

How to use Semtech SX1276 LoRa Module with Arduino LoRa.h library

Enric Puigvert Coromina
Universitat de Barcelona
FOXES

Introducció

En aquest document intern del grup de recerca del Departament d'Electrònica de la Facultat de Física de la Universitat de Barcelona s'explica com utilitzar el mòdul LoRa de Semtech SX1276 mitjançant la llibreria d'Arduino LoRa.h.

Material necessari

- Arduino UNO o similar (x2) + Cable USB
- Mòdul SX1276 de Semtech (x2)
- Cables Dupont (de protoboard)
- Oscil·loscopi o analitzador lògic (SPI sniffer)

Software necessari

- Arduino IDE

Instal·lació del software

1. Instal·lar Arduino IDE.
2. (En cas de utilitzar un Arduino saltar el pas 2, on s'explica com fer funcionar una placa ST amb Arduino)

Comencem instal·lant les llibreries necessàries per fer anar la ST NUCLEO64 L073RZ:

- a. A l'Arduino IDE anem a:

File -> Preferences -> Additional Boards Manager URLs i cliquem a la icona .

- b. Copiem l'adreça:

https://github.com/stm32duino/BoardManagerFiles/raw/master/package_stmicroelectronics_index.json

- c. Cliquem Ok. (Tanquem l'Additional Boards Manager)
- d. Cliquem Ok. (Tanquem preferències)
- e. Anem a

Tools -> Boards -> Board Manager

I busquem:

STM32

Instal·lem el Package:

STM32 MCU based boards

- f. Un cop instal·lat el Package, tanquem el quadre de diàleg.
- g. Tornem a clicar a Tools i posem els paràmetres tal com diu la Figure 1:

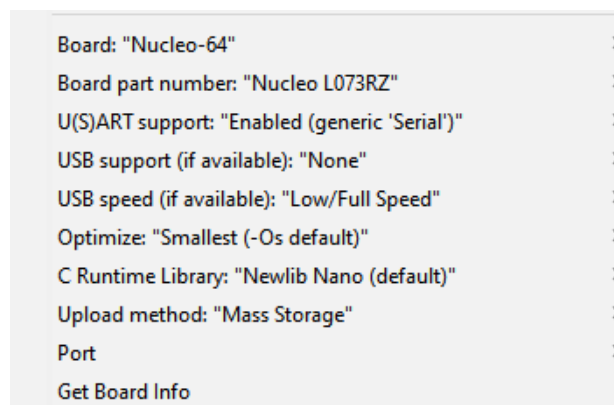


Figure 1. Paràmetres de la board.

- h. Provar si funciona l'exemple Blink d'Arduino. S'hauria de veure que el led de la placa fa pampallugues.

Ara ja tenim la configuració de la placa feta!

3. Ara hem d'instal·lar les llibreries del LoRa:
 - a. Anem a:
Tools -> Library Manager
 - b. Busquem LoRa
 - c. Instal·lem el Package LoRa by Sandeep Mistry. Hauria de posar que es compatible amb el Semtech SX1276/77/78/79.
 - d. Un cop instal·lat tancar el quadre de diàleg.

I ja tenim la llibreria del LoRa instal·lada!

Pinout de la placa

Documentació

En cas d'utilitzar com a microprocessador la ST NUCLEO64 L073RZ, el seu pinout està al següent enllaç:

<https://os.mbed.com/platforms/ST-Nucleo-L073RZ/>

Els datasheets de les diferents plaques es troben adjunts a aquest document.

El mòdul SX1276 funciona mitjançant protocol SPI i certs pins d'interruptió. El pinout de la placa de desenvolupament està ubicat al següent enllaç:

<https://os.mbed.com/components/SX1276MB1xAS/>

Versions

Cal notar que nosaltres disposem de la versió americana de la placa que és la SX1276MB1LAS. L'única diferència amb l'europea es que la freqüència de l'antena d'alta freqüència es de 915MHz i no de 868MHz. Les dues versions permeten també una freqüència de 433MHz que es universal.

Connexió

Per connectar-la a l'Arduino, es fa en mode de Shield. Això vol dir que es col·loca la placa directament als connectors superiors de la placa Arduino tal i com es mostra a la Figure 2:



Figure 2. Col·locació del mòdul LoRa amb la placa Arduino (o equivalent).

Mencionar que de tots els pins disponibles al mòdul, es fan servir, amb aquesta llibreria els marcats a la Table 1:

Semtech SX1276/77/78/79	Arduino
VCC	3.3V
GND	GND
SCK	SCK
MISO	MISO
MOSI	MOSI
NSS	10
NRESET	9
DIO0	2

Table 1. Taula de pins del mòdul SX1276.
[<https://github.com/sandeepmistry/arduino-LoRa>]

Funcionament de la llibreria

La referència de la llibreria està ubicada al repositori de GitHub:

<https://github.com/sandeepmistry/arduino-LoRa>

<https://github.com/sandeepmistry/arduino-LoRa/blob/master/API.md>

Com es pot veure en l'API de la llibreria hi ha diverses funcions disponibles per fer que el sistema funcioni.

Inicialització

Importar les llibreries LoRa.h i SPI.h.

Primer de tot cal dir a la llibreria quins pins utilitzarem:

```
LoRa.setPins(csPin, resetPin, irqPin);
```

Es important fer primer el `LoRa.begin()`, ja que posa el sistema en els paràmetres per defecte, i després fer totes les configuracions que es vulgui.

També es important inicialitzar amb el següent codi per assegurar que el sistema s'inicialitza correctament:

```
if (!LoRa.begin(frequency)) {  
  Serial.println("LoRa init failed. Check your connections.");  
  while (true);          // if failed, do nothing  
}
```

Configuració del mòdul LoRa

Un cop inicialitzat el mòdul, podem començar a configurar-lo.

Potència de sortida (a l'antena) – `LoRa.setTxPower(txpower, powerMode)`

Tenim dos tipus de modes de potència: PABOOST i RFO.

- **PABOOST:** Es el mode de màxima potencia i per tant, el que podrà arribar a més potència d'antena i per tant, arribar mes lluny. El handicap que té es que el seu consum es dispara. Per aquest mode de potència tenim uns valors de potència d'antena de 2dBm a 20dBm.
`LoRa.setTxPower(txPower);`
- **RFO:** Es el mode de mínim consum. Desactiva els amplificadors interns i per tant, no tindrem tanta potència a la sortida de l'antena, i no arribarà tant lluny en la transmissió. Per aquest mode tenim disponible els valors de potència d'antena de 0dBm a 14dBm.
`LoRa.setTxPower(txPower, PA_OUTPUT_RFO_PIN);`

També es important configurar Over Current Protection (OCP), ja que si posem el mode de màxima potencia, hem d'augmentar aquest limitador. En cas de no configurar-lo correctament no funcionaria la transmissió/recepció ja que consumirem per sobre del nivell de limitació. En teoria, aquesta configuració la fa la funció `setTxPower()`.

Nota: txPower es el número de la potència d'antena (en dBm).

Spreading factor – `LoRa.setSpreadingFactor(SF)`

L'Spreading Factor es un paràmetre del LoRa que marca en quants chirps s'envia la portadora de les dades per segon (mirar com funciona el LoRa per entendre ->

<https://www.youtube.com/watch?v=dxYY097QNs0>).

En resum, aquest paràmetre canvia la forma en què s'envien les dades, posant per davant el data rate (per tant posar mes dades per segon) o la seguretat de l'enviament.

Si volem enviar 100 dades, i posem un número petit de spreading factor (7, per exemple), fem molts chirps per segon i per tant, podem enviar moltes dades i per tant, consumir menys alhora d'enviar, ja que tindrem una finestra d'enviament molt petita. Però el receptor pot ser que perdi dades per culpa de les condicions atmosfèriques o de l'ambient.

En canvi, si posem un spreading factor alt, s'enviaran molt pocs chirps, i per tant, el data rate disminuirà i tardarem molt mes en enviar les dades. Però com que tenim el sistema enviant les dades durant mes temps, li donem mes oportunitats al receptor a rebre i recuperar les dades i per tant, la transmissió es més segura, tot i que consumeix molt mes.

Recordo que la tecnologia LoRa, es un tipus de transmissió de dades sense cables basat en NO connexió, i per tant, no tenim acknowledge ni res per l'estil.

Per entendre això recomano mirar el següent enllaç:

<https://www.thethingsnetwork.org/article/how-spreading-factor-affects-lorawan-device-battery-life>

En el cas de FOXES, hem fet estudis amb Spreading Factor de 7 (el mínim dins de l'especificació LoRa), ja que el consum es el mes determinant. Tot i així, es podria valorar, en cas d'aconseguir l'energy harvesting, de canviar d'Spreading Factor.

Bandwidth – LoRa.setSignalBandwidth(BW)

El bandwidth de la transmissió. Per regulació europea, el màxim que podem assolir es de 250KHz tot i que LoRa sigui capaç de transmetre a 500KHz.

SyncWord – LoRa.setSyncWord(syncWord)

Clau de pas per a fer una “sincronització” entre emissor i receptor. Bàsicament els sistemes propers no haurien de fer cas a transmissions que no tinguin la seva clau de pas o syncWord. No es necessari per a funcionar.

Interrupcions de transmissió/recepció – LoRa.onTxDone(onTxDone)/onReceive(onReceive)

Inicialització de les funcions de callback o d'interrupció per la fi de transmissió o la recepció. Es posa entre els parèntesis el nom de la funció d'interrupció que s'ha definit per processar les interrupcions de transmissió i recepció.

La interrupció de transmissió es donarà quan el sistema buidi la FIFO de sortida quan estigui en mode TX.

La interrupció de recepció es donarà quan el sistema tingui plena la FIFO de entrada quan estigui en mode RX.

Per mes informació mirar el datasheet del mòdul SX1276. Sobretot el diagrama de blocs del sistema, el procés de transmissió i el de recepció.

Transmissió

Per poder fer la transmissió cal seguir aquests passos:

1. Posar el sistema en mode stand-by. Cal fer això ja que de normal estarem en mode sleep (baix consum), i cal posar-lo en un mode que permeti escriure a la FIFO del sistema.

```
LoRa.idle()
```

2. Generació del paquet:

```
LoRa.beginPacket()
```

3. Escripció de dades:

```
LoRa.write(value_array, length); //Envia la dada en raw.
```

```
LoRa.print(string_array, length); //Envia la dada com a string.
```

Es important no sobrepassar el límit de dades de paquet (255 bytes).

4. Finalització del paquet:

```
LoRa.endPacket(async);
```

Async es una variable booleana que fa que el sistema esperi (o no) la finalització de l'enviament. En cas d'activar-la (true), el programa queda penjat en un while-loop fins que arriba la interrupció de TX.

Quan posem la instrucció de finalització del paquet, el sistema es posa en mode TX i envia la dada.

Recepció

Per poder fer la recepció cal seguir aquests passos:

1. Posar el sistema en mode recepció:

```
LoRa.receive();
```

2. Esperar interrupció de RX:

```
void onReceive(int packetSize){}
```

En aquest cas haurem de esperar a que arribi la interrupció de recepció que haguem definit a la configuració inicial.

3. Lectura de dades:

```
String message = "";  
while (LoRa.available()) {  
    message += (char)LoRa.read();  
}
```

Un cop som dins de la interrupció iniciem un while-loop que es mantindrà actiu sempre que quedin dades a la FIFO del mòdul LoRa. Dins d'aquest while anirem llegint les dades.

Important que el buffer de recepció sigui una String, ja que es la forma més efectiva de llegir els bytes i imprimir-los per terminal.

L'altre opció seria directament utilitzar una array de bytes com a buffer de recepció.

En cas de voler més informació de com programar el mòdul LoRa, mirar la referència de la llibreria.

Descodificació de la llibreria (SPI)

Com hem dit anteriorment, aquesta llibreria funciona per protocol SPI. Això vol dir, que podem fer un “sniffing” del bus de dades per veure què s’està enviant, en cada funció, al mòdul LoRa.

Així doncs, per ajudar en l’aprenentatge del funcionament del sistema, s’adjunta a aquest document, un llibre d’Excel ([LoRa decoded](#)) on es descodifiquen totes les ordres que es dona al sistema en el programa d’Arduino [LoRa_sniff_spi.ino](#) (adjuntat, també a la carpeta [./Arduino/LoRa_sniff_spi](#)).

El programa té uns defines on es pot controlar si es printen dades per consola, i quina part del funcionament del sistema es vol veure. Al document Excel, hi ha un full de càlcul per cada una de les següents parts:

- Setup
- TX
- RX

Es important, entendre que si es vol veure amb un oscil·loscopi digital o un analitzador lògic el que s’ha escrit al document Excel, sempre caldrà passar per la secció de Setup per a que el sistema quedi configurat i funcional.

Així doncs, al document Excel, hi trobarem diverses columnes tal i com esta descrit seguidament:

- MOSI (RAW-HEX): El que rebem directament a l’analitzador lògic pel bus MOSI:
 - o CMD+REG: Primer byte. Representa el comando (MSB) i el número de registre. Si el comandament es 1, vol dir que estem escrivint el registre. Si es 0, el llegim.
Exemple:
 Rebem per MOSI → 0xA2 = 10100010. Estem escrivint el registre 0x22
 Rebem per MOSI → 0x12 = 00010010. Estem llegint el registre 0x12
 - o VALUE: Valor que volem escriure al registre (en cas de estar en mode escriptura). En cas de lectura no s’utilitza.
- MOSI OP (RAW-DEC): El que rebem per MOSI en decimal. Es fa això per següents operacions.
- MOSI (HEX-CMD_FILTD): El que rebem per MOSI desglossat en comando, registre i valor:
 - o CMD: Comandament. W = write = escriptura. R = read = lectura.
 - o REG: Valor en hexadecimal de l’adreça del registre que es vol llegir/escriure.
 - o VALUE: Valor que es vol escriure al registre REG. En cas de lectura no s’utilitza.

Pel bus MISO no fem la descodificació en comandaments ja que el mòdul LoRa no envia cap comandament, sinó que respon amb dades (en cas de lectura). Per tant, s’ha fet 3 columnes on es converteix el valor en hexadecimal a decimal i binari.

Finalment, es mostra el registre que s’està utilitzant a la transferència amb el seu nom i la funció i paràmetres que s’ha modificat en aquesta.

Consultar el document Excel per a mes informació.

Conclusions

Per a la programació del mòdul Semtech LoRa s’ha fet enginyeria inversa a través de la tècnica “sniffing” d’una llibreria Arduino per estalviar temps. Amb l’ajuda del Datasheet del sistema s’ha pogut veure totes les transferències que utilitza el sistema per poder configurar-se, enviar dades i rebre-les.