

Prueba de conexión entre Arduino y Sigfox

Librerías por usar

Nombre	Fecha de modificación	Tipo	Tamaño
AceWire	12/01/2022 15:27	Carpeta de archivos	
Isigfox	12/01/2022 17:28	Carpeta de archivos	
SimpleTimer	12/01/2022 17:28	Carpeta de archivos	
Tinyfox	12/01/2022 15:26	Carpeta de archivos	
Tsensors	12/01/2022 17:28	Carpeta de archivos	
readme	12/01/2022 15:11	Documento de te...	1 KB

Código

```
////
```

```
Isigfox *Isigfox = new WISOL();
```

```
typedef union{
```

```
    uint16_t number;
```

```
    uint8_t byte[2];
```

```
}
```

```
UINT16_T;
```

```
void setup() {
```

```
    // Init serial connection between Arduino and Modem
```

```
    Serial.begin(96000);
```

```
    // WISOL modem test
```

```
    Isigfox->initSigfox();
```

```
    Isigfox->testComms();
```

```
    Isigfox->getZone();
```

```
}
```

```
void loop() {
```

```
    // obtenemos la temperatura que nos da un tipo de dato float
```

```
    float voltage = 1.45;
```

```
    int bateria = 100;
```

```
    Serial.println(voltage);
```

```
    Serial.println(bateria);
```

```

//hacemos un arreglo de bytes para enviar byte por byte mas adelante
byte *float_byte = (byte*)&voltage;
byte *int_byte = (byte*)&bateria;

// indicamos el tamaño de nuestro mensaje sabiendo que el maximo tamaño es de 12 bytes
const uint8_t payloadSize = 9;
uint8_t buf_str[payloadSize];
buf_str[0] = float_byte[0];
buf_str[1] = float_byte[1];
buf_str[2] = float_byte[2];
buf_str[3] = float_byte[3];
buf_str[4] = int_byte[0];
buf_str[5] = int_byte[1];
Send_Pload(buf_str, payloadSize);
}

```

```

void Send_Pload(uint8_t *sendData, const uint8_t len){
    // No downlink message require
    recvMsg *RecvMsg;

```

```

    RecvMsg = (recvMsg *)malloc(sizeof(recvMsg));
    Isigfox->sendPayload(sendData, len, 0, RecvMsg);
    for (int i = 0; i < RecvMsg->len; i++) {
        Serial.print(RecvMsg->inData[i]);
    }
    Serial.println("");
    free(RecvMsg);

```

```

// If want to get blocking downlink message, use the folling block instead
/*
recvMsg *RecvMsg;

```

```

RecvMsg = (recvMsg *)malloc(sizeof(recvMsg));

Isigfox->sendPayload(sendData, len, 1, RecvMsg);

for (int i=0; i<RecvMsg->len; i++){

    Serial.print(RecvMsg->inData[i]);

}

Serial.println("");

free(RecvMsg);

*/

// If want to get non-blocking downlink message, use the folling block instead

/*

Isigfox->sendPayload(sendData, len, 1);

timer.setTimeout(46000, getDLMsg);

*/

}

```

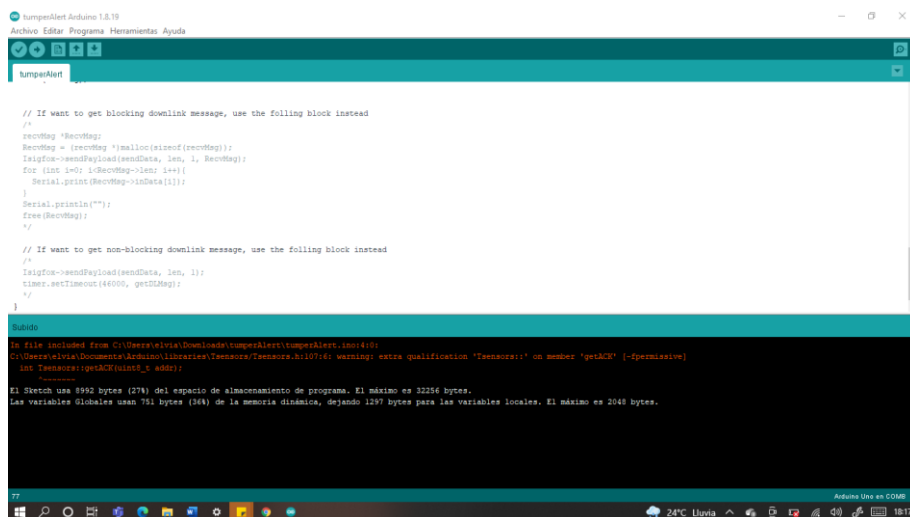


Ilustración 1 envió de código a Arduino

Conexión física de Arduino con Thinxtra:



Device

Time	Delay (s)	Seq Num	Data / Decoding	Base station reception attributes			Callbacks	Location
				Station	RSSI (dBm)	Freq (MHz)		
2021-08-11 14:36:16	2.6	3758	9a99054200c030bf	7C66	-129.00	920.8538		

Conclusión de prueba:

El código compila y la tarjeta se habilita mostrando una luz azul, sin embargo los mensajes no se envían posiblemente por falta de señal en la ubicación donde las pruebas con las librerías fueron realizadas.