

Programmers Guide Sonmicro SM130 RFID module

1. *Installing I2C firmware on Sonmicro SM130 RFID module.*

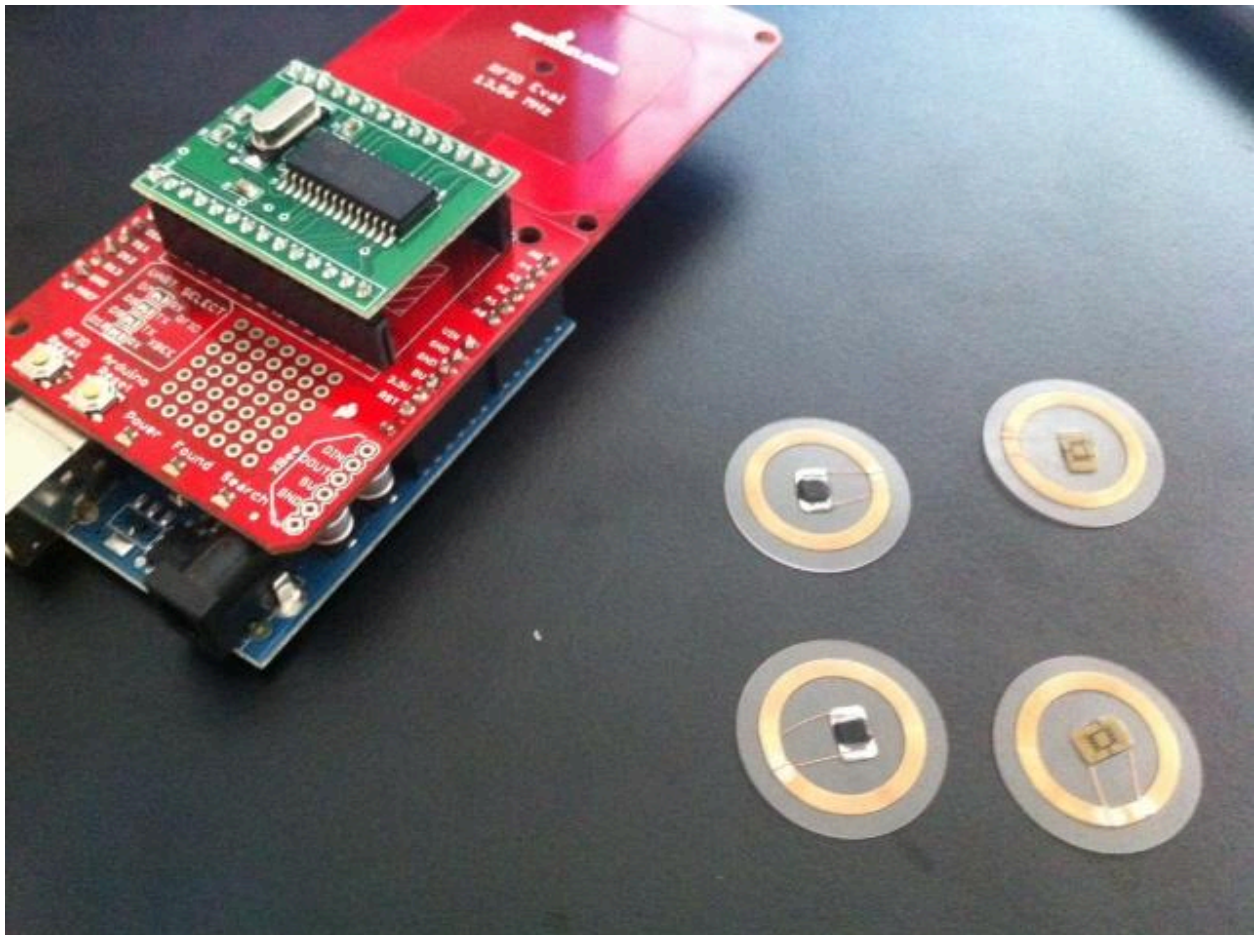
By Joost Aanen & Collaborators

This project was done with the Sonmicro SM130 Module, the Sparkfun RFID evaluation shield and an arduino deumilanove.

Arduino Shield: <http://www.sparkfun.com/products/10406>

SM130 RFID Module: <http://www.sparkfun.com/products/10126>

RFID MIFARE tags: <http://www.sparkfun.com/products/10128>



Before upgrading to new firmware try your standard module with the standard UART protocol and this example code for arduino:

http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Dev/Arduino/Shields/RFID_Eval_v10_Write.pde

It might already be what you are looking for and you will not have to upgrade. For this code to work however you will have to download the NewSoftSerial library for arduino:

<http://arduiniana.org/NewSoftSerial/NewSoftSerial10c.zip>. Download, unzip and place in arduino library folder for it to work.

Upgrade

If you still want to upgrade download the upgrade software tool from Sonmicro:

http://www.sonmicro.com/en/downloads/Mifare/SM130_FU.zip In order to use it you will have to unzip the file first using winrar or winzip.

firmware

You can find the firmware upgrade for I2C here:

http://www.joostaanen.com/RFID/SM130/Firmware/i2c_28_b1.rme

and to restore your SM130 back to factory UART the firmware can be downloaded here:

<http://www.joostaanen.com/RFID/SM130/Firmware/um13d1.rme>

A firmware file that supports anti-collision (for multiple tag reading) can be downloaded here:

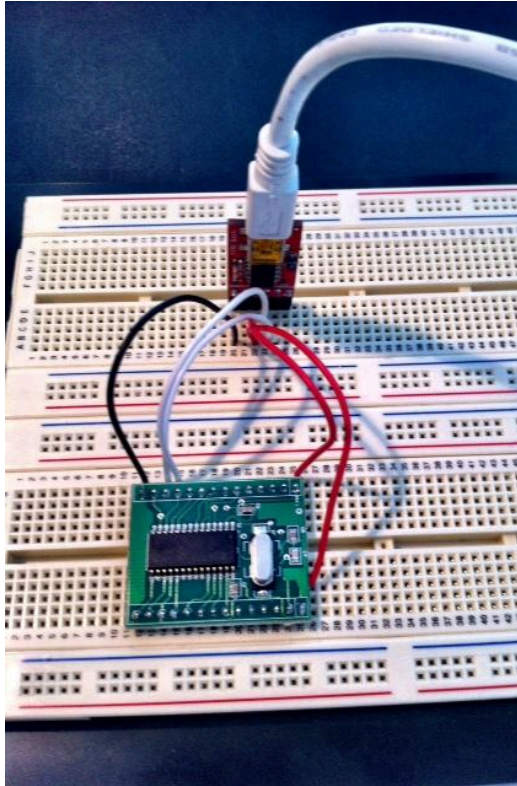
<http://www.joostaanen.com/RFID/SM130/Firmware/um13g.rme>

Connect the SM130 module to USB through a FTDI cable (easier) or a chipless arduino.

Beware what way round you place the SM130. A clear view can be found a the table below.

Name	Arduino (colour in the photo)	FTDI Colour	SM130 pin #
Vcc/+5V	5V (red and yellow)	red	1 and 28
GND	Gnd (green)	black	14
UART RX	Digital 0 (white)	orange	11
UART TX	Digital 1 (blue)	yellow	12

Via FTDI cable:



Via Chipless Arduino:

http://www.mcqn.net/mcfilter/archives/computers/upgrading_the_firmware_on_the_sm130_rfid_module.html

Either way, run the firmware upgrader tool from sun micro (link at the top of this document) and select the .rme file the you wish to install. Make sure you have the right COM port selected. Click auto upgrade and you're done. (For a more elaborate description visit http://www.mcqn.net/mcfilter/archives/computers/upgrading_the_firmware_on_the_sm130_rfid_module.html)

Before you put the module back into the RFID shield make sure that the SM130 module has the correct I2C slave address. This should be 0x42.

To check this you have to download a different tool from the Sonmicro website: http://www.sonmicro.com/en/downloads/Mifare/smrfid_mifare_v1_2_1.zip unzip the file and run.

Set the baud rate in the left top corner to 19200 and make sure you have the right com port selected.

While your SM130 module is still connected through the chipless arduino or FTDI cable click 'hardware commands' in the top menu and click 'read I2C adress'. If this returns 42 (or 0x42) everything is good. To make sure you can cick 'set I2C adress' in 'hardware commands' and set the adress to 42 (or 0x42).

Now reconnect your module to the RFID shield and upload the I2c program to the arduino. For several example programs can be found in Marc Boone's library:

<https://github.com/marcboon/RFIDuino/zipball/master> unpack this zip and place it in the libraries folder of arduino. Make sure to set the baud rate in the serial monitor to 115200 as is specified in the example code.

Run the example codes provided in this document, it will work :).

2. Reading Multiple Tags in RF field.

With a special firmware (UART) function the SM130 will be able to halt tags in the RF field after reading. this tag will become passive until it is taken from and reenters the RF field. this way large numbers of tags (entering the RF field one by one) can be read. The difference between the this firmware and the factory set firmware is that the HALT command instead of of halting the PICC halts the active tags in the RF field. Unfortunately this firmware does not support I2C.

The firmware for this can be found here:

<http://www.joostaanen.com/RFID/SM130/Firmware/um13g.rme>

Follow the instruction on how to install new firmware above to rewrite the um13g.rme (UART) firmware on the SM130

A modified example code now including a HALT command after a tag is found can be downloaded here:

http://www.joostaanen.com/RFID/SM130/Coding/RFID_multiple_tags_example.pde/ make sure to set the baud rate of your serial window to 19200.

3. Going from evaluation shield to prototype

This section describes the next step to take after you've got everything working with the evaluation shield. It involves increasing the reading distance of the RF field, getting rid of the evaluation shield, getting rid of the USB cable and switching to an Xbee connection, and moving to a smaller Arduino.

If everything is working and you want to actually build your actual design the whole assembly will have to become smaller and ready to be integrated into your product.

The first (optional) step will be to switch communication methods from USB cable to a wireless Xbee connection. there are a lot of good Xbee communication tutorials online so I will not go

into this. Here is an arduino code provided that communicates both through USB and Xbee: http://www.joostaanen.com/RFID/SM130/Coding/RFID_multiple_tags_Xbee_example.pde make sure to set the baud rate of your serial window to 19200.

The second step is to get rid of the evaluation shield. to do this you will have to switch to an external RF antenna. In my design I will build the antenna into a fruit bowl, so I chose the RF antenna coil from RFsolutions. I ordered it online at Element14 (spark fun sadly doesn't sell these): <http://au.element14.com/rf-solutions/ant-1356m/antenna-coil-rfid-13-56mhz/dp/1304031> (Element14 has different prices in every country. The link goes to the Australian website, make sure you go to the website of your own country).

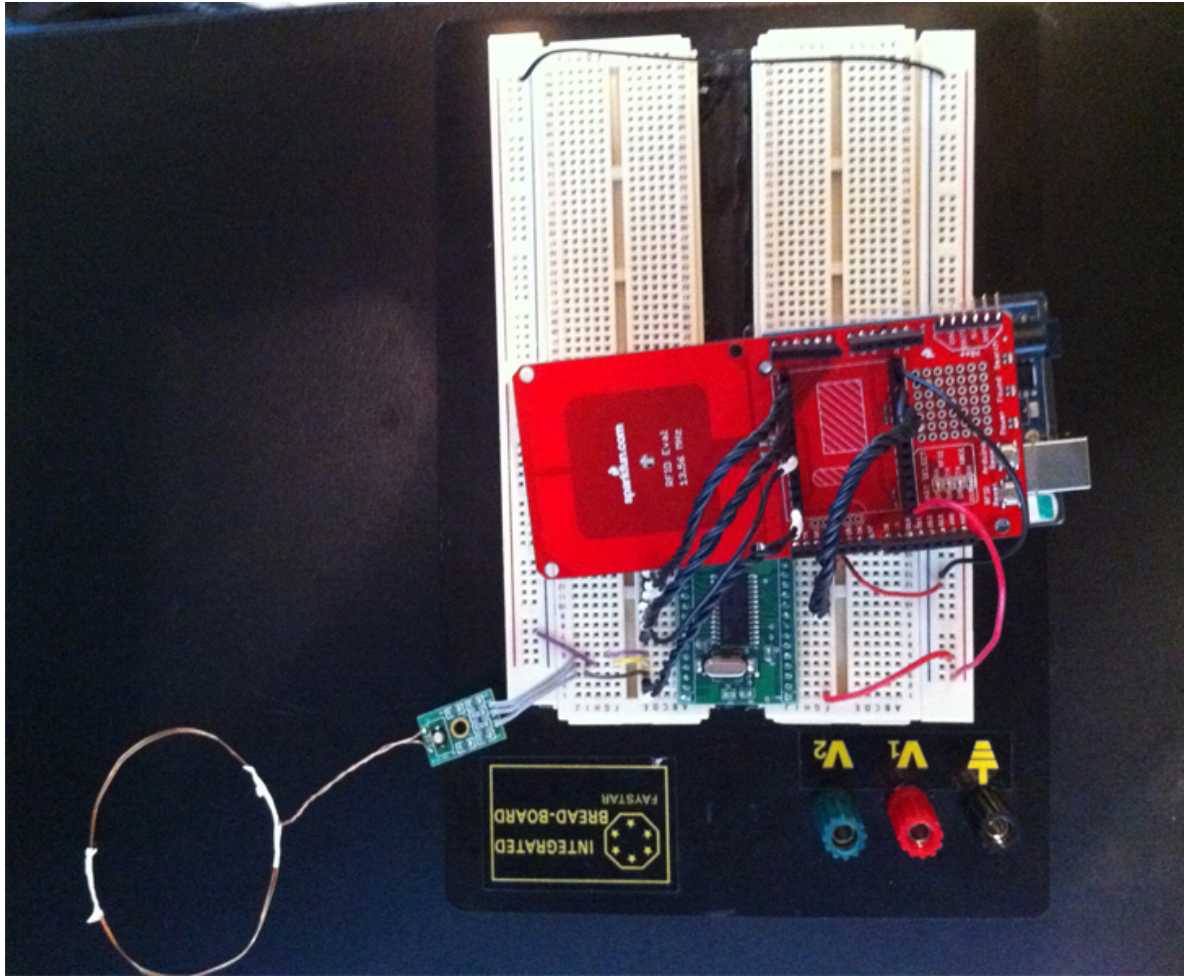
To get rid of the evaluation shield you will have to know exactly what it does and what connections it makes between the Arduino and the SM130. therefore you will need to take a look at the schematics: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Dev/Arduino/Shields/RFID_Eval_13.56MHz-v14.pdf

This schematic shows all the components related to the evaluations shield. however it shows a weird view of the SM130 because not all the pins are pictured in the right order. it notes the pin numbers so using that I provided a more clear overview:

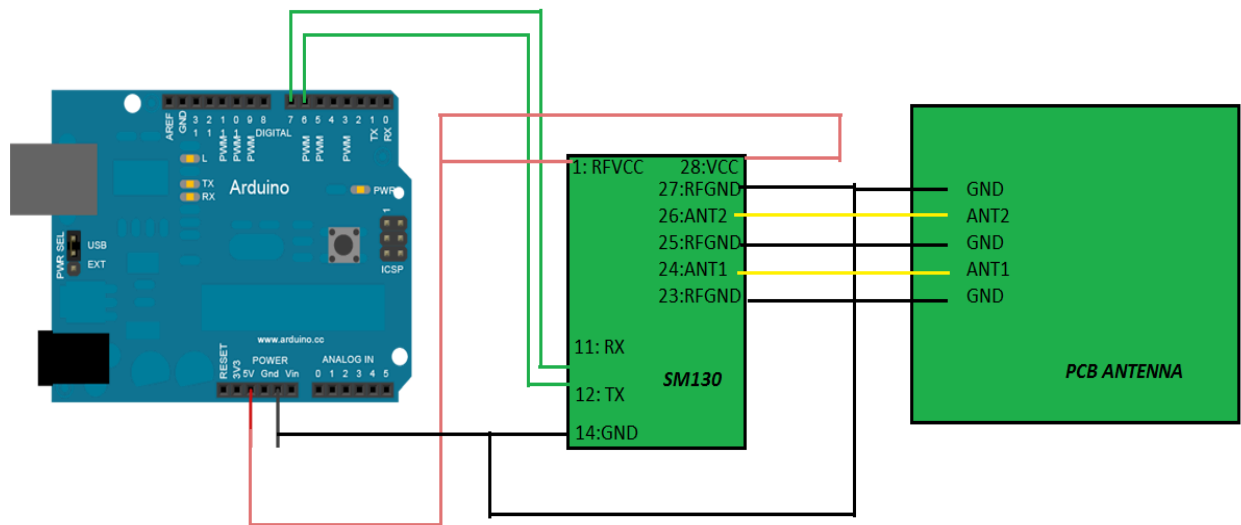
<http://www.joostaanen.com/RFID/SM130/scematic SM130.pdf/>

Using a breadboard you can connect all the pins from the SM130 to the evaluation shield for now. Switching the ANT1 on pin 24 and the ANT2 on pin 26 with the external antenna will allow us to test the antenna.

I've noticed that this antenna gives me a longer range too (5 cm).



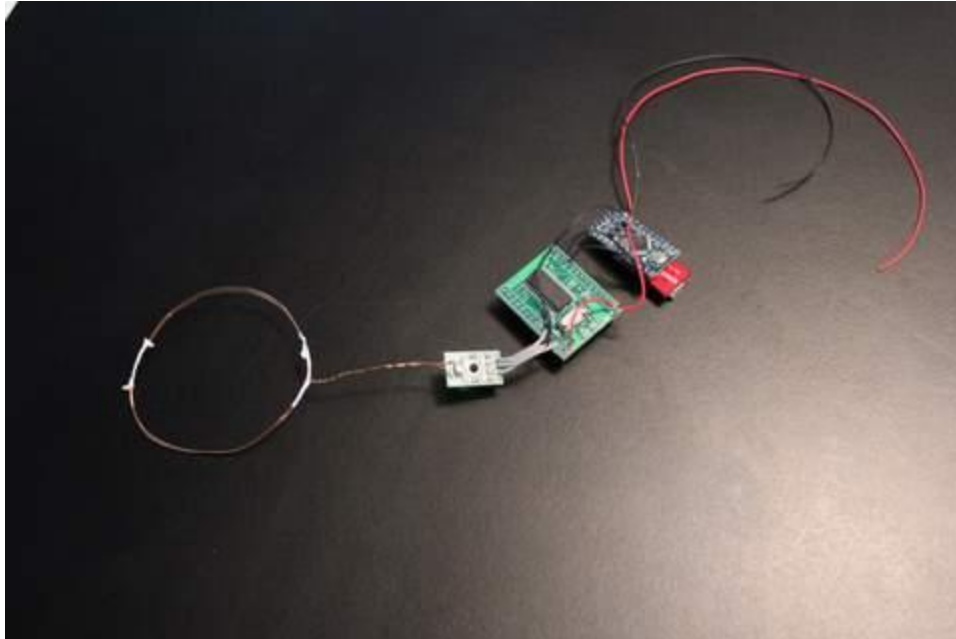
The next step would be to completely get rid of the board and connect the SM130 directly to the Arduino. I've managed to do this through this very easy setup:



(source: http://www.avdm.nl/wp-content/uploads/2011/03/Arduino_sm130_scheme.png)

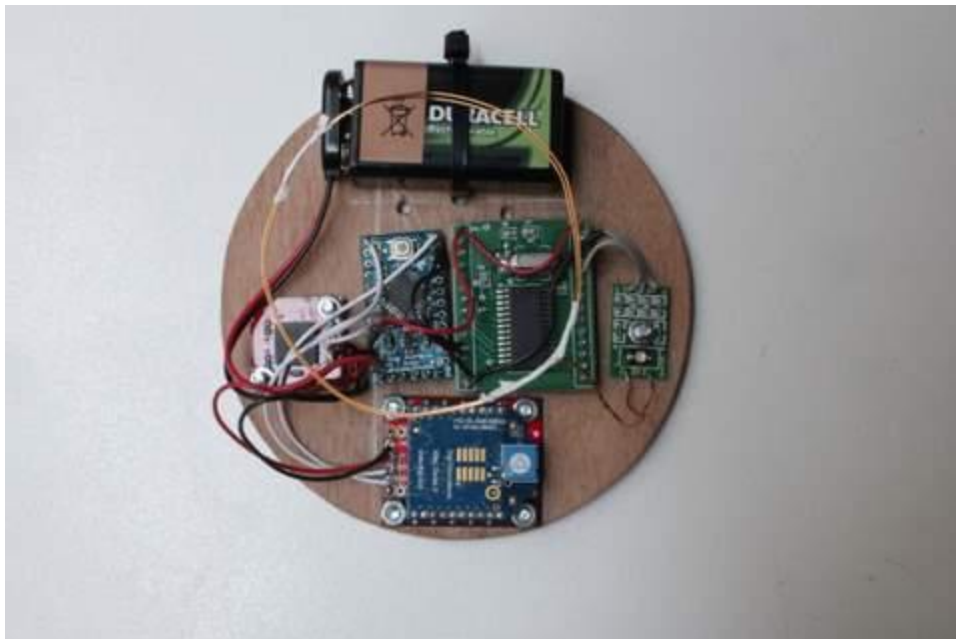
If you want to continue RFID projects and take it to a higher level at might we worthwhile to check out <http://www.openbeacon.org/>. They have a lot interesting full scale RFID project going.

The next step I took was to go to a smaller arduino, a 5v arduino pro mini.



It is exactly the same as with the bigger arduino. The mini pro is a lot more delicate though so be careful not to touch the chip.

The final setup looked like this (including a tilt compensated compass)



Documentation

optional:I2C Usage is explained in the following application note.

<http://www.sonmicro.com/en/downloads/Mifare/AN601.pdf>

SM130 Datasheet:

http://www.sonmicro.com/en/downloads/Mifare/ds_SM132_a7.pdf

Article by Marc Boone:

<https://trac.mediamatic.nl/devcamps/wiki/RfiDuino>

Questions: