

Apéndice C - Código para el Mote con Almacenamiento Interno

En este documento se encuentra el detalle del código para el Mote que se diseñó para realizar el almacenamiento interno de datos en la memoria Micro-SD, a diferencia del código detallado en los anexos b y d, para este caso no va a ser necesario ningún tipo de antena ya que no va a realizarse ningún tipo de conexión inalámbrica. Este Mote fue desarrollado en respuesta al incidente de tener ventanas de tiempo en las que se perdían los datos ya que los Mote con comunicación inalámbrica no estaban realizando el envío de la trama de datos, para poder cumplir con el objetivo de realizar un almacenamiento continuo de datos sin ningún tipo de pausa en su actividad debida a temas diferentes a alimentación energética.

En el Anexo B se encuentra en detalle el paso a paso de la instalación de los requerimientos para que la placa Linkit ONE pueda trabajarse en el entorno de desarrollo de Arduino IDE bajo una computadora con sistema operativo Windows.

El código completo se puede encontrar en el archivo con extensión (.INO) llamado (Mote Almacenamiento SD), 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 del almacenamiento en la tarjeta Micro-SD y la de los sensores de temperatura y humedad y el de luz solar. En el código se encuentra de la inclusión de las librerías descrito se puede apreciar en la *Figura 1*.

```

#include <LTask.h>
#include <LBattery.h>
#include <Wire.h>
#include "Arduino.h"
#include "SI114X.h"
#include <TH02_dev.h>
#include "LDHT.h"
#include <LSD.h>
#include "RTCLib.h"
#include <LDateTime.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 las Constantes para la identificación de los puertos de la placa Linkit ONE que van a utilizar cada uno de los dos sensores utilizados en este Mote, igualmente se puede encontrar la declaración del tipo de datos que van a tener las variables encargadas de representar los contadores. 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 N 5
#define CICLOMAX 3
#define TIEMPOMAX 300000

datetimeInfo t;
unsigned int rtc1;

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=0.0,temp1=0.0, temp2=0.0 ;
SI114X SI1145 = SI114X();
//String hora = "", fecha="";

```

Figura 2. Definición de Puertos y Variables

Una vez se tienen definidos los parámetros de las constantes y demás librerías que van a ser utilizadas en el cuerpo del código, inicializar las funciones del puerto serial, apertura del puerto declarado para los sensores y configuración del tiempo de inicio de la medición, estas líneas de código pueden verse en la *Figura 3*.

```
void setup()
{
  LTask.begin();
  Serial.begin(57600);
  dht.begin();
  pinMode(13, OUTPUT);
  DateTime now;
  now.year = 2019;
  now.mon = 04;
  now.day = 06;
  now.hour = 12;
  now.min = 25;
  now.sec = 00;
  LDateTime.setTime(&now);

  if (!LSD.begin()){
    Serial.println("Error inicializando SD");
  }
  else {
    Serial.println("SD inicializada OK");
  }

  while (!SI1145.Begin())
  {
    Serial.println("SI1145 is not ready!");
    delay(1000);
  }
  Serial.println("SI1145 inicializado");
}
```

Figura 3. Ciclo de Configuración

La descripción del ciclo que va a realizar el código donde se define 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 van a ser guardados en un archivo que se crea en la memoria SD, al que también se le determina un nombre específico y tipo de archivo.