**GPS and GSM**

#include <SoftwareSerial.h>

#include <TinyGPS.h>

int state = 0;

const int pin = 10;

float gpslat, gpslon;

TinyGPS gps;

SoftwareSerial sgps(4, 5);

SoftwareSerial sgsm(2, 3);

void setup()

{

sgsm.begin(9600);

sgps.begin(9600);

}

void loop()

{

while (sgps.available())

{

int c = sgps.read();

if (gps.encode(c))

{

gps.f\_get\_position(&gpslat, &gpslon);

}

}

if (digitalRead(pin) == HIGH && state == 0) {

sgsm.print("\r");

delay(1000);

sgsm.print("AT+CMGF=1\r");

delay(1000);

/\*Replace XXXXXXXXXX to 10 digit mobile number &

ZZ to 2 digit country code\*/

sgsm.print("AT+CMGS=\"+8801673077335\"\r");

delay(1000);

//The text of the message to be sent.

sgsm.print("https://www.google.com/maps/?q=");

sgsm.print(gpslat, 6);

sgsm.print(",");

sgsm.print(gpslon, 6);

delay(1000);

sgsm.write(0x1A);

delay(1000);

state = 1;

}

if (digitalRead(pin) == LOW) {

state = 0;

}

**}**

**Water sensor**

**Code:1**

#define Grove\_Water\_Sensor 8 // Attach Water sensor to Arduino Digital Pin 8

#define LED 9 // Attach an LED to Digital Pin 9 (or use onboard LED)

void setup() {

pinMode(Grove\_Water\_Sensor, INPUT); // The Water Sensor is an Input

pinMode(LED, OUTPUT); // The LED is an Output

}

void loop() {

/\* The water sensor will switch LOW when water is detected.

Get the Arduino to illuminate the LED and activate the buzzer

when water is detected, and switch both off when no water is present \*/

if( digitalRead(Grove\_Water\_Sensor) == LOW) {

digitalWrite(LED,HIGH);

}else {

digitalWrite(LED,LOW);

}

}

**Code:2**

const int waterSens = A0; //define water sensor  
const int led = 9;//define led to pin 9  
int waterVal; //define the water sensor value

void setup() {  
pinMode(led, OUTPUT); //set led as an output  
pinMode(waterSens, INPUT);//set water sensor as an input  
Serial.begin(9600);  //start the serial port at 9600 bauds

}

void loop() {  
  waterVal = analogRead(waterSens); //read the water sensor  
    
  Serial.println(waterVal); //print the value of the water sensor to the serial monitor  
    
if (waterVal <= 0){  
  digitalWrite(led, HIGH);//if the water sensor senses water turn the led on  
}  
else{  
  digitalWrite(led, LOW);//if it doesn't sense anything turn the led off  
}  
}

**Ultrasonic Sensor**

**Fig: 2 Ultrasonic sensor**

**Code: 1**

|  |  |
| --- | --- |
| 0  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | // defines pins numbers  const int trigPin = 9;  const int echoPin = 10;    // defines variables  long duration;  int distance;    void setup() {  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output  pinMode(echoPin, INPUT); // Sets the echoPin as an Input  Serial.begin(9600); // Starts the serial communication  }    void loop() {  // Clears the trigPin  digitalWrite(trigPin, LOW);  delayMicroseconds(2);    // Sets the trigPin on HIGH state for 10 micro seconds  digitalWrite(trigPin, HIGH);  delayMicroseconds(10);  digitalWrite(trigPin, LOW);    // Reads the echoPin, returns the sound wave travel time in microseconds  duration = pulseIn(echoPin, HIGH);    // Calculating the distance  distance= duration\*0.034/2;    // Prints the distance on the Serial Monitor  Serial.print("Distance: ");  Serial.println(distance);  } |

**Code2:**

1. /\*
2. \* Ultrasonic Sensor HC-SR04 and Arduino Tutorial
3. \*
4. \* by Dejan Nedelkovski,
5. \* www.HowToMechatronics.com
6. \*
7. \*/
8. // defines pins numbers
9. const int trigPin = 9;
10. const int echoPin = 10;
11. // defines variables
12. long duration;
13. int distance;
14. **void** setup() {
15. pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
16. pinMode(echoPin, INPUT); // Sets the echoPin as an Input
17. Serial.begin(9600); // Starts the serial communication
18. }
19. **void** loop() {
20. // Clears the trigPin
21. digitalWrite(trigPin, LOW);
22. delayMicroseconds(2);
23. // Sets the trigPin on HIGH state for 10 micro seconds
24. digitalWrite(trigPin, HIGH);
25. delayMicroseconds(10);
26. digitalWrite(trigPin, LOW);
27. // Reads the echoPin, returns the sound wave travel time in microseconds
28. duration = pulseIn(echoPin, HIGH);
29. // Calculating the distance
30. distance= duration\*0.034/2;
31. // Prints the distance on the Serial Monitor
32. Serial.print("Distance: ");
33. Serial.println(distance);
34. }