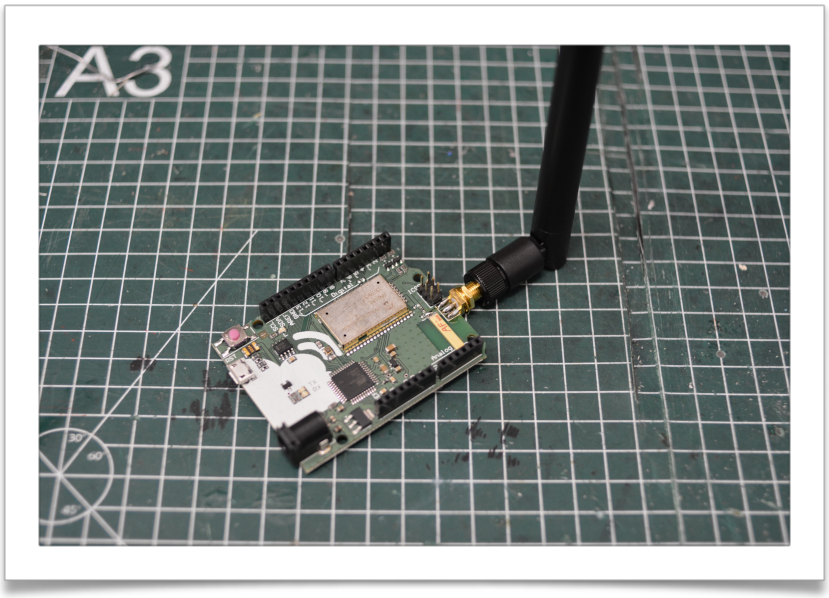
LoRa Quick Start Guide



The Things Uno

Tweetonig
Rotterdam
(English)
v1.0 - written for Things Uno v4

Index

Specifications

CPU: ATmega32u4

Digital I/O channels	20
PWM channels	7
Analog input channels	12
DC current per I/O	40 mA
DC current 3.3V pin	50 mA
Flash memory	32 KB (- reserved bootloader space)
SRAM (static RAM)	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz

Wireless Communication Module: RN2483 / Microchip

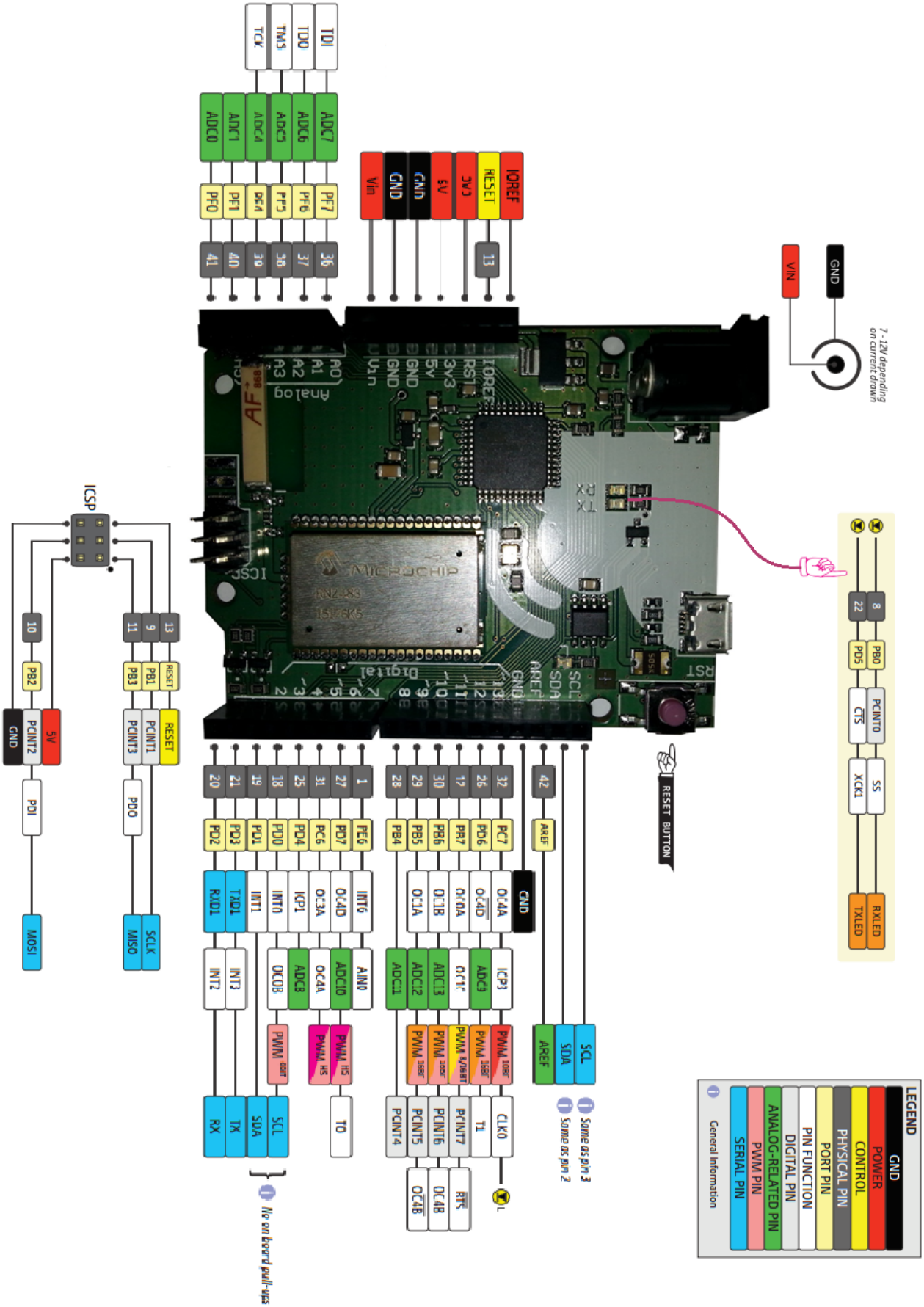
Low-Power Long Range Transceiver Module

Operating Frequencies:	433 MHz en 868 MHz
Receiving sensitivity:	up to -148 dBm
Transmitting power:	adjustable to +14 dBm
Modulation:	LoRa WAN
Range:	10 km coverage at suburban 5 km coverage at urban area

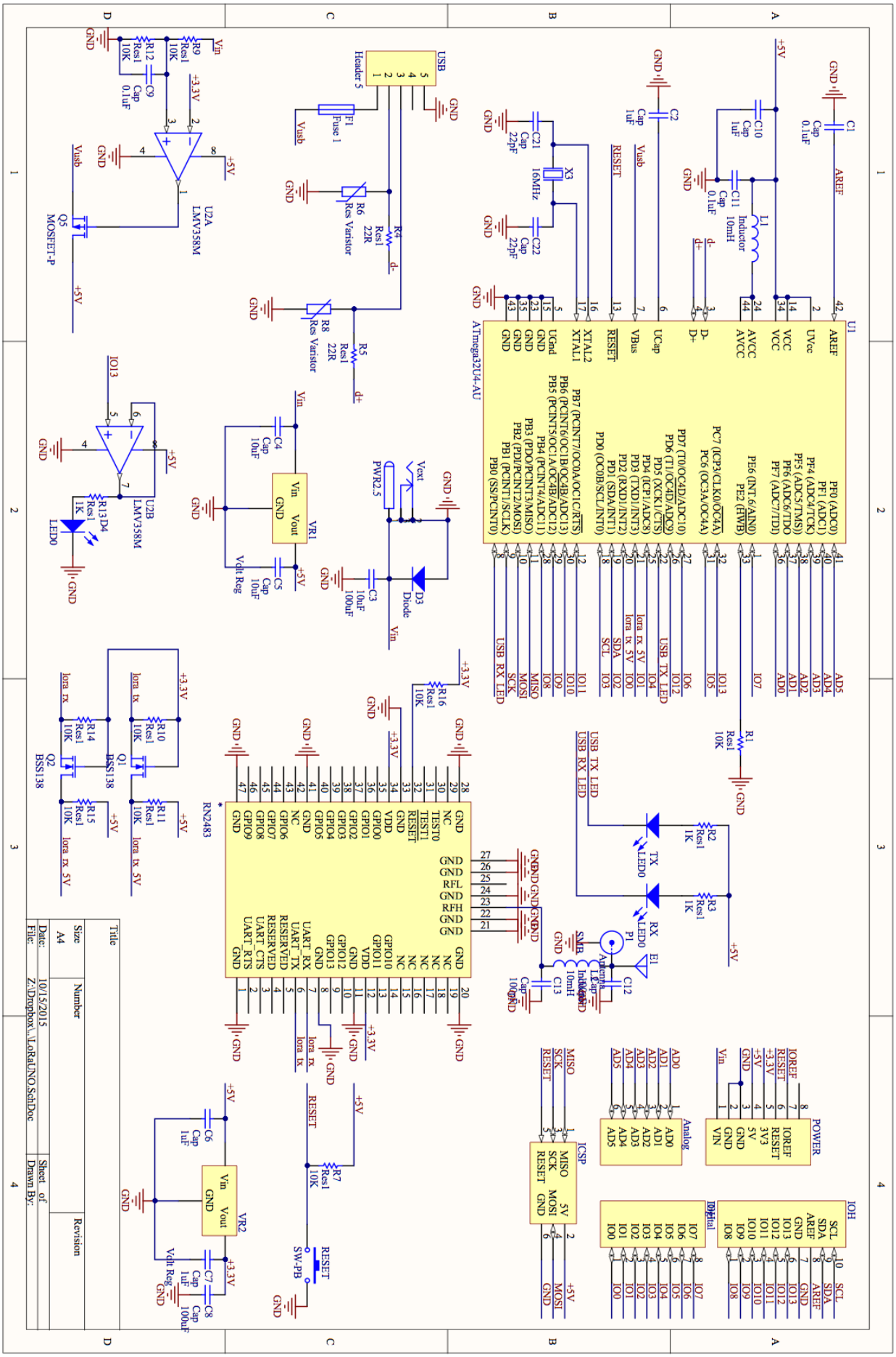
Development Board: Things Uno

Operating voltage:	5V (behind voltage regulator)
Input voltage power connector:	7-12V
Programming voltage (USB):	5V
Length:	68.6 mm
Width:	53.3 mm

Pin layout


LEGEND
GND
POWER
CONTROL
PHYSICAL PIN
PORT PIN
PIN FUNCTION
DIGITAL PIN
ANALOG-RELATED PIN
PWM PIN
SERIAL PIN
General Information
RESET BUTTON
7 - 12V depending on current drawn
VIN
GND
ICSP
Some as pin 3
Some as pin 2
No on board pull-ups

Schematics


ATmega32U4-AU
RN2483
POWER
Analog
Digital
ICSP
Title
Size
Number
Revision
A4
Date: 10/15/2015
Sheet of
File: Z:\Dropbox\...\LoRaUNO.SchDoc
Drawn By:

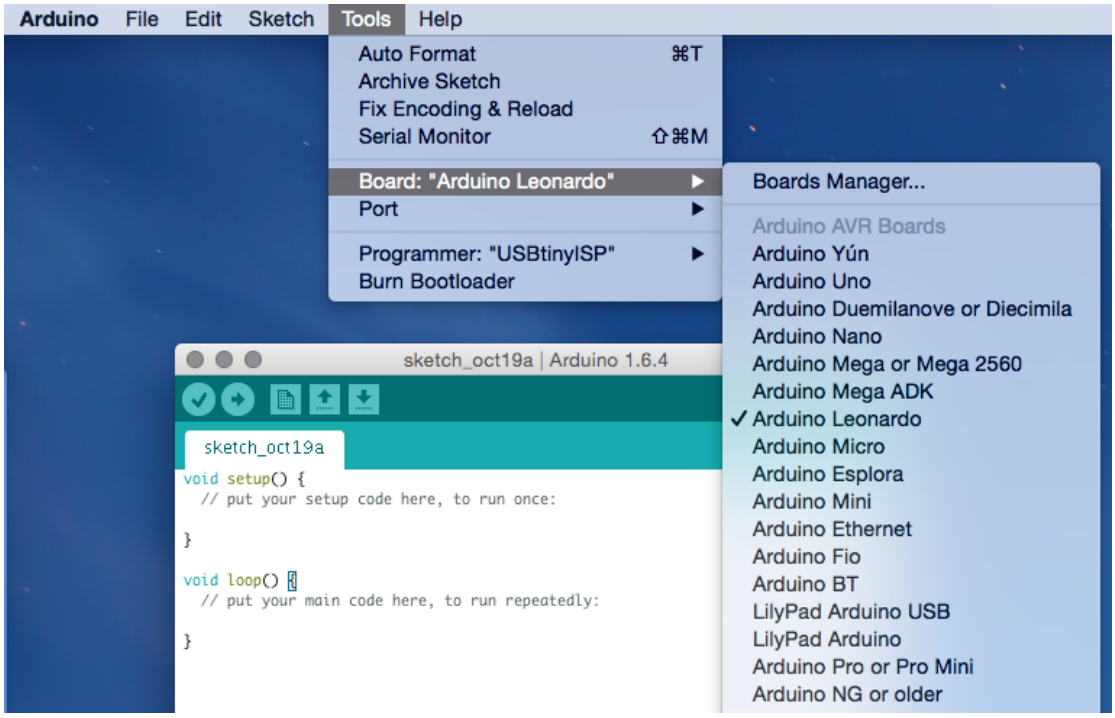
Configuration

Download Arduino test sketch + LoRa library over here: http://bit.ly/1W2j3ZS

Arduino

When plugging the Things Network Uno in your computer you must select the “Arduino Leonardo” board from the tools > board selection.

Test the blink example code to test communication.



Baud rate

To communicate with the RN2483 the baud rate needs to be set to 57600 bps. Do this with “*Serial1.begin(57600);”* in your code.

Now the hardware is ready to configure the next important setups:

LoRa setup

DevUEI (MAC-address)

The MAC-address is expressed in DevUEI. This because LoRa uses MAC EUI-64 standard. The address is build up with 16 hexadecimal characters and can be picked up from the RN2483 by writing sys get hweu over serial TX. For the full syntax use the example below. It's important that the MAC-address is coming back from the RN2483. Wait 1 second and listen to the RX serial bus to receive the MAC-address.

Example:

```
Serial1.write("sys get hweui\r\n");
delay(1000);
while(Serial1.available()) Serial.write(Serial1.read());
```

DevAddr (Device Address)

The assigned device address specifies a unique number in the Lora network. This address must be assigned to the RN2483. DevAddr is also the network address is the Lora network. When the RN2483 receives no DevAddr, it must be configured by *"Serial1.write ("mac set devaddr **assigned address** \ r \ n ");"*. After processing the data, the RN2483 sends feedback if the communication process was successful. There can then be listened to the RX of the ATmega in order to recover the status value.

Example:

```
Serial1.write("mac set devaddr network address\r\n");
delay(1000);
while(Serial1.available()) Serial.write(Serial1.read());
```

IMPORTANT: After a correct setup you should receive "ok".

NwkSKey (Network Session Key)

A device-specific network session key, which is used to compute the data integrity MIC code (Market Identifier Codes), and encryption/decryption of payload field between the Backend and the device. To set and write the NwkSKey to the RN2483 we need to repeat the description above this paragraph. To Set the NwkSKey it's important to use *"Serial1.write("mac set Nwkskey ??????????????????????????????????\r\n");"*. The NwkSKey always contains 32 characters. Again to receive the right feedback from the RN2483, wait for one second to receive 'ok'.

For example:

```
Serial1.write("mac set
nwkskey ????????????????????????????????
\r\n");
delay(1000);
while(Serial1.available())
Serial.write(Serial1.read());
```

AppSKey (Application Session Key)

One or more device-specific application session keys used for end encryption of the payload field for certain application ports. If this AppSKey is not provided to the Backend, then the network cannot access the payload information. To Set the AppSKey it's important to use *"Serial1.write("mac set appskey ????????????????????????????????\r\n");".* The AppSKey always contains 32 characters. Again to receive the right feedback from the RN2483, wait for one second to receive 'ok'.

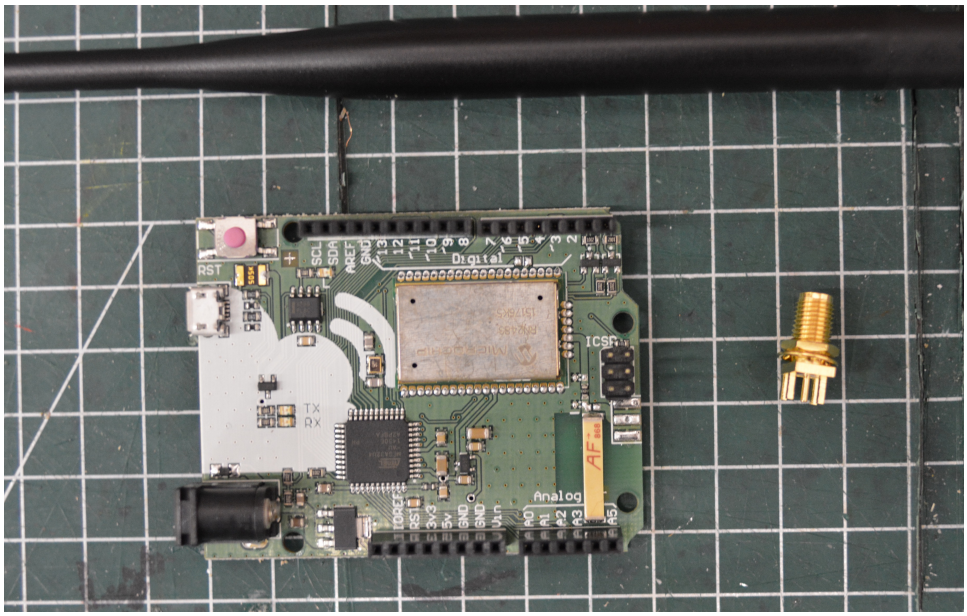
For example:

```
Serial1.write("mac set
appskey ????????????????????????????????
\r\n");
delay(1000);
while(Serial1.available())
Serial.write(Serial1.read());
```

External antenna

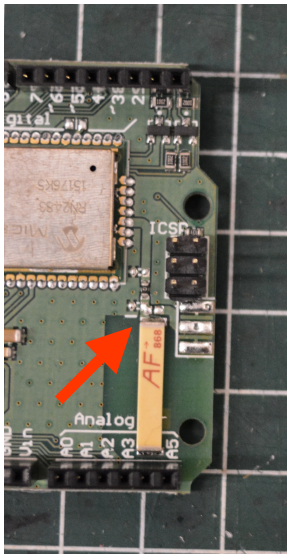
The Things Uno has an onboard smd antenna. In order to expand the range it is possible to add an external antenna. To do this you will need three things:

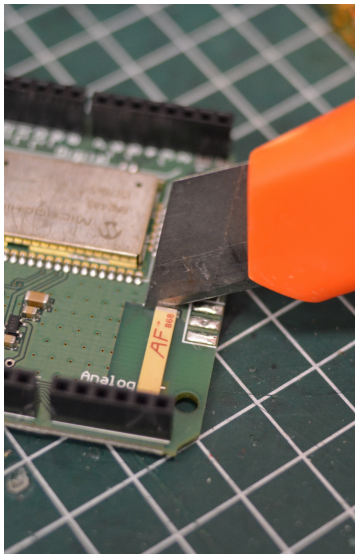
1. The Things UNO
2. A 1.6 mm spaced side SMA connector (like Mouser: 712-CONREVSMA003.062)
3. An antenna. (like Mouser: 712-ANT-868-CWHWRRPS)

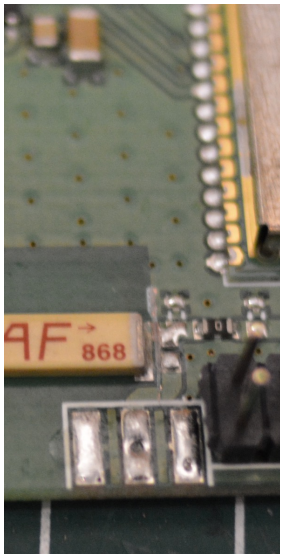


Step 1 - Cutting the connection with the on board antenna.

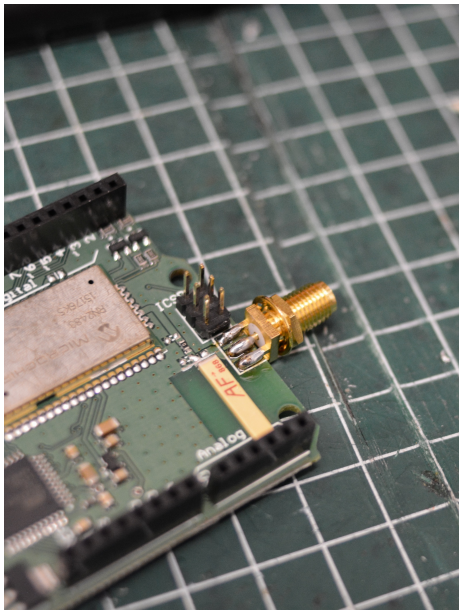
Find the white stripe mark on the PCB and cut the trace with a sharp knife.
(please do this with care)

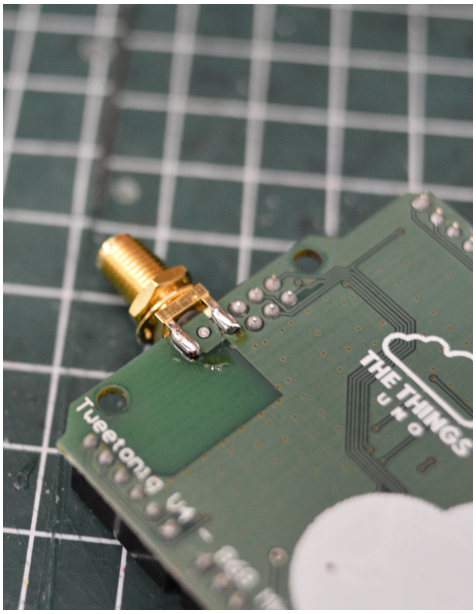




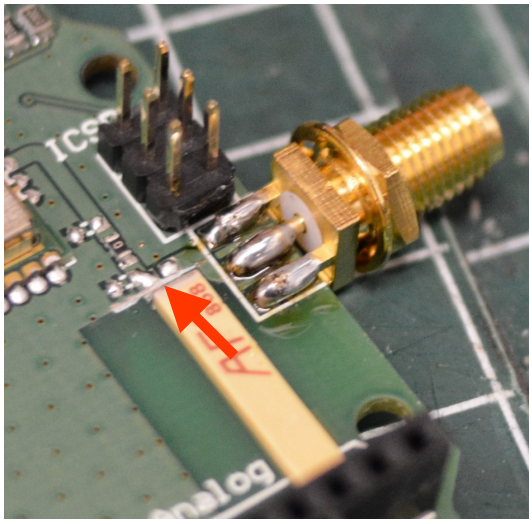


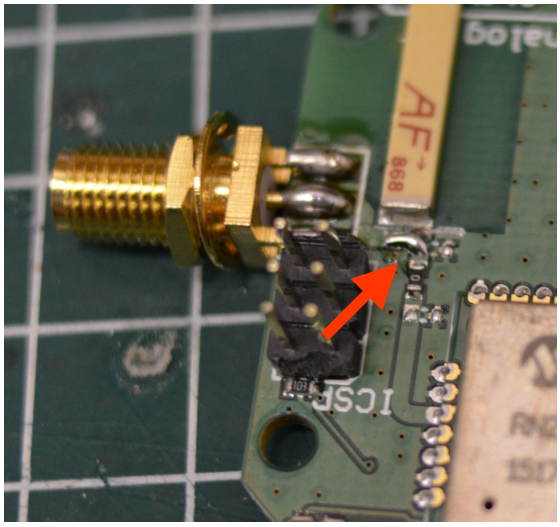
Step 2 - Solder the connector onto the pads.





Step 3 - Solder the “jumper” to lead the circuit to the new antenna.
Left without the jumper, right with jumper in place.





That’s it! Enjoy your TTN Uno.