#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <SoftwareSerial.h>

#include <TinyGPS.h>

#define ALCOHOL\_SENSOR\_PIN A0

#define RELAY\_PIN 8

#define BUZZER\_PIN 7

#define BUTTON\_PIN 2

LiquidCrystal\_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2 rows

SoftwareSerial gpsSerial(10, 11); // RX, TX

TinyGPS gps;

bool engineLocked = false;

void setup() {

pinMode(ALCOHOL\_SENSOR\_PIN, INPUT);

pinMode(RELAY\_PIN, OUTPUT);

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(BUTTON\_PIN, INPUT\_PULLUP);

lcd.begin(16, 2);

lcd.print("Engine Lock System");

delay(2000);

lcd.clear();

Serial.begin(9600);

gpsSerial.begin(9600);

}

void loop() {

checkAlcoholLevel();

if (digitalRead(BUTTON\_PIN) == LOW) {

delay(1000); // Debouncing delay

engineLocked = !engineLocked;

updateDisplay();

delay(1000); // Debouncing delay

}

if (engineLocked) {

lockEngine();

} else {

unlockEngine();

}

displayGPSInfo();

}

void checkAlcoholLevel() {

int alcoholValue = analogRead(ALCOHOL\_SENSOR\_PIN);

if (alcoholValue > 500) { // Adjust this threshold based on your sensor readings

engineLocked = true;

digitalWrite(BUZZER\_PIN, HIGH);

delay(2000); // Buzz for 2 seconds

digitalWrite(BUZZER\_PIN, LOW);

updateDisplay();

delay(5000); // Wait for 5 seconds to prevent rapid locking/unlocking due to sensor noise

}

}

void lockEngine() {

digitalWrite(RELAY\_PIN, HIGH);

}

void unlockEngine() {

digitalWrite(RELAY\_PIN, LOW);

}

void updateDisplay() {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Engine: ");

lcd.print(engineLocked ? "Locked " : "Unlocked");

lcd.setCursor(0, 1);

lcd.print("Alcohol: ");

lcd.print(analogRead(ALCOHOL\_SENSOR\_PIN));

}

void displayGPSInfo() {

while (gpsSerial.available() > 0) {

if (gps.encode(gpsSerial.read())) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Lat: ");

lcd.print(gps.location.lat(), 6);

lcd.setCursor(0, 1);

lcd.print("Lon: ");

lcd.print(gps.location.lng(), 6);

delay(1000); // Display GPS info for 1 second

updateDisplay(); // Restore the main display

}

}

}