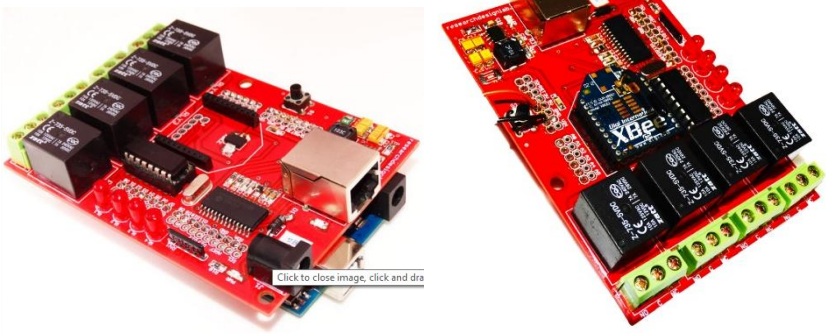


INTERNET OF THINGS (IoT)

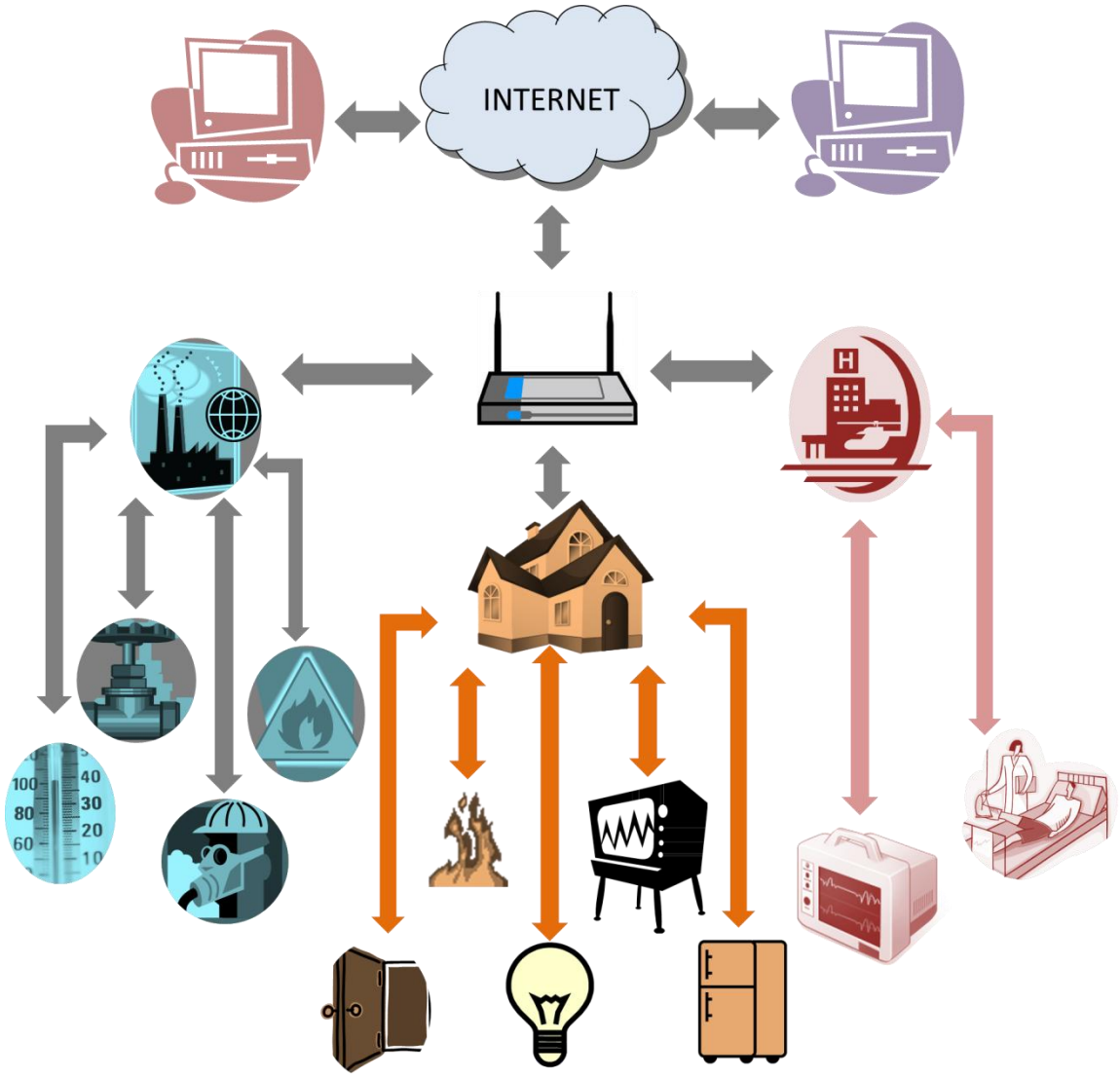
WEB SHIELD

SMART INTERNET APPLICATION DEVELOPMENT KIT

Contents

OVERVIEW

INTERNET OF THINGS (IoT)



The Internet of Things (IoT) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure.

Ethernet Web shield is a smart communication shield for controlling and monitoring embedded application via internet. Internet is the media and economic way for communication between anywhere from to geographical location. This shield is compatible with Arduino open source and make easy way for customized application development. This shield has got four relay 10AMP.The board by ULN2003 IC. The board works on 5V.

FEATURES:

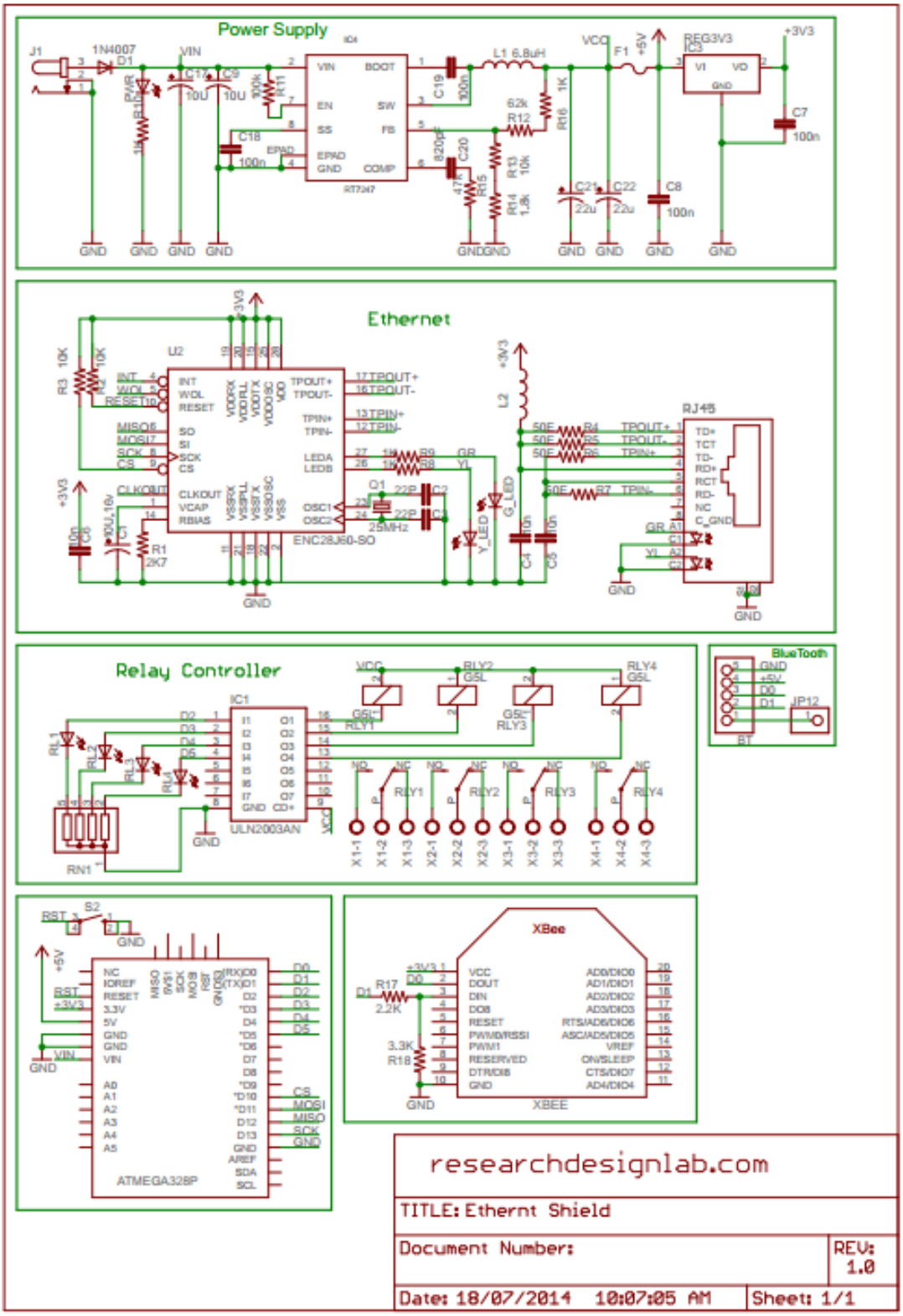
1. IEEE 802.3 compatible.
2. Supports Full and Half-Duplex modes.
3. Integrated MAC and 10Base-T PHY
4. SPI Interface with Clock Speeds Up to 20 MHz.
5. With this Ethernet Shield, your Arduino board can be connected to internet.
6. Stackable by accepting prototype shield or any other boards with Arduino compatible interface.
7. LED indication for relay & power supply.
8. Design based on highly proven IC ULN2803 as driver.
9. DC-DC Converter.

Ethernet Controller (ENC28J60)

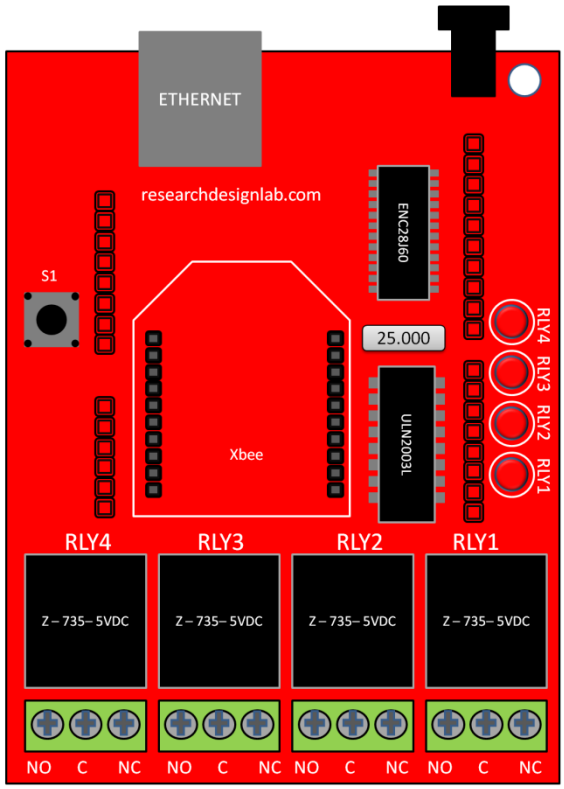
The Ethernet Controller (ENC28J60) is a so called SPI device and uses the SPI pins (10, 11, 12, 13) of your Arduino.

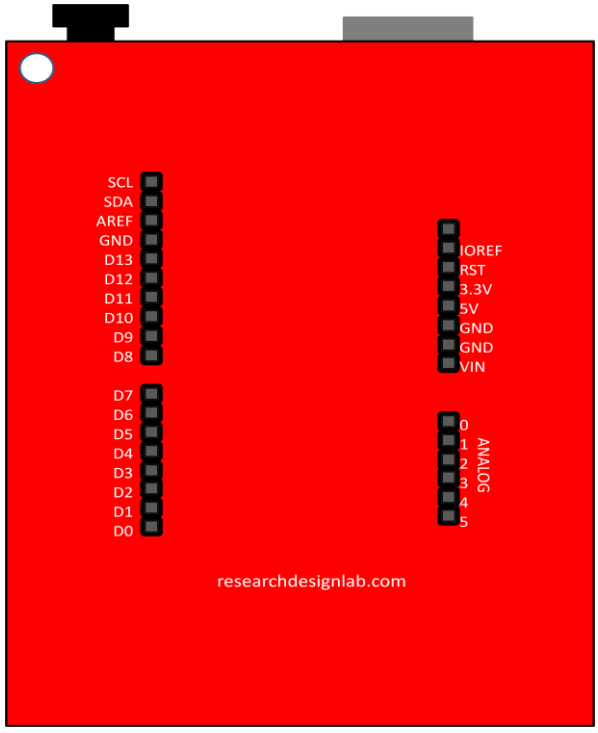
- SS stands for Slave Select, used to enable or disable the slave device (the Ethernet module in this case).
- MOSI stands for Master Output Slave Input, or in other words: Arduino OUTPUT (data from Arduino to Ethernet Controller).
- MISO stands for the opposite, Master Input Slave Output, or: Arduino INPUT (data from Ethernet Controller to Arduino).
- SCK is the clock used for SPI timing.

CIRCUIT DIAGRAM

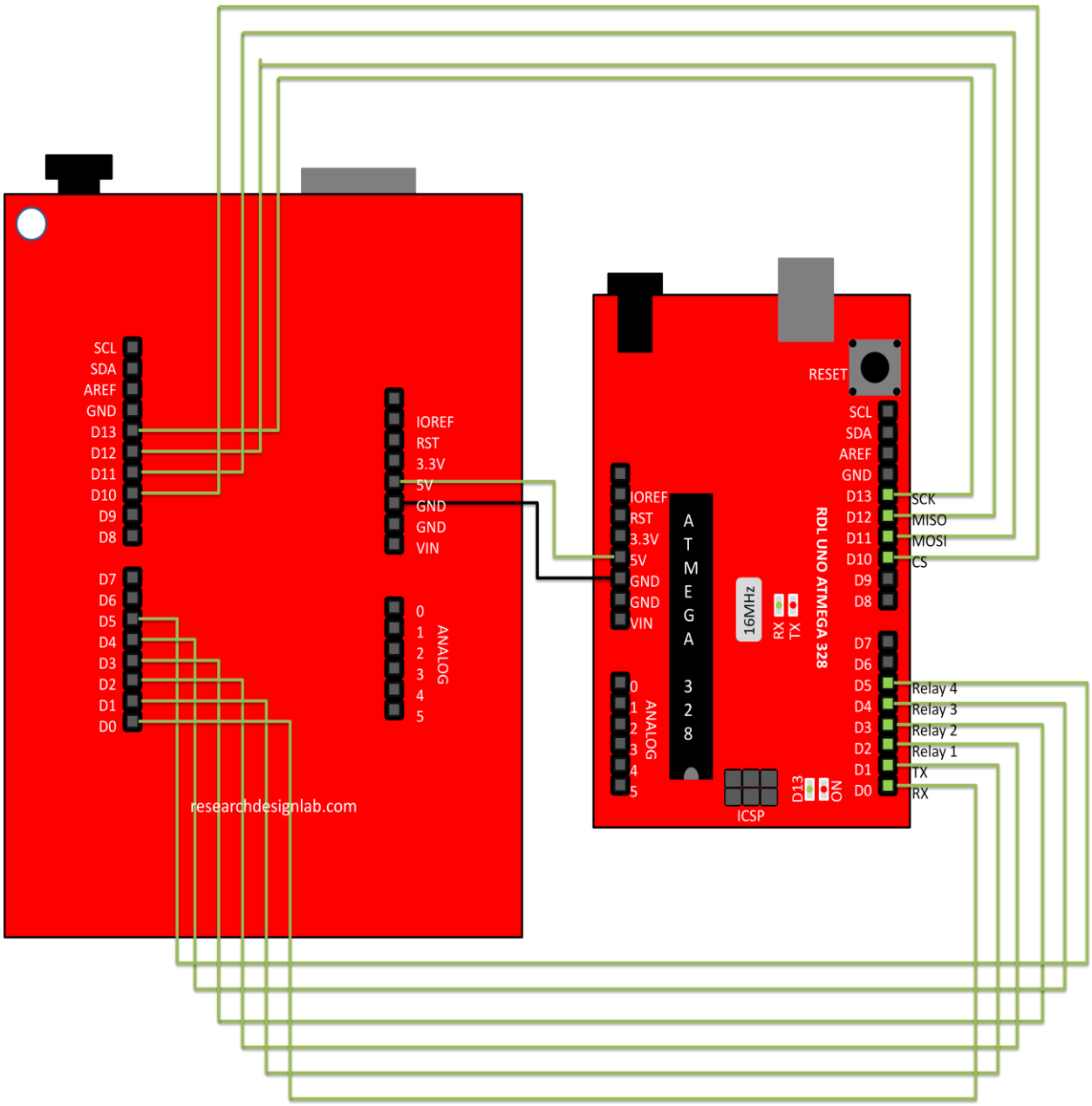


SCHEMATIC OF WEB SHIELD

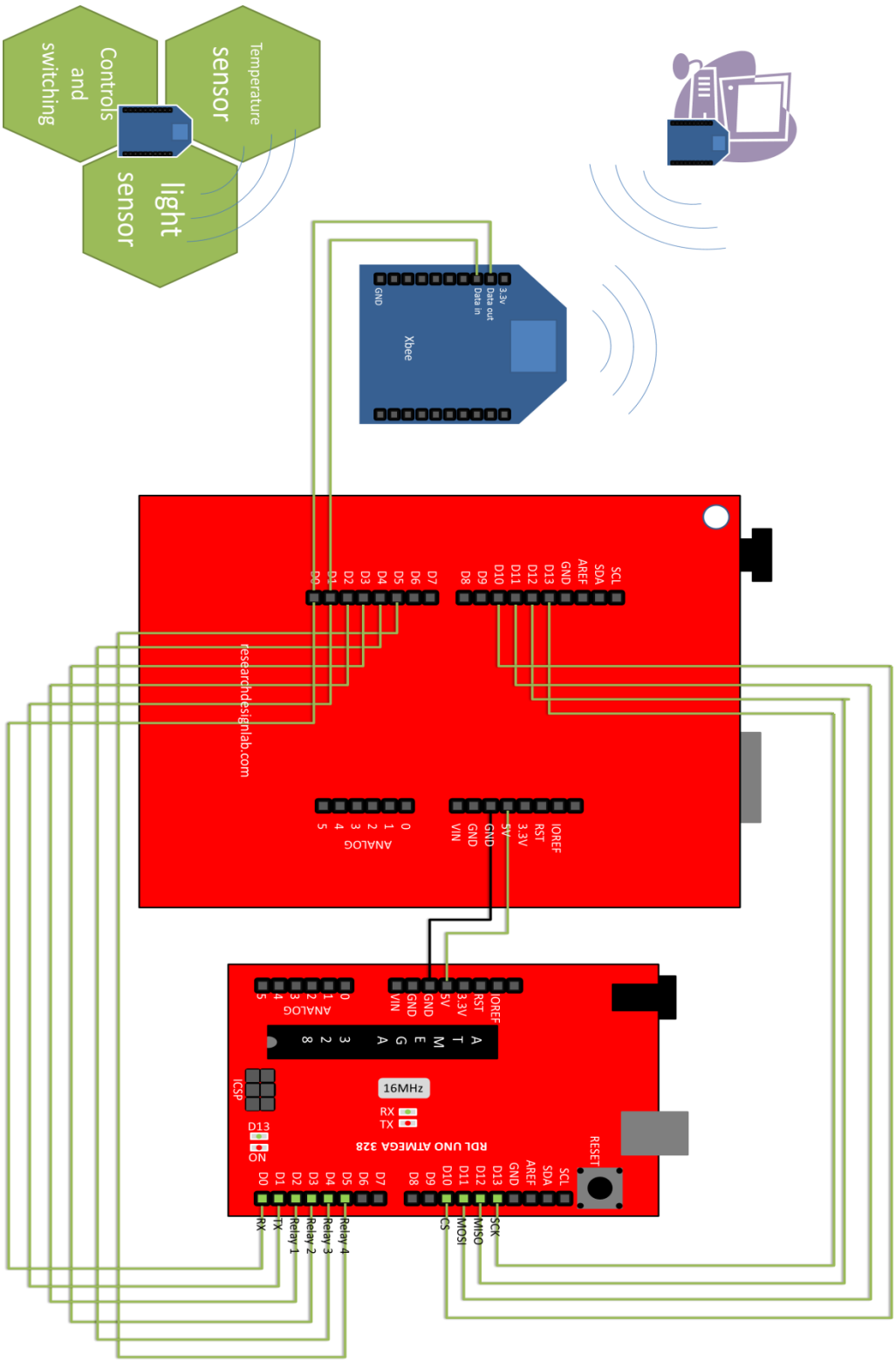




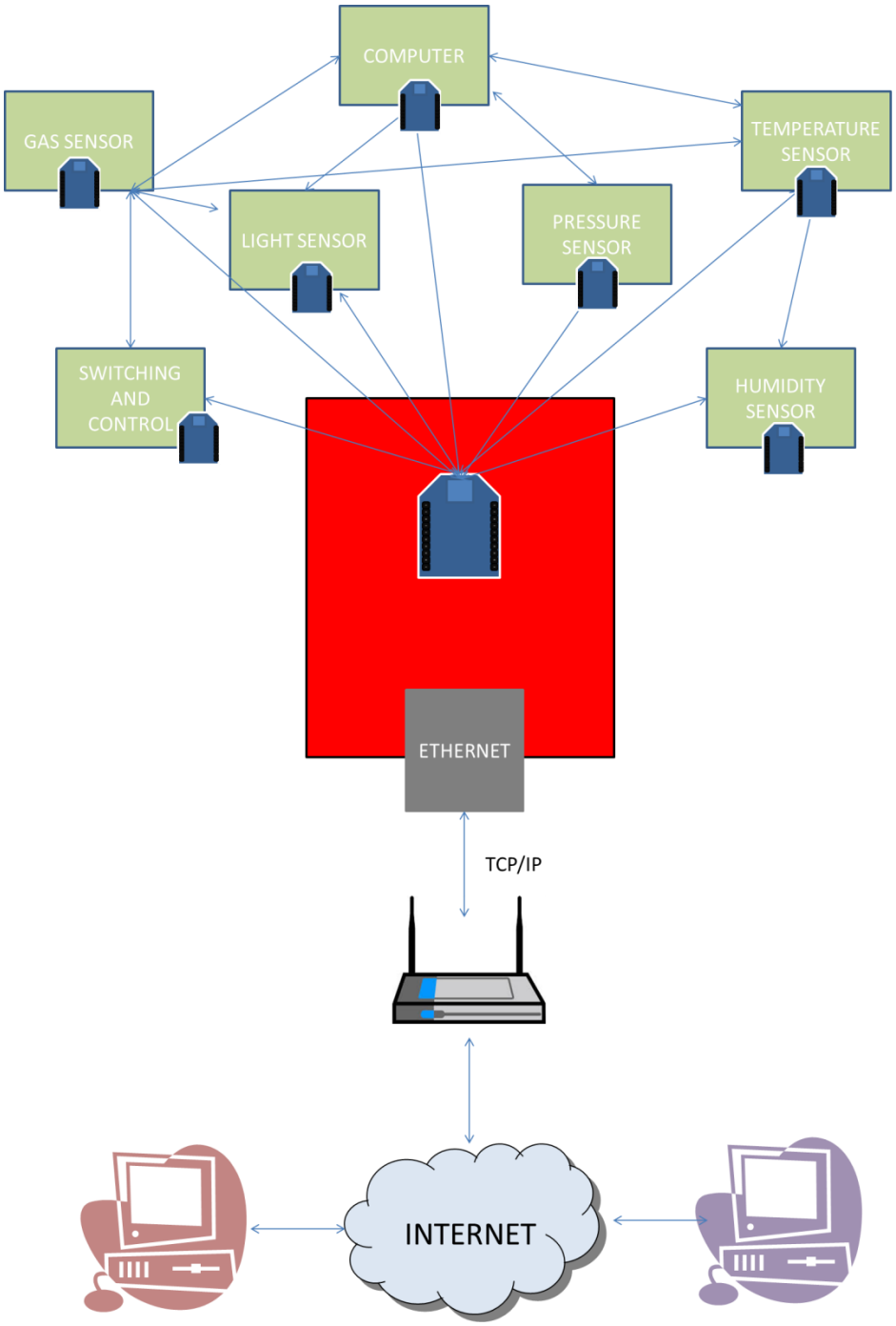
INTERNAL CONNECTION OF WEBSHIELD WITH UNO BOARD



INTERNAL CONNECTIONS OF XBEE WITH WEBSHIELD



XBEE – WEBSHIELD



MODULE SETUP

step 1 : Download - ETHER_28J60 and Ethershield libraries from Github.com
https://github.com/muanis/arduino-projects/tree/master/libraries
step 2 : Import ETHER_28J60 and Ethershield libraries into the Arduino1.0.6 .
step 3: Connect UNO board to PC then compile and dump the code (given in code section of this document)to Arduino board before mounting web shield.

step 4: Mount Arduino onto web shield ,connect 12 V DC power supply to web shield.
step 5: Connect PC and web shield with Ethernet cable for direct control(local LAN network).

or

Connect web shield to the router through Ethernet cable(internet)
*the IP address and port used in the code for web shield should be port forwarded in the router so as to make it online accessible.

step 6: Enter IP address(Ex: http://192.168.1.15) into the browser specified in the code and enter.(to follow up refer screen shots)(**this holds good for local LAN control)

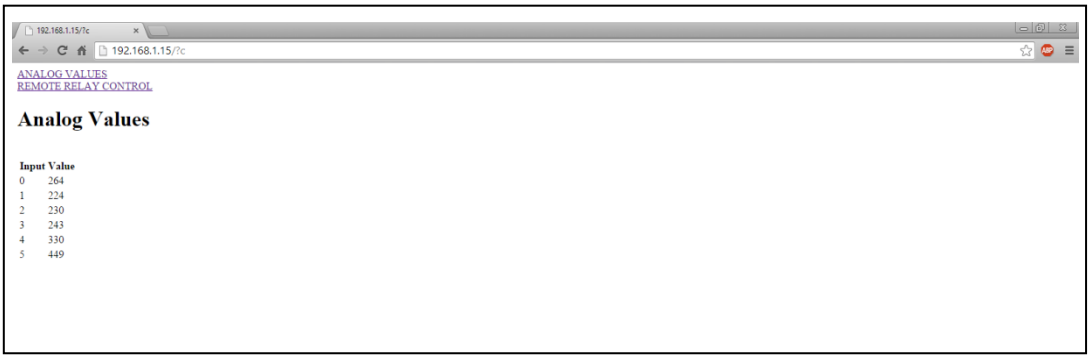
Or

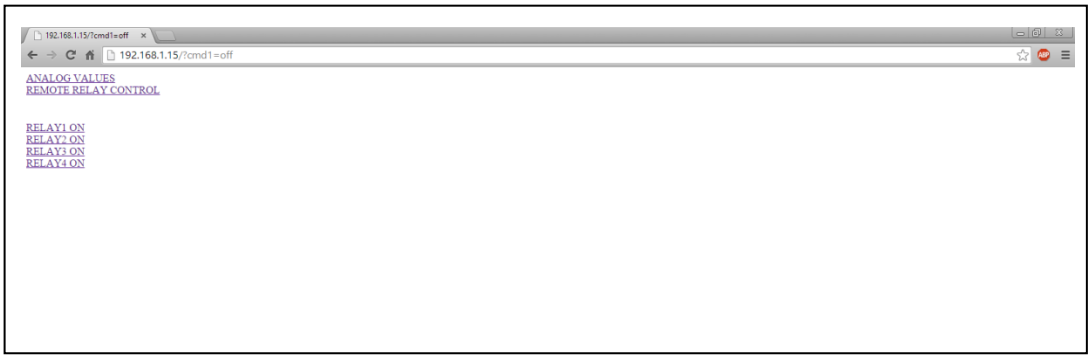
Port forward 192.168.1.15:80 IP in your respective router settings so that Webshield will be accessed online

Note : *need to have to Dynamic DNS and port forwarding knowledge to access through internet.
*if one has dedicated IP/ VPN owned, they can directly access within their private network.

step 7 :after port forwarding, use “internet IP”(static/public IP Address) to access web shield online.

SCREENSHOTS





Code:

To control onboard relay and to display analog pin values in web browser

```
#include "etherShield.h"
#include "ETHER_28J60.h"

static uint8_t mac[6] = {0x54, 0x55, 0x58, 0x10, 0x00, 0x24};

static uint8_t ip[4] = {192, 168, 1, 15};   // the IP address for your board. Check your home hub
                                            // to find an IP address not in use and pick that
int outputPin1 = 2;
int outputPin2 = 3;
int outputPin3 = 4;
int outputPin4 = 5;                         // this or 10.0.0.15 are likely formats for an address
                                            // that will work.

static uint16_t port = 80;                              // Use port 80 - the standard for HTTP

ETHER_28J60 e;
char flag1=0,flag2=0,flag3=0,flag4=0;

void setup()
{
  e.setup(mac, ip, port);
  pinMode(outputPin1, OUTPUT);
pinMode(outputPin2, OUTPUT);
pinMode(outputPin3, OUTPUT);
pinMode(outputPin4, OUTPUT);
}

void loop()
{
  char *params;
  if (params = e.serviceRequest())
  {
    e.print("<A HREF='?c'>ANALOG VALUES</A></BR>");
    e.print("<A HREF='?cmd1=off'>REMOTE RELAY CONTROL</A></BR>");
    if (strcmp(params, "?c") == 0)
    {
   analog();
    }
   e.print("<H1>       </H1><br/>");
    if (strcmp(params, "?cmd1=on") == 0)
    {
    digitalWrite(outputPin1, HIGH);
```

```
 flag1=1;
display();
}
else if (strcmp(params, "?cmd1=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin1, LOW);
flag1=0;
display();
}
if (strcmp(params, "?cmd2=on") == 0)
{
digitalWrite(outputPin2, HIGH);
flag2=1;
display();
}
else if (strcmp(params, "?cmd2=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin2, LOW);
flag2=0;
display();
}
if (strcmp(params, "?cmd3=on") == 0)
{
digitalWrite(outputPin3, HIGH);
flag3=1;
display();
}
else if (strcmp(params, "?cmd3=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin3, LOW);
flag3=0;
display();
 }
 if (strcmp(params, "?cmd4=on") == 0)
 {
 digitalWrite(outputPin4, HIGH);
  flag4=1;
  display();
  }
else if (strcmp(params, "?cmd4=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin4, LOW);
flag4=0;
display();
}
```

```
    e.respond();
}
  delay(100);
}

void analog()
{
    e.print("<H1>Analog Values</H1><br/><table>");
    e.print("<tr><th>Input</th><th>Value</th></tr>");
    for (int i = 0; i < 6; i++)
    {
      e.print("<tr><td>");          e.print(i);          e.print("</td><td>");          e.print(analogRead(i));
e.print("</td></tr>");
    }
    e.print("</table>");

}

void display()
{

if(flag1==0)
e.print("<A HREF='?cmd1=on'>RELAY1 ON</A></BR>");
else
e.print("<A HREF='?cmd1=off'>RELAY1 OFF</A></BR>");

if(flag2==0)
e.print("<A HREF='?cmd2=on'>RELAY2 ON</A></BR>");
else
e.print("<A HREF='?cmd2=off'>RELAY2 OFF</A></BR>");

if(flag3==0)
e.print("<A HREF='?cmd3=on'>RELAY3 ON</A></BR>");
else
e.print("<A HREF='?cmd3=off'>RELAY3 OFF</A></BR>");

if(flag4==0)
e.print("<A HREF='?cmd4=on'>RELAY4 ON</A></BR>");
else
e.print("<A HREF='?cmd4=off'>RELAY4 OFF</A></BR>");

}
```