<https://makeradvisor.com/10-iot-development-boards/>

|  |  |
| --- | --- |
|  |  |

**#include <SoftwareSerial.h>**

**//Create software serial object to communicate with SIM800**

SoftwareSerial **SIM800**( 8, 7 ); **// TX =8, RX = 7**

**void setup() {**

  Serial.**begin**(9600);

  while(!Serial);

**SIM800**.**begin**(9600);

  delay(1000);

}

RECIBIR SMS

if(**SIM800**.available()){

    Serial.write(**SIM800**.read());

  }

char SMS\_In[1000];  
int i = 0;  
while (**SIM800**.available() > 0 && i<1000)  
  {  
    SMS\_In[i++] = **SIM800**.read();     
  }

ENVIAR SMS

**//Set SMS ASCII**

**SIM800**.write("**AT+CMGF=1\r\n**");

delay(1000);

**SIM800**.write("**AT+CMGS=\"+50370020016\"\r\n**");

delay(1000);

**SIM800**.write("**TEST**");

delay(1000);

**//Send Ctrl+Z / ESC de finaizado**

**SIM800**.write((char)26);

delay(1000);

Librearia Seeeduino\_GPRS

Note : La libreria assume que TX esta conectado a D8 y RX a D7 al Arduino.

* Clear steps relevant to installing Arduino library is available at : <https://www.arduino.cc/en/Guide/Libraries>
* Seeeduino\_GPRS library is available for download at : <https://github.com/Seeed-Studio/Seeeduino_GPRS>

Once library is installed in Arduino IDE File menu, Examples section you will find “Seeeduino\_GPRS” library and withing examples you will find “GPRS\_SendSMS” example which reads as follows :

note: the following pins has been used and should not be used for other purposes.

  pin 8   // tx pin

  pin 7   // rx pin

  pin 9   // power key pin

  pin 12  // power status pin

by lawliet.zou(lawliet.zou@gmail.com)

\*/

#include <gprs.h>;

#include <SoftwareSerial.h>;

GPRS gprs;

void setup() {

  Serial.begin(9600);

  while(!Serial);

  Serial.println("GPRS - Send SMS Test ...");

  gprs.preInit();

  delay(1000);

  while(0 != gprs.init()) {

      delay(1000);

      Serial.print("init error\r\n");

  }

  gprs.**sendSMS**("07194XXXXX","hello,world");

}

void loop() {

  //nothing to do

}

If you go through Seeeduino library you will notice that it is possible to send commands directly for any advanced use cases. For examples there are library methods such as :

* sendCmdAndWaitForResp(const char\* cmd, const char \*expectedResp, unsigned timeout)
* sendCmd(const char\* cmd)
* waitForResp(const char \*resp, unsigned int timeout)

Hence, you could simply correctly rewrite the SMS sending application as below (reinvent the wheel) :

[?](https://www.ayomaonline.com/programming/quickstart-sim800-sim800l-with-arduino/)

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14 | if(0 != gprs.sendCmdAndWaitForResp("AT+CMGF=1\r\n", "OK", DEFAULT\_TIMEOUT)) { // Set message mode to ASCII      ERROR("ERROR:CMGF");      return;  }  delay(500);  if(0 != gprs.sendCmdAndWaitForResp("AT+CMGS=\"07194XXXXX\"\r\n","&gt;",DEFAULT\_TIMEOUT)) {      ERROR("ERROR:CMGS");      return;  }  delay(1000);  gprs.serialSIM800.write(data);  delay(500);  gprs.serialSIM800.write((char)26); |