

TA0013

ARDUINO RFID UNO

STARTER KIT



Overview – TA0013

This Arduino Uno ultimate project kit includes comprehensive range of components to get you started in building and experimenting with Arduino projects. The “RFID Starter Kit for Arduino Uno” comes in a handy case and contains over 30 state-of-the-art components, devices and modules. This is a huge value starter kit includes lots of common electronic components and module; Uno development board, LED's, resistors, sensors, LED displays, RFID readers, relays, stepper motors and more. For complete package list, please refer to our website.

The wide range of peripherals included in the kit ensures the number of different experiments and applications you can build.

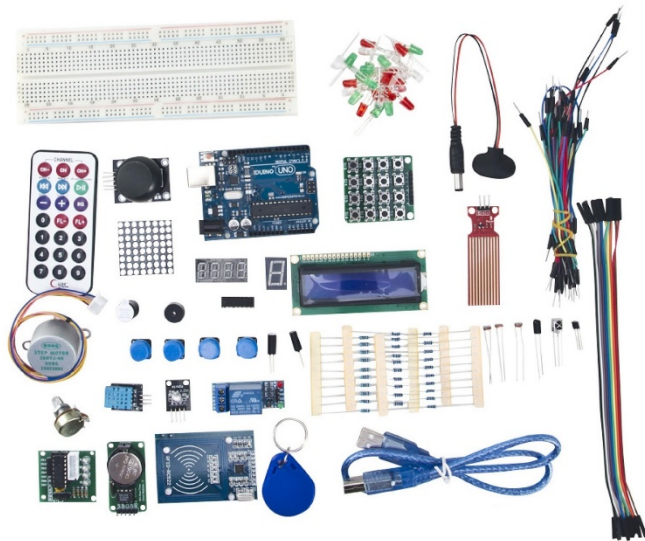
In this instruction, we will walk through IR remote + the receiver, and MFRC522 13.56Mhz IC by MIFARE, as described at:

- <https://www.auselectronicsdirect.com.au/arduino-rfid-uno-starter-kit>

If you would like to purchase additional RFID cards of fobs they can be found at the following:

- <https://www.auselectronicsdirect.com.au/arduino-rfid-read-and-write-module-kit>

Let's get started!



RFID Fundamentals

RFID stands for: *Radio Frequency Identification*

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 need 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.

Each tag has a unique ID number that is read from the sensor and a microcontroller takes it from there. RFID cards/tags cannot hold data like a card with a magnetic strip, such as a metro card which keeps track of when it expires or runs out of money. However, a microcontroller can hold the data instead and have values increase and decrease for each time a recognizable card is swiped.

In addition, unlike a magnetic strip card, RFID sensor do not have to make direct physical contact with the card and can sense a card from over an inch away.

Preparation for RFID project

Prepare the following:

- The red PCB board with a WIFI graphic (outward waves) on it, is the sensor.
- The white card is the card.
- The blue circle attached to a keychain is the tag.
- The Arduino Uno
- Some male to female jumper wires (male to male jumper wires if you a breadboard)



1 X RFID Module



1 X RFID White card



1 X RFID keychain

Wiring

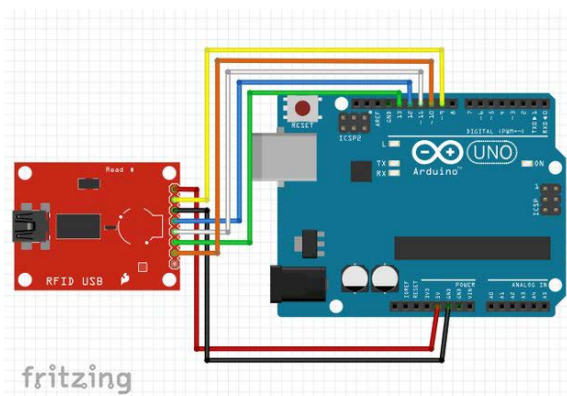
The following table shows the needed connections between the RFID and the Arduino Uno

| RFID Module | Arduino UNO Development Board |
|-------------|-------------------------------|
| Vcc | 5V |
| RST | 9 |
| GND | GND |
| MISO | 12 |
| MOSI | 11 |
| SCK | 13 |
| NSS | 10 |

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. RST and IRQ are user specified. The bottom pin is not being used and it still works perfectly. (That pin is labelled as IRQ)

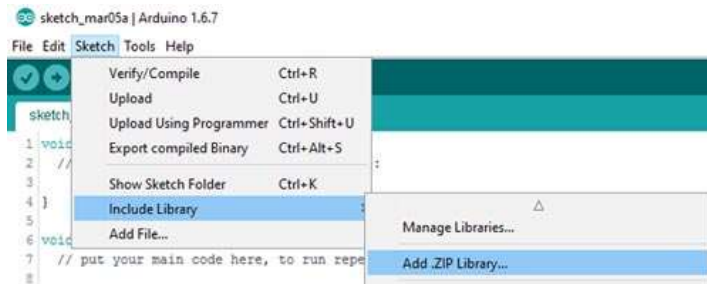


RFID CODE

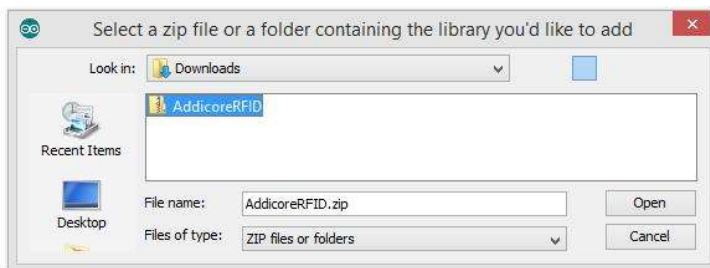
Download the ZIP library at:

<https://www.auselectronicdirect.com.au/arduino-rfid-uno-starter-kit>

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



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:

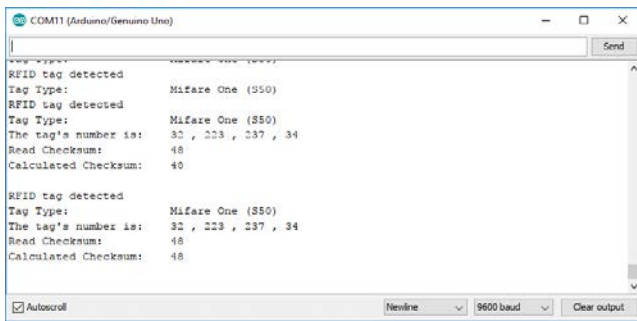
Library added to your libraries. Check "Include library" menu

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

Program the Arduino

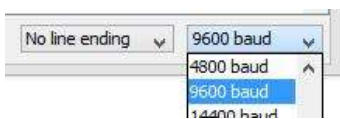
After installing the AddicoreRFID library in the steps above the library will be available to use in sketches but any example sketches included with the library may not be shown in File > Examples until after restarting the Arduino IDE.

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



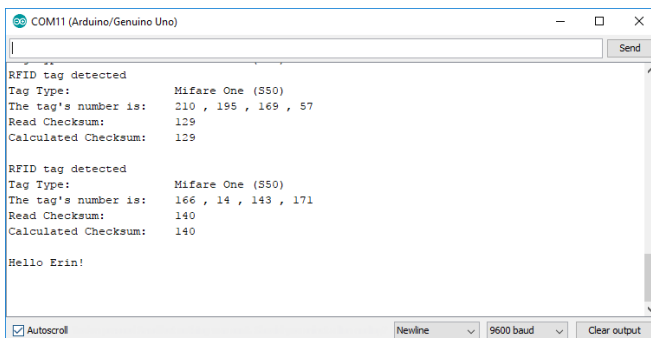
Reading RFID Tags

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.

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 line prints out the data stored in the card, in this case an ID identifying the specific tag scanned.

Note the number 6 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 6 but you will need to write down the number preceding the first comma of the tag's ID number 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] == 32) //You can change this to the first byte of you
{
  Serial.println("\nHello Craig!\n");
} else if (str[0] == 166) { //You can change this to the first byte of y
  Serial.println("\nHello Erin!\n");
}
Serial.println();
delay(1000);
,
```

Change the number in the following line of code to the number you wrote down above, in my case it is 6.

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 Erin!”

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:

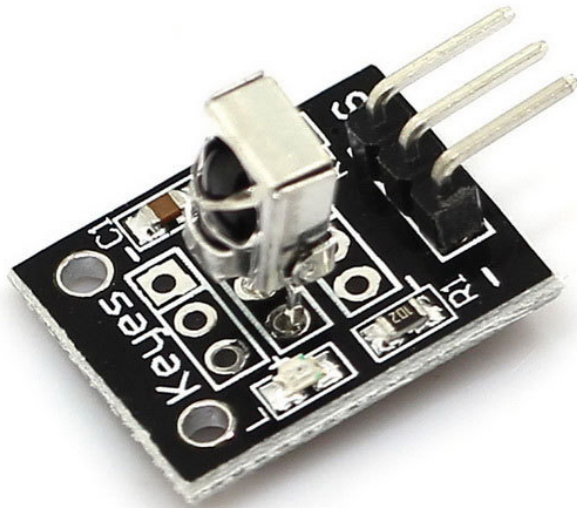
```
RFID tag detected
Tag Type:           Mifare One (S50)
The tag's number is: 166 , 14 , 143 , 171
Read Checksum:     140
Calculated Checksum: 140

Hello Erin!
```

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 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.

Infrared Remote Control



This infrared sensor module has three pins:

- Signal
- GND
- VCC

Wiring the Signal pin to Arduino 11. GND to GND and VCC to 5V.

First, download the library and the code and place the library into the Arduino's library folder. Then upload the code to the Arduino and open the serial monitor. From here you will see "ready to decode" and other stuff but when you press a button you will see "other button" pop up and after a second a number sequence will appear below it.



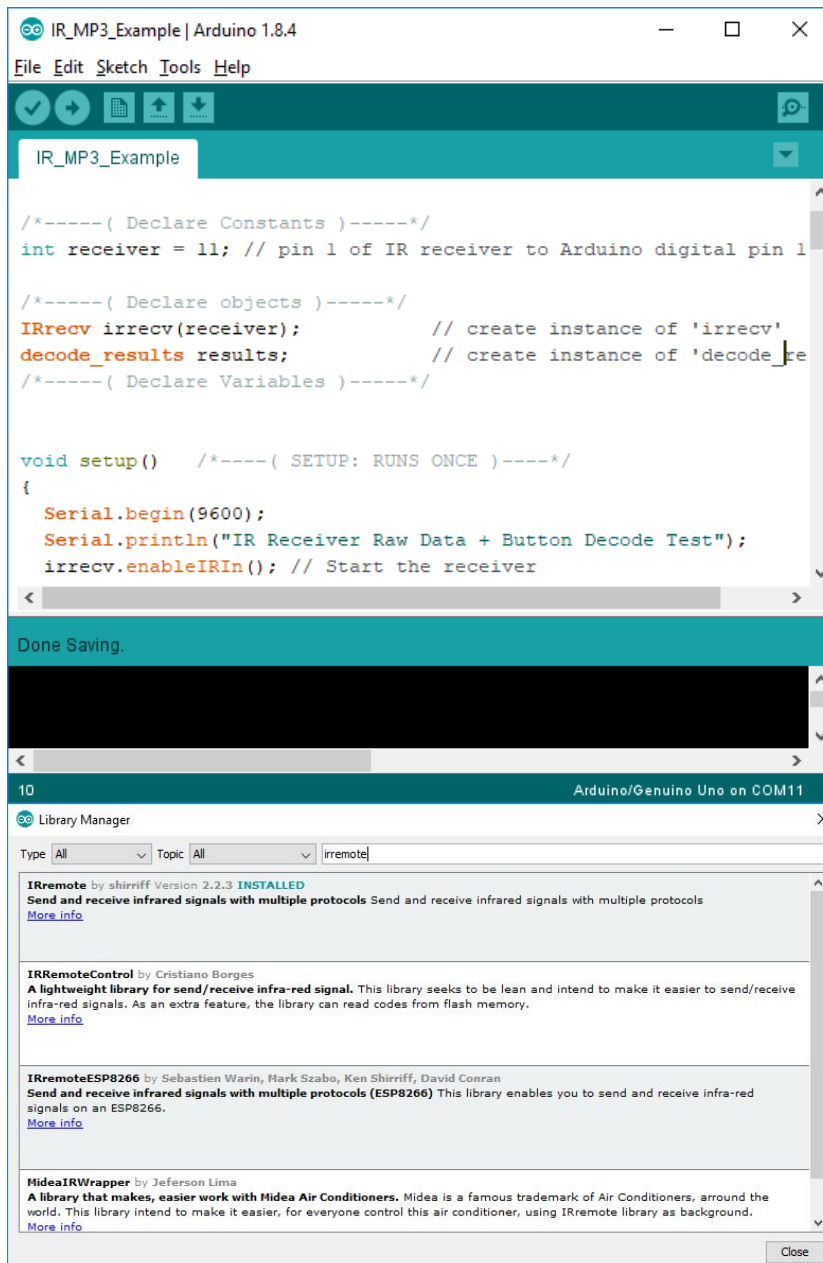
Each button on the remote has a different number sequence. Press the POWER button on the remote about 3 times and watch to see the number sequence that shows up. Then locate in the code...

```
case 0xFFE318261B:
```

```
Serial.println("POWER ");
```

```
break;
```

If you see where it says 0xFFE318261B it is saying when an IR number sequence is E318261B then serial print POWER. Now you replace E318261B with whatever you got when you pressed the power button in the serial monitor. NEVER EVER change or mess with the 0xFF, just put your number sequence after that. Then one by one press each button and copy the number sequence that shows up in the serial monitor after the correct 0xFF.



Open the sketch and upload to your Arduino Development board

If this is your first remote control project, then you might need to import remote control library. Follow - Sketch | Include Library | Manage Libraries.

Search "irremote" in the search bar and install the package.

Aim the remote receiver and press the buttons and check the corresponding code for each button. Open the Serial Monitor as the last experiment with RFID and see what are the unique code for each button.