SRF Shield

Document version 001

CISECO

1. Description

The Ciseco SRF shield instantly transforms any Arduino style board into a fully wireless device. There are no jumpers to worry about, no configuration to be done, simply plug in and begin coding.

The shield utilises the world's best value, secure wireless module, the SRF. You can securely exchange data with all other Ciseco radio devices, including the ultra-long range ARF.

Designed for ease of use, the shield uses normal ASCII when transferring data, requiring no library or complex software. This means all your memory space is for code, not to drive the radio. All settings can be accessed or changed via standard text based AT commands.

The shield has extra pads to allow for configurations such as; Over the Air Programming of your micro, low power sleep states and adding an external antenna to extend the range.

The SRF has flexible frequency and power settings, to cater for all global radio regulations; these are easily set in software.

The surface mount module used on the shield is available separately for those wishing to manufacture wireless products.

2. Features

- ETSI, FCC, CTick, CE compliant in all applicable ISM bands
- Long range, up to 5Km
- Plug 'n Play
- Supports "over the air programing" of the host Arduino (or other) microcontroller
- External antenna support for whip, SMA or u.FL connectors
- 2 low power sleep modes
- Fully supports Open LLAP IoT nodes via free to download source code
- Support for 65,000 networks (PANID)
- User control of duty cycle
- Heartbeat LED to indicate operation or digital RSSI level (software selectable)
- Supports point to point, point to multipoint, multipoint to point and multipoint to multipoint
- Veroboard friendly top layout

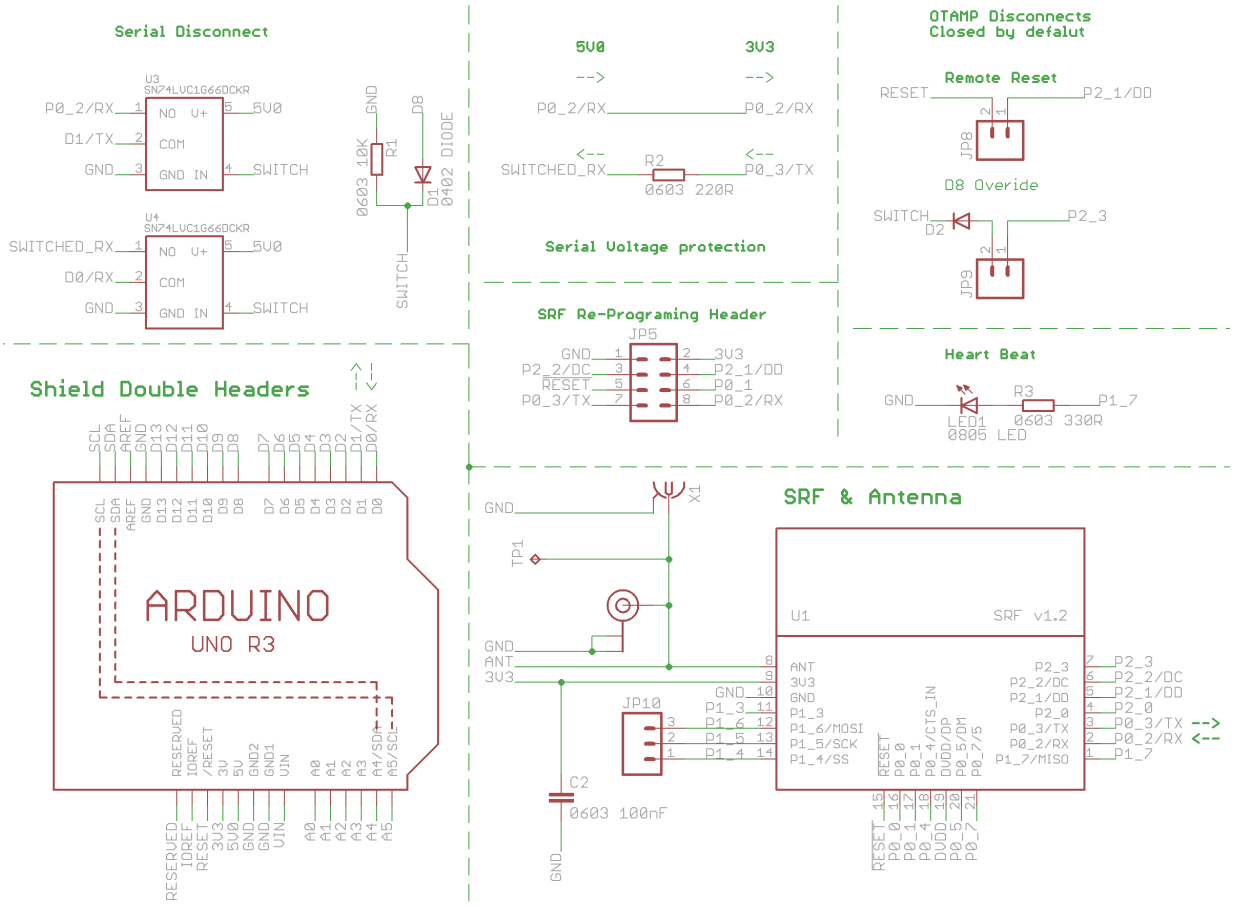
3. Applications

- Add wireless to any existing Arduino project
- Uploading programs from the Arduino IDE code editor "over the air"
- Remote Sensors
- Remote Actuators
- Remote controlled robots
- Remote Telemetry

4. Technical specs

Modulation:	GFSK
Encryption standard: (pass phrase set via AT command)	128 bit AES
Over the air data rate: (changeable via AT commands)	250Kbps
Serial baud rate: (changeable via AT commands)	115.2Kbps
Operating Voltage:	3.3v (2v – 3.9v)
Power Consumption:	33mA (transmitting)
	19mA (receiving)
Light sleep	0.147mA (147uA)
Deep sleep	0.002mA (0.2uA)
Tested range (actual will vary with environment)	
To a device with a chip antenna:	100-150m
To a device with a wire whip:	200-300m
Wire whip to wire whip:	700-1Km
9dB Yagi to wire whip:	1.2-1.6Km
9dB Yagi to 9dB Yagi:	3-5Km
Length:	68.6 mm
Width:	53.35 mm
Height:	12 mm
Weight:	16 grams

5. Circuit Schematic



6. Basic Hardware configuration

Simply plug your shield into your Arduino, you are now ready to begin coding.

7. Arduino Code Examples

A basic radio “Hello, World” example is printed directly on the PCB. To get started open the Arduino IDE, start a new sketch, type in the code from the PCB and Upload the sketch to your Arduino or simply copy and paste this code into the Arduino IDE code editor:

```
void setup() {
  pinMode(8, OUTPUT);
  digitalWrite(8, HIGH);
  Serial.begin(115200);

  Serial.println("Hello, World");
}

void loop() {
}
```

With another radio device listening (eg an SRF Stick in a PC) you will receive “Hello, World” when you power up.

Here is a code example for receiving a command to switch the on-board LED on and off.

```
void setup() {
  pinMode(8, OUTPUT);
  pinMode(13, OUTPUT);
  digitalWrite(8, HIGH);
  Serial.begin(115200);
}
void loop() {
  if (Serial.available() {
    char c = Serial.read();
    if (c == '0'){
      digitalWrite(13, LOW);
    } else if (c == '1') {
      digitalWrite(13, HIGH);
    }
  }
}
```

(continued from pg2) with a PC connected device (eg an SRF Stick) open the Arduino Serial monitor and send a 1

character, the on board LED attached to pin 13 will turn on. Send a 0 to turn it off again.

8. Changing an AT value (code snippet)

It's easy to change the operational configuration from within software. Here is an example how to change the default network PANID from 5AA5 to 2345.

```
Serial.print("+++");
delay(1000);
Serial.println("ATID2345");
Serial.println("ATDN");
```

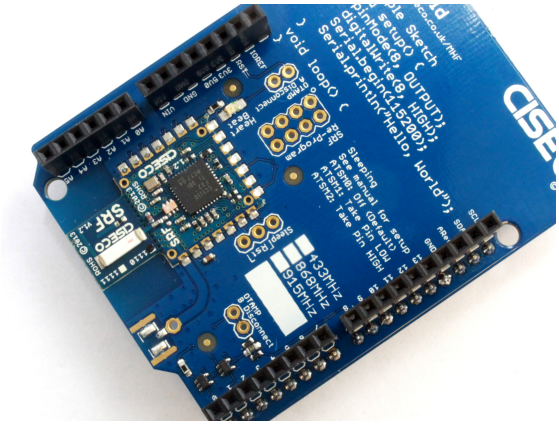
9. Advanced Options

9.1. Additional headers

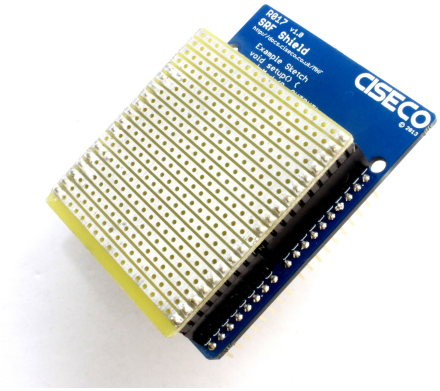
It is possible to add standard 2.54mm (0.1") connectors to the shield for easy prototyping (pic1.). The connectors are spaced so that standard Vero or similar style board can be utilised (pic2.).

It is worth noting that covering the antenna section could reduce the maximum range. To cater for this, an external antenna can be added.

Pic1.



Pic2.



9.2. AT Configuration commands

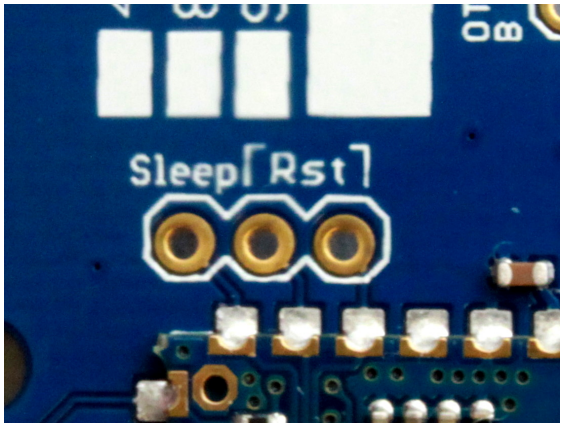
This device like all Ciseco wireless modules uses standard AT commands to change all settings. A full list and explanation of these AT commands can be found at:

http://openmicros.org/index.php/articles/88-ciseco-product-documentation/260-srf-configuration

9.3. Restoring Factory Settings

To return your SRF Shield to its factory settings, connect (short) the two pins marked RST at initial power up. The setting will remain in this mode until either a. The power is cycled, or b. The settings are changed AND committed to flash memory.

Pic3.



9.4. Over the Air Programming (OTAMP)

OTAMP allows you to wirelessly reprogram your Arduino shaped device. The shield is by default set to 115,200 bps as standard; the boot loader on your micro has to match (eg the Arduino UNO boot loader)

To do this you will require another Ciseco radio plugged into your computer and it's ATRC, ATRI and ATRP settings configured for OTAMP

The default OTAMP settings to "target" a standard SRF Shield are:

```
ATRI --
ATRC 3
ATBD 1C200
```

For a more detailed guide on OTAMP please see this page:

http://openmicros.org/index.php/articles/84-xrf-basics/150-otamp

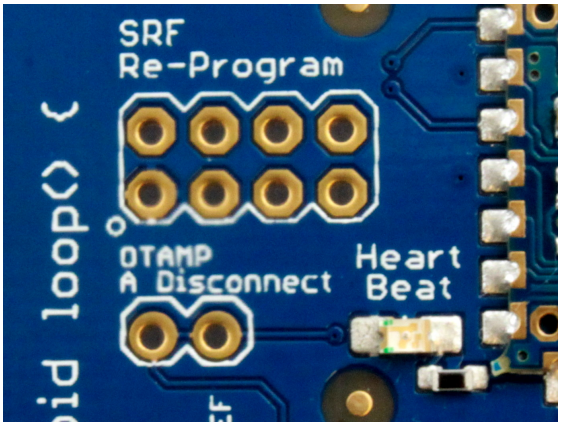
9.5. Disconnecting OTAMP

OTAMP is enabled by default.

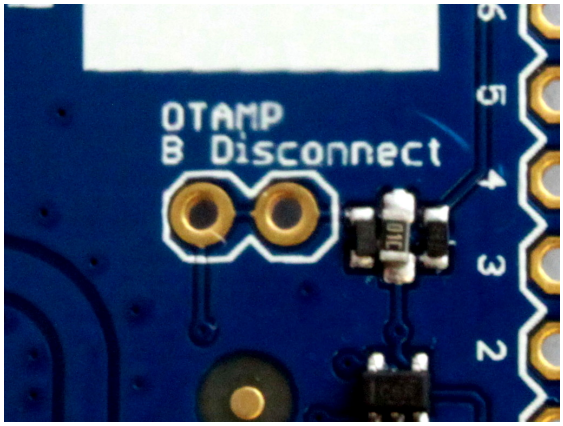
To disable OTAMP cut the two copper traces labelled "A Disconnect" and "B Disconnect" as shown in Pic4. And Pic5.

There is a small copper trace between the two solder holes, carefully cut this trace with a craft knife.

Pic4.

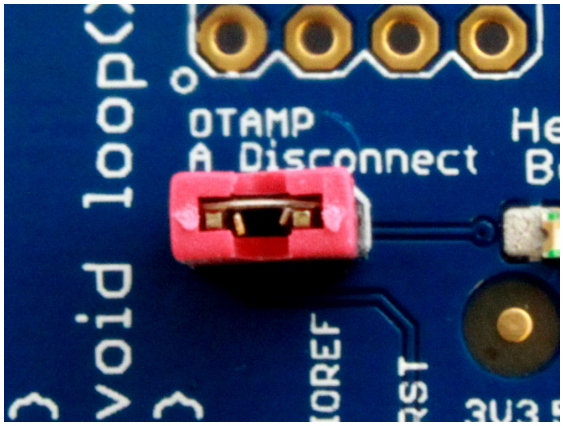


Pic5.

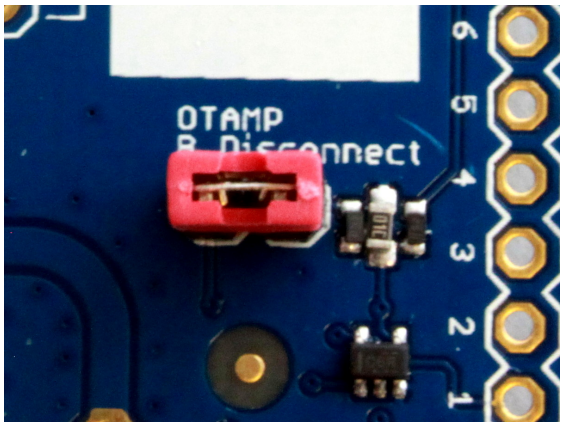


If you wish to re-enable OTAMP at any time, you can solder standard two pin 2.54mm(0.1") pitch headers into the holes and add a jumper link as shown in Pic 6. & Pic 7.

Pic6.



Pic7.

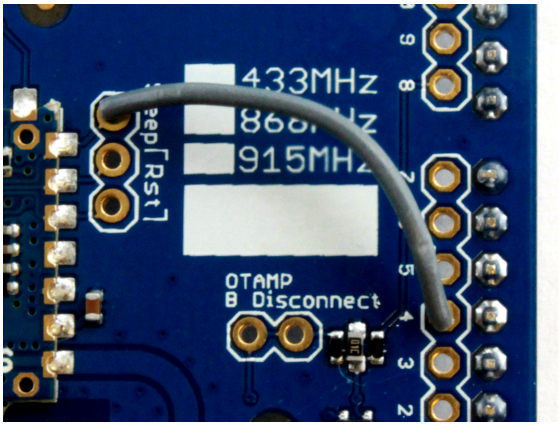


9.6. Sleep modes to save power

To reduce power consumption you can put the SRF Shield into a low power sleep state. This requires a connection to an Arduino I/O pin, to control when the device is to sleep or be awake.

On the SRF shield there is a pin labelled Sleep (Pic 8.) You will need to solder a link between this and your chosen Arduino I/O pin. In the picture below, a link has been soldered between D4 and the Sleep pin. (This happens to match the Ciseco RFu-328 device)

Pic8.



After you have connected the hardware pin, configure the radio module for sleep, this is done via the ATSM command.(see earlier link in 9.2.)

The following modes are available:

ATSM0 = No sleep, the SLEEP pin has no effect

ATSM1 = Light sleep: when the SLEEP pin is set high or un-connected the SRF will run; when the sleep pin is set low the SRF will sleep (power consumption when sleeping of around 150uA)

ATSM2 = Deep sleep: when the SLEEP pin is set low the SRF will run, when the sleep pin is un-connected or set high the SRF will sleep. This is the sleep mode with the lowest sleeping power consumption (<0.2uA)

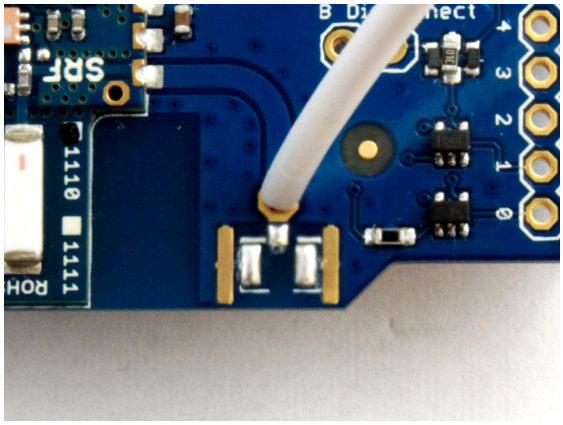
9.7. External antennas

If you wish to extend the range of your SRF Shield you can add an external antenna. There are three ways supported by the PCB.

9.8. Wire Whip

This is the simplest and often most effective way to increase range. Solder the correct length of wire (82.2mm by default) to the hole indicated below.

Pic9.



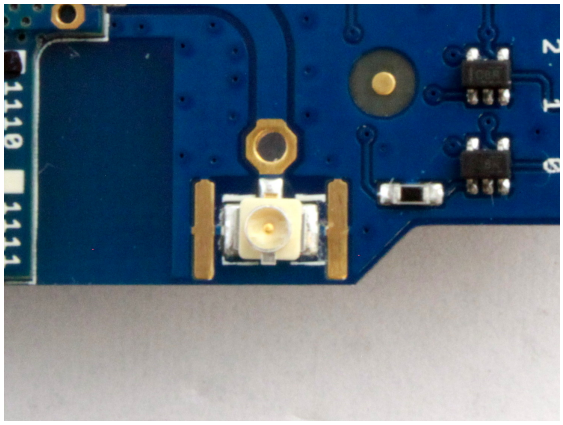
The Length is based on your configured frequency:

- 868MHz = 82.2mm
- 915MHz = 77.9mm

9.9. U.FL connector

You can solder a U.FL connector to the pads indicated below. The pad layout is for a Hirose standard SMD part.

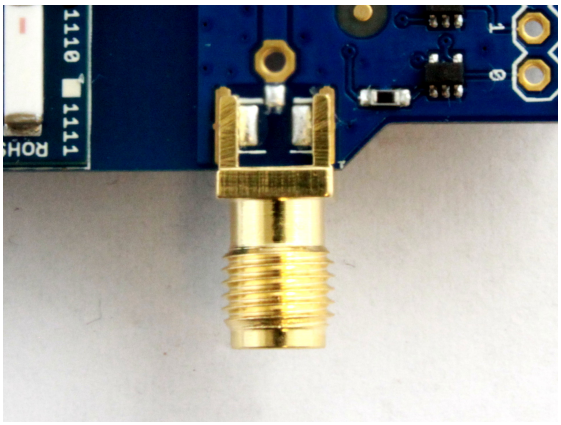
Pic10.



9.10. SMA connector

You can solder an SMA connector to the pads indicated below

Pic11.



NOTE: Take care not to exceed your local regulations transmit limit (via high gain antennas) OR operate a frequency which is not an allowed ISM band in your country.

10. SRF Flash Update

Your SRF Shield will have been shipped with the most recent firmware. Most firmware updates are normally minor, **we therefore strongly suggest, unless otherwise instructed, not to re-flash devices.**

Our firmware release notes, detail any changes that have been made, they can be found here.

http://openmicros.org/index.php/articles/84-xrf-basics/224-firmware-release-notes

The SRF Shield uses the "SerialPassThru-Vx.xx-24MHz-UARTSRF.bin" version of our firmware (where x.xx is the version number)

This can be downloaded from our git-hub repository here:

https://github.com/CisecoPlc/XRF-Firmware-downloads

Updating the firmware is done via XCM, our windows based configuration utility. You will require a separate USB to TTL Serial device; this **must** be a 3V3 device.

You will need to solder on a header to the programming pins.

Using appropriate jump wires, connect the following pins between your TTL Serial cable and the SRF Shield.

SRF Shield Programming Pin	SRF Shield Name	Serial Pin Name
1	GND	GND
2	3V3	3V3
7	P0_3/TX	RX
8	P0_2/RX	TX

Once everything is connected see the following guide on using XCM for firmware updates

http://openmicros.org/index.php/articles/84-xrf-basics/111-firmware-updating-how-to

11. Separating radio traffic

There are two common methods of restricting wireless data to groups of wireless devices.

11.1: Your radio device will have been set to a PANID or network of 5AA5 by default. You could for example put some wireless devices on AAAA and some on AAAB. Devices set for the AAAA network will ignore traffic on the AAAB network and vice versa.

Ciseco radios have a unique function, to extend range or bridge networks you can put a wireless device into repeater mode. When in this mode two PANID's are used. The traffic from one PANID is rebroadcast on the other PANID.

Note: Because all devices are half duplex you can add a rebroadcast delay time to prevent simultaneous "over broadcast"

11.2: In some regions it is possible to use more than one frequency. In this scenario traffic even on the same PANID will not be heard by another device on a different frequency.