MECHATRONICS

Lab #3

11/12/20

Brent Fujii | Phong Nguyen

ENGR 10

Dr. AH Tabrizi

TABLE OF CONTENTS

Introduction

In our recent arduino uno experiment, we learned how to connect the circuits onto the redboard by using jumper wires attached to the prototyping board. On the prototyping board we allow using our resistor, led lights, diode, wires, temperature sensor to allow the functions running through the board. Arduino Uno powered through using a usb cord which connects the redboard to a desktop using open source electronics prototyping using software and hardware components. It allows us to read the inputs while turning the components to outputs, by connecting the wires properly we can send messages to the computer to instruct and control the Arduino by programming C++ language.

Equipment

- 1 Red Arduino Board as shown in *figure 1.* to the right
- 1 Magic Chassis Buggy
- 1 LCD screen, refer to *figure 1.*

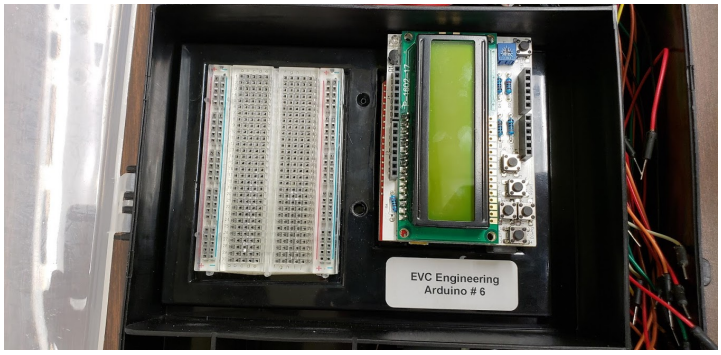

Figure 1. Arduino redboard with the LCD screen already attached on the right.

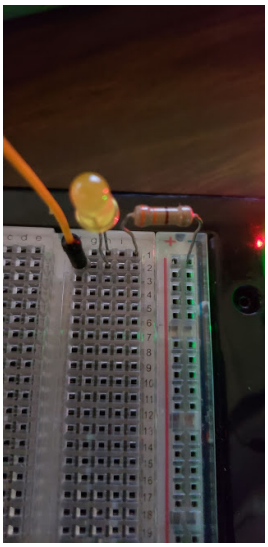

Figure 2. A circuit setup to light up the yellow LED.

- 2 diodes
- 2 transistors
- 3 resistors (one shown to the left in *figure 2.)*
- 1 red LED
- 1 yellow LED shown in *figure 2.* to the left.
- Temperature sensor
- 17 wires as two are shown in *figure 3 below*

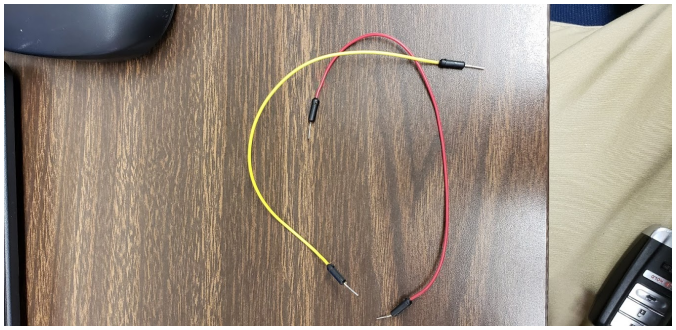

Figure 3. Two wires before being used in a circuit on the board.

Process

Blinking Led Light and Red Light

1. Connecting the red jumper wires from 5 volts to the positive prototype board, black jumper wire to ground and negative.

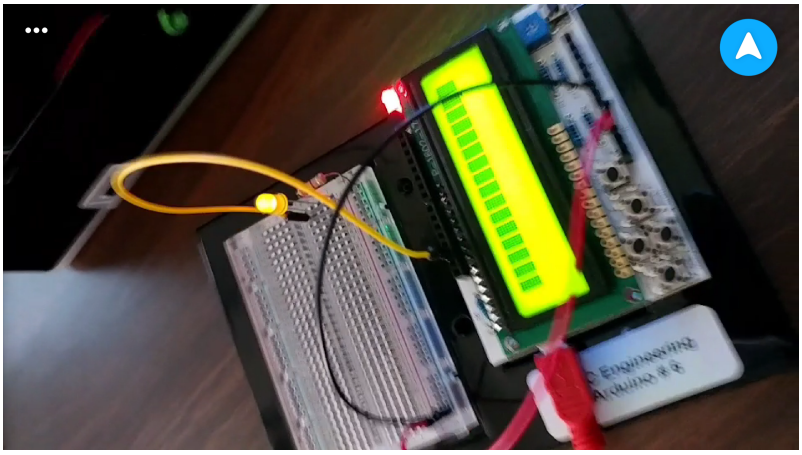

Figure 4. Circuit for LED lights

2. Use a resistor connected to the negative prototype board to the column and connect jumper wires to the redboard to the positive resistor shown in *figure 4* above.
3. Program the code to turn on led light by using high or low.
4. Red led lights share the same resistor and connect jumper wire to positive.

Temperature Sensor and LCD Display

1. Connect the temperature sensor from the front side, first use ground, 2nd signal, 3rd + voltage.
2. Use jumper wires to connect to negative to ground, positive to voltage, signal to any analog in from A1 to A5. (circuit shown in *figure 5* below)
3. Program the code to turn the voltage on and temperature on the LCD.
4. Program the LCD greetings using 2 columns and 16 rows.

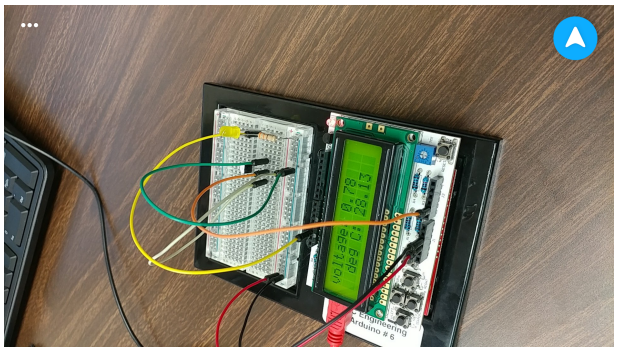

Figure 5. Circuit for temperature reading

Running Two Motors

1. Connect jumper wire negative to emitter, transistor base to resistor, and jumper wires to base, resistor and redboard.
2. Connect the motor to diode from negative motor, positive running same row of the diode to the positive jumper wires prototype board.
3. Repeat the steps to run the 2nd motors. (Full setup for both motors shown in *figure 6* below)

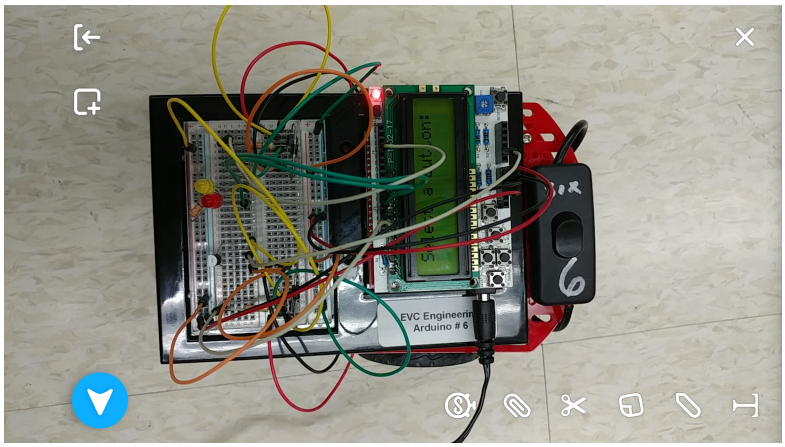

Figure 6. Circuits for two motors

Code

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);          // the lcd screen is using these pins

const int temperaturePin = 1;                 //labels the pin numbers throughout the code
const int yled = 10;
const int rled = 11;
const int r = 13;
const int l = 3;

void setup() {
 lcd.begin(16, 2);                             // The lcd have two columns and 16 rows
 lcd.clear();                                  // Use clear to clear all the junk on the screen
 lcd.setCursor(16,0);                          // We start off the cursor from first column
 lcd.print("Good Morning Dr.Tabrizi");        // We type using the first column row to greet on lcd
 lcd.setCursor(25,1);                          // 2nd column 25 row on lcd
 lcd.print("Give us an A");                    // We type using the first column row to greet on lcd
 lcd.setCursor(0,0);                           //sets the cursor position

  for (char i=0; i<39; i++)                    //loop to scroll the LCD screen to the left
  {
   lcd.scrollDisplayLeft();
   delay(400);
  }
  lcd.clear();                                 //Use clear to clear all the junk on the screen
 pinMode(l, OUTPUT);
 pinMode(r, OUTPUT);
```

```
  Serial.begin(9600);
}

void loop() {
  int x;
  x = analogRead (0);

  lcd.setCursor(0,0);
  lcd.print("Select a button:");  // lcd screen will show select a button
  lcd.setCursor(6,1);              //  show the select a button words

if (x < 60) {
  lcd.print ("Right ");            // set by pressing up button the function will run accordly
    analogWrite(r, 190);          // turn on right motor on
    delay(1600);                  // delay time in milliseconds(1.6 seconds)
    digitalWrite(r, LOW);         // turn off the right motor
  }

else if (x < 200) {
  lcd.print ("Up    ");           // set by pressing up button the function will run accordly
    analogWrite(l, 190);          // turn on the left motor to speed 190
    analogWrite(r, 140);          // turn on the right motor to speed 140
    delay(1200);                  // delay time milliseconds(1.2 seconds)
    digitalWrite(r, LOW);         // turn off right motor
    digitalWrite(l, LOW);         // turn off left motor
  }

else if (x < 400){
  lcd.print ("Down  ");           // set by pressing down button the function will run accordly
    pinMode(yled, OUTPUT);        // yellow led light
```

```
  digitalWrite(yled, HIGH);   // turn on yellow led on high
  delay(300);                 // delay time in milliseconds(0.3 seconds)
  digitalWrite(yled, LOW);    // yellow led turn off
  delay(300);                 // delay time in milliseconds(0.3 seconds)
 }

else if (x < 600){
 lcd.print ("Left  ");        // set by pressing left button the function will run accordly
analogWrite(l, 190);          // turn on the left motor to speed 190
  delay(1600);                // delay time in milliseconds (1.6 seconds)
  digitalWrite(l, LOW);       // turn off left motor
 }

else if (x < 800){
 lcd.print ("Select");                    // set by pressing up button the function will run accordly
  float voltage, degreesC, degreesF;
  voltage = getVoltage(temperaturePin);   //Equations to turn voltage into temperatures
  degreesC = (voltage - 0.5) * 100.0;
  degreesF = degreesC * (9.0/5.0) + 32.0;

  lcd.clear();
  lcd.setCursor(2,0);                     //Displays results on LCD screen
  lcd.print("voltage: ");                 // lcd print words (voltage)
  lcd.print(voltage);
  lcd.setCursor(2,1);                     //  We type using the first column row to greet on lcd
  lcd.print("  deg C: ");
  lcd.print(degreesC);
  delay(3000);                            // delay time in milliseconds (3 seconds)
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("  deg F: ");
```

```
   lcd.println(degreesF);

  if(degreesC >= 28){              //Loop to light red LED when temperature gets over 28 degrees C
    pinMode(rled, OUTPUT);
    digitalWrite(rled, HIGH);  // red led light turns on
  }else
  {
    digitalWrite(rled, LOW);  // left led light turn off
  }
  delay(3000);                     // delay time in milliseconds (3 seconds)
 }
}
float getVoltage(int pin)          // For Temp
{
  return (analogRead(pin) * 0.004882814);
}
```

Conclusion

In this Arduino Uno project we operate using many components to run from the redboard, prototype board to the computer and program the software to successfully execute our requirements. We were able to program the car to turn left counterclockwise by pressing the left key button and right key button to turn the car right clockwise.Select button to greet on the LCD screen with red light, giving us the voltage and temperature when it reaches a certain degree. Down button for the yellow led light, up button making the car travel forward at least one and half feet. During the experiment we performed, we noticed both motors run at a different speed as we use the left analog write 190 and make the right analog write 140 to run properly straight line. While performing a right and left turn we notice when analogwrite was set below 50, the motor wouldn't start spinning because there wasn't enough torque. We also notice the temperature won't work properly when plugging the jumper wires into the pins of 8, 9, 4, 5, 6, and 7 due to the LCD screen manufacturing needing those pins.