

PART 1 - How to develop a project – NOT how to make it.

NO CODE – THIS IS NOT AN INSTRUCTABLE

just the methods to get there with mistakes and observations !

WARNING – includes noted missteps and oversights!

We're going to develop an informal, battery powered order collection system for country fairs - that keeps track of orders, notifies cooks of outstanding quantities, informs customers of highest order ready, allows server to release orders from the queue.

1. Work out your requirements

- Three or four connected devices (at unknown distances – approx. 10m) working together
- 20x4 character LCD with keypad or buttons, or LCD/Touch screens
- 4x7-seg LED display(s) and/or 32x8 dot matrix modules
- Serial ticket printer(s)
- Wireless comms capability in less than desirable cabling environments.
- Node 1 – LCD & keypad collects sales/order details – assigns a queue number
- Node 2 – displays orders outstanding for collection.
- Node 3 – displays number of orders in the queue on large 7-seg display
- Node 4 – (optional) display the highest 'ready for collection' order number, refreshed when collected.

2. Develop a specification

- Run time of > 6 hours on rechargeable batteries. USB recharge.
- WiFi or LoRa wireless communications.
- Simple idiot-proof user-interfaces.
- Low cost
- Splash & spill proof

3. Choose hardware

- Arduino UNO original or clones
- SURE Electronics 4x7-segment SPI displays
- SURE Electronics (0832) 32x8 LED matrix displays
- Generic 20x4 LCD display (HD44780)
- LiPo battery & charge modules.
- SX-1276 LoRa modules (Reliable Mode) chosen for the small footprint and flexibility of one-to-many messaging. (using [RADIOHEAD](#)) library for RF95 modules
- Everything will fit in an Arduino Uno – low-cost and easy to source.

CONSIDER BREAKING YOUR PROJECT INTO SUB-PROJECTS – TO PROVE THEM INDIVIDUALLY

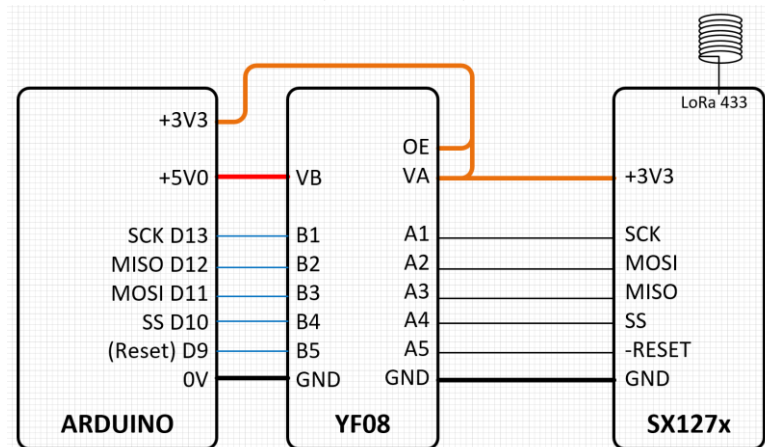
4. Wire up your test platforms

- Controller(s)
- LCD & keypad / buttons
- LoRa Radio modules

Careful to **observe I/O voltages & levels**, and check for shorts between jumpers and/or tracks.

This is the Arduino + SX-1276 LoRa sub-project.

(FIRST draft)



Yes, create a drawing from the beginning - to immortalise the *ideas* you're starting with...

The level shifter drawn (B -> A) to comply with chip spec (A=3V3 / B=5V), and to simplify wiring.

Note the signal flow from left to right

5. Assemble, Compile and test the fundamental elements – separately with mini projects

- **Buttons**
For simplicity - an R2R ladder
- **Keypad**
Key matrix using the standard Arduino keypad library.
- **LCD / touch**
using the standard Arduino LCD libraries, or your own if preferred
- **LoRa Comms**
using an off-the-shelf LoRa library(using [RADIOHEAD](#)) library for RF95 modules

These may be tested in any order, but at some stage you'll want to combine elements for more comprehensive testing. **DON'T RUN BEFORE YOU CAN WALK !**

TROUBLESHOOTING the LoRa SX127x subproject...

SIMPLE MISTAKES – I should have been better prepared!

I started with a couple of Arduino UNOs and LoRa modules... wired up as shown above – but I was getting the LoRa modules *failed to initialise*...

FIRST – I swapped MISO & MOSI pins going to the SX12176 – easily fixed.

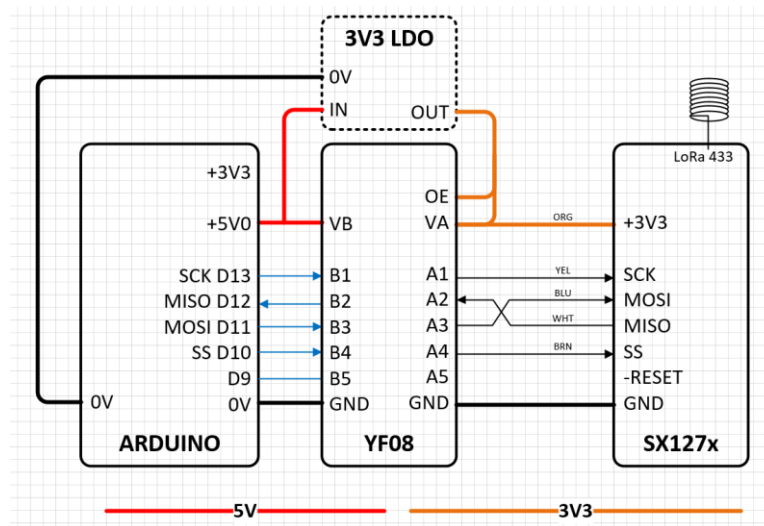
THEN - Back to the datasheet and the library files...

```
/// Caution: Power supply requirements of the RFM module may be relevant in some circumstances:
/// RFM95/96/97/98 modules are capable of pulling 120mA+ at full power, where Arduino's 3.3V line can
/// give 50mA. You may need to make provision for alternate power supply for
/// the RFM module, especially if you wish to use full transmit power, and/or you have
/// other shields demanding power. Inadequate power for the RFM is likely to cause symptoms
/// such as:
/// - reset's/bootups terminate with "init failed" messages
```

D'oh – a problem is known- when the 3V3 rail is unable to supply the start-up current of the LoRa module.

So I must step away from using the Arduino's 3V3 output, and find a separate 3V3 regulator (sigh – more later)

(SECOND draft)



Revised schematic with added 3V3 LDO regulator & corrected pinouts...

Yes, the Arduino 5V can supply enough current in this project.

Oops again... this is called debugging !

While the setup 'sort of' worked – *sometimes*...

As it turned out – the extra regulator wasn't the issue . With or without they behaved the same ??!

But here was something else 'not quite right', and some of the Serial.print() were cut short...

This might be a sign of interrupt issues... I went off for some more research.

Deep inside **RH_RF95.h** (different file for other modules & builds...)

```
For Arduino connections-  
///  
interrupt 0 pin D2 <==> DIO0 (interrupt request out)  
Note - the pins used are defined in the low-level module setup / configuration.
```

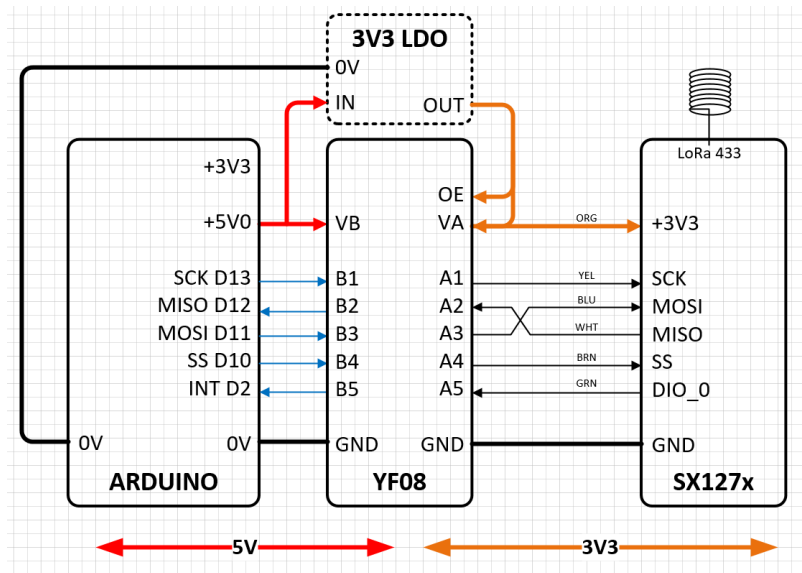
So another wire mod in the prototypes (sigh again)

Because I wasn't using the LoRa RESET pin (expected in early thinking), that went...

I repurposed the B5-A5 level converter for the interrupt.

Coincidentally – I also found two unsoldered wires (everyone does it!)

(THIRD draft)



Revised schematic with INTERRUPT line added
3V3 LDO regulator retained for technical accuracy

This is still called debugging !

Now I'm in trouble... Less happening than before! The boards initialise correctly, but...

Checked all the wiring, some exploration needed.

Lifted the wires on B5 (restored to previous setup)... slightly better...

EUREKA!

The silkscreen on the LoRa module was smeared – I had connected to **DIO_1** – should have been **DIO_0**

```

A reply was sent and acknowledged by the
CLIENT

Message was received and acknowledged to
CLIENT ID: 2 -> Hello there!
A reply was sent and acknowledged by the
CLIENT

Message was received and acknowledged to
CLIENT ID: 2 -> Hello there!
A reply was sent and acknowledged by the
CLIENT

Message was received and acknowledged to
CLIENT ID: 2 -> Hello there!
A reply was sent and acknowledged by the
CLIENT

Sending a message to the SERVER...
Got reply from server ID 1: FROM SERVER:
Message received!

Sending a message to the SERVER...
Got reply from server ID 1: FROM SERVER:
Message received!

Sending a message to the SERVER...
Got reply from server ID 1: FROM SERVER:
Message received!

Sending a message to the SERVER...
Got reply from server ID 1: FROM SERVER:
Message received!

Sending a message to the SERVER...
Got reply from server ID 1: FROM SERVER:
Message received!

Sending a message to the SERVER...
Got reply from server ID 1: FROM SERVER:
Message received!

```

And awaaay we go!

Success – now the next piece of the puzzle.

This is the real world.

This LoRa subproject didn't come pre-wired in a box. You have to think.

If you're not up to it. Study, find a knowledgeable friend, or give up now.

To be continued...