

Apéndice B – Código para el Mote Wi-Fi

Este documento contiene en detalle la totalidad del código que se usó para configurar el Mote con comunicación Wi-Fi, el código que aquí se presenta es basado en el lenguaje de programación C++, este código es compilado en el aplicativo software Arduino IDE, como sabemos la placa Linkit One es compatible con Arduino, el código para este Mote se basa en que realiza la recepción de los datos provenientes de los sensores y son procesados para darles el valor en las unidades que corresponda según la variable que se está censando, estos datos son organizados en una estructura definida para poder ser presentados de una forma agradable y fácil de interpretar, una vez los datos están listos para ser transmitidos se realiza el emparejamiento o conexión con el aplicativo web Ubidots que es la plataforma donde finalmente van a ser visualizados los datos.

Requerimientos de Software

Para poder cargar el código a la placa Linkit One con éxito es necesario tener una serie de programas instalados en una computadora, estos requisitos van a ser descritos y llevados a cabo en el siguiente paso a paso.

1er Paso: Se va a realizar la instalación del Arduino IDE, este aplicativo software se encuentra disponible en la página web del Fabricante Arduino, siempre se podrá encontrar la versión más actual del software y disponible para los sistemas operativos Windows, Mac OS y Linux. El enlace para la descarga es el siguiente:

<https://www.arduino.cc/en/main/software>

Una vez se descarga el aplicativo de instalación se debe ejecutar y comenzar el proceso de instalación, ahí se debe configurar los parámetros de instalación del software y ubicación de la carpeta contenedora del aplicativo.

2do Paso: En este paso vamos a instalar el Linkit ONE SDK, para esto es necesario que se instale un controlador de puerto COM USB para que la placa pueda ser reconocida por el Arduino IDE. En el portal de desarrollo MediaTek Labs se encuentra una guía con el paso a paso de como instalar el controlador diferenciando los sistemas operativos Windows en versiones 7 a 10, Mac OS y Linux. Vamos a encontrar esta información detallada en el siguiente enlace:

<https://docs.labs.mediatek.com/resource/linkit-one/en/getting-started/get-started-on-windows/install-the-arduino-ide-and-linkit-one-sdk>

3er Paso: En este paso se debe actualizar el Firmware de la placa SKD Linkit ONE, normalmente el firmware es cargado a la placa en su versión más reciente al salir de la fábrica, pero para ahorrarnos problemas a futuro vamos a dejar actualizado su Firmware siguiendo unos sencillos pasos descritos en el siguiente enlace:

<https://docs.labs.mediatek.com/resource/linkit-one/en/getting-started/get-started-on-windows/update-your-boards-firmware>

Requerimientos de Hardware

Es importante conocer que la placa Linkit One cuenta con modos diferenciados de funcionamiento según el tipo de comunicación que se desee realizar, para esto cuenta con interruptores en la placa que configuran cada uno de los tipos de operación y el puerto que se va a dedicar para la alimentar la placa de procesamiento, para el caso de este documento se va a trabajar

con la comunicación vía Wi-Fi y con alimentación de energía por el puerto Micro-USB por lo cual la disposición de los interruptores de la placa deben fijarse como se ven en la *Figura 1*.

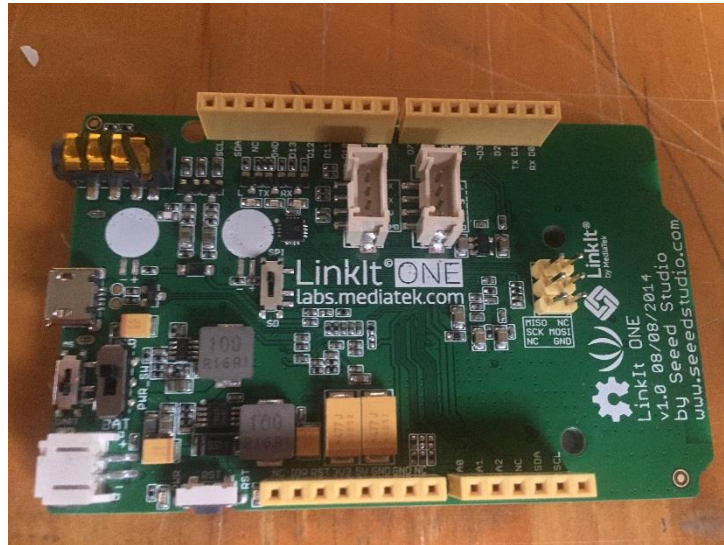


Figura 1. Disposición de los interruptores de la Placa Linkit ONE

El código debe ser cargado a la placa Linkit ONE realizando la conexión física de la placa mediante el puerto Micro-USB, al momento de dejar la placa con sus conexiones físicas realizadas en el Mote se debe oprimir el botón de reinicio, para que cargue desde cero el código y se inicie con la operación en ese momento.

Igualmente es importante que se realice la conexión de la antena Wi-Fi que viene dentro de la caja que contiene la placa Linkit ONE, en la parte posterior de la placa se encuentra el puerto señalado donde se debe realizar la conexión de la antena.

Una vez se culminan estos pasos anteriores con éxito se puede pasar a desarrollar las líneas de código necesarias para controlar el Mote, cada tipo de Mote tiene una variante que se destaca por el tipo de comunicación para la transmisión de los datos recolectados, para este caso vamos a realizar el código del Mote con comunicación vía Wi-Fi.

El código completo se puede encontrar en el archivo con extensión (.INO) llamado (Mote Wi-Fi), 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 Wi-Fi 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 2*.

```
#include <LTask.h>
#include <LWiFi.h>
#include <LWiFiClient.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 2. 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 red inalámbrica de conexión a internet, la clave y el tipo de seguridad de la conexión Wi-Fi, 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. 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 WIFI_AP "USTA_Administrativo"
#define WIFI_PASSWORD "#soytomasino#"
#define WIFI_AUTH LWIFI_WPA // choose from LWIFI_OPEN, LWIFI_WPA, or LWIFI_WEP.
// #define WIFI_AP "AURORA ZABALA"
// #define WIFI_PASSWORD "auroza52"

// These are the variables you will want to change based on your IOT data streaming account / provider
#define WEBSITE "things.ubidots.com"
#define URL "/api/v1.6/collections/values/?token=BBFF-WDXjl8XdQjfyqxxqs8cX4rrByXlpKy&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 5

```

Figura 3. Definición de 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(38400);
    pinMode(13, OUTPUT);
    dht.begin();
    delay(1000);
    Wire.begin();
}

```

Figura 4. Configuración de los Puertos utilizados.

La descripción de 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 como llegan los datos a Ubidots para que puedan ser presentados de forma acertada.