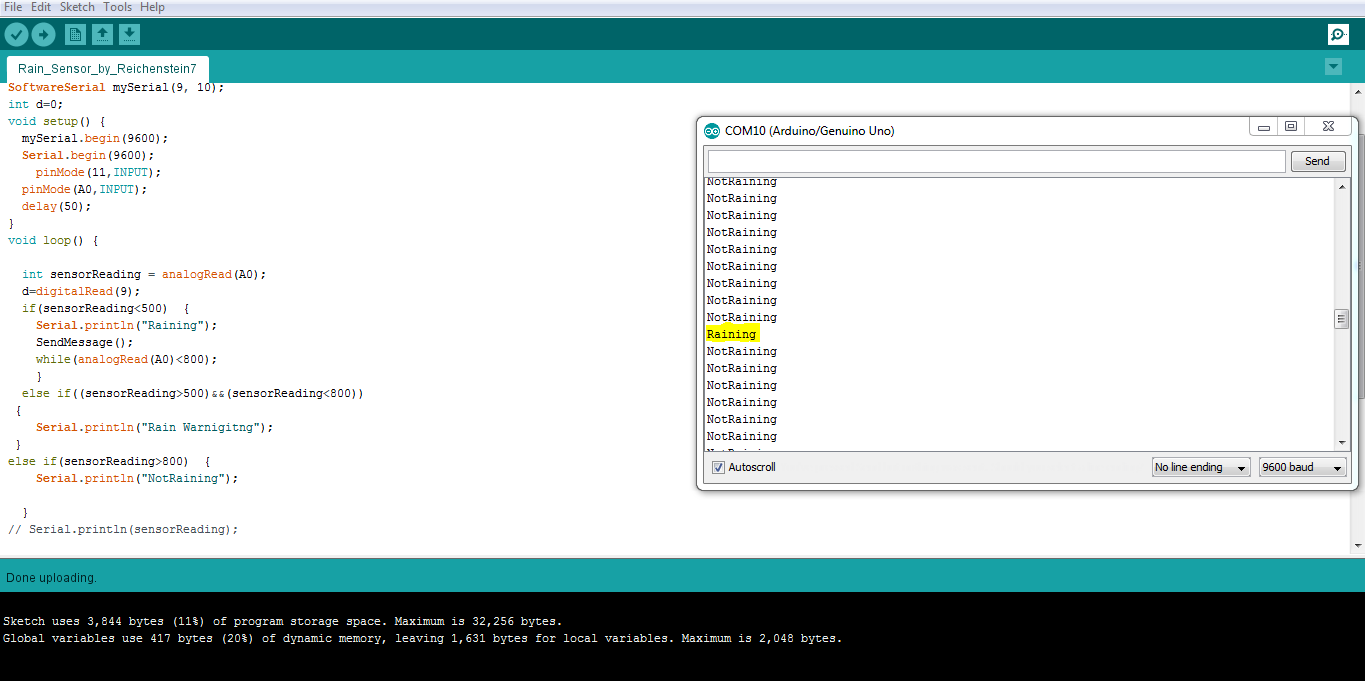
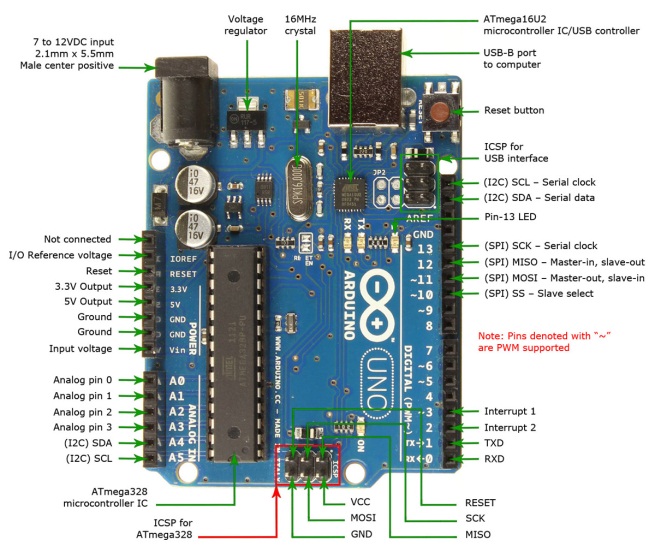
**Arduino-Interface-with-MySQL-for-storing-RFID-access-details (ARDUINO UNO,ETHERNET SHIELD, RFID MFRC522 & MYSQL)**

**ARDUINO :**

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

**ETHERNET SHIELD**

The Arduino Ethernet Shield 2 connects your Arduino to the internet in mere minutes. Just plug this module onto your Arduino Board, connect it to your network with an RJ45 cable (not included) and follow a few simple steps to start controlling your world pharmacy through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino Boards are already fueling people’s creativity all over the world, every day.

\*Requires an Arduino Board (not included)

* Operating voltage 5V (supplied from the Arduino Board)
* Ethernet Controller: W5500 with internal 32K buffer
* Connection speed: 10/100Mb
* Connection with Arduino on SPI port

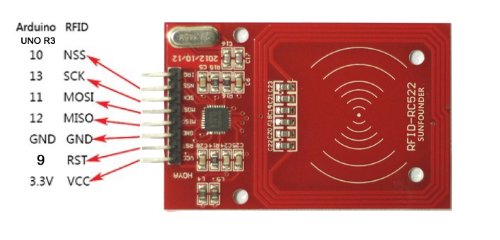
For More Details <https://www.arduino.cc/en/Main/ArduinoEthernetShield>

**MFRC522 RFID MODULE & CARD**

There are cheap RFID modules that can read and write Mifare's tags and being sold at several web stores, like eBay and included with many "starter kits" nowadays. Simply search RFID-RC522 (MF-RC522). The microcontroller and card reader uses SPI for communication (chip supports I2C and UART protocols but not implemented on library) (Maybe someone implements?). The card reader and the tags communicate using a 13.56MHz electromagnetic field. (ISO 14443A standard tags)

Datasheet for the chip that used in modules can be found at:

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



**PROJECT DESCRIPTION:**

Here We are going to connect Arduino UNO, RFID (MFRC522) & Ethernet Shield with MYSQL Database.

So for that first we should connect our Arduino Board with the Ethernet Shield & RFID Module.

By using the RFID Module we are going to scan our RFID card and tag which are allow or not. And by using our Ethernet shield we are going to send that data to our MYSQL Database which is connect through a pho page. Bellow we provided the code for PHP as well as for Arduino. You also can go through our video to better understanding how to create a database in MYSQL and how to connect with PHP and Arduino.See pin Configuration bellow

**NOTE-BECAUSE WE ARE USING TWO SPI DEVICES THAT’S WHY WE HAVE TO CHANGE OUR SS PIN FOR RFID MODULE .**

Reader Uno Mega Nano v3 Leonardo Pro Micro

\* Signal Pin Pin Pin Pin Pin Pin

\* -----------------------------------------------------------------------------------------

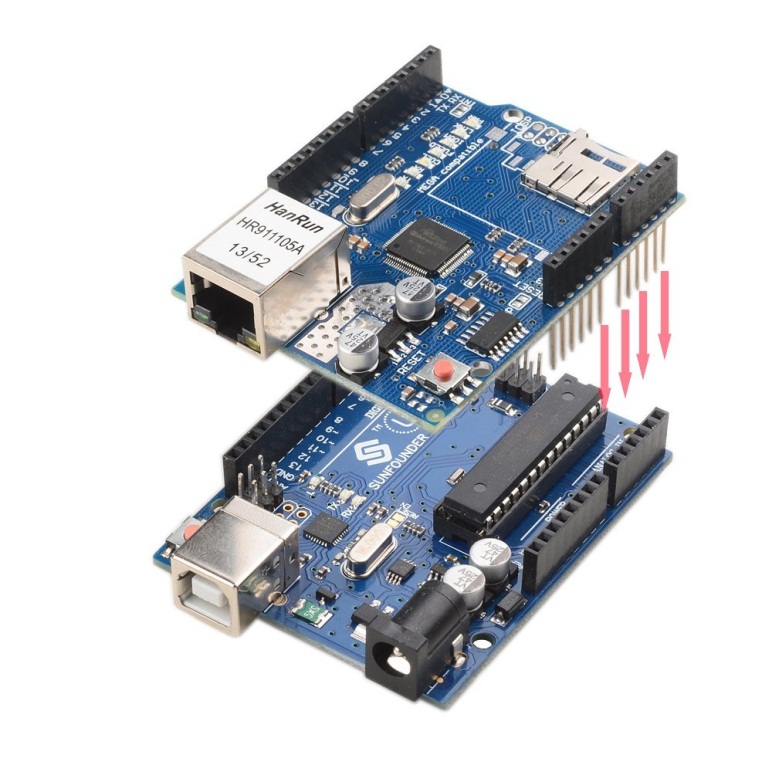
\* RST/Reset RST 9 9 D9 RESET/ICSP-5 RST

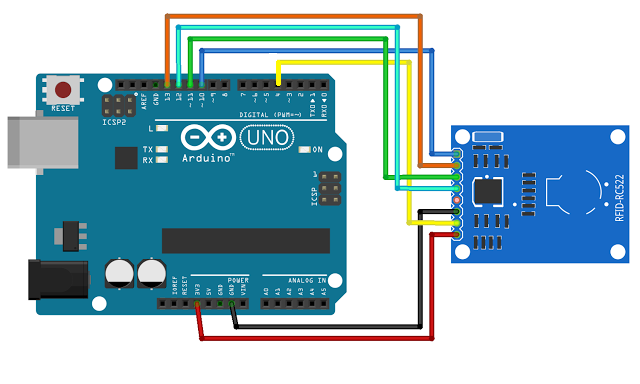
\* SPI SS SDA(SS) 4/10 4/53 D10 10 10

\* SPI MOSI MOSI 11 / ICSP-4 51 D11 ICSP-4 16

\* SPI MISO MISO 12 / ICSP-1 50 D12 ICSP-1 14

\* SPI SCK SCK 13 / ICSP-3 52 D13 ICSP-3 15





**CODE: (PHP)**

|  |
| --- |
| <?php  class rfid{ |
| public $link=''; |
| function \_\_construct($allow, $id){ |
| $this->connect(); |
| $this->storeInDB($allow, $id); |
| } |
|  |
| function connect(){ |
| $this->link = mysql\_connect('localhost','deligenc\_rfid','rfid@234') or die('Cannot connect to the DB'); |
| mysql\_select\_db('deligenc\_rfid') or die('Cannot select the DB'); |
| } |
|  |
| function storeInDB($allow, $id){ |
| $query = "insert into rfid set rfid='".$id."', allow='".$allow."'"; |
| $result = mysql\_query($query) or die('Errant query: '.$query); |
| } |
|  |
| } |
| if($\_GET['allow'] != '' and $\_GET['id'] != ''){ |
| $rfid=new rfid($\_GET['allow'],$\_GET['id']); |
| } |
|  |
| /\*class rfid{ |
|  |
| function show(){ |
| $fh=fopen('test.txt','a+'); |
| fwrite($fh,'hi'); |
| return 'hi'; |
| } |
| } |
| $rfid=new rfid(); |
| $rfid->show(); |
| \*/ |
| ?> |

**CODE: (ARDUINO)**

#include<SPI.h>

#include<MFRC522.h>

#include<SoftwareSerial.h>

#include <Ethernet.h>

#define SS\_PIN 4 //FOR RFID SS PIN BECASUSE WE ARE USING BOTH ETHERNET SHIELD AND RS-522

#define RST\_PIN 9

#define No\_Of\_Card 3

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

char server[] = "www.XXXXXXXXXXXX.com"; //YOUR SERVER

IPAddress ip(192, 168, 6, 177);

EthernetClient client;

SoftwareSerial mySerial(8,9);

MFRC522 rfid(SS\_PIN,RST\_PIN);

MFRC522::MIFARE\_Key key;

byte id[No\_Of\_Card][4]={

{44,153,22,219}, //RFID NO-1

{112,224,72,84}, //RFID NO-2

{151,94,80,84} //RFID NO-3

};

byte id\_temp[3][3];

byte i;

int j=0;

void setup()

{

Serial.begin(9600);

mySerial.begin(9600);

SPI.begin();

rfid.PCD\_Init();

for(byte i=0;i<6;i++)

{

key.keyByte[i]=0xFF;

}

if (Ethernet.begin(mac) == 0) {

Serial.println("Failed to configure Ethernet using DHCP");

Ethernet.begin(mac, ip);

}

delay(1000);

Serial.println("connecting...");

}

void loop()

{int m=0;

if(!rfid.PICC\_IsNewCardPresent())

return;

if(!rfid.PICC\_ReadCardSerial())

return;

for(i=0;i<4;i++)

{

id\_temp[0][i]=rfid.uid.uidByte[i];

delay(50);

}

for(i=0;i<No\_Of\_Card;i++)

{

if(id[i][0]==id\_temp[0][0])

{

if(id[i][1]==id\_temp[0][1])

{

if(id[i][2]==id\_temp[0][2])

{

if(id[i][3]==id\_temp[0][3])

{

Serial.print("your card no :");

for(int s=0;s<4;s++)

{

Serial.print(rfid.uid.uidByte[s]);

Serial.print(" ");

}

Serial.println("\nVALID");

Sending\_To\_DB();

j=0;

rfid.PICC\_HaltA(); rfid.PCD\_StopCrypto1(); return;

}

}

}

}

else

{j++;

if(j==No\_Of\_Card)

{

Serial.println("inVALID");

Sending\_To\_DB();

j=0;

}

}

}

// Halt PICC

rfid.PICC\_HaltA();

// Stop encryption on PCD

rfid.PCD\_StopCrypto1();

}

void Sending\_To\_DB() //CONNECTING WITH MYSQL

{

if (client.connect(server, 80)) {

Serial.println("connected");

// Make a HTTP request:

client.print("GET /rfid/rfid\_read.php?allow="); //YOUR URL

if(j!=No\_Of\_Card)

{

client.print('1');

}

else

{

client.print('0');

}

client.print("&id=");

for(int s=0;s<4;s++)

{

client.print(rfid.uid.uidByte[s]);

}

client.print(" "); //SPACE BEFORE HTTP/1.1

client.print("HTTP/1.1");

client.println();

client.println("Host: www.XXXXXXXXXX.com");

client.println("Connection: close");

client.println();

} else {

// if you didn't get a connection to the server:

Serial.println("connection failed");

}

client.stop();

}