PROGRAM AND EXPLANATION

GPS INTERFACING

CODE:

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
float lat = 28.5458,lon = 77.1703; // create variable for latitude and longitude object
SoftwareSerial gpsSerial(3,4);//rx,tx
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);
TinyGPS gps; // create gps object
void setup(){
Serial.begin(9600); // connect serial
gpsSerial.begin(9600); // connect gps sensor
lcd.begin(16,2);
}
void loop(){
   while(gpsSerial.available()){ // check for gps data
   if(gps.encode(gpsSerial.read()))// encode gps data
   {
   gps.f_get_position(&lat,&lon); // get latitude and longitude
   // display position
   lcd.clear();
   lcd.setCursor(1,0);
   lcd.print("GPS Signal");
   lcd.setCursor(1,0);
   lcd.print("LAT:");
   lcd.setCursor(5,0);
   lcd.print(lat);
   lcd.setCursor(0,1);
   lcd.print(",LON:");
   lcd.setCursor(5,1);
   lcd.print(lon);
   }}
 String latitude = String(lat,6);
 String longitude = String(lon,6);
  Serial.println(latitude+";"+longitude);
  delay(1000);
}
```

GPS INTERFACING PROGRAM EXPLANATION

- First we have to include the TinyGPS++ library in our Arduino IDE.
- Open Arduino IDE and under File navigate to and upload the code. Once, the code is uploaded open the serial monitor and you will and the data are now more readable like Location: Latitude, Longitude along with Date/Time.
- The GPS Module is now integrated.

GSM INTERFACING

CODE:

```
const int Input1=8;
int State1=0;
void setup(){
  Serial.begin(9600);
  pinMode(Input1, INPUT);
}
void loop(){
State1= digitalRead(Input1);
 if(State1 == HIGH){
 sendsms();
delay(2000);
}}
void sendsms(){
 Serial.println("AT\r");
 delay(1000);
 Serial.println("AT+CMGF=1\r");
 delay(1000);
  Serial.println("AT+CMGS=\"XXXXXXXXXX\"\r");
  delay(1000);
  Serial.println("MESSAGE 1");
  delay(1000);
  Serial.println((char)26);
  delay(100);
}
```

GSM INTERFACING PROGRAM EXPLANATION

- Input1 is a constant integer type, constants won’t change and it is set to pin number 8 of Arduino. State1 is a variable, variables will change and it is assigned with 0 value**.**
- Put your setup or configuration code in the setup function, it will only run once during the startup.
- Initialize the Input1 pin as an input. Set the Baudrate of Serial communication as 9600.
- Put your main code in void loop() function to run repeatedly. Read the state of the State1 variable value and check if the switch is pressed which is connected pin number 8 of arduino.
- If it is, then it will call GSM function to send SMS to the programmed mobile number.
- This function is used to establish communication between arduino and GSM module by AT commands to send SMS to the provided mobile number.
- The GSM Module is now integrated.

GYROSCOPE SENSOR INTERFACING

CODE:

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);
#include <Wire.h>
#include <MPU6050.h>
#define period 10000
MPU6050 mpu;
int count=0;
char okFlag=0;
byte degree[8] = {
 0b00000,
 0b00110,
 0b01111,
 0b00110,
 0b00000,
 0b00000,
 0b00000,
 0b00000
};
void setup()
{
 lcd.begin(16,2);
 lcd.createChar(0, degree);
 Serial.begin(9600);
 Serial.println("Initialize MPU6050");
 while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
 {
   lcd.clear();
   lcd.print("Device not Found");
   Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
   delay(500);
 }
 count=0;
 mpu.calibrateGyro();
 mpu.setThreshold(3);
 lcd.clear();
 lcd.print("MPU6050 Interface");
 lcd.setCursor(0,1);
 lcd.print(" Circuit Digest");
 delay(2000);
 lcd.clear();
}
void loop()
{
   lcd.clear();
   lcd.print("Temperature");
   long st=millis();
   Serial.println("Temperature");
   while(millis()<st+period)
   {
     lcd.setCursor(0,1);
     tempShow();
```

```
  }

  lcd.clear();
  lcd.print("Gyro");
  delay(2000);
  st=millis();
  Serial.println("Gyro");
  while(millis()<st+period)
  {
    lcd.setCursor(0,1);
    gyroShow();
  }
  lcd.clear();
  lcd.print("Accelerometer");
  delay(2000);
  st=millis();
  Serial.println("Accelerometer");
  while(millis()<st+period)
  {
    lcd.setCursor(0,1);
    accelShow();
  }
}
void tempShow()
{
  float temp = mpu.readTemperature();
  Serial.print(" Temp = ");
  Serial.print(temp);
  Serial.println(" *C");
  lcd.clear();
  lcd.print("Temperature");
  lcd.setCursor(0,1);
  lcd.print(temp);
  lcd.write((byte)0);
  lcd.print("C");
  delay(400);
}
void gyroShow()
{
 //lcd.setCursor(0,0);
 lcd.clear();
 lcd.print(" X     Y     Z");
 Vector rawGyro = mpu.readRawGyro();
 Vector normGyro = mpu.readNormalizeGyro();
 lcd.setCursor(0,1);
 lcd.print(normGyro.XAxis,1);
 lcd.setCursor(6,1);
 lcd.print(normGyro.YAxis,1);
 lcd.setCursor(12,1);
 lcd.print(normGyro.ZAxis,1);
 Serial.print(" Xnorm = ");
 Serial.print(normGyro.XAxis);
 Serial.print(" Ynorm = ");
 Serial.print(normGyro.YAxis);
 Serial.print(" Znorm = ");
 Serial.println(normGyro.ZAxis);
 delay(200);
}
void accelShow()
{
// lcd.setCursor(0,0);
 lcd.clear();
 lcd.print(" X     Y     Z");
 Vector rawAccel = mpu.readRawAccel();
 Vector normAccel = mpu.readNormalizeAccel();
 lcd.setCursor(0,1);
 lcd.print(normAccel.XAxis,1);
 lcd.setCursor(6,1);
 lcd.print(normAccel.YAxis,1);
 lcd.setCursor(12,1);
 lcd.print(normAccel.ZAxis,1);
 Serial.print(" Xnorm = ");
 Serial.print(normAccel.XAxis);
 Serial.print(" Ynorm = ");
 Serial.print(normAccel.YAxis);
 Serial.print(" Znorm = ");
 Serial.println(normAccel.ZAxis);
 delay(200);
}
```

GYROSCOPE SENSOR INTERFACING PROGRAM EXPLANATION

- Here we have used this MPU6050 library to interface it with Arduino. So first of all, we need to download the MPU6050 library from GitHub and install it in Arduino IDE.
- In coding, we have included some required libraries like MPU6050 and LCD.
- In setup function, we initialize both devices and write welcome message over LCD void setup()
- In loop Function, we have called three functions in every 10seconds for displaying temperature, gyro, and accelerometer reading on LCD.
- MPU6050 gyro and accelerometer both are used to detect the position and orientation of any device. Gyro uses earth gravity to determine the x,y and z- axis positions and accelerometer detects based on the rate of the change of movement.