# Modulo SIM900

Derek Yair Curtidor Contreras
316030032@upjr.edu.mx

12 de marzo de 2019

## 1. Primeros pasos

Esta tarjeta está basada en el módulo SIM900, y la configuraremos y controlaremos vía UART utilizando los comandos AT.
Antes de conectar nada al Arduino, vamos a colocar los jumpers que tiene la tarjeta de forma que utilicemos para comunicarnos los pines 7 y 8.
Se introduce una tarjeta nano sim en el módulo.

Para poder alimentar a los módulos es necesario utilizar una fuente externa, ya que, de lo contrario, si alimentamos únicamente con el cable USB alimentaremos solo a los módulos, mas no a la tarjeta SIM, en mi caso utilicé una fuente de alimentación de 2V a 1.5A.

Por recomendación, muchas páginas de internet recomiendan que se alimenten ambos módulos conectando la fuente de alimentación al Arduino, en mi caso esto no funcionaba por motivo que aun desconozco, así que la única opción que se encontró fue alimentar el Arduino mediante el cable USB y el módulo GPRS con la fuente externa.
Para empezar a familiarizarme con el módulo sim900 se subió al Arduino un código sencillo, en el cual por medio del monitor serial, enviaríamos comandos AT para hacer funcionar el módulo, por ejemplo hacer llamadas, enviar mensajes, verificar el estado de la red, etc.
El primer diseño que se utilizo fue el siguiente (El software donde se hicieron los diseños (Fritzing) no tiene el módulo sim900 por lo que se utilizó un cellular shield, por su similitud con el módulo GPRS):

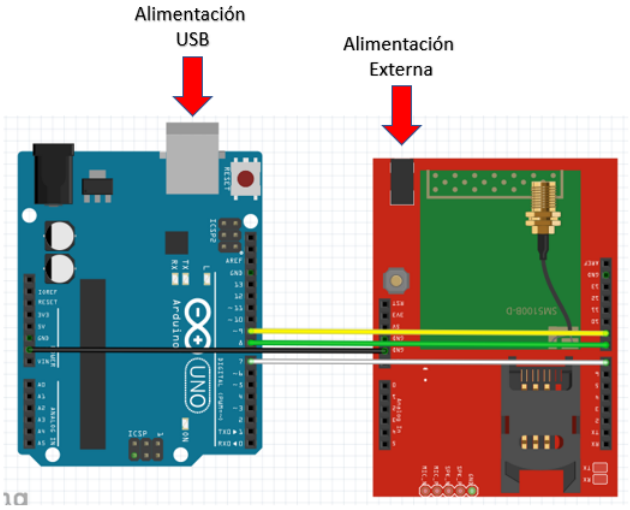


Figura 1: Diseño simple para pruebas del SIM900

El código para empezar a hacer pruebas con los comandos AT es el siguiente:

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8); //Seleccionamos los pines 7 como Rx y 8 como Tx

void setup()
{
 SIM900.begin(19200);
 Serial.begin(19200);
 delay(1000);
}

void loop()
{
 //Enviamos y recibimos datos
 if (Serial.available() > 0)
 SIM900.write(Serial.read());
 if (SIM900.available() > 0)
 Serial.write(SIM900.read());
}
```

## 1.1. Problemas presentados y consideraciones

- Como se mencionó arriba muchas páginas de internet mencionan que es recomendable alimentar ambas tarjetas desde el mismo Arduino, en este caso al hacerlo de dicha manera presentaba el problema de que la tarjeta SIM no encendía, o no siempre, por lo que la solución fue alimentar el módulo GPSR directamente con la fuente externa.

- A la hora de hacer pruebas desde el monitor serial hay que checar que la velocidad a la que se trabaja sea la misma en la que se especifica el programa, en nuestro caso 19200, y que la impresión este en modo retorno de carro".

## 2. Subir datos de Sensor DS18B20 a ThingSpeak

Una vez que entendida la naturaleza del módulo sim900 y resueltos los primeros conflictos presentados, fue hora de subir los primeros datos censados a un servidor. Por comodidad y para primeros ejemplos, se utilizó la página "ThingSpeak"la cual se orienta a IoT y es cómoda y fácil de usar. Se creo un nuevo çanal.[en] el cual se testearían los primeros ejemplos del sensor.

Una vez creado el canal al cual se subirían los datos, se descargó un código ejemplo de internet, el cual mediante códigos AT hace la conexión a la página ThingSpeak, lee los datos del sensor, y sube dichos datos a la plataforma, los cuales se muestran automáticamente en una gráfica. El sensor utilizado es un sensor digital de temperatura ( DS18B20). El diseño para el primer censado de datos es el siguiente:

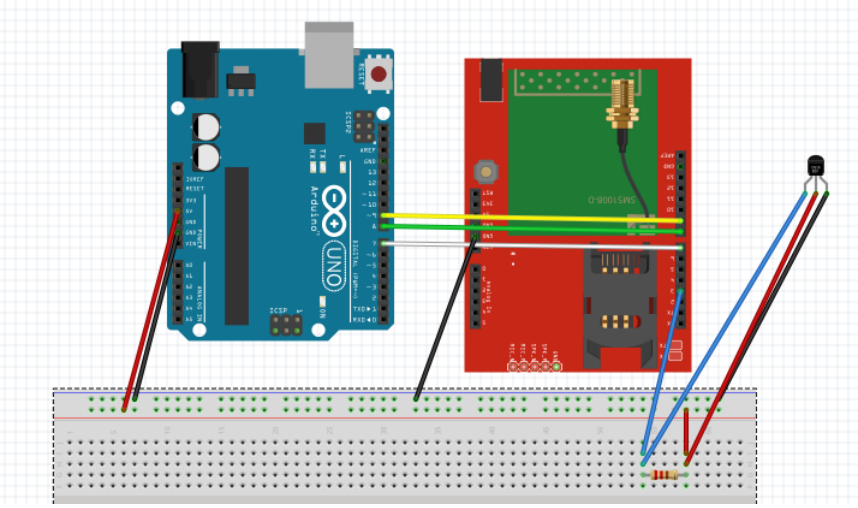

Figura 2: Diseño del primer circuito para subir datos a un servidor

El codigo que se cargó al Arduino fue el siguiente:

```
#include <SoftwareSerial.h>
#include <String.h>
SoftwareSerial Sim900Serial(7, 8);//Configaracion de los pines serial por software
// Incluya las bibliotecas que necesitamos para controlar el sensor ds18b20
#include <DallasTemperature.h>
#include <OneWire.h>
// El cable de datos esta conectado al puerto 4 del Arduino
#define ONE_WIRE_BUS 4
//Configure una instancia oneWire para comunicarse con cualquier dispositivo OneWire
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float temperatura=0.00;
void setup(){
  Sim900Serial.begin(19200);//Arduino se comunica con el SIM900 a una velocidad de 19200bps
  Serial.begin(19200);//Velocidad del puerto serial de arduino
  sensors.begin();//Inicializamos el sensor de temperatura
```

```
  //Encendido del modulo por sofware
  digitalWrite(9, HIGH);
  delay(1000);
  digitalWrite(9, LOW);
  delay(40000);//Tiempo prudencial para el escudo inicie sesion de red con tu operador
}

void loop(){
  comandosAT();//Llama a la funcion comandosAT
  if(Sim900Serial.available())//Verificamos si hay datos disponibles desde el SIM900
  Serial.write(Sim900Serial.read());//Escribir datos
}

void comandosAT(){
  Sim900Serial.println("AT+CIPSTATUS");//Consultar el estado actual de la conexion
  delay(2000);
  Sim900Serial.println("AT+CIPSTATUS");//Consultar el estado actual de la conexion
  delay(2000);
  Sim900Serial.println("AT+CIPMUX=0");//configurar el dispositivo para una conexion IP unica o multiple 0=unica
  delay(3000);
  mostrarDatosSeriales();
  Sim900Serial.println("AT+CSTT=\"internet.itelcel.com\",\"webgprs\",\"webgprs2002\"");
  //comando configura el APN, nombre de usuario y contrasenia."gprs.movistar.com.ar","wap","wap"->Movistar Arg.
  delay(1000);
  mostrarDatosSeriales();
  Sim900Serial.println("AT+CIICR");//REALIZAR UNA CONEXION INALAMBRICA CON GPRS O CSD
  delay(3000);
  mostrarDatosSeriales();
  Sim900Serial.println("AT+CIFSR");// Obtenemos nuestra IP local
  delay(2000);
  mostrarDatosSeriales();
  Sim900Serial.println("AT+CIPSPRT=0");//Establece un indicador '>' al enviar datos
  grados();
  delay(3000);
  mostrarDatosSeriales();
  Sim900Serial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"");
  //Indicamos el tipo de conexion, url o direccion IP y puerto al que realizamos la conexion
  delay(6000);
  mostrarDatosSeriales();
  Sim900Serial.println("AT+CIPSEND");//ENVIA DATOS A TRAVES DE una CONEXION TCP O UDP
  delay(4000);
  mostrarDatosSeriales();
  String datos="GET https://api.thingspeak.com/update?api_key=TXIHR2QE5W0IQDKZ&field1=0" + String(temperatura);
  Sim900Serial.println(datos);//Envia datos al servidor remoto
  delay(4000);
  mostrarDatosSeriales();
  Sim900Serial.println((char)26);
  delay(5000);//Ahora esperaremos una respuesta pero esto va a depender de las condiones de la red y este valor quiza debamos modific
  Sim900Serial.println();
  mostrarDatosSeriales();
  Sim900Serial.println("AT+CIPSHUT");//Cierra la conexion(Desactiva el contexto GPRS PDP)
  delay(5000);
  mostrarDatosSeriales();
}

void mostrarDatosSeriales(){//Muestra los datos que va entregando el sim900
  while(Sim900Serial.available()!=0)
  Serial.write(Sim900Serial.read());
}

void grados(){//Funcion para la lectura del sensor de temepratura
  sensors.requestTemperatures(); // Envia el comando para obtener temperaturas
  temperatura=sensors.getTempCByIndex(0);/// getTempCByIndex(0) se refiere al primer
  //sensor si es que tubieramos mas de uno conectado en el cable
  Serial.print("La temperatura es: ");
  Serial.print(temperatura);//Imprime la temperatura
  Serial.println(" Grados centigrados");
}
```

## 2.1. Resultados del primer sensado de datos

A continuación se muestra cómo es que se grafican los datos en la página de ThingSpeak asi como los resultados impresos en el monitor serial:

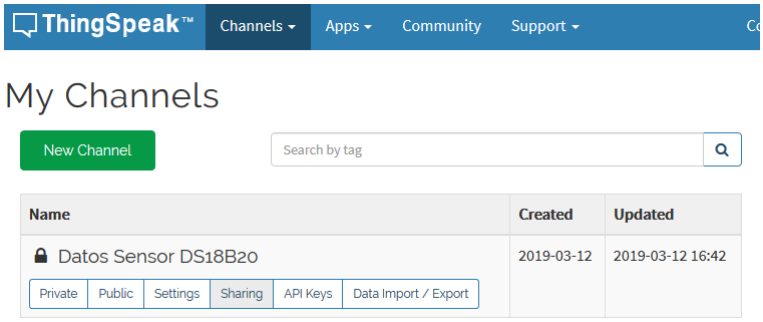


Figura 3: Canal creado para almacenar los datos del sensor DS18B20

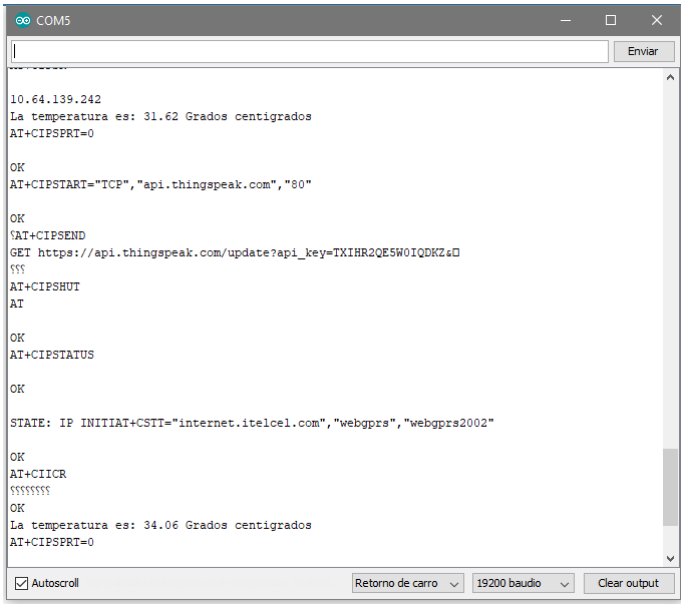


Figura 4: Impresiones del monitor serial

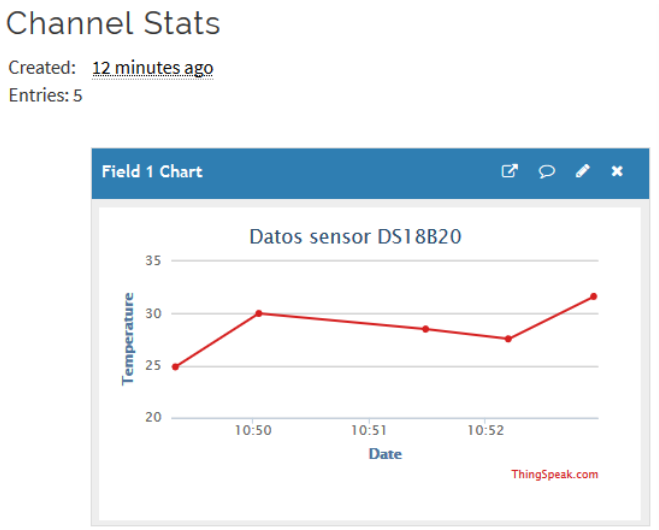


Figura 5: Gráfica con los valores obtenidos