RFIDuino kit

Introduction

RFIDuino is a Arduino shield which can read and write 13.56 MHz ISO-14443A (Mifare) RFID tags. These RFID tags are commonly used in all kinds of plastic cards like access badges, public transport passes, bank cards, library cards, loyalty cards and so on.

The RFIDuino board plugs upside-down on top of a Arduino, maintaining a 20mm board to board spacing. Mounted upside-down, the top is free from electronic components, so the antenna surface can be mounted close to a non-conductive enclosure. Typical reading distance is 3 cm.

RFIDuino interfaces to Arduino using the I2C bus. This two-wire serial bus uses pin A4 and A5 of the Arduino, and is handled by the Wire library. Two other signals are used for a hardware RESET and a DREADY signal, connecting to IO3 and IO4 of the Arduino. Together with 5V and GND there are a total of 6 connections between RFIDuino and Arduino.
By using the I2C bus, the serial port of Arduino remains free for other applications, like connecting to a host computer or other peripherals.

RFIDuino is based on the SM130 RFID module from SonMicro. Check the SM130 datasheet for specifications and command set.

Note for Arduino Mega users: the Arduino Mega moved SDA/SCL from A4/A5 to a new ‘communications’ header. You will have to manually connect SDA and SCL to your RFIDuino, or solder two wires on the bottom of your Arduino Mega board, one from A4 to SDA, and one from A5 to SCL. In the latter case, don’t use A4/A5 for anything else.

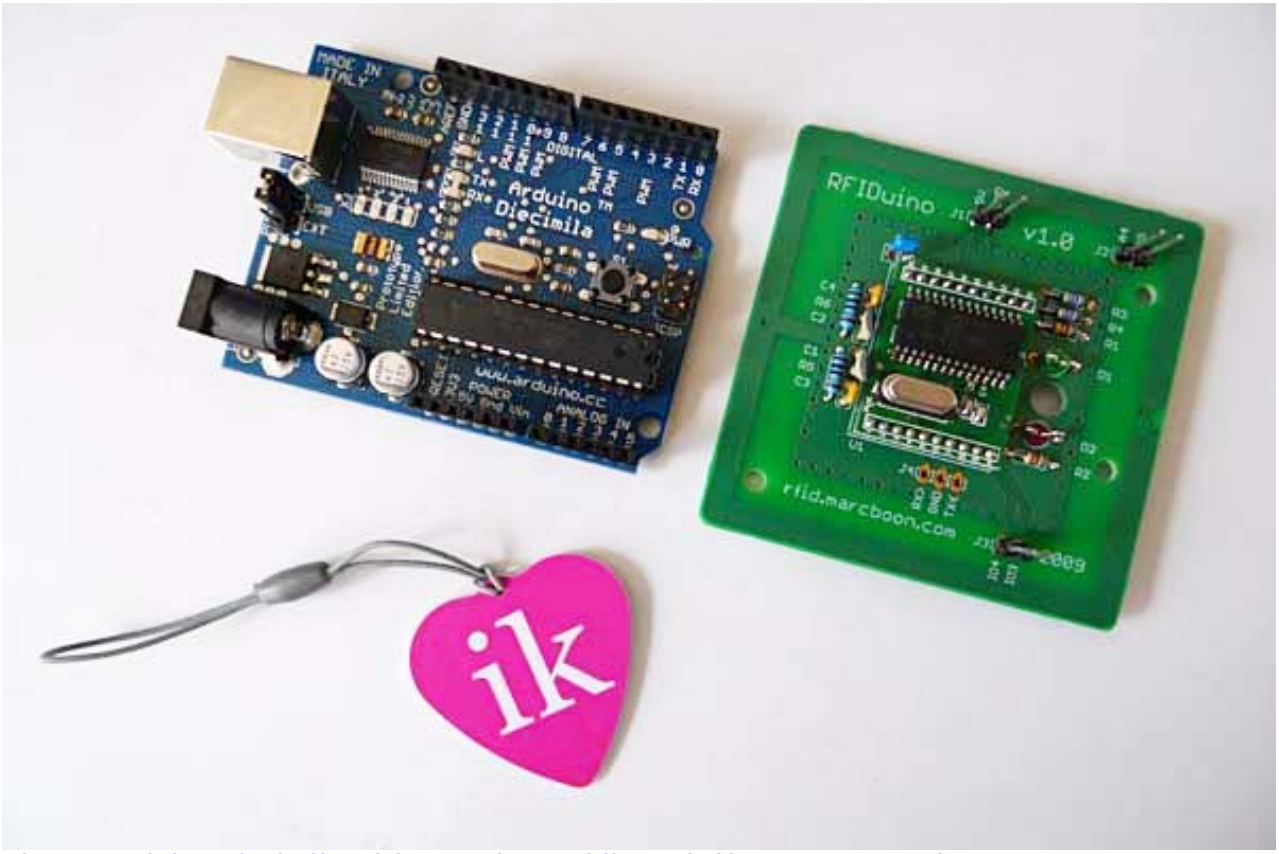

Fig. 1 - Arduino Diecimila with RFIDuino and ikTag (Mifare 1K RFID tag)

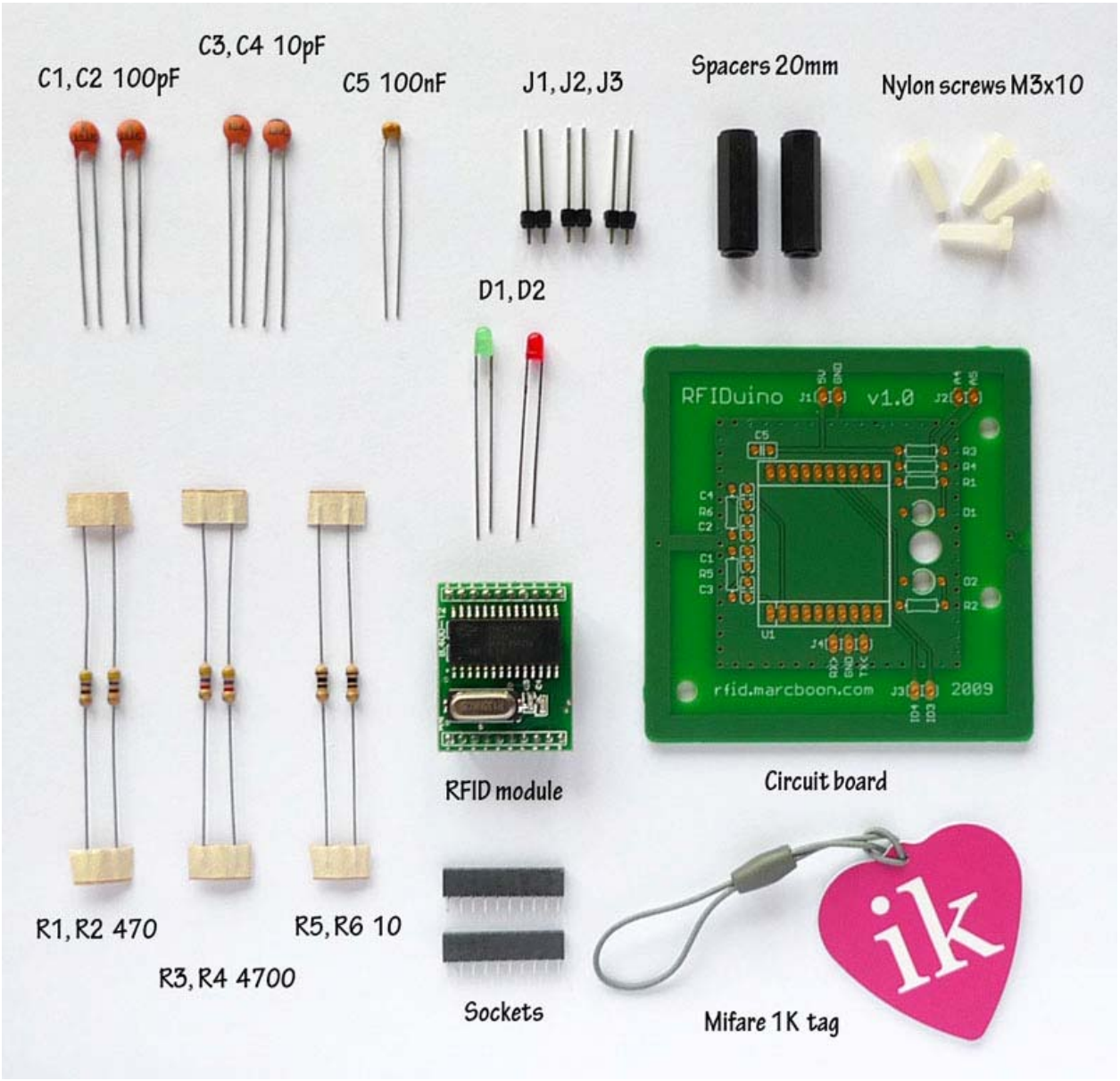

Fig. 2 – RFIDuino kit contents

Kit contents

- Printed circuit board
- Components:
  2x 100pF capacitor (C1, C2)
  2x 10pF capacitor (C3, C4)
  1x 100nF capacitor (C5)
  2x 470Ω resistor (R1, R2)
  2x 4700Ω resistor (R3, R4)
  2x 10Ω resistor (R5, R6)
  1x green 3mm LED (D1)
  1x red 3mm LED (D2)
  1x RFID module (U1)
  2x socket for RFID module
  3x 2-pin header (J1, J2, J3)
  2x threaded nylon spacer 20mm
  4x nylon screw M3x10
- Mifare 1K tag, courtesy of Mediamatic.net

Assembly instructions

All parts are through-hole mounted, at the side with the white text printed on it.

Start by mounting the resistors. The legs can be bent at a right angle using your fingers, so they will have the correct 0.3" spacing.
R1 and R2 have a value of 470Ω and are color-coded with *yellow - violet – brown – gold*.
R3 and R4 have a value of 4700Ω and are color-coded with *yellow - violet – red – gold*.
R5 and R6 have a value of 10Ω and are color-coded with *brown - black – black – gold*.

After soldering the resistors, and trimming their legs using the wire cutter, mount the capacitors.
C1 and C2 have a value of 100pf, and a orange body marked *101K*.
C3 and C4 have a value of 10pf, and a orange body marked *10J*.
C5 has a value of 100nf, and a yellow body marked *104*.

The LEDs D1 (green) and D2 (red) are mounted with their head sticking through a hole in the board, so they are visible from the top when the board is mounted upside-down on the Arduino. Using the long-nose pliers, hold one of the wires close to the head of the LED, and bend the long end outward 180 degrees, making a U-turn. Do the same with the other leg, so both are parallel to the head of the LED. Bent like this, the LED can be mounted with its head just sticking through the 3mm hole, and the legs sticking through the solder pads. Make sure the LEDs are mounted in the right orientation. The longest leg (anode) should be on the outside of the board.

Fig. 3/4 – Bending the legs of the LEDs

Finally, the sockets for the RFID module and the pin headers are mounted. Make sure they are aligned vertically, otherwise you will have problems plugging the board onto your Arduino. Note: J4 is left empty.

Carefully check the solderpads for any shorts, using a magnifying glass. You may want to clean the solder flux residue by wiping the pads with a cotton swab dipped in thinner.

When all parts are soldered, and double-checked for proper placement, plug in the RFID module. Make sure pin 1, which is near the crystal, is at the bottom-left corner, marked U1. See Fig.1 for proper placement. Plugging in the module reversed can permanently damage it!

Fix the nylon spacers to the opposite corners of the board (the third hole is not used) using two nylon screws. Then plug the board upside-down on your Arduino, while carefully checking the proper alignment of the pin headers, and secure it with the two remaining screws.

Testing

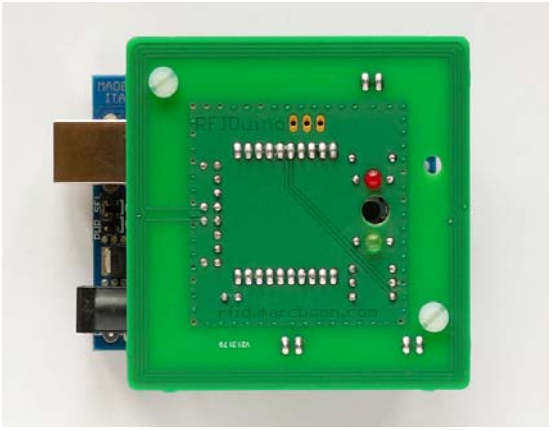

Fig 5. Mounted RFIDuino, top view

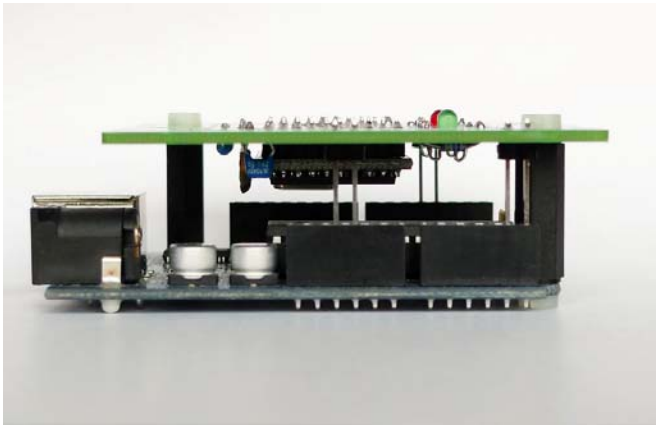

Fig 6. Mounted RFIDuino, side view

Open the Arduino IDE on your computer, and connect the Arduino. The test software can be downloaded here: http://marcboon.com/rfiduino/code/rfiduino.pde.

After uploading the code to your Arduino, open the Serial Monitor and set it to 115200 baud. After reset, the version of the SM130 module (I2C 2.8) should be displayed in the Serial Monitor window. If it doesn't, something is wrong. After printing the version number, the RFIDuino goes into SEEK mode. The green led on the RFIDuino indicates it is seeking tags. When a RFID tag is placed over the antenna, the red LED should light up, and the type and id of the tag is displayed in the Serial Monitor window.

Besides this simple test program, there is also a complete SM130 library for Arduino, including some demo programs, demonstrating reading and writing the tag's memory. The SM130 library and examples can be downloaded from http://marcboon.com/rfiduino/code/SM130v1.zip. After downloading, unzip the SM130 folder in the hardware/librarires folder of your Arduino installation, then restart the Arduino IDE. The examples can be found in File | Sketchbook | Examples | Library-SM130. API documentation is here: http://marcboon.com/rfiduino/code/SM130/index.html.

Troubleshooting

A common problem, especially with Macbooks, is insufficient power from the USB port. This will cause malfunction of the RFIDuino. Part of the problem is that the Arduino advertises itself to the host computer as consuming not more than 90mA. This value is hardcoded in the USB descriptor of the Arduino. However, the actual power consumption of the Arduino/RFIDuino combo is higher, about 200mA. This can shutdown the USB power on some systems. Using an external power supply for your Arduino will solve this issue.

Another issue is the reading distance of tags. It depends heavily on the size of the antenna in the tag. The supplied ikTags have a small antenna, so they might not work properly unless placed on the corners of the board, where the tag antenna has maximum coverage with the pcb antenna. Standard credit-card sized tags have a reading distance of about 3-5 cm across the whole surface of the RFIDuino antenna.