Arduino Program

Main File

```
/*
*   Main file for full Arduino Program
*/
#include <Wire.h>

char BT;
bool Secure = 0;
bool RunningLights = 0;

/*Initialization function
  initialize each module
  set ISR's
*/
void setup() {
  // Open serial communications and wait for port to open:
  //Serial.begin(9600);
 // while (!Serial) {
 //   ; // wait for serial port to connect. Needed for Leonardo only
 // }
  //Keep power ON
  pinMode(A0, OUTPUT);
  digitalWrite(A0, HIGH);

  //Initiate Modules
  BluetoothInit();
  LEDInit();
  LIDARInit();
  MicroSDInit();

  //Interrupt on BT Receive
  attachInterrupt(digitalPinToInterrupt(2), BTMessageReceive, CHANGE);
  //Play welcome message
  Welcome();
}

/*Arduino version of a main() function
  currently empty
  most modules will work as an ISR
  Might be used for the LIDAR module to take timed measurements
*/
void loop() {
  delay(500);
  //Check for text message to return coordinates
  if(CheckMessages()){
    Receive(); //clear Serial
```

```
    sendMessage();
  }

 if(RunningLights == 0){
  RunningOff();
 } else if(RunningLights == 1){
  RunningOn();
 }

  //Check Lights, Security and LiDAR each time through loop
  LightCheck();
  getDistance();

  //If Security is turned out, switch state and enter Security Mode
  if(BT=='C'){
    Secure = 1;
    SecurityMode();
  }
}
```

LED File

```
/*
 * Test program to turn on LEDs for project
 * Start with just a simple LED connected to a digital pin
 * Note: For project implementation, these will need to be initialized
with the internal pullup resistor because of BJT
 * Author: Jeff Scherer
 * Date: July 8, 2021
*/

//LED constants for digital pins
#define RIGHT 6     //Right Signal Connected to Digital Pin 4
#define LEFT 4      //Left Signal Connected to Digital Pin 5
#define Rear_RUN A3  //Rear Running Lights Connected to Digital Pin 6
#define Front_RUN A1  //Front Running Lights Connected to Digital Pin 7
#define BRAKE 7     //Brake Lights connected to Digital Pin 8

int LightLogic = A7;

//Initialize all LED outputs
void LEDInit() {
  pinMode(LEFT, OUTPUT);
  pinMode(RIGHT,OUTPUT);
  pinMode(Rear_RUN,OUTPUT);
  pinMode(Front_RUN,OUTPUT);
  pinMode(BRAKE,OUTPUT);
```

```
}

//Turn on Running Lights
void RunningOn(void){
  digitalWrite(Front_RUN,HIGH);
  digitalWrite(Rear_RUN,HIGH);
}

//Turn off Running Lights
void RunningOff(void){
  digitalWrite(Front_RUN,LOW);
  digitalWrite(Rear_RUN,LOW);
}

//Signal Left
void LeftSignal(void){
  digitalWrite(LEFT,HIGH);
  delay(500);
  digitalWrite(LEFT,LOW);
  delay(500);
}

//Signal Right
void RightSignal(void){
  digitalWrite(RIGHT,HIGH);
  delay(500);
  digitalWrite(RIGHT,LOW);
  delay(500);
}

//Turn on Brake Lights
void AmBraking(void){
  digitalWrite(BRAKE,HIGH);
}

//Turn off Brake Lights
void NotBraking(void){
  digitalWrite(BRAKE,LOW);
}

//Check if brake button is pressed
void LightCheck(void){
  int Logic = analogRead(LightLogic);
  if(Logic < 115){
    AmBraking();
  } else if( Logic > 115 && Logic < 135){
    NotBraking();
  } else if( Logic > 135 && Logic < 155){
    RightSignal();
    AmBraking();
```

```
    } else if( Logic > 160 && Logic < 180){
      NotBraking();
      RightSignal();
    } else if( Logic > 230 && Logic < 260){
      LeftSignal();
      AmBraking();
    } else if( Logic > 315){
      NotBraking();
      LeftSignal();
    }
  }
```

LiDAR File

```
#include <Wire.h>
#include <LIDARLite.h>
#include <LIDARLite_v3HP.h>
#include <LIDARLite_v4LED.h>

LIDARLite lidarLite;
int cal_cnt = 0;
int count = 0;
int dist[100];
int Compare = 0;
int Diff = 0;
int Closer = 0;


void LIDARInit(){
  lidarLite.begin(0, true);
  lidarLite.configure(0);
}

//get a new distance measurement from LiDAR
//Compare new distance with last distance
//if new distance is 500cm (5m) closer, increment counter
//increment 5 time, trigger alert
void getDistance(){
  if(count == 0){
    dist[count] = lidarLite.distance();   //every 100 measurements,
correct bias
  } else {
    dist[count] = lidarLite.distance(false);
  }
  //compare new distance with last distance
  Compare = dist[count] - dist[count-1];
  //if 1m closer, increment counter
  if(Compare < -250){
    Closer++;
    //if 25m closer, alert cyclist
    if(Closer == 5){
      Car();
      Closer = 0;
    }if(Compare > -200){
      Closer = 0;
    }
  }
  count++;
  count = count % 100;
}
```

MicroSD File

```
#include <SPI.h>
#include <SD.h>
#include "TMRpcm.h"

//SD Specific constants
File myFile;
const int chipSelect = 10;
TMRpcm Speaker;

//MicroSD Initialization Test
void MicroSDInit(void){
  Speaker.speakerPin=9;
  Speaker.setVolume(5);
  Speaker.stopPlayback();
  Speaker.quality(1);
  Speaker.loop(0);

  if (!SD.begin()) {
    return;
  }
}

void Story(void){
  Speaker.play("FreeHeineken.wav");
}

void Welcome(void){
  Speaker.play("Hey.wav");
}

void Warning(void){
  Speaker.play("warning1.wav");
}

void Alarm(void){
  Speaker.play("5.wav");
}

void Bell(void){
  Speaker.play("bell1.wav");
  delay(1000);
}

void Car(void){
  Speaker.play("App.wav");
}
```

SIM File

```
#include <SoftwareSerial.h>
SoftwareSerial SimSerial(8,5); //RX, TX
String PhoneNumber = "2267970765";

void SimInit(void){
  SimSerial.begin(19200); // the GPRS baud rate
  //delay(15000);//wait for SIM module to boot up
  SimSerial.println("AT+CMGF=1");//Because we want to send the SMS in
text mode
  delay(100);
  SimSerial.println("AT+CGPSSSL=0");//no gps certificate
  delay(100);
  //commented out already set up
  //SimSerial.println("AT+CGPSHOT");//start gps in Hot mode
  //delay(5000);
}

//Reads serial data from the SIM5360
char * Receive(void){
  char x[151];
  static char* reply = x;
  int i = 0;
  while(SimSerial.available()){
    delay(2);
    reply[i] = SimSerial.read();
    i++;
  }
  reply[i] = '\0';
  return(reply);
}

//Reads GPS and sends it to user through sms
void sendMessage(void){
  int colon;
  int commas[8];
  int i = 0;
  char Lat[13];
  char Long[13];
  char Date[8];
  char Time[10];
  char Alt[7];
  char Speed[7];
  char Course[7];
  char *reply;

  SimSerial.println("AT+CGPSINFO");//Check GPS
  for(int j=0; j<1000;j++){
```

```
    delay(1000);
    if(SimSerial.available()){
      break;
    }
  }
  reply = Receive();

 int sizem = strlen(reply); //find the ":"
  for(i = 0; i < sizem; i++){
    if(reply[i] == 58){     //":" ascii
      colon = i;
      break;
    }
  }

  if(i>=sizem){//return if no ":"
    return;
  }

  i++;
  int a = 0;
  for(i; i < sizem; i++){ //find all the ","
    if(reply[i] == 44){   //"," ascii
      commas[a] = i;
      a++;
    }
  }

  if((commas[1] - colon == 1)||(commas[3] - commas[2] == 1)){
    SimSerial.print("AT+CMGS=\"+");//send sms message
    SimSerial.print(PhoneNumber);//send sms message
    SimSerial.println("\"");//send sms message
    delay(100);
    SimSerial.println("GPS Location Not Found");
    delay(100);
    SimSerial.println((char)26);//the ASCII code of the ctrl+z is 26
    delay(100);
    return;
  }

  for(int j = 0; j<14; j++){    //Latitude
    if(reply[colon+1+j]==44){
      Lat[j] = "\0";
      break;
    }
    Lat[j]=reply[colon+1+j];//colon+1
  }

  for(int j = 0; j<14; j++){    //Longitude
    if(reply[commas[1]+1+j]==44){
```

```
    Long[j] = "\0";
    break;
  }
  Long[j]=reply[commas[1]+1+j];
}
Long[0]= 45;

for(int j = 0; j<11; j++){    //Date
  if(reply[commas[3]+1+j]==44){
    Date[j] = "\0";
    break;
  }
  Date[j]=reply[commas[3]+1+j];
}

for(int j = 0; j<11; j++){    //Time
  if(reply[commas[4]+1+j]==44){
    Time[j] = "\0";
    break;
  }
  Time[j]=reply[commas[4]+1+j];
}

for(int j = 0; j<8; j++){    //Altitute
  if(reply[commas[5]+1+j]==44){
    Alt[j] = "\0";
    break;
  }
  Alt[j]=reply[commas[5]+1+j];
}

for(int j = 0; j<8; j++){    //speed
  if(reply[commas[6]+1+j]==44){
    Speed[j] = "\0";
    break;
  }
  Speed[j]=reply[commas[6]+1+j];
}

for(int j = 0; j<8; j++){    //course
  if(reply[commas[7]+1+j]==46){
    Course[j] = reply[commas[7]+1+j];
    Course[j+1]=reply[commas[7]+2+j];
    Course[j+2] = "\0";
    break;
  }
  Course[j]=reply[commas[7]+1+j];
}

SimSerial.println("AT+CMGS=\"+2267970765\"");//send sms message
```

```
  delay(100);
  SimSerial.print("Latitude: ");
  SimSerial.println(Lat);//the content of the message
  SimSerial.print("Longitude: ");
  SimSerial.println(Long);//the content of the message
  SimSerial.print("Date: ");
  SimSerial.println(Date);//the content of the message
  SimSerial.print("Time: ");
  SimSerial.println(Time);//the content of the message
  SimSerial.print("Alt: ");
  SimSerial.println(Alt);//the content of the message
  SimSerial.print("Speed: ");
  SimSerial.println(Speed);//the content of the message
  SimSerial.print("Course: ");
  SimSerial.println(Course);//the content of the message

  delay(100);
  SimSerial.println((char)26);//the ASCII code of the ctrl+z is 26
  delay(100);
}

//Check SMS Messages returns 1 if there is a message from the correct
number else returns 0
int CheckMessages(void){
  char *reply;
  char number[12];
  Receive();//clear serial
  SimSerial.println("AT+CMGL=\"ALL\"");//check messages//+CMGL=\"ALL\"
  delay(100);
  reply = Receive();
  int sizem = strlen(reply); //find the "+"
  int pluscount = 0;
  int i;
  for(i = 0; i < sizem; i++){
    if(pluscount == 2){
      break;
    }
    if(reply[i] == 43){     //"+" ascii
      pluscount++;
    }
  }
  if(i>=sizem){//return if no "+"//return if no message
    return 0;
  }
  SimSerial.println("AT+CMGD=0,4");//delete messages
  delay(100);
  for(int j = 0; j<13; j++){  //number
    if(reply[i+j] == 34){       //"ascii
      number[j]=char(0); //NULL ascii
      break;
```

```
      }
      number[j]=reply[i+j];
    }

    if (strlen(number)== 10){ ////////////normal
      if(strcmp(number,&PhoneNumber[0])){
        return 0;
      }
      //Serial.println("match");
      return 1;
    }

    else if (strlen(number)== 11){////////////long distance
      char TempNumber[12];
      TempNumber[11] = char(0);
      sprintf(TempNumber, "1%s", &PhoneNumber[0]);
      if(strcmp(number,TempNumber)){
        return 0;
      }
      return 1;
    }
    else{
      return 0;
    }
  }
```

Security File

```
int Security = A6;

//Check vibration sensor
int SecurityCheck() {
  int Check = analogRead(Security);
  if(Check > 100){
    Secure = 1;
    Speaker.stopPlayback();
    return 1;
  }
  return 0;
}

//check vibration, send warning, signal alarm
void SecurityMode(){
  SimInit();
  while(Secure){
    if(SecurityCheck()){
      Warning();
      delay(3000);
      for(int i = 0; i<100000;i++){
        if(SecurityCheck()){
          Alarm();
          sendMessage();
          delay(10000);
          if(!Secure){
            break;
          }
        }
      }
    }

  }
}
```

Bluetooth File

```
#include "SoftwareSerial.h"

//Set digital input/output pins
//cannot use Rx Tx pins on Arduino because of serial communication
complications
SoftwareSerial mySerial(2,3);  //Rx, Tx

void BluetoothInit(){
  mySerial.begin(9600);
}

//Receive message interrupt
//execute functions based on user input on ap
void BTMessageReceive(void){
  //receive message
  BT = char(0);
  BT = mySerial.read();
  //Serial.println("message: ");
  //turn on running lights
  if(BT == 'A'){
    if(RunningLights == 0){
      RunningLights = 1 ;
    } else if(RunningLights == 1){
      RunningLights = 0;
    }
  }
  //ring bell
  if(BT == 'B'){
    Bell();
  }
  //turn off security
  if(BT == 'D'){
    Secure = 0;
    Speaker.stopPlayback();
    Speaker.play("Unlocked.wav");
  }
  //Power off
  if(BT == 'E'){
    digitalWrite(A0, LOW);
  }
}
```