

Apéndice D – Código para el Mote GSM

Este documento contiene en detalle la totalidad del código que se usó para configurar el Mote con comunicación GSM, este Mote como característica principal tiene que los datos son enviados a través de la red celular del chip del prestador de servicios de Telecomunicaciones que se le inserte a la placa Linkit ONE, como sabemos este chip debe tener habilitada una línea con disponibilidad de consumo de datos, para la operación de este Mote entonces fue necesario activar un plan de consumo de datos del operador de telefonía móvil Movistar, la trama de datos que es enviada por el Mote hacia el portal Web Ubidots, es de un tamaño en aproximadamente 1 Kb por lo cual no tiene problemas con el ancho de banda que consume de la red celular.

El código completo se puede encontrar en el archivo con extensión (.INO) llamado (Mote GSM), ahí se encuentra en su totalidad de cada una de las líneas de código necesarias para correr el programa y cargarlo a la placa Linkit One.

El código inicia incluyendo las librerías necesarias, las librerías son otros programas o scripts con códigos para realizar una tarea definida, en este caso las librerías que se van a utilizar son para traer el código de funcionamiento base de la conexión a la red de Telefonía Móvil y la de los sensores de temperatura y humedad y el de luz solar. En el código se encuentra descrito como se puede apreciar en la *Figura 1*.

```

#include <LGPRS.h>          //include the base GPRS library
#include <LGPRSClient.h>    //include the ability to Post and Get information using HTTP
#include <LGPRSUdp.h>
#include <LTask.h>
#include <LBattery.h>
#include <Wire.h>
#include "Arduino.h"
//#include "SI114X.h"
#include <TH02_dev.h>
#include "LDHT.h"
#include <Digital_Light_TSL2561.h>

```

Figura 1. Inclusión de Librerías

De la misma manera se definen constantes que van a ser utilizadas en el desarrollo del código. Para este caso se define la identificación de los puertos de conexión de los sensores y la asignación de la identificación de las variables que van a ser llevadas a Ubidots y la identificación del usuario que se crea. La definición de las constantes se cita después de la inclusión de las librerías, en la *Figura 2* se puede observar cómo quedan designadas en el código.

```

#define DHTPIN 6           // what pin we're connected to
#define DHTTYPE DHT11      // using DHT11 sensor
#define WEBSITE "things.ubidots.com"
#define URL "/api/v1.6/collections/values/?token=BBFF-WDXj18XdQjfyqxxqs8cX4rrByXlpKy&force=true"
#define idvariable1 "5ad56f47642ab62d1469a761" //temperatura
#define idvariable2 "5ad56f6a642ab62c9796cd7b" //humedad
#define idvariable3 "5ad56f88642ab62d50ae0bea" // Vis
#define idvariable4 "5ad56fa3642ab62d50ae0bfc" //UV
#define idvariable5 "5ad56fc0642ab62cd28bec22" //temperatura cristal
#define idvariable6 "5ad56fda642ab62d50ae0c13" // bateria
#define N 5 // cuantos datos toma por ciclo
#define CICLOMAX 3
#define DELAYLOOP 14000 //cada cuanto toma dato

//valor real delayloop 594000
boolean exito = false;
int counter = 0;
int counter1=0;
LDHT dht(DHTPIN,DHTTYPE);
float tempC=0.0,Humi=0.0, Vis=0.0, IR=0.0,UV_1=0.0,temp1=0.0, temp2=0.0 ;
float sum_1=0;
//SI114X SI1145 = SI114X();

char payload[360]; // Reserve a char to store the Ubidots data. Account for 60 bytes per variable.
char le[4];
String response;
int temp_p;

```

Figura 2. Definición de Variables y Constantes

Una vez se tienen definidos los parámetros de las constantes y demás librerías que van a ser utilizadas en el cuerpo del código, se necesita declarar la configuración de la velocidad del puerto serial, y demás configuración con respecto a los puertos de configuración. Esto se puede ver reflejado en la *Figura 3*.

```
void setup()
{
  LTask.begin();
  Serial.begin(19200);           // setup Serial port
  pinMode(13, OUTPUT);
  dht.begin();
  delay(1000);
  Wire.begin();
  TSL2561.init();

  Serial.println("Attach to GPRS network"); // Attach to GPRS network - need to add timeout
  while (!LGPRS.attachGPRS("web.colombiamovil.com.co","", "")) {
    delay(2000);
  }
  Serial.println("GPRS attached!");
  delay(10000);
}
```

Figura 3. Ciclo de Configuración

La descripción de ciclo que va a realizar el procesamiento de los datos provenientes de los sensores, así como la presentación de los valores para cada una de las variables que se quieren traer de los sensores, al igual que la encapsulación de los valores en las variables y la definición de la estructura de como llegan los datos a Ubidots para que puedan ser presentados de forma acertada.