Drexel University

# Memo #6

**To:** ECE 303

**From**: David Schmidt

**CC**: Other recipients

## **Arduino Mega 2650 Lab final All systems together**

## **Task:** Wrap all systems from the quarter together into one system. This includes RFID locking system with buzzer, low coolant and temp interlocks, LCD display showing motor percentage and temperature, Lights connected ultrasonic sensor distance, Lights for bright, dim and off. All these lights and motor percentage is controlled by a wireless IR remote to the Arduino. Finally the entire system should run and have indications through MATLAB software.

**Equipment:** Mega Arduino 2650 starter kit

2-10k ohm resistor

Jumper wires

Bread Board

Power unit

H-bridge

2 5v LED 10mA

5 v buzzer max 32 mA

**Equipment Continued:**

5v motor fan. 200mA

5v water level sensor 20mA.

3.3 IR receiver 23 mA,

LCD screen max 32mA

Ultrasonic distance sensor 15mA.

Temperature and Humidity sensor 0.3mA

Adding up all the current we get a current **budget = 332.3mA**

This amount of current is too much for the Arduino Mega 2560 to handle so we use a power unit to supply current to the fan motor

A picture containing indoor, clothes, cluttered

Description automatically generated

**GUI:**

Distance, LEDs, Headlights Motor speed, coolant and temperature were all sent to MATLAB where I made an app to indicate all the signals.

Graphical user interface, application

Description automatically generated

Code of this app will be attached to the HW submission.

Each of these indications were controlled by sensor input coming from the Arduino in real time.

Chart, radar chart

Description automatically generated

**CODE:** Code is on the following pages and the Arduino sketch will be attached with the submission

#define echoPin 42

#define trigPin 44

#include <MFRC522.h>

#include <require\_cpp11.h>

#include <SPI.h>

#define SS\_PIN 53

#define RST\_PIN 4

#include <DHT.h>

#define DHTPIN 12

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

#include <LiquidCrystal.h>

#define SIGNAL\_PIN 13

LiquidCrystal lcd(2,3,8,9,10,11);

MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance.

#include <IRremote.h>

int value = 0;

// defines variables

long duration; // variable for the duration of sound wave travel

int distance; // variable for the distance measurement

const int forward=5;

const int backward = 7;

int val = 0;

int out=0;

char temp;

float t = 0;

const byte IR\_RECEIVE\_PIN = 49;

String instr = "distance ";

String outstr=", Output speed: ";

String rotation ="water level";

int led1 = 24;

int led2 = 26;

int led3 = 28;

int b = 0;

const int buzzer = 6;

int light = 0;

int move1 = 100;

int tlight = 0;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT

pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed

pinMode(forward,OUTPUT);

pinMode(26,OUTPUT);

pinMode(buzzer, OUTPUT);

analogWrite(26,120);

lcd.begin(16,2);

SPI.begin(); // Initiate SPI bus

mfrc522.PCD\_Init(); // Initiate MFRC522

pinMode(led1,OUTPUT);

pinMode(led2,OUTPUT);

pinMode(led3,OUTPUT);

IrReceiver.begin(IR\_RECEIVE\_PIN, ENABLE\_LED\_FEEDBACK);

int b = 0;

dht.begin(); // initialize the sensor

}

void loop() {

// Clears the trigPin condition

digitalWrite(trigPin, LOW);

while(b != 1)

{

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

b = 0;

lcd.setCursor(0,1);

lcd.print("ID card?");

return;

}

if ( ! mfrc522.PICC\_ReadCardSerial())

{

b = 0;

}

//Show UID on serial monitor

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) == "33 89 D3 18") //change here the UID of the card/cards that you want to give access

{

b = 1;

noTone(buzzer);

delay(500);

}

else {

b = 0;

tone(buzzer,1000);

delay(500);

}

}

if (IrReceiver.decode())

{

light = IrReceiver.decodedIRData.command;

IrReceiver.resume();

}

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2; // Speed of sound wave divided by 2 (go and back)

val=Serial.parseInt();

temp = Serial.read();

if(light == 12){

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

tlight = 0;

}

else if(light == 24){

analogWrite(led1,255);

analogWrite(led2,255);

analogWrite(led3,255);

tlight = 1;

}

else if(light ==94){

analogWrite(led1,135);

analogWrite(led2,135);

analogWrite(led3,135);

tlight = 2;

}

else if(light ==70){

if(move1 <=224)

{

move1 = move1+(255\*0.10);

analogWrite(forward,move1);

light = 0;

}

}

else if(light ==21){

if(move1 >= 25)

{

move1 = move1-(255\*0.10);

analogWrite(forward,move1);

light = 0;

}

}

t = dht.readTemperature();

t = 33;

float move2 = move1;

move2 = 100\*move2/255;

value = analogRead(SIGNAL\_PIN); // read the analog value from sensor

Serial.println(String(distance)+" "+String(tlight)+" "+String(move2)+" "+String(value)+" "+String(t));

lcd.clear();

lcd.setCursor(0,0);

lcd.print(String(move2)+"%");

lcd.setCursor(0,1);

lcd.print(String(t)+"C");

if(t > 50){

b = 0;

lcd.print("System too hot, shutting down");

analogWrite(forward,0);

delay(2000);

}

}