
| --- | --- | --- |
|  |  | …(2) |

**Using a Switch with Pull-Up or Pull-Down Resistors**

A diagram of a circuit

Description automatically generated with medium confidence

Figure 6: Switch with (a) pull-up resistor or (b) pull-down resistor.

**4ai.** No, the voltage in the bare wire may be close to zero but cannot be absolutely zero. Any conductor in an electro-magnetic field will develop an electric potential perpendicular to the field, and this is how antennas essentially work. The bare wire is not isolated from such fields, which is why it will always have stray voltage.

**4aii.** No, since the bare wire is not connected to a power source, it cannot be at 5V potential.

**4aiii.** There is no way to predict the voltage within the bare wire, unless it is measured.

**4b.** In the case of switch arrangement shown in Figure 5 (a), when the switch is open, the voltage at DI will be raised to VCC (presumably 5V), and when the switch is closed the voltage at DI will fall to GND (0V).

**4c.** Without the pullup resistor and the opened switch, the DI pin will be at floating voltage, which is induced from surrounding environmental conditions. If we try to read the voltage programmatically, we will get random (garbage) values.

**4d.** In the case of switch arrangement shown in Figure 5 (b), when the switch is open, the voltage at DI will be at the level of GND (0V), and when the switch is closed the voltage at DI will rise to VCC (presumably 5V).

**4e.** Without the pulldown resistor and the opened switch, the behavior at DI pin will be the same as in the case of 4c above, DI will be at floating voltage and generate garbage values.

**Working with a Transistor**

The circuit in Figure 3 was used for this section of the lab. The code from Appendix A was modified to keep the LED on for 1000 ms and off for 500 ms (66.66% Duty Cycle).

**5a.** Transistor delay and rise times were measured by placing the probes as shown in Figure 4.

A diagram of a circuit board

Description automatically generated

Figure 7: Probe locations on the LED-transistor circuit

The oscilloscope’s channel 1 was connected to the base and channel 2 was connected to the collector. Figure 7 and Figure 8 show the recorded signals for delay and rise times as result of the signal generated on digital pin 12 of the Arduino UNO board.

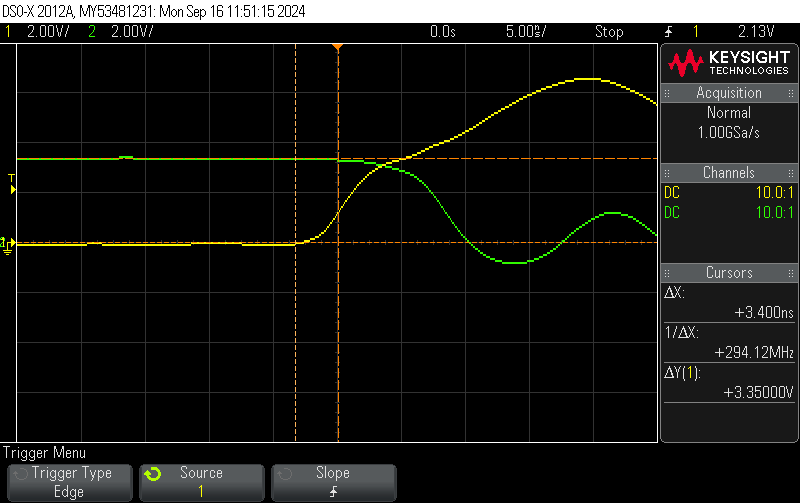


Figure 8: Transistor delay

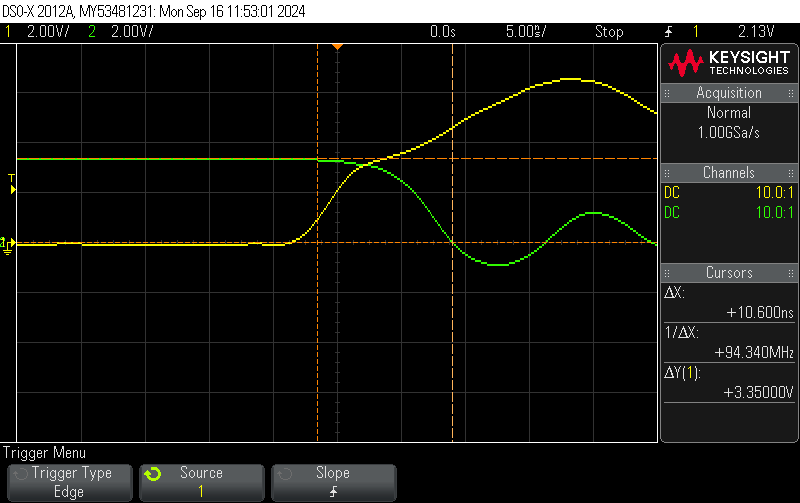


Figure 9: Transistor rise time

From Figure 8, transistor delay is 3.4 ns and from Figure 9, transistor rise time is 10.6 ns.

Transistor’s turn on time is given by:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | …(3) |

**5b.** Transistor storage and fall times were measured using the setup given in Figure 7.

A screen shot of a graph

Description automatically generated

Figure 10: Transistor storage time

A screen shot of a graph

Description automatically generated

Figure 11: Transistor fall time

From Figure 10, transistor delay is 150.1 ns, and from Figure 11, transistor rise time is 27 ns.

Transistor’s turn off time is given by:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | …(4) |

**5c.** The switching characteristics of 2N3904 transistor as given in the provided datasheet [3] from Fairchild Semiconductors are enlisted in Table 1

Table 1: 2N3904 Switching Characteristics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Test condition** | **Min** | **Max** | **Units** |
| td | Delay time | VCC = 3.0V, VBE = 0.5V  IC = 10mA, IB1 = 1.0mA |  | 35 | ns |
| tr | Rise time |  | 35 | ns |
| ts | Storage time | VCC = 3.0V, IC = 10mA,  IB1 = IB2 = 1.0mA |  | 200 | ns |
| tf | Fall time |  | 50 | ns |

Table 2: Measured switching characteristics of 2N3904

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Value** | **Units** |
| td | Delay time | 3.5 | ns |
| tr | Rise time | \_\_ | ns |
| ts | Storage time | 150.1 | ns |
| tf | Fall time | 27 | ns |

Comparing the measured values listed in Table 2 with the maximum values from the datasheet. All the values are within their limits.

**5d.** Current applied to transistor’s base at logic High (5V) level considering the series resistor:

Here, the 1K

**5e.** The current limiting resistor can be used to prevent damage to transistor (from excess current at transistor base) as well as the microcontroller (from excess current draw). The tested current at base (IB) at base-emitter saturation voltage as mentioned in the transistor’s datasheet [3] is 5mA, and the permissible current draw from DIO pin on UNO R4 [1] is 8mA. A resistor of 1KΩ at logic HIGH (5v) will limit the current to 5mA, keeping the whole operation under safe limits. In absence of this resistor the current draw when the transistor is in saturation region (switched on) would go outside the designed current specifications and could harm the microcontroller and the transistor. For example, in the case of UNO R3, the current through DIO pin can be up to 20mA [1], and not limiting this sort of current will likely kill the transistor.

**5f**. Measured voltage-drop across the LED when LED is on is \_\_\_\_ v.

**5g.** Since the 150-ohm resistor is in series with the LED, the current flowing through the LED is same as the current flowing through the resistor (measured to be 150.3Ω). So:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**BONUS: Variable LED Brightness**

In the circuit represented by Figure 4, a 10K potentiometer is connected in series as shown in the Figure 5. When the potentiometer value is changed the intensity of the LED changes proportionally. For several resistance values, voltage drop across LED and the current through LED is noted in the Table 3.

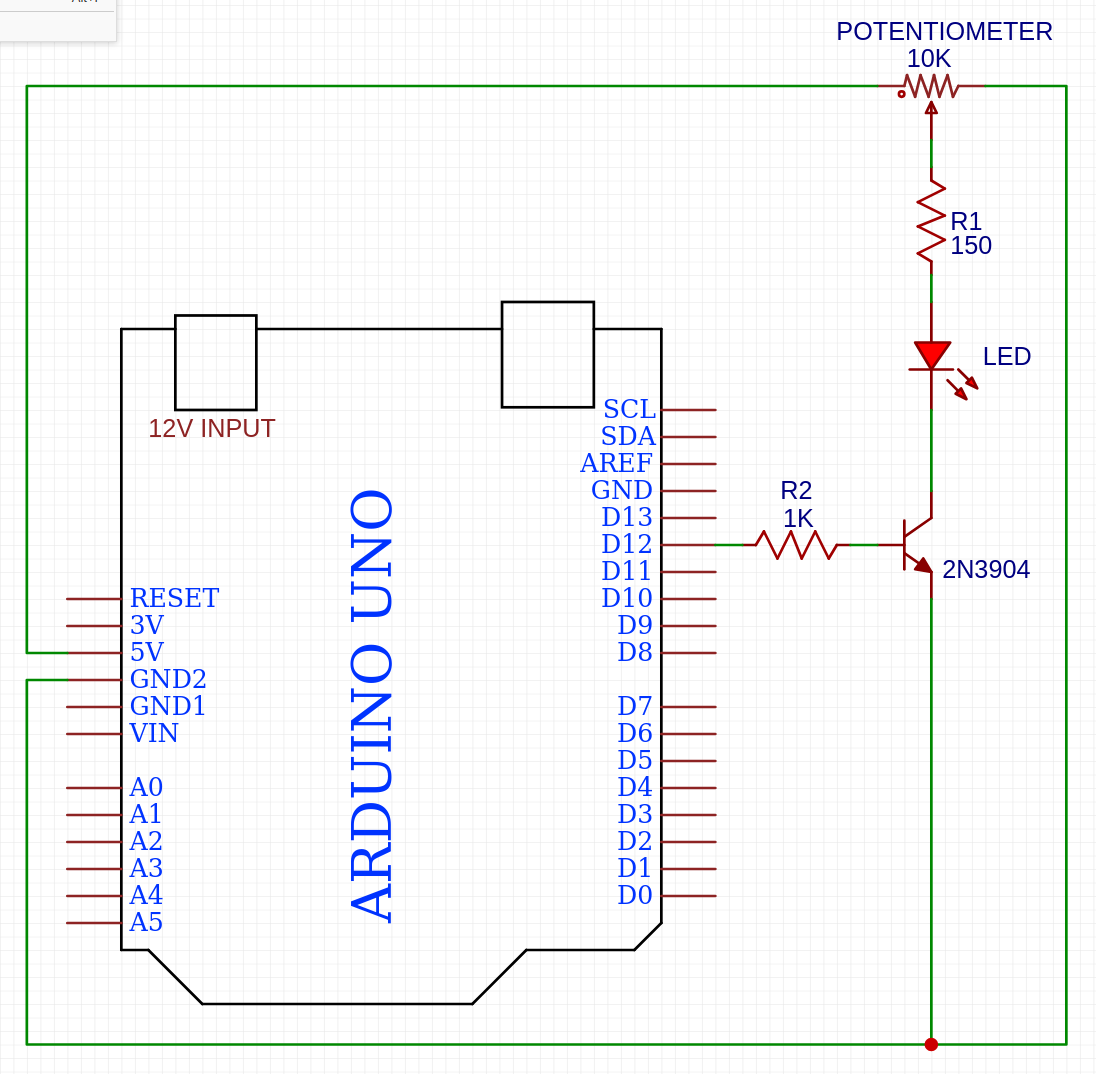


Figure 12: Addition of Potentiometer to existing LED circuit

Table 3: Corelating Potentiometer values with LED voltage drops and currents

|  |  |  |
| --- | --- | --- |
| **Pot Value (KΩ)** | **Vdrop (Volts)** | **ILED (mA)** |
| 1.04 | 2.76 | 2.65 |
| 2.02 | 2.86 | 1.41 |
| 4.03 | 2.92 | 0.72 |
| 6.58 | 2.96 | 0.45 |

It is observed that the LED intensity is lowered proprortionally with the resistance values of the potentiometer on the circuit. This behavior can also be utlized to change the blinking rate of the led. The analogue value from the potentiometer output can be used to linearly scale the opertaing period of LED in the arduino code, thereby controlling its blinking frequecy.

**Citations**

|  |  |
| --- | --- |
| [1] | Arduino LLC, "Arduino UNO R4 Minima data sheet," Arduino Docs, [Online]. Available: https://docs.arduino.cc/resources/datasheets/ABX00087-datasheet.pdf. |
| [2] | Arduino LLC, "Software | Arduino," Arduino, [Online]. Available: https://www.arduino.cc/en/software. |
| [3] | Fairchild semiconductor, "Lab Components: 1359\_1734-Mechatronic Systems," Lawrence Technological University, [Online]. Available: https://lawrencetech.instructure.com/courses/17830/files/5802728/download?wrap=1. |

**Appendix A: Code to Blink LED**

// setup global macros as per lab2 instructions:

#define LED\_PIN 12

#define ON\_TIME 500 // in milliseconds

#define OFF\_TIME 1500 // in milliseconds

// the setup function runs once when the board is reset or powered

void setup() {

// initialize digital pin LED\_PIN as an output.

pinMode(LED\_PIN, OUTPUT);

}

// the loop function runs repeatedly forever

void loop() {

digitalWrite(LED\_PIN, HIGH); // turn the LED on (HIGH is the voltage level)

delay(ON\_TIME); // wait for ON\_TIME in ms

digitalWrite(LED\_PIN, LOW); // turn the LED off by making the voltage LOW

delay(OFF\_TIME); // wait for OFF\_TIME in ms

}