<https://randomnerdtutorials.com/esp32-sim800l-send-text-messages-sms/>

SEND SMS

https://circuitdigest.com/microcontroller-projects/sending-sms-using-esp8266

<https://www.engineersgarage.com/esp8266/send-text-message-with-nodemcu/>



INSTALAR

* Insertar SIM de teléfono
* Librería ESP32
* Librería Wire
* Librería TinyGsmClient

#include <Wire.h>

#include <TinyGsmClient.h>

void setup() {

int PW\_KEY 4; // pin modem enable

int RST 5; // pin modem reset

int POWER\_ON 23; // pin modem power

pinMode(PW\_KEY, OUTPUT);

pinMode(RST, OUTPUT);

pinMode(POWER\_ON, OUTPUT);

digitalWrite(PW\_KEY, LOW);

digitalWrite(RST, HIGH);

digitalWrite(POWER\_ON, HIGH);

// Set GSM module baud rate and UART pins

int RX = 26;

int RX = 27;

SerialAT.begin(115200, SERIAL\_8N1, RX, TX);

delay(3000);

modem.restart(); // or modem.init();

}

**Enviar SMS**

if(modem.sendSMS("+351XXXXXXXXX", "Enviando mensaje"));

{

Serial.println("Enviando");

}

**Recibir SMS**

String str3="";

while(modem.available()) {

char c = Serial.read();

str3 += c;

}

Cargar código

* Escoja en **Tools**> **Board**y seleccione **ESP32 Dev module**
* Seleccione el Puerto con **Tools**> **Port**y selecciona el COM al ESP32
* Presiona botón **Upload**

RECEIVING

<http://www.jamesrobertson.eu/arduino/2019/aug/31/receiving-sms-using-the-ttgo-t-call.html>

#include **<Arduino.h>**

#include **<Wire.h>**

#include **<sim800.h>**

#include **<WiFi.h>**

#include **<gprs.h>**

#define T\_CALL

#if defined(T\_CALL)

#define UART\_TX 27

#define UART\_RX 26

#define SIMCARD\_RST 5

#define SIMCARD\_PWKEY 4

#define SIM800\_POWER\_ON 23

#else

#define UART\_TX 33

#define UART\_RX 34

#define SIMCARD\_RST 14

#define SIMCARD\_PWKEY 15

#define SIM800\_POWER\_ON 4

#endif

#define UART\_BANUD\_RATE 9600

#define I2C\_SDA 21

#define I2C\_SCL 22

#define IP5306\_ADDR 0X75

#define IP5306\_REG\_SYS\_CTL0 0x00

HardwareSerial hwSerial(1);

GPRS gprs(hwSerial, SIMCARD\_PWKEY, SIMCARD\_RST, SIM800\_POWER\_ON);

**bool** setPowerBoostKeepOn(**int** en)

{

Wire.beginTransmission(IP5306\_ADDR);

Wire.write(IP5306\_REG\_SYS\_CTL0);

**if** (en)

Wire.write(0x37); // Set bit1: 1 enable 0 disable boost keep on

**else**

Wire.write(0x35); // 0x37 is default reg value

**return** Wire.endTransmission() == 0;

}

**void** wifi\_test()

{

WiFi.mode(WIFI\_STA);

WiFi.disconnect();

**int** n = WiFi.scanNetworks();

**if** (n == 0) {

Serial.println("no networks found");

**return**;

}

Serial.println("Found " + String(n) + " networks");

}

**void** setup()

{

Serial.begin(115200);

#if defined(T\_CALL)

Wire.begin(I2C\_SDA, I2C\_SCL);

**bool** isOk = setPowerBoostKeepOn(1);

String info = "IP5306 KeepOn " + String((isOk ? "PASS" : "FAIL"));

Serial.println(info);

#endif

hwSerial.begin(UART\_BANUD\_RATE, SERIAL\_8N1, UART\_RX, UART\_TX);

Serial.println("Setup Complete!");

wifi\_test();

**if** (gprs.preInit()) {

Serial.println("SIM800 Begin PASS");

} **else** {

Serial.println("SIM800 Begin FAIL");

}

Serial.println("Test motor ...");

**int** i = 3;

**while** (i--) {

hwSerial.print("AT+SPWM=0,1000,1\r\n");

delay(2000);

hwSerial.print("AT+SPWM=0,0,0\r\n");

}

delay(200);

**if** (0 != gprs.init()) {

Serial.println("Not checked out to SIM card...");

**while** (1);

}

// Switch audio channels

hwSerial.print("AT+CHFA=1\r\n");

delay(2);

hwSerial.print("AT+CRSL=100\r\n");

delay(2);

hwSerial.print("AT+CLVL=100\r\n");

delay(2);

hwSerial.print("AT+CLIP=1\r\n");

delay(2);

Serial.println("Init success");

}

**bool** isCallIn = false;

**void** loop()

{

**if** (hwSerial.available()) {

**const** **char** \*s = hwSerial.readStringUntil('\n').c\_str();

**if** (strstr(s, "OK" ) != NULL) {

Serial.println("SIM OK");

} **else** **if** (strstr(s, "+CPIN: NOT READY") != NULL) {

Serial.println("SIM +CPIN: NOT READY");

} **else** **if** (strstr(s, "+CPIN: READY") != NULL) {

Serial.println("SIM +CPIN: READY");

} **else** **if** (strstr(s, "+CLIP:") != NULL) {

Serial.printf("SIM %s\n", s);

} **else** **if** (strstr(s, "RING") != NULL) {

delay(200);

hwSerial.write("ATA\r\n");

Serial.println("SIM RING");

} **else** **if** (strstr(s, "Call Ready") != NULL) {

Serial.println("SIM Call Ready");

} **else** **if** (strstr(s, "SMS Ready") != NULL) {

Serial.println("SIM SMS Ready");

} **else** **if** (strstr(s, "NO CARRIER") != NULL) {

Serial.println("SIM NO CARRIER");

} **else** **if** (strstr(s, "NO DIALTONE") != NULL) {

Serial.println("SIM NO DIALTONE");

} **else** **if** (strstr(s, "BUSY") != NULL) {

Serial.println("SIM BUSY");

} **else** {

Serial.print(s);

}

Serial.println("==========");

}

**if** (Serial.available()) {

String r = Serial.readString();

hwSerial.write(r.c\_str());

}

}