

30
MAR
2011

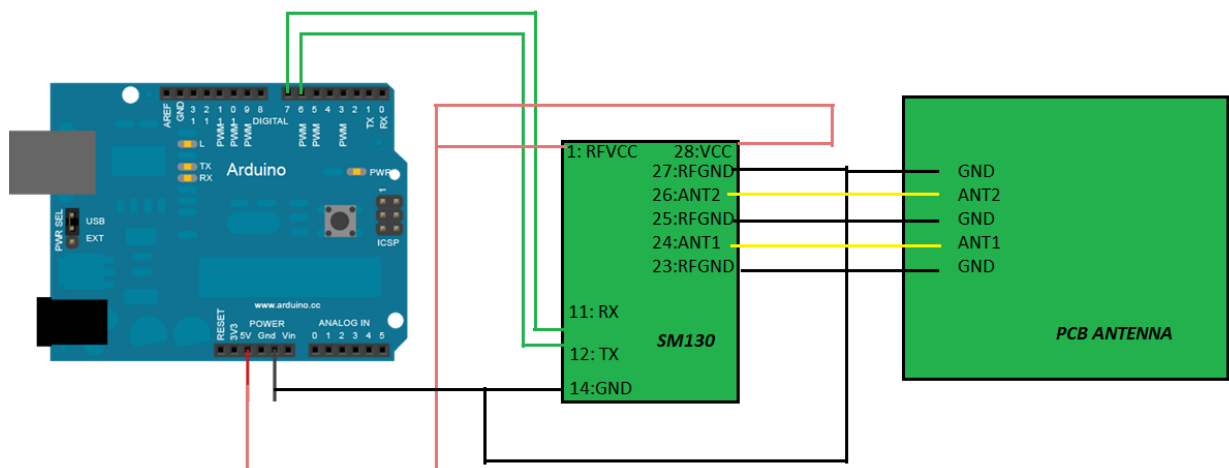
The communication between an Arduino and a SM130 RFID reader through UART

Since both arduinos and RFID-technology become more populair, I decided to put some experiences about the communication between an arduino and a RFID-reader online, in this case the Sonmicro SM130 13.56 MHz RFID Mifare r/w Module.

This module supports UART and i2c. UART is the easiest way to communicate with an arduino, since there is a NewSoftSerial-library that makes it possible to use as many serial communications as needed. Furthermore, the SM130 is preloaded with UART-firmware, which means that a firmware flash is needed for using i2c.

Together with the SM130, we need an Antenna: the Sonmicro 13.56 MHz Mifare PCB Antenna.

Four wires are needed for the connection between the arduino and the SM130; five wires are needed for a connection between the antenna and the SM130. The connection scheme can be found in the figure on the right.



(http://www.avdm.nl/image/arduino_sm130_scheme.png)

The connections between the input-arduino and the SM130 are as following:

- From the 5V on the Arduino to SM130 on:
 - Pin 1:RFVCC
 - Pin 28:VCC
- From the GND on the Arduino to SM130 on:
 - Pin 14:RFGND
 - Pin 27:RFGND
 - Pin 25:RFGND
 - Pin 23:RFGND
- From digital-pin 6 on the Arduino to SM130 on:
 - Pin 12:UART TX
- From digital-pin 7 on the Arduino to SM130 on:
 - Pin 11:UART RX

For the connection between the SM130 and the antenna, we need three wires connected to the ground, and two wires connected to Pin 24 on Pin 26, according to the figure. It turns out that it does not matter whether the antennas ant1 or ant2 is connected to pin 24 or 26.

There is a protocol for the communication between the arduino and the SM130, which can found on the SM130s datasheet

(http://www.sonmicro.com/en/downloads/Mifare/ds_SM130.pdf

(http://www.sonmicro.com/en/downloads/Mifare/ds_SM130.pdf)). The only command needed for scanning a tag is: FF 00 01 82 83.

These five bytes consists of:

- 0xFF: the first byte for indicating a new frame
- 0x00: a reserved byte for future use; it always has to be 0x00
- 0x01: the length of the frame
- 0x82: the real command: seek for tag
- 0x83: Checksum byte

These bytes need to be printed one by one over the serial port to the SM130. As an answer, a lot of hexadecimal code will be received, for instance: FF 00 06 82 02 D4 5A 8D 55 9A. In this case, the found RFIDtag is: **D45A8D55**.

All this knowledge together results in the following simple arduinosketch:

```
//Simple communication sketch for the SM130 RFID-reader.
//The sketch reads a tag and sends it over the Serial port.
//Made by A. van der Meer (AvdM.nl)

#include <NewSoftSerial.h>

NewSoftSerial mySerial(6, 7);

String temp;
String reader;
String tag;
int i=0;

void setup() {
  Serial.begin(57600);
  mySerial.begin(19200);//begin mySerial at 19200(standard baudrate for the SM130)
}

void loop() {

  mySerial.print((char)0xFF); //first byte for indicating a new frame
  mySerial.print((char)0x00); //a reserved byte for future use; it always has to be 0x00
  mySerial.print((char)0x01); //the length of the frame
  mySerial.print((char)0x82); //the real command: seek for tag
  mySerial.print((char)0x83); //CSUM byte

  delay(500); //a small delay is needed for giving the rfidreader time

  while (mySerial.available() > 0) {
    temp=String(mySerial.read(), HEX);
    reader=reader+temp; //compile the answer of the rfidreader
  }

  if(reader!="ff02824cd0"){//ff02824cd0 is the answer if no tag is found
    tag=reader.substring(17,25); //distract the tagid from the answer
    Serial.println(tag); //print the found tag over the Serial port (usb)
  }
  reader="";
}
}
```

select an article

all

tech

hobby

Smart Clothespin v2

(https://www.avdm.nl/articles/tech/clothespin_v2)

De dankbaarheid van een Kip

(https://www.avdm.nl/articles/hobby/de_dankbaarheid_van_een_kip)

Duifjes

(<https://www.avdm.nl/articles/hobby/duifjes>)

Simpele standalone RSS-notifier

(<https://www.avdm.nl/articles/tech/arduinorssnotifier>)

The communication between an Arduino and a SM130 RFID reader through UART