ECE 303-062 - LAB 6: The final Project - Tech Memo
To: TA Jeffrey Gregorio
From: Quoc Thinh Vo – 14419634
Date: 11/30/2020

Introduction

In this half-term project, students will be developing an electric vehicle testbed. This will utilize just about everything students have learned throughout the term. Students are suggested to think about how to structure the project before just writing code and building circuits.

This project has the following areas that have been introduced to students throughout the term:

1. Collision avoidance testing

2. Dashboard development

3. Security and Remote Control

4. Sensor implementation

Project Design

The design was an implementation of sensors and motor using microcontroller driven system (Arduino Mega 2560).

The project can be divided into five different main functions:
1. Collision Avoidance:
In this part of the project, students will be simulating a collision avoidance system using the ultrasonic sensor and servo motor in the provided kit. The closer an object gets to the ultrasonic sensor, the servo rotates to a different position.
In my particular design, when the object is moving close to 10cm, the motor speed threshold value will be set to 0. When the object is moving close 30cm, the motor speed threshold value will be set to 100. When the object is moving close 50cm, the motor speed threshold value will be set to 175, and when the object is moving with a distance greater than 50cm, the value of motor speed value will be set to 250.
The motor speed can also be controlled by user using the IR remote control. When the decreasing button is pressed, the motor speed will be decreased by 10% and in reverse, when the increasing button is pressed, the motor speed will be increased by 10%.

2. Graphical User Interface
For this part of the project, students are to use a graphical user interface.
I used Python to produce my interface. The designed GUI has the following displays:

- Distance to object (in centimeters)
- Distance LEDs based on distance (Red, Yellow, and Green)

- Headlights (off/dim/bright) that controlled by the IR remote
- Motor Speed when object is moving
- Coolant level and alarm when it is lower than 10.
- Temperature display and alarm when it is greater than 90 Celsius degrees.

3. Interlocks
In this part of the project, students need to have the system shut down in the event of a high temperature or low water level alarm. The system will activate the buzzer to notify user that it is under critical conditions.

4. Security
In this part of the project, students are to implement a security measure to prevent unauthorized personnel from starting the system.
In my design, when the fob is scanned the UID tag is wrong and therefore, it will not activate the system. When the key is scanned the UID is right and the system is activated to function as designed.
When user scans a wrong key, the system will activate the buzzer to notify user that it is locked due to incorrect input key.

5. Displays:
Beside the values that have been displayed on the GUI, an additional LED was implemented in order to display the motor speed, coolant level, and system temperature.

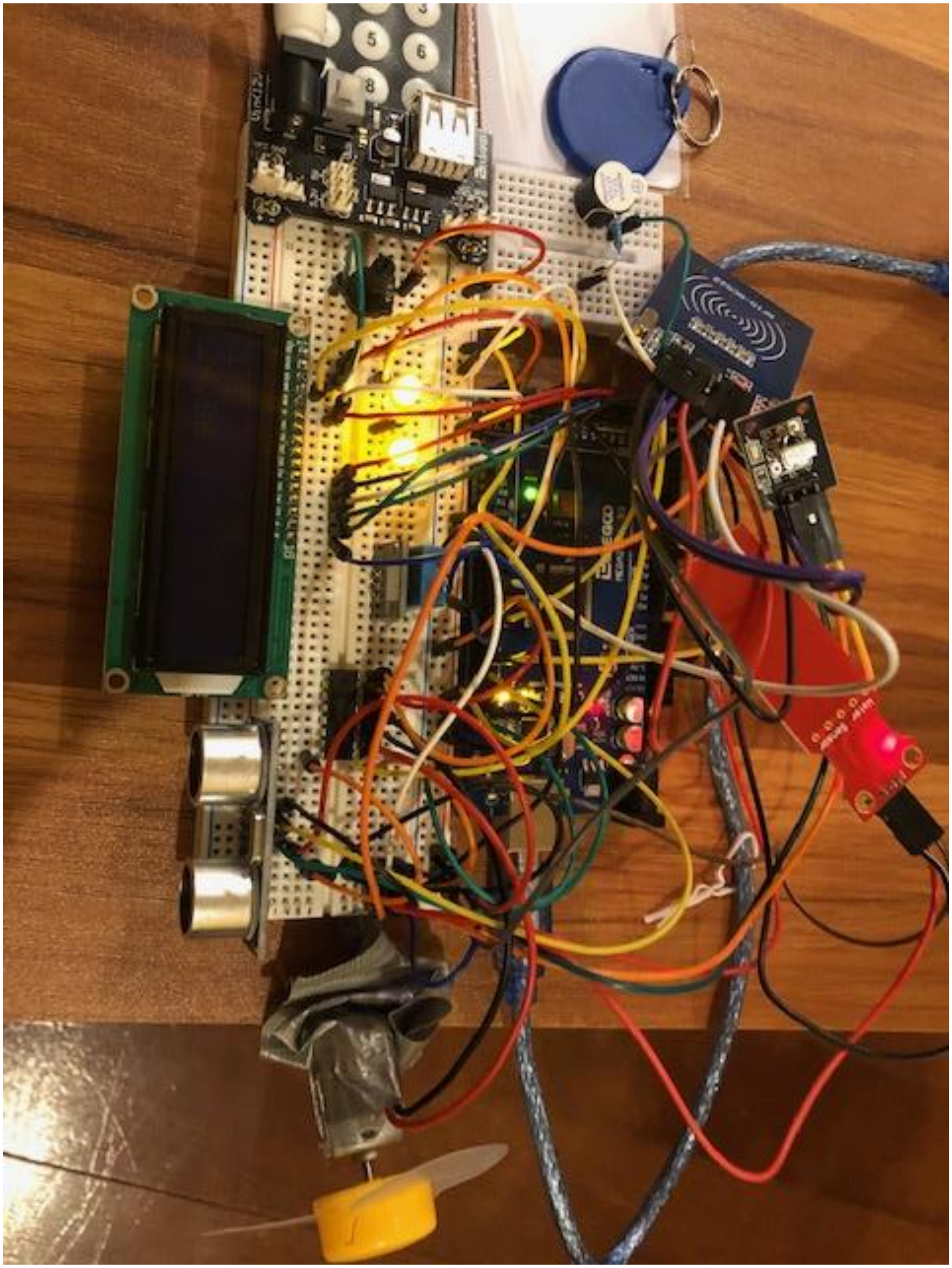


Figure 1: Photo of real system implementation

Technical Activity

Sensors and electronic parts that were used to design the project:

1. DHT11 Temperature and humidity sensor
2. Ultrasonic Sensor
3. Water Level Detection Sensor
4. L293D bridge
5. IR remote control
6. MEGA 2560 Controller Board
7. RC522 RFID Module
8. LCD1602 Module
9. Fan Blade and 3-6V Motor
10. Active buzzer
11. IR receiver Module
12. IR Receiver Module
13. Power Supply Module
14. Yellow LEDS
15. Resistors

Project Code

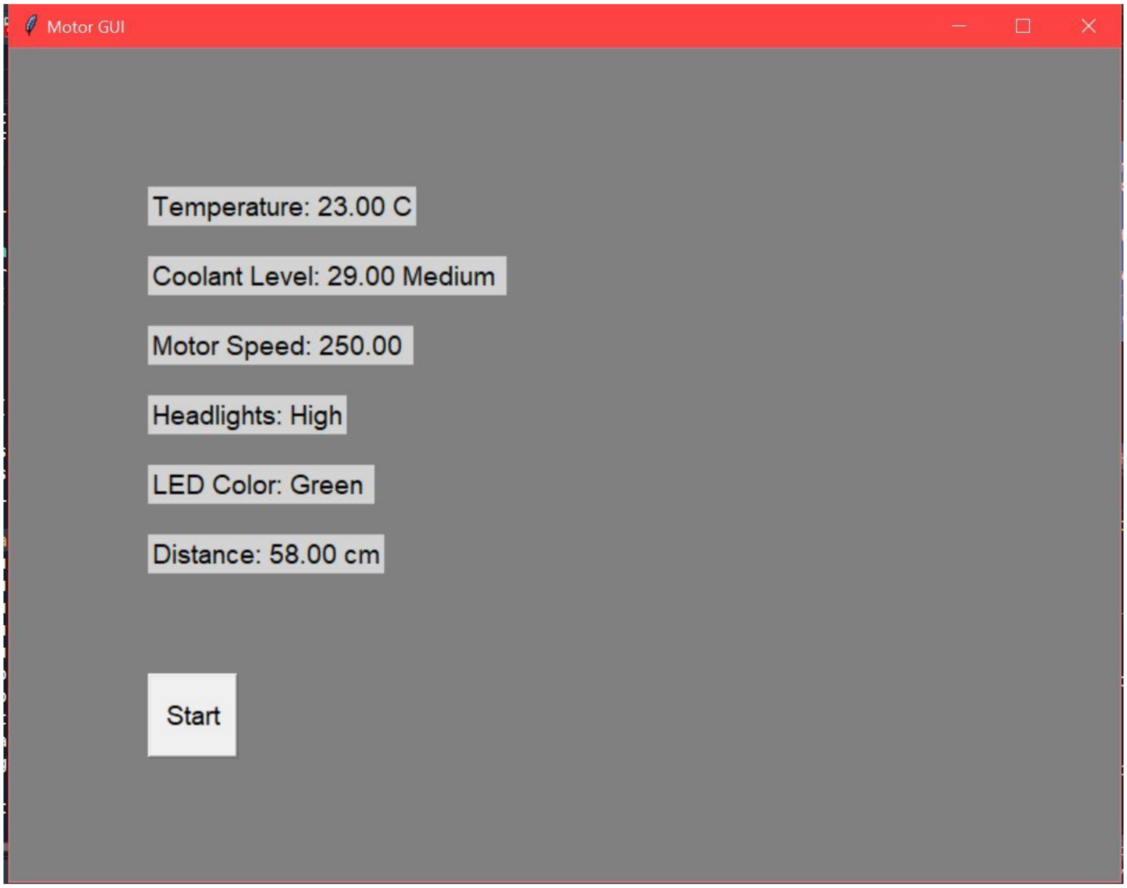
The Arduino IDE and Python source code have been attached to the report at the end as appendix. There are also project.ino and project.py file that have been attached in the submitted zip file.

Results

A result video has been submitted to demonstrate the behavior of the designed system.

The design has successfully functioned as intended. In testing environment, when object was moving close to the ultrasound sensor, the motor was able to slow down until it stopped working. The LED color indicated on GUI also displayed a correct output in comparison to the distance.

When user scan the correct key, the system was able to activate and in reverse, it was locked up due to in correct key scan with sound alarm.



Temperature: 23.00 C

Coolant Level: 29.00 Medium

Motor Speed: 250.00

Headlights: High

LED Color: Green

Distance: 58.00 cm

Figure 2: Sample of GUI displayed

When the coolant level was lower than 10 or the temperature was greater than 90 Celsius degrees, the system would also activate the sound alarm to notify the critical conditions.

The LCD was able to display the motor speed, coolant level, and system temperature. The implementation of IR remote was also successful. The remote was able to increase or decrease the motor speed by 10% for each button hit. It also set the 2 LEDs, that was implemented in place of headlights, to off, dim or high using the analog write to change the LED intensity.

Discussion

The project was an implementation of several components that have been introduced throughout the term. It was easier to implement and test all small units before building the whole system. Design the system required students to understand each of the components and to have the general idea of the working system before starting to build it. Putting together a system requires students the ability to design and connect wires, understand the MEGA 2560 board, function timer and general ideas of circuits. In order to calculate the current that was drawn from source, we can use the Ohms law with 5V source and 2k Ohms resister that were use. That will genuinely protect the circuit and prevent overflow. In the future, the system can be scaled and re-designed for better functioning. We can implement more functions and perfect the current functions. Overall, the project was an interesting design that required students to understand each of the weekly lecture plus the ability to research and develop a final design.

Appendix

Arduino Source Code

```
#include <LiquidCrystal.h>
#include <dht.h>
#include <IRremote.h>
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 53
#define RST_PIN 4
MFRC522 mfrc522(SS_PIN, RST_PIN);

IRrecv irrecv(10);
decode_results results;
long lastPressTime=0;
int state = LOW;
LiquidCrystal lcd(31,30,37,35,34,32);

dht DHT;
#define DHT11_PIN 8

#define SLOW 16769055
#define FAST 16748655
#define ONE 16724175
#define TWO 16718055
#define THREE 16743045

int alarm;
int flag;
int buzzer = 46;//the pin of the active buzzer
float control=1.0;
int intensity=255;


int coolant=0;
int respin=A5;

const int forward = 7;

int MotorThres=0;
int motorSpeed=0;

int trigPin = 11;    // Trigger
int echoPin = 12;    // Echo
long duration, cm, inches;

void setup() {
  pinMode(forward,OUTPUT);
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzer,OUTPUT);

  irrecv.enableIRIn();
  pinMode(26,OUTPUT);
  analogWrite(26,120);
  lcd.begin(16,2);
  SPI.begin();      // Init SPI bus
  mfrc522.PCD_Init();   // Init MFRC522
  delay(4);

}

void loop() {
  //Card key set up
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }
  //Show UID
  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) == "3A DD D7 81") {
        flag =1;
       delay(1000);
```

```
 }


if(content.substring(1) != "3A DD D7 81"){
   Serial.println(0);
   Serial.println(0);
   Serial.println(0);
   Serial.println(0);
   Serial.println(0);
   flag =0;
   int i;
   for(i=0;i<200;i++){
     digitalWrite(buzzer,HIGH);
     delay(1);//wait for 1ms
     digitalWrite(buzzer,LOW);
     delay(1);//wait for 1ms
   }
   delay(1000);
 }
 while(flag){
     digitalWrite(trigPin, LOW);
     delayMicroseconds(5);
     digitalWrite(trigPin, HIGH);
     delayMicroseconds(10);
     digitalWrite(trigPin, LOW);
     pinMode(echoPin, INPUT);
     duration = pulseIn(echoPin, HIGH);
     //Convert time into a distance
     cm = (duration/2) / 29.1;
     if (cm<=10){
       MotorThres=0;
     }else if (cm<=30){
       MotorThres=100;
     }else if (cm<50){
       MotorThres=175;
     }else{
       MotorThres=250;
     }
     //Remote Control
      if (irrecv.decode(&results)){
       if (results.value==SLOW){
         if (control > 0 ){
           control = control - 0.1;
         }else {
           control = 0;}
        }
        else if (results.value==FAST){
         if (control <1){
           control = control + 0.1;
         }else{
           control =1;}
         }
         else if(results.value==ONE){
           intensity=0;
         }else if(results.value==TWO){
           intensity=50;
         }else if(results.value==THREE){
           intensity=255;
         }
       irrecv.resume();
       }
       analogWrite(2,intensity);
       analogWrite(3,intensity);

       motorSpeed=MotorThres*control;
       analogWrite(forward,motorSpeed);

       int chk = DHT.read11(DHT11_PIN);
       coolant=analogRead(respin);
       delay(50);
       if (coolant <=10 || DHT.temperature>=90){
         alarm=1;
       }else {
         alarm=0;
       }
       if (alarm){
         int i;
         for(i=0;i<200;i++){
            digitalWrite(buzzer,HIGH);
            delay(1);//wait for 1ms
            digitalWrite(buzzer,LOW);
            delay(1);//wait for 1ms
         }
       }
       Serial.print(cm);
       Serial.println();
       delay(50);
       Serial.print(DHT.temperature);
       Serial.println();
       delay(50);
       Serial.print(coolant);
       Serial.println();
```

```
        delay(50);
        Serial.print(motorSpeed);
        Serial.println();
        delay(50);
        Serial.print(intensity);
        Serial.println();

        lcd.setCursor(0,0);
        lcd.print(motorSpeed);
        lcd.setCursor(9,0);
        lcd.print(coolant);
        lcd.setCursor(0,1);
        lcd.print(DHT.temperature);

        delay(1000);
  } // end of while(flag)
}// end of loop()
```

Python Source Code

```
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 23 01:04:52 2020

@author: Quoc Thinh Vo
"""

import serial
import time
import tkinter as tk
import tkinter.font as tkFont
from tkinter import *
from tkinter import messagebox
import matplotlib.pyplot as plt


plt.close('all')
root=tk.Tk()
root.geometry("800x600")
root.title("Motor GUI")
root.configure(bg='grey')

figure1=plt.Figure(figsize=(12,6), dpi=100)
```

```
figure1.subplots_adjust(hspace=0.5)
fontStyle = tkFont.Font(family="Lucida Grande", size=14)
var5=StringVar()
label = Label( root, textvariable=var5, font = fontStyle, bg='lightgrey')
label.place(x=100,y=100)

figure1.subplots_adjust(hspace=0.5)
fontStyle = tkFont.Font(family="Lucida Grande", size=14)
var4=StringVar()
label = Label( root, textvariable=var4, font = fontStyle, bg='lightgrey')
label.place(x=100,y=150)

figure1.subplots_adjust(hspace=0.5)
fontStyle = tkFont.Font(family="Lucida Grande", size=14)
var3=StringVar()
label = Label( root, textvariable=var3, font = fontStyle, bg='lightgrey')
label.place(x=100,y=200)

figure1.subplots_adjust(hspace=0.5)
fontStyle = tkFont.Font(family="Lucida Grande", size=14)
var2=StringVar()
label = Label( root, textvariable=var2, font = fontStyle, bg='lightgrey')
label.place(x=100,y=250)


figure1.subplots_adjust(hspace=0.5)
fontStyle = tkFont.Font(family="Lucida Grande", size=14)
var1=StringVar()
label = Label( root, textvariable=var1, font = fontStyle, bg='lightgrey')
label.place(x=100,y=300)


figure1.subplots_adjust(hspace=0.5)
fontStyle = tkFont.Font(family="Lucida Grande", size=14)
var=StringVar()
label = Label( root, textvariable=var, font = fontStyle, bg='lightgrey')
label.place(x=100,y=350)

def startCallBack():
    arduino=serial.Serial('COM7',9600,timeout=5)
    time.sleep(1)
    Dist=[]
    Temp=[]
    Coolant=[]
    Motor=[]
    Headlight=[]
```

```
arduino.reset_output_buffer()
arduino.reset_input_buffer()
arduino.write(b'1')

for i in range(10001):
    a=(arduino.readline().decode("utf-8"))
    b=(arduino.readline().decode("utf-8"))
    c=(arduino.readline().decode("utf-8"))
    d=(arduino.readline().decode("utf-8"))
    e=(arduino.readline().decode("utf-8"))
    Dist.append(float(a))
    Temp.append(float(b))
    Coolant.append(float(c))
    Motor.append(float(d))
    Headlight.append(float(e))

    """Distance"""
    var.set("Distance: {:.2f} cm".format(Dist[-1]))
    """Distance"""

    """Distance LED"""
    if (float (a) <= 10):
        LED = "Red"
    elif (float (a) <= 30):
        LED = "Yellow"
    else:
        LED ="Green"

    var1.set("LED Color: {} ".format(LED))
    """Distance LED """

    """Headlights"""
    if (float(e) == 0):
        var2.set("Headlights: Off".format(Headlight[-1]))
    elif (float(e) ==150):
        var2.set("Headlights: Dim".format(Headlight[-1]))
    elif(float(e) ==255):
        var2.set("Headlights: High".format(Headlight[-1]))
    """Headlights"""

    """Motor Speed"""
    var3.set("Motor Speed: {:.2f} ".format(Motor[-1]))
    """Motor Speed"""

    """Coolant LEVEL"""
```

```
        if (float(c) <= 10):
            var4.set("Coolant Level: {:.2f} - LOW Warning".format(Coolant[-1]))
        elif (float(c) <=300):
            var4.set("Coolant Level: {:.2f} Medium ".format(Coolant[-1]))
        else:
            var4.set("Coolant Level: {:.2f} High".format(Coolant[-1]))
        """Coolant LEVEL"""

        """TEMP ALARM"""
        if (float(b) >= 90):
            var5.set("Alarm: Temperature is over 90 C")
        else:
            var5.set("Temperature: {:.2f} C".format(Temp[-1]))
        """TEMP ALARM"""
        time.sleep(1)

    arduino.close()
    return

start_button = Button(root,text="Start", font=fontStyle,
height=2,width=5,command=startCallBack)
start_button.place(x=100,y=450)
root.mainloop()
```