

# LoRa Setup Writeup

Henry Mitchell

## I. INTRODUCTION

The purpose of this experiment was to examine the direct wireless communication between two devices with SX1276RF1KAS Transceivers and Arduino Uno boards. As it stands, no gateway has been set up (though that is in the works), so source code that allows direct communication between two nodes was used. This experiment implemented an approach using SX1276RF1KAS Transceivers and Arduino Uno boards. Communication was established between the two units before sensors were attached to the client unit, and the data from those sensors were sent to the server unit.

## II. MATERIALS

The following were used in developing this system:

- SX1276RF1KAS  $\times$  2
  - These are the LoRa units which transmit and receive the data from one Arduino to another
- 915MHz Antenna (yellow)  $\times$  2
  - These are the antennae which came with the LoRa units, and are plugged into the “HF” port on the LoRa units
- Arduino Uno board  $\times$  2
  - These are the micro-controllers which determine which data to collect and send
- USB 2.0 Type B cable  $\times$  2
  - These are to connect the Arduino units to a computer, in order to upload instructions to them
- Jumper Wires
  - These are to connect the Arduino and LoRa units, as well as the Arduino units and sensors
- Required Arduino libraries
  - These are the SPI library and RadioHead library.

The pin connections used in the setup of this system are shown in table I.

Purpose	LoRa	Arduino
Power supply	2 (VDD_RF)	3.3 V
	22 (VDD_ANA)	
	34 (VDD_FEM)	
Ground	32 (GND)	GND
SPI	1 (SCK)	D13
	3 (MOSI)	D11
	8 (MISO)	D12
	7 (NSS)	10
Digital I/O	12 (DIO0)	2
	5 (DIO1)	6
	17 (DIO2)	7
Reset	10 (NRESET)	8
RXTX	13 (RXTX)	3

TABLE I

PIN MAPPING FOR THE EXPERIMENT.

### III. CONNECTING SENSORS

Before establishing communication between the two units, sensors must be connected. As things currently stand, the client code is set up for two sensors. One of these is to be connected to pin A0, and the other is to be connected to A1. In future versions of the software, the number of sensors will not be hard-coded, but will simply allow for a plug-and-play style of use. Additionally, any functions that will be used to process the raw data from the sensors must be edited (they are currently called `fix_temp` and `fix_sound`, and are in `arduino/client/client.ino*`). Sensors will also need to be connected to the Arduino's 3.3 V and ground pins.

### IV. PROCEDURE

After all components have been connected as described, All that remains is to upload the source code. The unit with the sensors attached is the client unit, and will therefore have the client code (`arduino/client/client.ino`) uploaded to it. The other unit is the server unit, and will therefore have the server code (`arduino/server/server.ino`) uploaded to it. This is the unit for which it is useful to have the serial monitor open, in order to verify that it is receiving data from the client unit. In future iterations of the system, this is the unit which will upload to the database via the gateway.

\*All file paths are relative to the main git repository.