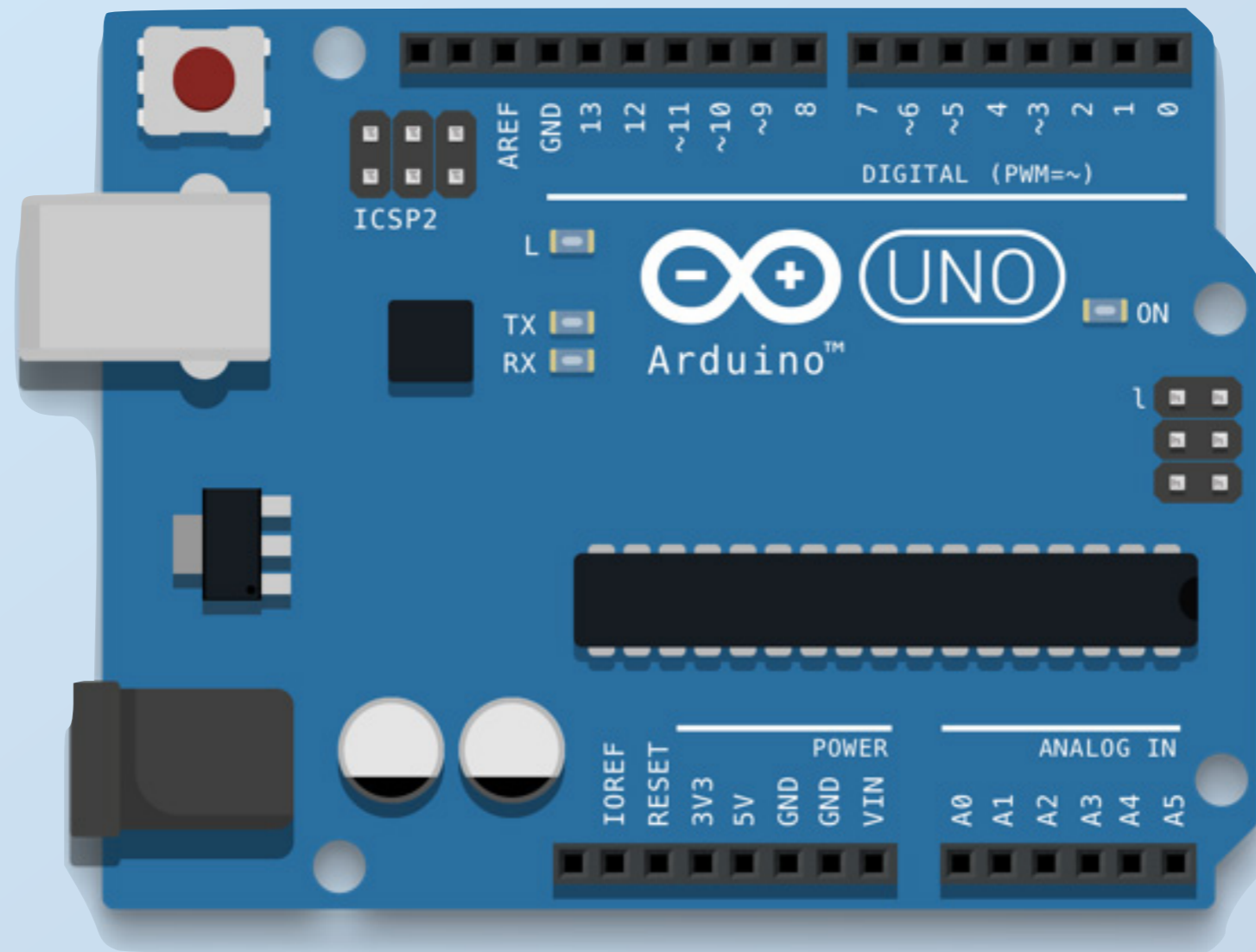


EtherSia

IPv6 Library for Arduino

What is Arduino?





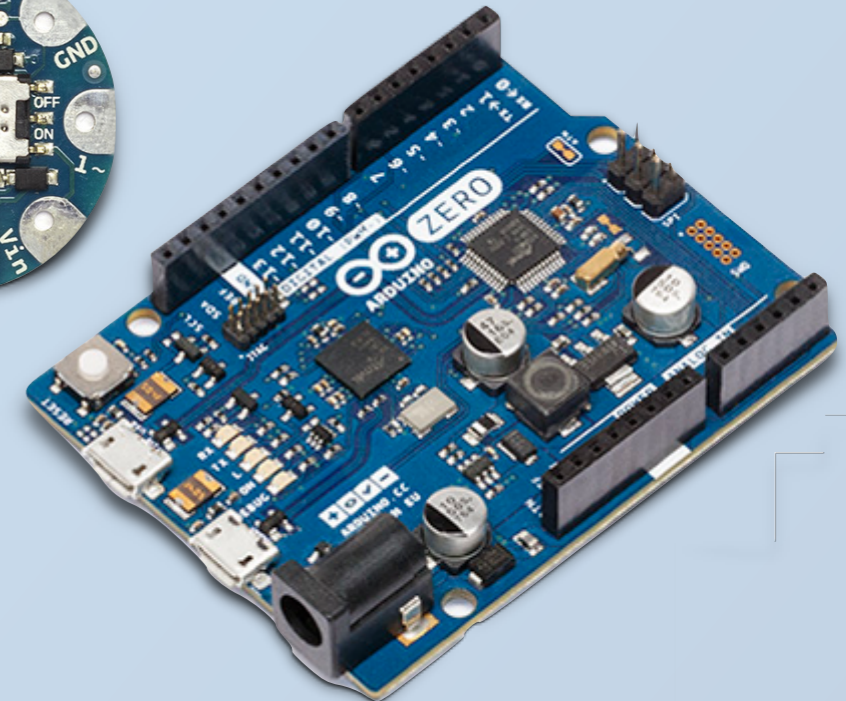
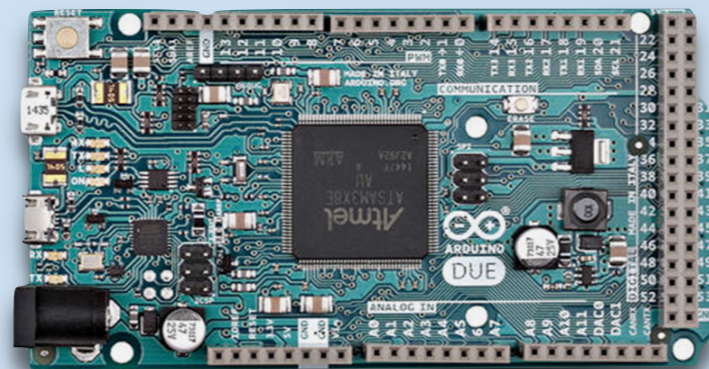
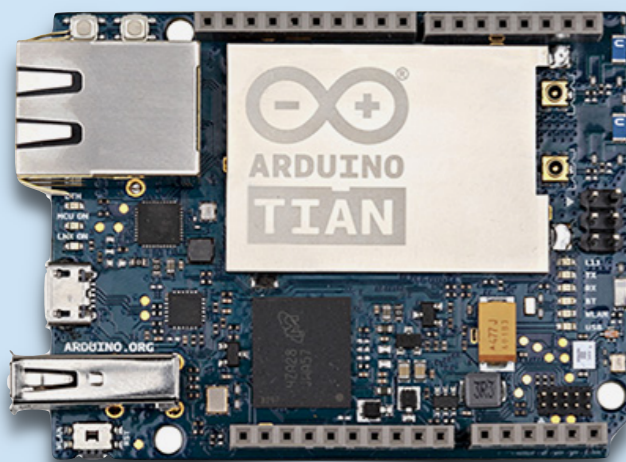
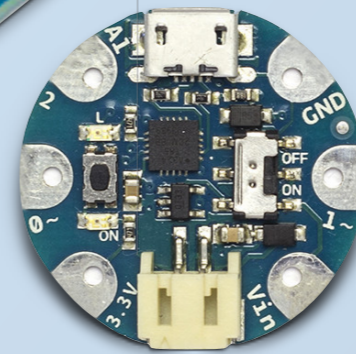
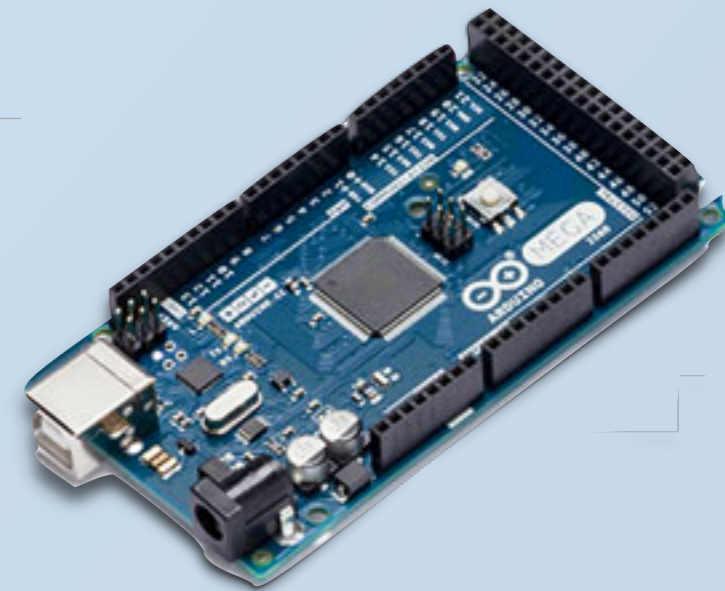
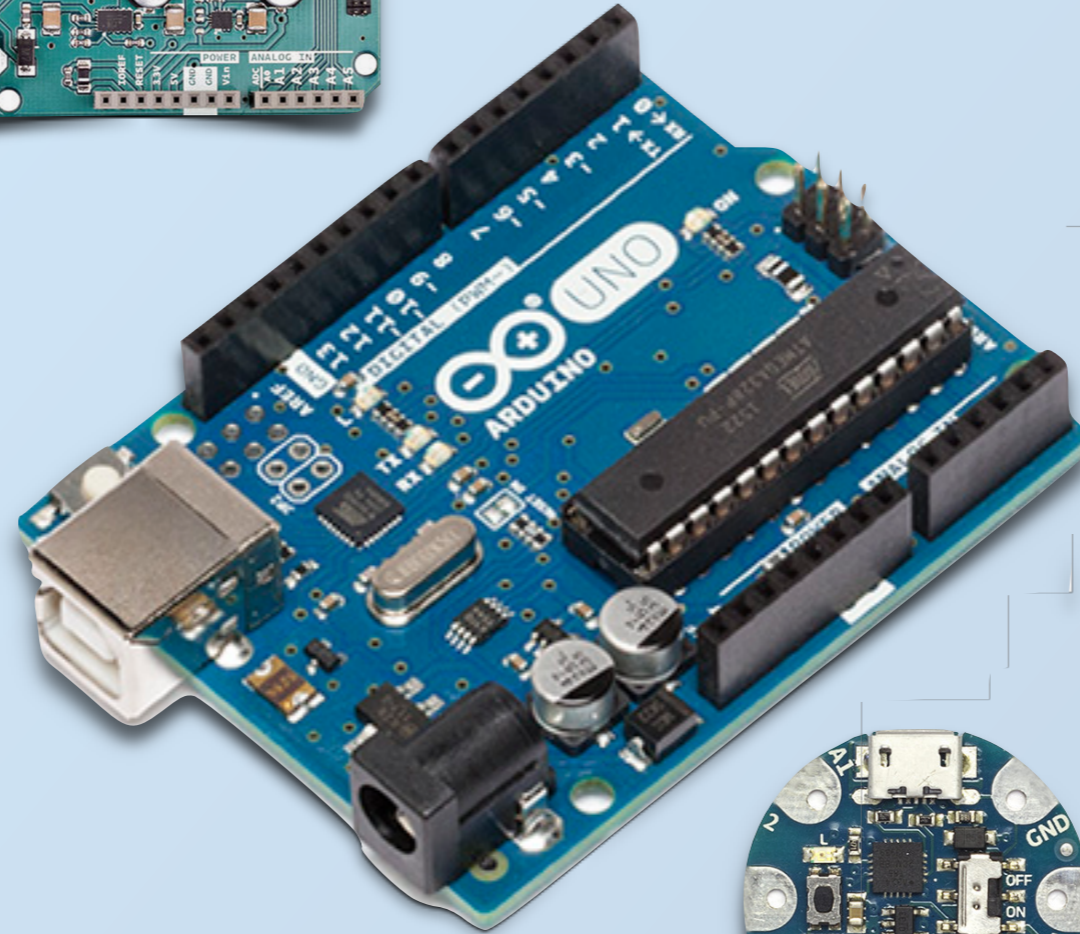
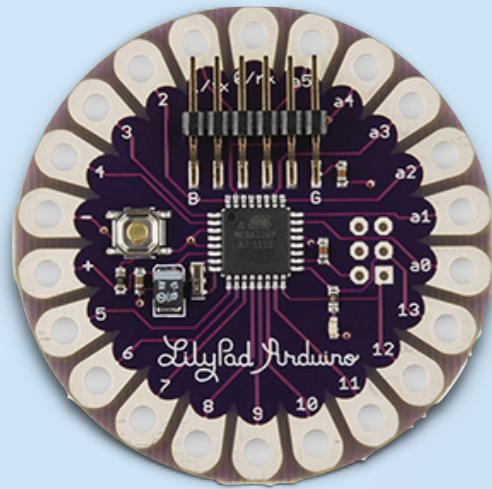
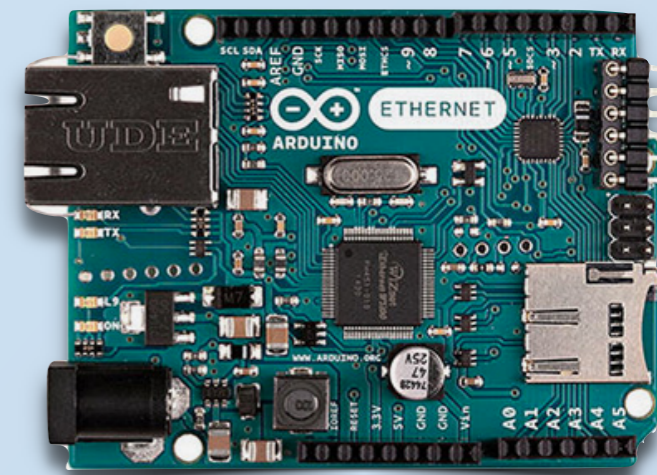
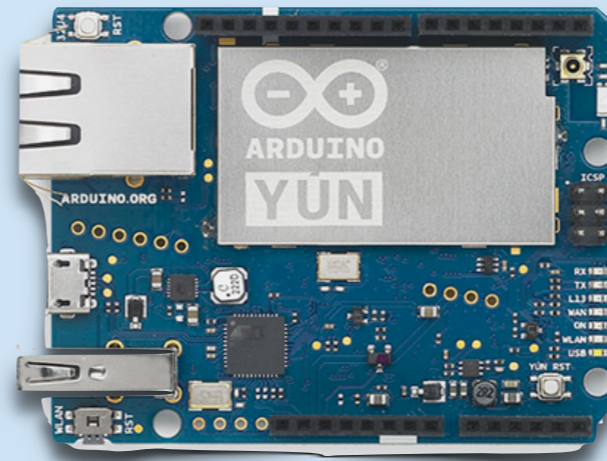
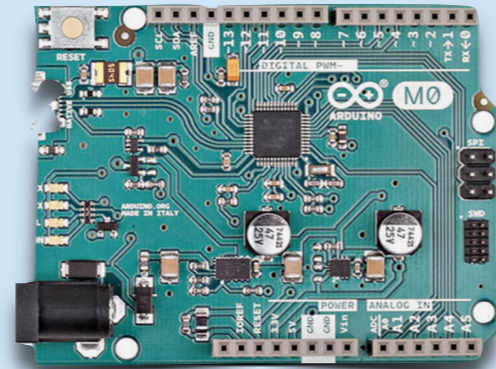
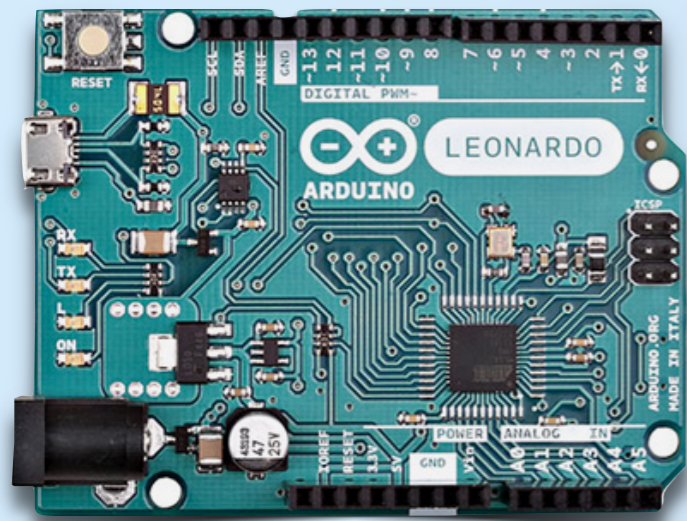
Blink

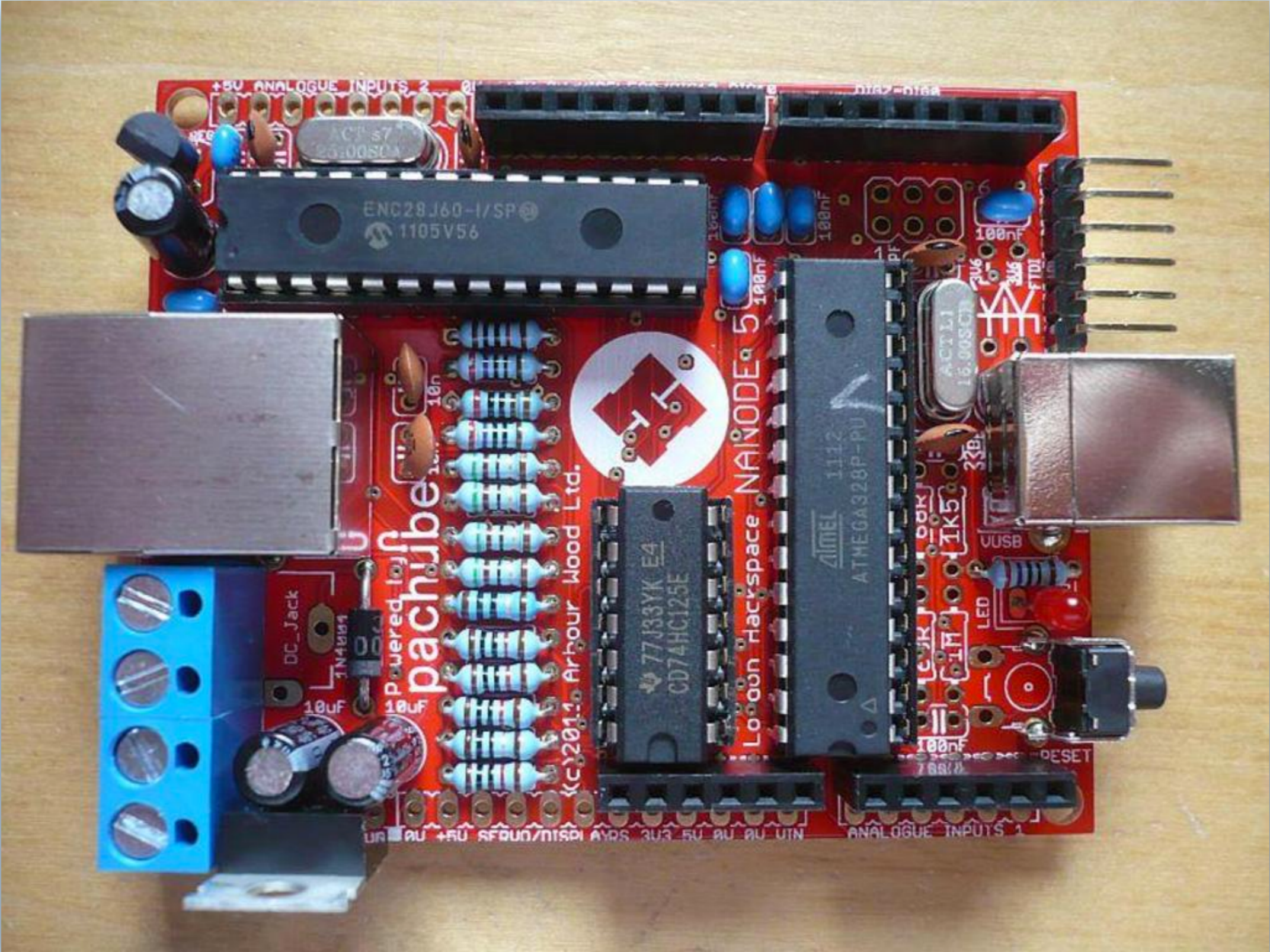
```
modified 2 Sep 2016
by Arturo Guadalupi

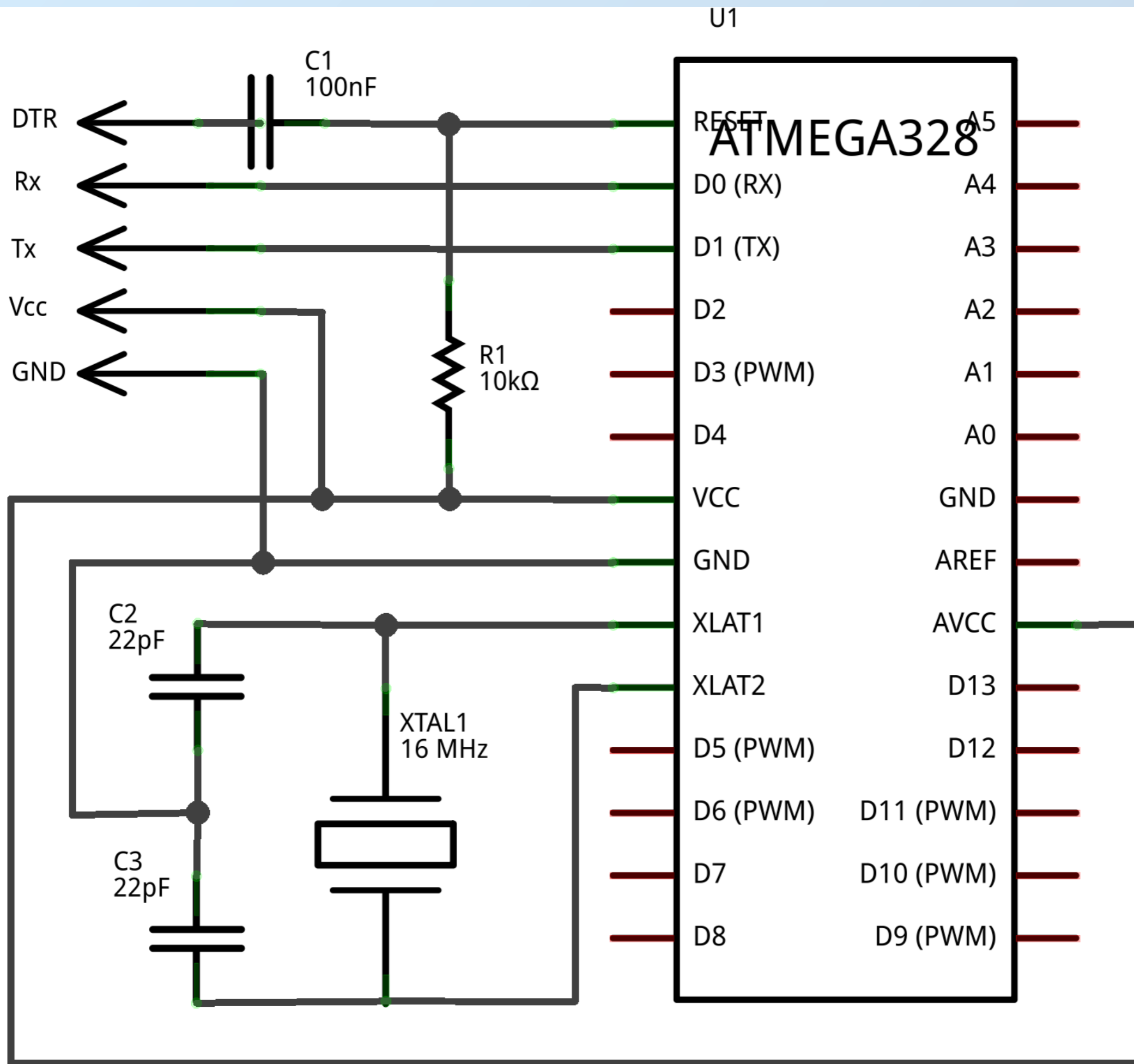
modified 8 Sep 2016
by Colby Newman
*/

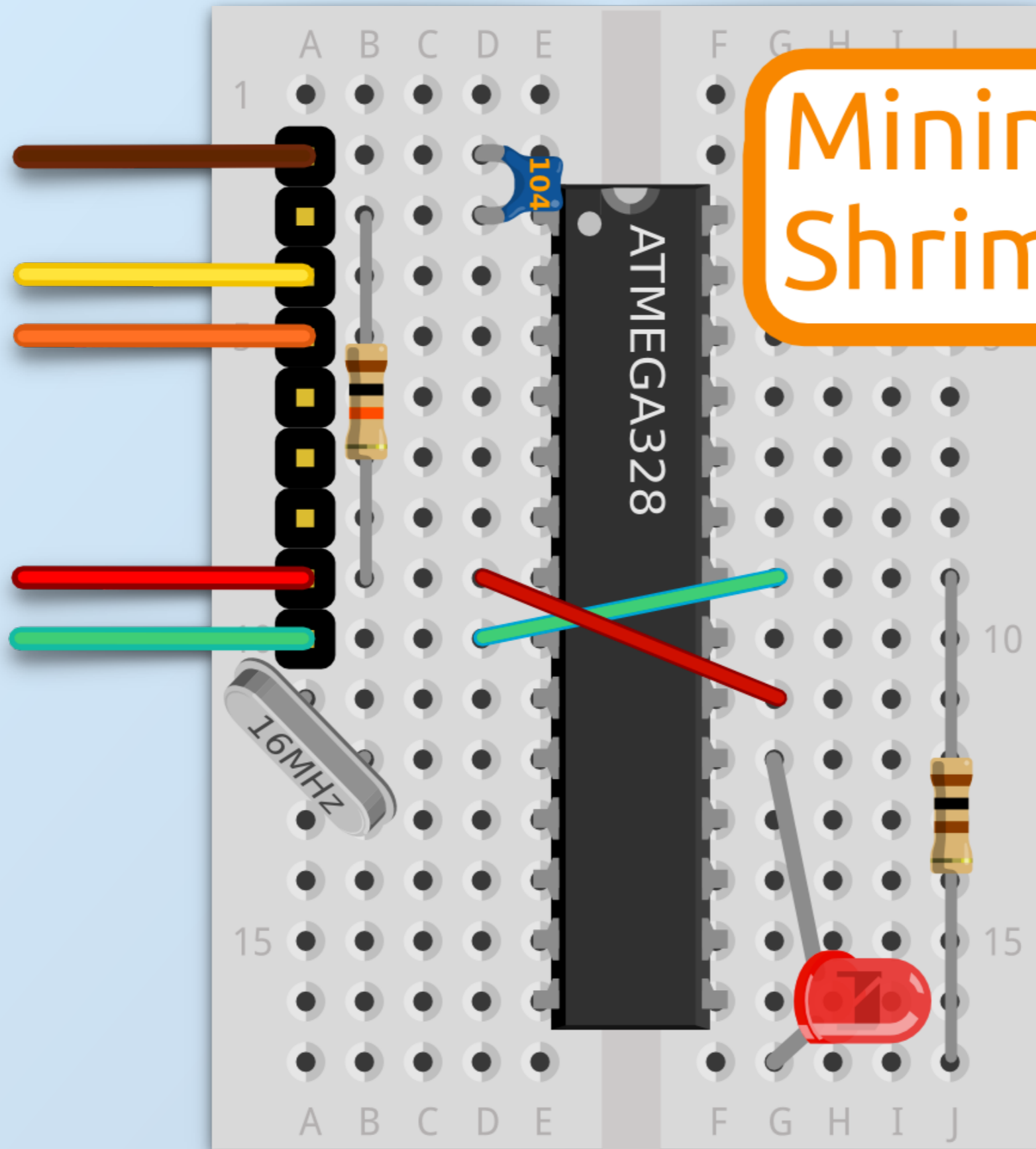
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```







Minimal Shrimp

Cheap!



New ENC28J60 Ethernet LAN Network Module
For Arduino SPI AVR PIC LPC STM32

£1.91

From Hong Kong

Buy It Now

Free Postage

70 sold

How minimal?

- 32kB of Programme Memory
- 2kB of RAM
- 1kB of EEPROM

Why would you
do that?

Why?

- As a Hobby
- A personal challenge
- Learn by doing
- How do memory sizes compare with IPv4?
- Reduce the number of @ipv6excuses

Doesn't that already exist?

- uIP / Contiki
 - uIPEthernet
 - Arduino-pIPv6Stack
- Some Arduino boards have CPU running Linux

EtherSia Features

- SLAAC
- Ping!
- UDP Server
- UDP Client
- DNS Client
- HTTP Server

EtherSia overview

1. Optimised for Arduino
2. Should be easy to use
3. Should work within the constraints of an Uno
4. Avoid using complex C features
5. Decouple the core from protocols where possible
6. Only use statically allocated memory

Basic example

```
/** Ethernet Interface */
EtherSia_ENC28J60 ether;

void setup() {
  MACAddress macAddress("aa:1c:5d:fe:e7:86");

  // Setup serial port for debugging
  Serial.begin(38400);
  Serial.println("[EtherSia Minimal]");

  // Configure the Ethernet interface
  if (ether.begin(macAddress) == false) {
    Serial.println("Failed to configure Ethernet");
  }

  Serial.print("Global address: ");
  ether.globalAddress().println();
}

void loop() {
  ether.receivePacket();
}
```

It Pings!

```
humfrn01 — BBC Off Network — ping6 2001:8b0:ffd5:3:a81c:5dff:fefe:e786 — 99x21
~ $ ping6 2001:8b0:ffd5:3:a81c:5dff:fefe:e786
PING6(56=40+8+8 bytes) 2001:8b0:ffd5:3:10e6:e25d:ce5c:a17c --> 2001:8b0:ffd5:3:a81c:5dff:fefe:e786
16 bytes from 2001:8b0:ffd5:3:a81c:5dff:fefe:e786, icmp_seq=0 hlim=64 time=3.049 ms
16 bytes from 2001:8b0:ffd5:3:a81c:5dff:fefe:e786, icmp_seq=1 hlim=64 time=3.018 ms
16 bytes from 2001:8b0:ffd5:3:a81c:5dff:fefe:e786, icmp_seq=2 hlim=64 time=2.333 ms
16 bytes from 2001:8b0:ffd5:3:a81c:5dff:fefe:e786, icmp_seq=3 hlim=64 time=2.164 ms
16 bytes from 2001:8b0:ffd5:3:a81c:5dff:fefe:e786, icmp_seq=4 hlim=64 time=2.331 ms
```


UDP Server example

```
EtherSia_ENC28J60 ether(8);

UDPSocket udp(ether, 1234);

void setup()
{
  MACAddress macAddress("d6:9c:e1:1c:0b:32");

  Serial.begin(115200);
  Serial.println("[EtherSia UDP Server]");
  macAddress.println();

  if (ether.begin(macAddress) == false) {
    Serial.println("Failed to configure Ethernet");
  }

  Serial.print("Our address is: ");
  ether.globalAddress().println();
}
```

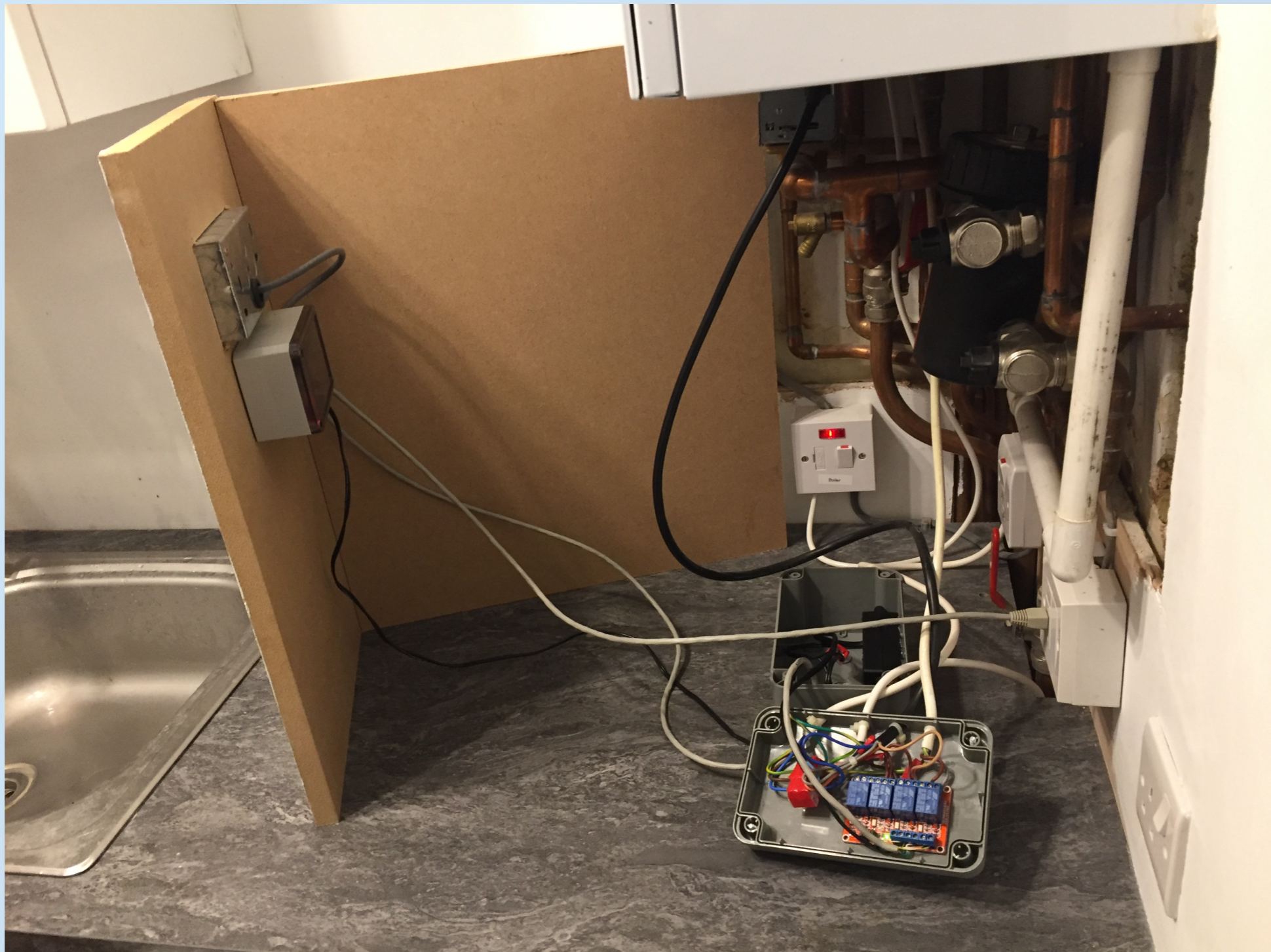
UDP Server example

```
void loop() {  
  ether.receivePacket();  
  
  if (udp.havePacket()) {  
    Serial.print("Received UDP from: ");  
    udp.packetSource().println();  
  
    Serial.print("Packet length: ");  
    Serial.println(udp.payloadLength(), DEC);  
  
    if (udp.payloadEquals("on")) {  
      Serial.println("** LED On **");  
      digitalWrite(NANODE_LED_PIN, LOW);  
      udp.sendReply("ok");  
    } else if (udp.payloadEquals("off")) {  
      Serial.println("** LED Off **");  
      digitalWrite(NANODE_LED_PIN, HIGH);  
      udp.sendReply("ok");  
    } else {  
      Serial.println("Invalid Message");  
      udp.sendReply("err");  
    }  
  }  
}
```

HTTP Server example

```
void loop() {  
    ether.receivePacket();  
  
    if (http.isGet(F("/"))) {  
        http.printHeaders(http.typeHtml);  
        http.println(F("<h1>Hello World</h1>"));  
        http.sendReply();  
    } else if (http.isGet(F("/text"))) {  
        http.printHeaders(http.typePlain);  
        http.println(F("This is some plain text"));  
        http.sendReply();  
    } else {  
        http.notFound();  
    }  
}
```


IPv6 Control of my Boiler





Boiler

Radiators

Under
Floor



[2001:8b0:ffd5:3:a81c:5dff:fefe:e786]



Boiler Controller

Radiators:

On

Toggle

Underfloor:

Off

Toggle

The Challenges

- Tiny amounts of RAM
- Keeping the codebase small
- Easy to read and write code
- Debugging

What's Next?

- TCP / HTTP Client
- MQTT
- COAP
- Over-the-air updates

github.com / njh / ethersia