MEMORANDUM

TO: Dr. Christopher Peters
FROM: Yelnur Abilakim
DATE: 29 Nov 2021
SUBJECT: ECE-303-063: Final

Summary
This final project involved us showing the fact that Arduino can interface with any device that uses various currents to control devices with a power supply. We design and implement a code and circuit that will integrate various components into one device. The following components are to be integrated in this circuit: a motor, an ultrasonic sensor, LEDs, an LCD display, and RFID scanners. This project combines all we have learned individually throughout the course and makes sure that we can implement them all together using Arduino code and the circuit.

Introduction
This project had us simulate a motor test bed that has an ultrasonic sensor, 2 white LEDs, a motor, and an L293D IC. The IC would take a PWM signal from the Arduino and allow current to go through from the 5V source to the motor and the ground with current proportional to the supplied signal. In addition, new components such as an IR receiver, a water sensor and a DHT11 will also be added to the circuit. All these sensors are relayed back into an LCD display which will output the readings from these various components. Along with the LCD displaying the information from the sensors, a Graphic Interface Unit (GUI) that is implemented independently will also be displaying the readings from the sensors. Also, there is an IR remote that will be used to light the two LEDs on the breadboard as well as accelerate and decelerate the motor.

Methods
The idea is to echo the ultrasonic sensor, read the time to get the pulse back and use math to get the distance. The distance will be recorded and prompted for display on the GUI. The LEDs are connected to the IC L293D. They are also connected to ground a resistor. The GUI displays a corresponding color to the distance of the motor. When an object forward distance is less than 7cm there will be a red display, half full power for between 7 and 20 cm there will be a yellow display, and finally full power when the distance is more than 20cm will display a green light on the GUI. The RFID scanner will only accept and make a sound of acceptance from the programmed RFID chip and will decline and make a sound of declination for any other RFID device that is scanned to it, whether it be an RFID fob or a keycard. The Water and Temperature sensor are also connected into the Arduino. The readings are displayed in various parts of the project. The LCD used in this project is used to display the Water and Temperature sensor as well as the motor speed. The GUI being used will also display the various components mentioned before. The IR receiver is also connected to the Arduino and is programmed to take input signals from a remote. The remote allows the user to turn on “headlights” which are modelled by two LEDs at two different frequencies based on how much voltage is going into the LEDs. The remote is

also able to increase the motor speed in increments to make the “fan” [DC Motor] accelerate, as well as decelerate.

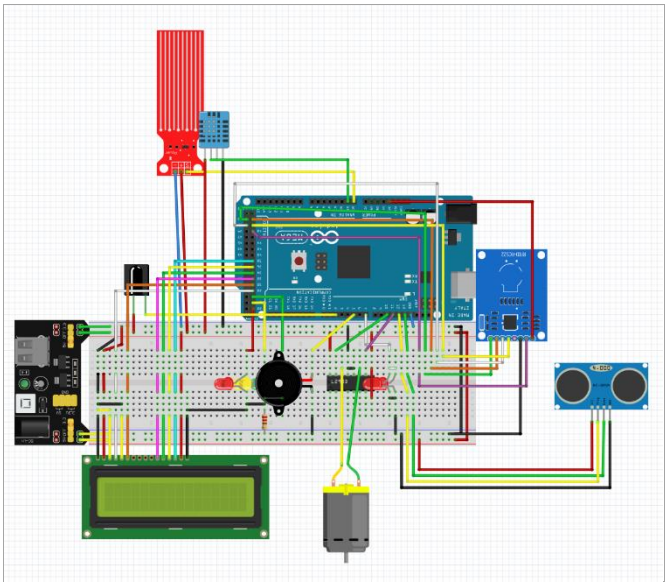

Figure 1. Circuit Schematic

Results

While few of the components were used individually in the past labs, the integration of various components together was a challenge. Ensuring each component was fittingly integrated on the Arduino and breadboard took a lot of time. Using Arduino’s various preexisting libraries helped keep the code concise and efficient. The GUI was written on Python. Connecting the GUI to the Arduino code with the new various components was also a challenge. However once again, the Arduino has shown just how powerful of a microcontroller it is by integrating all these various components into one program which can run continuously.

Appendices

Arduino code:

```
#include<LiquidCrystal.h>
#include<DHT.h>



#include <IRremote.h>
decode_results IRsignal;



#include<MFRC522.h>



#define DHTYPE DHT11

#define backIRButton 0xFF22DD
#define forwardIRButton 0xFFC23D
#define upIRButton 0xFF906F
#define downIRButton 0xFFE01F


#define waterPin   A0
#define piezo   10
#define whiteLED   11
#define backward   9
#define forward   8
#define trigPin   12
#define echoPin   13
#define redLedPin   22
#define yellowLedPin   24
#define greenLedPin   26
#define rs   28
#define en   30
#define d4   32
#define d5   34
#define d6   36
#define d7   38
#define DHTPIN 41
#define rfidRST   48
#define rfidSS   53
#define IRremoteSig  43
```

```
#define maxWater  360.0
#define minDist   7
#define maxDist   15
#define lowTone 100
#define highTone 1000
#define toneDuration 500

int val = 0;
int outval = 0;

float waterLevel;
uint8_t mtrSpeed = 0;
uint8_t prevMtrSpd = 0;
uint8_t headlightIntensity = 0;
float temperature = 0;
uint32_t prevIRValue = 0;
bool isAuth = 0;
String fobID = "A3 08 51 3A";


LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
DHT dht = DHT(DHTPIN, DHTYPE);

IRrecv irrecv( IRremoteSig);

MFRC522 mfrc522( rfidSS,  rfidRST);

void setup() {
  pinMode(forward, OUTPUT);
  pinMode(backward, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(waterPin, INPUT);
  pinMode(yellowLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
  pinMode( piezo, OUTPUT);
  pinMode( whiteLED, OUTPUT);

  irrecv.enableIRIn();
  irrecv.blink13(false);

  Serial.begin(115200);
```

```
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("SPD=");
  lcd.setCursor(8, 0);
  lcd.print("WTR=");
  lcd.setCursor(0, 1);
  lcd.print("TMP=");



  SPI.begin();
  mfrc522.PCD_Init();
  tone( piezo, lowTone, toneDuration);
  delay(toneDuration);

  tone( piezo, highTone, toneDuration);
  delay(toneDuration);
  noTone( piezo);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  float duration = pulseIn(echoPin, HIGH);


  if (mfrc522.PICC_IsNewCardPresent())
  {
    if (mfrc522.PICC_ReadCardSerial()){
      String content= "";
      byte letter;
      for (byte i = 0; i < mfrc522.uid.size; i++) {
         content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
         content.concat(String(mfrc522.uid.uidByte[i], HEX));
      }
      content.toUpperCase();
      if (content.substring(1) == fobID){
         isAuth = 1;
         tone( piezo, highTone, toneDuration);
         delay(toneDuration);
      }
      else   {
```

```
        isAuth = 0;
        tone( piezo, lowTone, toneDuration);
        delay(toneDuration);
      }
  }
}



float distance = (duration*.0343)/2;
lightUp(distance/100);
float waterLevel = (float)analogRead(waterPin)/maxWater;

if (!isnan(dht.readTemperature()))
  temperature = dht.readTemperature();




lcd.setCursor(4, 0);
lcd.print("    ");
lcd.setCursor(4, 0);
lcd.print((int)((float)mtrSpeed/2.5));
lcd.setCursor(12, 0);
lcd.print(waterLevel);
lcd.setCursor(4, 1);
lcd.print(temperature);

if (irrecv.decode(&IRsignal)) {
  irrecv.resume();
  //Serial.println(IRsignal.value);
  if (isAuth){
    switch (IRsignal.value){
      case backIRButton:
        if (mtrSpeed != 0)
          mtrSpeed = 0;
        break;
      case forwardIRButton:
        if (mtrSpeed != 250)
          mtrSpeed += 25;
        break;
      case downIRButton:
         if (headlightIntensity != 0) headlightIntensity -= 125;
         analogWrite( whiteLED, headlightIntensity );
         break;
      case upIRButton:
```

```
        if (headlightIntensity != 250) headlightIntensity += 125;
        analogWrite( whiteLED, headlightIntensity);
         break;
       default:
        break;
     }
     analogWrite(forward, mtrSpeed);
   }
   else{
    tone( piezo, lowTone, toneDuration);
    delay(toneDuration);
   }
 }


 Serial.print((float)distance/100);
 Serial.print(",");
 Serial.print((float)mtrSpeed/2.5);
 Serial.print(",");
 Serial.print(waterLevel);
 Serial.print(",");
 Serial.print(temperature);
 Serial.print(",");
 Serial.print(headlightIntensity);
 Serial.print(",");
 Serial.println(isAuth);

}

void lightUp(float dist)
{
 if (dist > 0.15){
  digitalWrite(greenLedPin, HIGH);
  digitalWrite(yellowLedPin, LOW);
  digitalWrite(redLedPin, LOW);
 }
 else if (dist > 0.07){
  digitalWrite(greenLedPin, LOW);
  digitalWrite(yellowLedPin, HIGH);
  digitalWrite(redLedPin, LOW);
 }
 else{
  digitalWrite(greenLedPin, LOW);
  digitalWrite(yellowLedPin, LOW);
  digitalWrite(redLedPin, HIGH);
 }
```

}

Python code:

```
import serial as sr
import re
import tkinter as tk
import time

baudRate = 115200
comName = "Com6"
unwantedChars = re.compile(r"[^\d,.]+")
numFields = 2

ledColors = ["Red", "Yellow", "Green"]
headLightColors = ["gray1", "gray70", "gray99"]

authStrings = ["denied. Please Scan your correct ID badge to control the testbed", "granted.
You may control the testbed"]

def _create_circle(self, x, y, r, **kwargs):
   return self.create_oval(x-r, y-r, x+r, y+r, **kwargs)
tk.Canvas.create_circle = _create_circle


def runProgram():
   ser = sr.Serial(comName, baudRate, timeout=1)
   while (True):
      curReadings = []
      strReading = unwantedChars.sub("",str(ser.readline()))
      if (strReading):
         curReadings = [float(elem) for elem in strReading.split(",")]
         distance.set("Distance = " + str(curReadings[0]) + " Meters")
         waterLevel.set("Water Level = " + str(curReadings[2]*100) + "%")
         motorSpdStr.set("Motor Speed: " + str(curReadings[1]) +"%")
         tempStr.set("Temperature = " + str(curReadings[3]) + "°C")
         ledIdx = (curReadings[0] > 0.15) + (curReadings[0] > 0.07)
         canvas.itemconfig(ledCircle, fill=ledColors[ledIdx])
         headLightIdx = (curReadings[4] > 0) + (curReadings[4] > 200)
         isAuth.set("Current access is " + authStrings[int(curReadings[5])])
         canvas2.itemconfig(headLight1, fill=headLightColors[headLightIdx])
         canvas2.itemconfig(headLight2, fill=headLightColors[headLightIdx])
      window.update()
   ser.close()

if __name__ == "__main__":
```

```
window = tk.Tk()
distance = tk.StringVar()
ledColor = tk.StringVar()
waterLevel = tk.StringVar()
motorSpdStr = tk.StringVar()
isAuth = tk.StringVar()
tempStr = tk.StringVar()
window.geometry("600x600")
window.title("Test Bed")

canvas = tk.Canvas(window, width = 120, height = 120)
canvas.place(relx = 0.8333, rely = 0.0, anchor=tk.CENTER)
canvas.pack()
start = tk.Button(window, text="Start Running", command = runProgram)
start.place(x=500, y=500)
distanceLabel = tk.Label(window, textvariable=distance)
distanceLabel.place(x = 90, y = 20)

ledCircle = canvas.create_circle(60, 60, 50)
canvas.itemconfig(ledCircle, fill = "Gray")

waterLabel = tk.Label(window, textvariable=waterLevel)
waterLabel.place(x = 90, y = 120)

motorSpdLabel = tk.Label(window, textvariable=motorSpdStr)
motorSpdLabel.place(x = 90, y = 160)

temperatureLabel = tk.Label(window, textvariable=tempStr);
temperatureLabel.place(x = 90, y = 200)

accessLabel = tk.Label(window, textvariable=isAuth)
accessLabel.place(x = 90, y = 380)

canvas2 = tk.Canvas(window, width = 240, height = 120)
canvas2.place(relx = 0.5, rely = 0.5, anchor=tk.CENTER)
headLight1 = canvas2.create_circle(60, 60, 50, fill="gray1")
headLight2 = canvas2.create_circle(180, 60, 50, fill="gray1")

window.mainloop()
```