

RFID Quick Start Guide: Arduino

Understanding RFID

RFID, or Radio Frequency Identification, is a system for transferring data over short distances (typically less than 6 inches). Often only one of the two devices needs to be powered, while the other is a passive device. This allows for easy use in such things as credit cards, key fobs, and pet collars as there is no need to worry about battery life. The downside is that the reader and the information holder (ie credit card) must be very close, and can only hold small amounts of data.

In this tutorial we will be using the MFRC522 13.56Mhz IC by MIFARE, as described at <http://www.addicore.com/product-p/126.htm>.

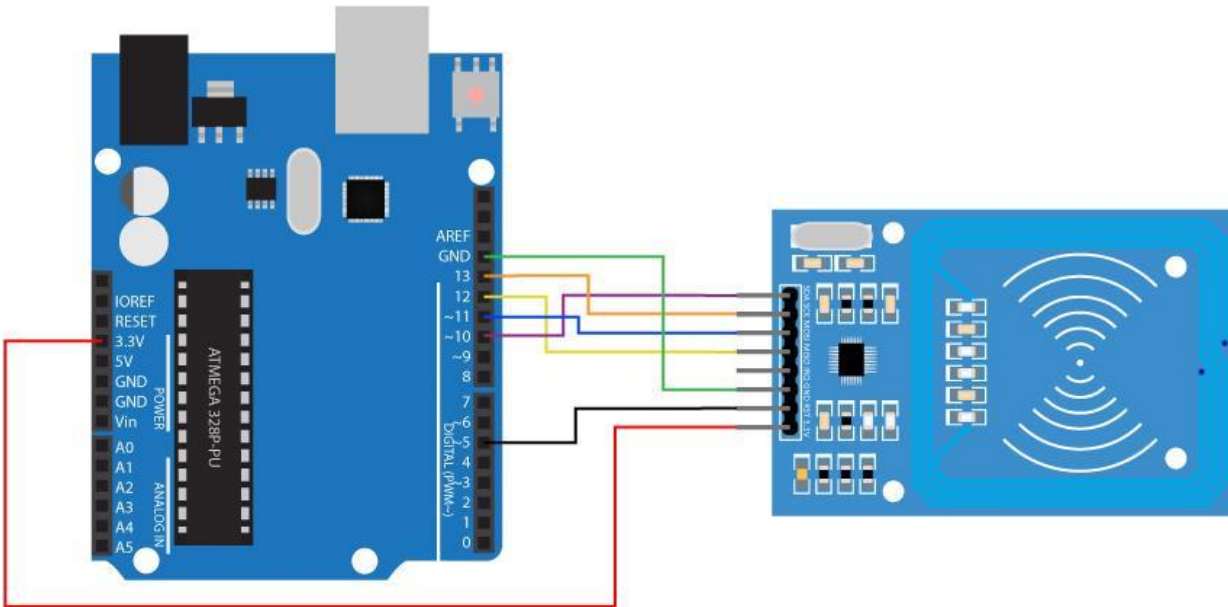
Wiring

The following table shows the needed connections between the RFID and the Arduino Uno
 Cautions:

*On the Arduino many of the pins are not swappable. Because this device uses the SPI bus, who's pins cannot be moved around, pins 11, 12, 13 must remain as shown. RST and IRQ are user specified.

*This device is NOT a 5 volt powered device. You MUST power it with 3.3 volts. If you do not, you risk overheating the RFID. Most Arduino boards include a 3.3V supply pin which can be used to power the RFID module. If 3.3 volts is not accessible, there are [LD33V regulators available at Addicore.com](#) that supply 3.3 volts.

RFID-RC522 Module	Arduino Uno
1 - SDA	Digital 10
2 - SCK	Digital 13
3 - MOSI	Digital 11
4 - MISO	Digital 12
5 - IRQ	--unconnected--
6 - GND	Gnd
7 - RST	Digital 5
8 - 3.3V	3.3v

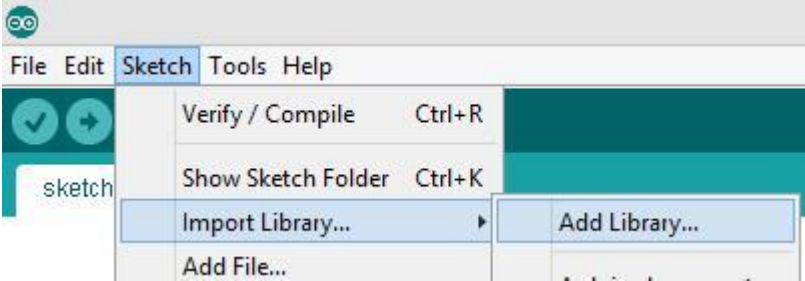


Adding the Library

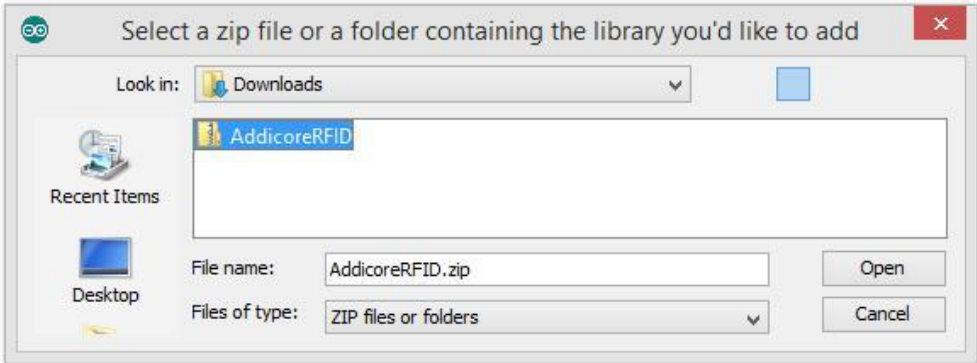
If you haven't already done so, the AddicoreRFID library needs to be added to your Arduino library depository.

1. Download the [AddicoreRFID library](#)
2. Starting with Arduino IDE version 1.0.5 you can automatically ☺ install additional libraries from within the Arduino IDE. If you currently use an Arduino IDE older than version 1.0.5 you can still install the library manually. Arduino provides instructions to do so that can be found at <http://arduino.cc/en/Guide/Libraries> (Scroll down the page until you find the section on “Manual Installation”).

Open the Arduino IDE and navigate to *Sketch > Import Library*. A drop down menu will appear. Click the option to “Add Library.”



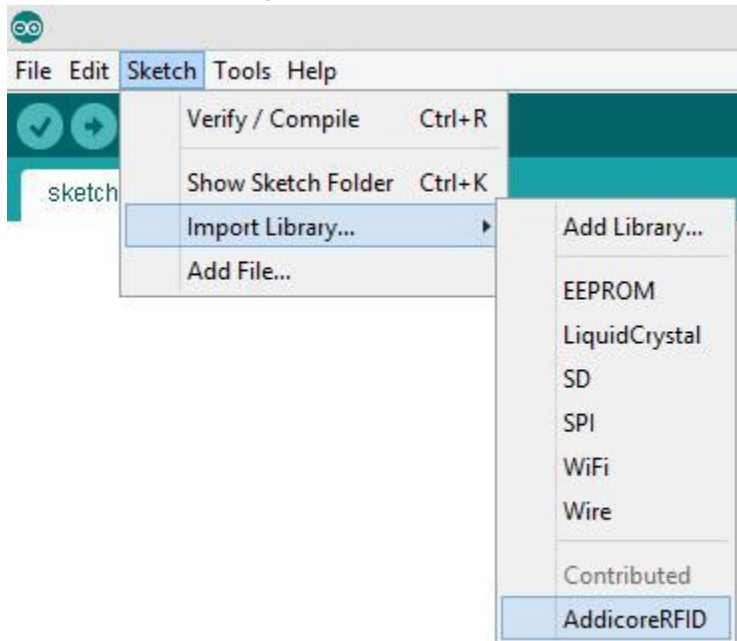
3. A window will open. Navigate to the location where the AddicoreRFID library you downloaded above is currently located and open it.



After the library has been installed the bottom left side of the IDE should show the following message:



4. You can confirm that the library has been installed by again navigating to *Sketch > Import Library*. The AddicoreRFID library should now show in the list of “Contributed” libraries.



Program the Arduino

After installing the AddicoreRFID library in the steps above the library will be available to use in sketches but any example sketches included with the library will not be shown in *File > Examples* until after restarting the Arduino IDE.

1. Restart the Arduino IDE by closing all currently open Arduino IDE windows and then opening the Arduino IDE.
2. When the Arduino IDE opens navigate to *File > Examples > AddicoreRFID*. Select the “Addicore_RFID_Example” sketch. This will open a sketch which we will use with the RFID module that we wired to our Arduino earlier.

A second option instead of opening the example sketch is to copy the code below into a new sketch.

```
// Example sketch to read the ID from an Addicore 13.56MHz RFID tag  
// as found in the RFID AddiKit found at:  
// http://www.addicore.com/RFID-AddiKit-with-RC522-MIFARE-Module-RFID-Cards-p/126.htm
```

```
#include <AddicoreRFID.h>  
#include <SPI.h>
```

```
#define uchar   unsigned char  
#define uint   unsigned int
```

```
//4 bytes tag serial number, the first 5 bytes for the checksum byte  
uchar serNumA[5];
```

```

uchar fifobytes;
uchar fifoValue;

AddicoreRFID myRFID; // create AddicoreRFID object to control the RFID module

////////////////////////////////////
//set the pins
////////////////////////////////////
const int chipSelectPin = 10;
const int NRSTPD = 5;

//Maximum length of the array
#define MAX_LEN 16

void setup() {
  Serial.begin(9600);           // RFID reader SOUT pin connected to Serial RX pin at 9600bps

  // start the SPI library:
  SPI.begin();

  pinMode(chipSelectPin,OUTPUT); // Set digital pin 10 as OUTPUT to connect it to the RFID /ENABLE
  pin
  digitalWrite(chipSelectPin, LOW); // Activate the RFID reader
  pinMode(NRSTPD,OUTPUT); // Set digital pin 10 , Not Reset and Power-down
  digitalWrite(NRSTPD, HIGH);

  myRFID.AddicoreRFID_Init();
}

void loop()
{
  uchar i, tmp, checksum1;
  uchar status;
  uchar str[MAX_LEN];
  uchar RC_size;
  uchar blockAddr; //Selection operation block address 0 to 63
  String mynum = "";

  str[1] = 0x4400;
  //Find tags, return tag type
  status = myRFID.AddicoreRFID_Request(PICC_REQIDL, str);
  if (status == MI_OK)
  {
    Serial.println("RFID tag detected");
    Serial.print(str[0],BIN);
    Serial.print(" , ");
    Serial.print(str[1],BIN);
    Serial.println(" ");
  }
}

```

```

//Anti-collision, return tag serial number 4 bytes
status = myRFID.AddicoreRFID_Anticoll(str);
if (status == MI_OK)
{
checksum1 = str[0] ^ str[1] ^ str[2] ^ str[3];
Serial.println("The tag's number is : ");
  //Serial.print(2);
  Serial.print(str[0]);
Serial.print(" , ");
  Serial.print(str[1],BIN);
Serial.print(" , ");
  Serial.print(str[2],BIN);
Serial.print(" , ");
  Serial.print(str[3],BIN);
Serial.print(" , ");
  Serial.print(str[4],BIN);
Serial.print(" , ");
  Serial.println(checksum1,BIN);

  // Should really check all pairs, but for now we'll just use the first
  if(str[0] == 156) //You can change this to the first byte of your tag by finding the card's ID
  through the Serial Monitor
  {
    Serial.print("Hello Craig!\n");
  } else if(str[0] == 244) { //You can change this to the first byte of your tag by finding the card's ID
  through the Serial Monitor
    Serial.print("Hello Erin!\n");
  }
  Serial.println();
  delay(1000);
}

myRFID.AddicoreRFID_Halt(); //Command tag into hibernation
}

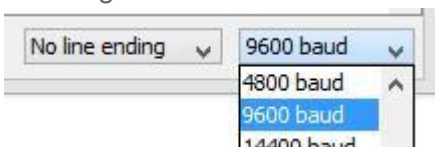
```

3. Now connect your Arduino to your computer and upload the code.

Reading RFID Tags

With your Arduino programmed with the “Addicore_RFID_Example” sketch and with it still plugged into your computer open the serial monitor by clicking the Serial Monitor icon or pressing *CTRL+SHIFT+M*.

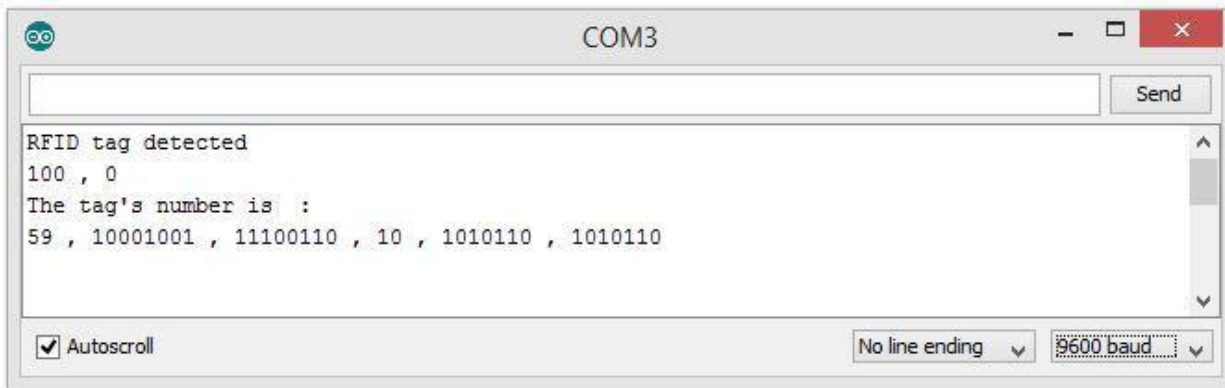
If it isn't already selected set the serial baud rate to 9600 baud by clicking on the drop down selection in the lower right corner of the Serial Monitor window as shown below and selecting 9600 baud.



Now take one of the RFID cards or fobs that came in your RFID AddiKit and hold it near the white graphic printed on your RFID-RC522 module as shown below:



Once the module has read the RFID tag your Serial Monitor should show something similar to the following:



When the RFID reader senses a card, it starts by printing “RFID tag detected” on the Serial Monitor. The next line verifies the card type. In normal use, this would be unnecessary. The next two lines print out the data stored in the card, in this case an ID identifying the specific tag scanned.

Note the number 59 from the screenshot above. This is the first byte of the scanned tag’s ID. The tag you scan will have a different ID than the one used in this example. In the case of this example I will write down 59 but you will need to write down the number preceding the first comma of the tag’s ID that your Serial Monitor shows. Close the Serial Monitor and find in the Addicore_RFID_Example code the following lines of code:

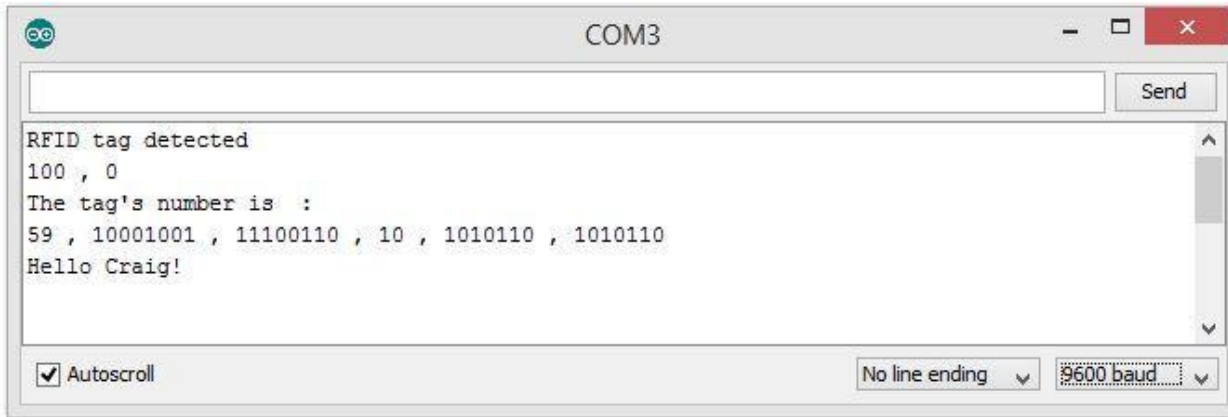
```
// Should really check all pairs, but for now we'll just use the first
if(str[0] == 156) //You can change this to the first byte of your tag
{
  Serial.print("Hello Craig!\n");
} else if(str[0] == 244) { //You can change this to the first byte of your tag
  Serial.print("Hello Erin!\n");
}
```

Change the number in the following line of code to the number you wrote down above, in my case it is 59.

```
if(str[0] == 59) //You can change this to the first byte of your tag
```

You can change the text on the next full line of code to whatever you want to display when you scan the same RFID tag. I will leave mine to say “Hello Craig!”

Upload your changed code to your Arduino, reopen the Serial Monitor, and then scan the same RFID tag as before. Now when you scan the tag the Arduino will recognize the ID and will display the desired text. Below is an example of what mine looks like:



Cautions:

- *Make sure to keep the card close to the reader for enough time. A fast swipe by can cause errors in the printing to the serial monitor.
- *The more you print to the serial monitor, the longer the card or fob has to be near it. This is because compared to the read and write on the card, the serial monitor is slow, and bogs down the system.
- *The sketch shown only validates part of the card ID number. In many systems this will be sufficient. In others, you may want to validate more or all of the ID number. In such systems, consider checking after reading the card in a switch statement.
- *Don't forget to set the correct baud rate on the serial monitor.

Notes & Comments by the Author

This guide has been as simple as I could possibly make it and still have a working setup. This is by no means an exhaustive explanation or use of the module; Rather, this is meant to get you started.

Any errors found in this document can be sent to support@addicore.com.

Reference

The following resources are ones that we found useful, to be taken with a grain of salt. They are subject to change without notice. Use at your own risk.

www.grantgibson.co.uk/2012/04/how-to-get-started-with-the-mifare-mf522-an-and-arduino/

<https://github.com/miguelbalboa/rfid>

http://www.nxp.com/documents/data_sheet/MFRC522.pdf

Copyright

The AddiKit and Addicore names and logos are trademarks of the AddiKit series of Addicore kits and of Addicore LLC. Other products and company names mentioned in this document are trademarks of their respective companies.

All images and illustrations contained in this document are property of Addicore LLC and can not be reused without written permission from Addicore LLC. Addicore is more than happy to grant you permission to use these images for non-commercial uses if you ask and receive permission from us. Images for illustration purposes only. Actual products may vary from illustrations.

The code in this guide is open under the GNU General Public License version 2 as published by the Free Software Foundation.