# Circuit Documentation

## Summary

This circuit integrates various components with an Arduino UNO microcontroller to perform multiple functions. It includes sensors for environmental monitoring (BMP180, DHT11, Smoke sensor, Soil Moisture Sensor, and Photocell), a SIM900a GSM module for communication, and an LCD I2C Display for output. The circuit is powered by a 9V battery, regulated to 5V using a 7805 voltage regulator, and includes capacitors for voltage smoothing. The Arduino UNO serves as the central processing unit, interfacing with the sensors and modules to collect data, control outputs, and communicate with external devices.

## Component List

* **Arduino UNO**: A microcontroller board based on the ATmega328P, with a variety of digital and analog I/O pins.
* **BMP180 Breakout**: A digital barometric pressure sensor that can be used to measure temperature, pressure, and altitude.
* **9V Battery**: Provides power to the circuit.
* **7805 Voltage Regulator**: A voltage regulator that outputs a stable 5V from a higher voltage input.
* **Soil Moisture Sensor**: Measures the moisture level in the soil.
* **Photocell (LDR)**: A light-dependent resistor that changes resistance based on light intensity.
* **Resistor**: A 10kΩ resistor, likely used for pull-up/down configurations or voltage division.
* **SIM900a GSM Module**: A GSM/GPRS module for cellular communication.
* **Electrolytic Capacitors**: Used for decoupling and filtering noise from the power supply.
* **LCD I2C Display**: A display module for showing information, interfaced via I2C.
* **DHT11 Humidity and Temperature Sensor**: A sensor for measuring ambient temperature and humidity.
* **Smoke Sensor**: Detects the presence of smoke in the environment.
* **2.1mm Barrel Jack with Terminal Block**: Connectors for power input to the circuit.

## Wiring Details

### Arduino UNO

* **GND**: Connected to the ground pins of the SIM900a GSM module, BMP180 Breakout, LCD I2C Display, DHT11 sensor, Smoke sensor, Soil Moisture Sensor, and 7805 Voltage Regulator.
* **D11**: Connected to the 5V Rx pin of the SIM900a GSM module.
* **D10**: Connected to the 3.3 Rx pin of the SIM900a GSM module.
* **5V**: Connected to the NULL pin of the DHT11 sensor and pin 0 of the Photocell (LDR).
* **A5**: Connected to the SDA pins of the BMP180 Breakout and LCD I2C Display.
* **A4**: Connected to the SCL pins of the BMP180 Breakout and LCD I2C Display.
* **A1**: Connected to pin 1 of the Photocell (LDR).
* **A0**: Connected to the SIG pin of the Soil Moisture Sensor.

### BMP180 Breakout

* **GND**: Connected to the Arduino UNO's GND.
* **3.3V**: Not connected in the provided net list.
* **IO**: Not connected in the provided net list.
* **SCL**: Connected to the Arduino UNO's A4.
* **SDA**: Connected to the Arduino UNO's A5.

### 9V Battery

* **+**: Connected to the INPUT pin of the 7805 Voltage Regulator.
* **-**: Connected to the GND pin of the 7805 Voltage Regulator.

### 7805 Voltage Regulator

* **INPUT**: Connected to the + pin of the 9V Battery.
* **OUTPUT**: Connected to the VCC pins of the Smoke sensor, Soil Moisture Sensor, and the POS pin of the 2.1mm Barrel Jack with Terminal Block.
* **GND**: Connected to the GND pins of the Smoke sensor, Soil Moisture Sensor, and the NEG pin of the 2.1mm Barrel Jack with Terminal Block.

### Soil Moisture Sensor

* **VCC**: Connected to the OUTPUT pin of the 7805 Voltage Regulator.
* **GND**: Connected to the Arduino UNO's GND.
* **SIG**: Connected to the Arduino UNO's A0.

### Photocell (LDR)

* **pin 0**: Connected to the 5V pin of the Arduino UNO.
* **pin 1**: Connected to the A1 pin of the Arduino UNO.

### Resistor

* **pin1**: Not connected in the provided net list.
* **pin2**: Connected to the GND pin of the Arduino UNO and the GND pin of the DHT11 sensor.

### SIM900a GSM Module

* **GND**: Connected to the Arduino UNO's GND.
* **RS 232 Receive**: Not connected in the provided net list.
* **RS 232 Transmit**: Not connected in the provided net list.
* **VCC MCU**: Not connected in the provided net list.
* **5V Tx**: Not connected in the provided net list.
* **5V Rx**: Connected to the D11 pin of the Arduino UNO.
* **3.3 TX**: Not connected in the provided net list.
* **3.3 Rx**: Connected to the D10 pin of the Arduino UNO.

### Electrolytic Capacitors

* **-**: Connected to the GND of the circuit.
* **+**: Connected to the VCC of the circuit.

### LCD I2C Display

* **GND**: Connected to the Arduino UNO's GND.
* **VCC**: Not connected in the provided net list.
* **SDA**: Connected to the Arduino UNO's A5.
* **SCL**: Connected to the Arduino UNO's A4.

### DHT11 Humidity and Temperature Sensor

* **VDD**: Not connected in the provided net list.
* **DATA**: Not connected in the provided net list.
* **NULL**: Connected to the 5V pin of the Arduino UNO.
* **GND**: Connected to the Arduino UNO's GND.

### Smoke Sensor

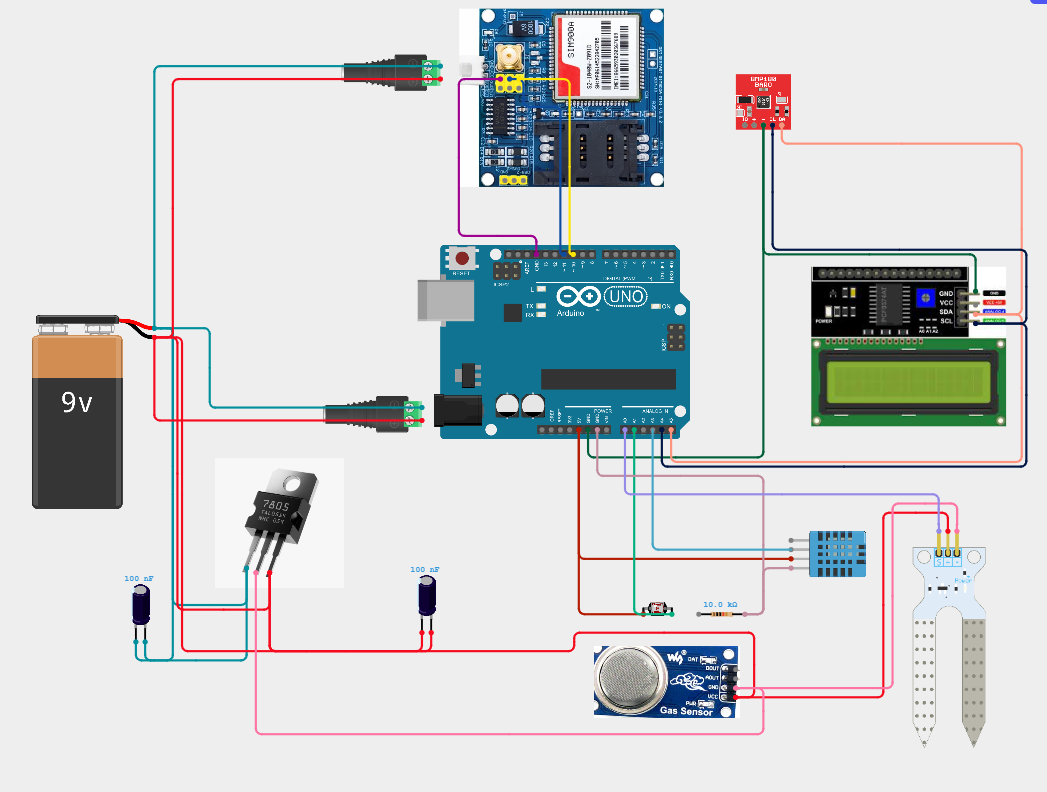
* **DO**: Not connected in the provided net list.
* **AO**: Not connected in the provided net list.
* **GND**: Connected to the Arduino UNO's GND.
* **VCC**: Connected to the OUTPUT pin of the 7805 Voltage Regulator.

### 2.1mm Barrel Jack with Terminal Block

* **POS**: Connected to the OUTPUT pin of the 7805 Voltage Regulator.
* **NEG**: Connected to the GND of the circuit.Code

## 

This code is a template and does not contain any functional instructions. It needs to be populated with initialization code in the setup() function and operational code in the loop() function to interact with the connected components and perform the desired tasks.



*//-----Electronics-project-hub>com------//*

***#include <SoftwareSerial.h>***

***#include <Wire.h>***

***#include <LiquidCrystal\_I2C.h>***

***#include <dht.h>***

***#include <Adafruit\_BMP085.h>***

Adafruit\_BMP085 bmp;

dht DHT;

LiquidCrystal\_I2C **lcd**(**0x27**, **16**, **2**);

SoftwareSerial **gsm**(**10**, **11**); *// RX, TX*

***#define DHT11\_PIN A3***

**int** chk;

**int** humi = **0**;

**int** temp = **0**;

**int** soil = **0**;

**int** light = **0**;

**int** BMP = **0**;

**int** gas = **0**;

boolean HT;

**void** **setup**()

{

gsm.begin(**9600**);

pinMode(A0, INPUT);

pinMode(A1, INPUT);

pinMode(A2, INPUT);

pinMode(A3, INPUT);

lcd.init();

lcd.backlight();

lcd.setCursor(**0**, **0**);

lcd.print("Please wait for");

lcd.setCursor(**0**, **1**);

lcd.print("60 seconds.");

delay(**20000**);

delay(**20000**);

delay(**20000**);

modem\_init();

data\_init();

internet\_init();

lcd.clear();

}

**void** **loop**()

{

chk = DHT.read11(DHT11\_PIN);

temp = DHT.temperature;

humi = DHT.humidity;

soil = analogRead(A0);

light = analogRead(A1);

gas = analogRead(A2);

BMP = bmp.readPressure();

lcd.clear();

lcd.setCursor(**0**, **0**);

lcd.print("Soil:");

soil = map(soil, **0**, **1023**, **100**, **0**);

lcd.print(soil);

lcd.print("%");

lcd.setCursor(**0**, **1**);

lcd.print("Light:");

light = map(light, **0**, **1023**, **0**, **100**);

lcd.print(light);

lcd.print("%");

delay(**3000**);

lcd.clear();

lcd.setCursor(**0**, **0**);

**switch** (chk)

{

**case** DHTLIB\_OK:

HT = true;

**break**;

default:

HT = false;

**break**;

}

**if** (HT == true)

{

lcd.print("Temp:");

lcd.print(temp);

lcd.print(" \*C");

lcd.setCursor(**0**, **1**);

lcd.print("Humidity:");

lcd.print(humi);

lcd.print("%");

}

**else**

{

temp = **0**;

humi = **0**;

lcd.print("Temp:");

lcd.print("No Data");

lcd.setCursor(**0**, **1**);

lcd.print("Humidity:");

lcd.print("No Data");

}

delay(**3000**);

lcd.clear();

lcd.setCursor(**0**, **0**);

lcd.print("Air Qlt: ");

gas = map(gas, **0**, **1023**, **0**, **100**);

lcd.print(gas);

lcd.print("%");

lcd.setCursor(**0**, **1**);

lcd.print("Pressure:");

**if** (!bmp.begin())

{

lcd.print("No Data");

BMP = **0**;

}

**else**

{

lcd.print(BMP);

}

lcd.print("Pa");

delay(**3000**);

Send\_data();

}

**void** **modem\_init**()

{

Serial.println("Please wait.....");

gsm.println("AT");

delay(**1000**);

gsm.println("AT+CMGF=1");

delay(**1000**);

gsm.println("AT+CNMI=2,2,0,0,0");

delay(**1000**);

}

**void** **data\_init**()

{

Serial.println("Please wait.....");

gsm.println("AT");

delay(**1000**); delay(**1000**);

gsm.println("AT+CPIN?");

delay(**1000**);

delay(**1000**);

gsm.print("AT+SAPBR=3,1");

gsm.write(',');

gsm.write('"');

gsm.print("contype");

gsm.write('"');

gsm.write(',');

gsm.write('"');

gsm.print("GPRS");

gsm.write('"');

gsm.write(**0x0d**);

gsm.write(**0x0a**);

delay(**1000**); ;

gsm.print("AT+SAPBR=3,1");

gsm.write(',');

gsm.write('"');

gsm.print("APN");

gsm.write('"');

gsm.write(',');

gsm.write('"');

gsm.print("bsnlnet"); gsm.write('"');

gsm.write(**0x0d**);

gsm.write(**0x0a**);

delay(**1000**);

gsm.print("AT+SAPBR=3,1");

gsm.write(',');

gsm.write('"');

gsm.print("USER");

gsm.write('"');

gsm.write(',');

gsm.write('"');

gsm.print(" ");

gsm.write('"');

gsm.write(**0x0d**);

gsm.write(**0x0a**);

delay(**1000**);

gsm.print("AT+SAPBR=3,1");

gsm.write(',');

gsm.write('"');

gsm.print("PWD");

gsm.write('"');

gsm.write(',');

gsm.write('"');

gsm.print(" ");

gsm.write('"');

gsm.write(**0x0d**);

gsm.write(**0x0a**);

delay(**2000**);

gsm.print("AT+SAPBR=1,1");

gsm.write(**0x0d**);

gsm.write(**0x0a**);

delay(**3000**);

}

**void** **internet\_init**()

{

Serial.println("Please wait.....");

delay(**1000**);

gsm.println("AT+HTTPINIT");

delay(**1000**); delay(**1000**);

gsm.print("AT+HTTPPARA=");

gsm.print('"');

gsm.print("CID");

gsm.print('"');

gsm.print(',');

gsm.println('1');

delay(**1000**);

}

**void** **Send\_data**()

{

lcd.clear();

lcd.print("Sending the data");

lcd.setCursor(**0**, **1**);

lcd.print("to Thingspeak...");

delay(**1500**);

gsm.print("AT+HTTPPARA=");

gsm.print('"');

gsm.print("URL");

gsm.print('"');

gsm.print(',');

gsm.print('"');

gsm.print("http:");

gsm.print('/');

gsm.print('/');

gsm.print("api.thingspeak.com/update?api\_key=xxxxxxxxxxxxx&field1=");

gsm.print(soil);

gsm.print("&field2=");

gsm.print(light);

gsm.print("&field3=");

gsm.print(gas); *// >>>>>> variable 3*

gsm.print("&field4=");

gsm.print(temp); *// >>>>>> variable 4*

gsm.print("&field5=");

gsm.print(humi); *// >>>>>> variable 5*

gsm.print("&field6=");

gsm.print(BMP); *// >>>>>> variable 6*

gsm.write(**0x0d**);

gsm.write(**0x0a**);

delay(**1000**);

gsm.println("AT+HTTPACTION=0");

delay(**1000**);

}