```
#include <SoftwareSerial.h>
#include <RH_ASK.h>
#include <SPI.h> // Not actualy used but needed to compile
#include <TinyGPSPlus.h>
SoftwareSerial ss(4, 5);//gps tx rx
SoftwareSerial mySerial(3,2);//gsm tx rx
RH_ASK driver;
TinyGPSPlus gps;
#define echoPin A1
#define trigPin A2
#define Relay1  9
#define Relay2  13
#define Buzzer  8
#define SoilMoisture A0
#define PushButton 12
#define IR 10
long duration;
int distance;
int ButtonState = 0;
byte gpsData = 0;
String Lat, Long;
int timer;
void displayInfo(){
  //Serial.print(F("Location: "));
  if (gps.location.isValid()){
    Lat = String(gps.location.lat());
    Long = String(gps.location.lng());
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(","));
    Serial.println(gps.location.lng(), 6);
  }
  else{
    Serial.print(F("INVALID"));
  }
```

```
}
void GSM_setup(){
    Serial.begin(9600);
    mySerial.begin(9600);
    Serial.println("Initializing...");
    delay(1000);
    mySerial.println("AT"); //Once the handshake test is successful, it will back to OK
    updateSerial();
    mySerial.println("AT+CMGF=1"); // Configuring TEXT mode
    updateSerial();
    mySerial.println("AT+CMGS=\"+916301083854\"");
    updateSerial();
    String textSMS1="Alert please help me: GPS LOCATION\nhttp://maps.google.com/?q=";
      textSMS1 += Lat;
      textSMS1 += ",";
      textSMS1 += Long;
    mySerial.print(textSMS1); //text content
    updateSerial();
    mySerial.write(26);
}
void updateSerial(){
  delay(500);
  while (Serial.available()) {
      mySerial.write(Serial.read()); //Forward what Serial received to Software Serial Port
    }
   while(mySerial.available()) {
      Serial.write(mySerial.read()); //Forward what Software Serial received to Serial Port
    }
}
void Distance() {
  long duration, distance;    // Trigger ultrasonic pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
```

```
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);   // Measure the pulse duration on echo pin
 duration = pulseIn(echoPin, HIGH);    // Calculate distance in centimeters
 distance = (duration / 2) / 29.1;    // Print distance to Serial Monitor
 Serial.print("Distance: ");
 Serial.print(distance);
 Serial.println(" cm");
 if(distance >10 && distance < 30)    {
   digitalWrite(Buzzer, HIGH);
   delay(50);
   digitalWrite(Buzzer, LOW);
   timer = distance * 10;
   delay(timer);
  }
}
void RF433_Receive(){
  uint8_t buf[12];
  uint8_t buflen = sizeof(buf);
  if (driver.recv(buf, &buflen)) // Non-blocking {
    Serial.print("Message: ");
    Serial.println((char*)buf);
    if((char*)buf ==1)  {
      digitalWrite(Buzzer,HIGH);
     }
   }
}
void SoilMoisutre(){
 int Moisture = analogRead(A0);
 int value = map(Moisture,1023,0,0,100);
 Serial.print("Moisture Level :");
 Serial.println(value);
 if(value > 50){
   digitalWrite(Relay1,HIGH);
  }
  else{
      digitalWrite(Relay1,LOW);
    }
}
void IRsensor(){
  int Motion = digitalRead(IR);
  Serial.print("Motion :");
  Serial.println(Motion);
  if(Motion == 0){
      digitalWrite(Relay2,HIGH);
    }
  else{
      digitalWrite(Relay2,LOW);
    }
}
void setup() {
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(SoilMoisture, INPUT);
  pinMode(IR,INPUT);
  pinMode(PushButton,INPUT);
  Serial.begin(9600);
  ss.begin(9600);
  if (!driver.init())
      Serial.println("init failed");
  //GSM_setup();
}
void loop() {
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
      displayInfo();
  ButtonState = digitalRead(12);
  if (ButtonState == HIGH) {
    GSM_setup();
  }
  RF433_Receive();
  SoilMoisutre();
  Distance();
  IRsensor();
  delay(500);
}
```