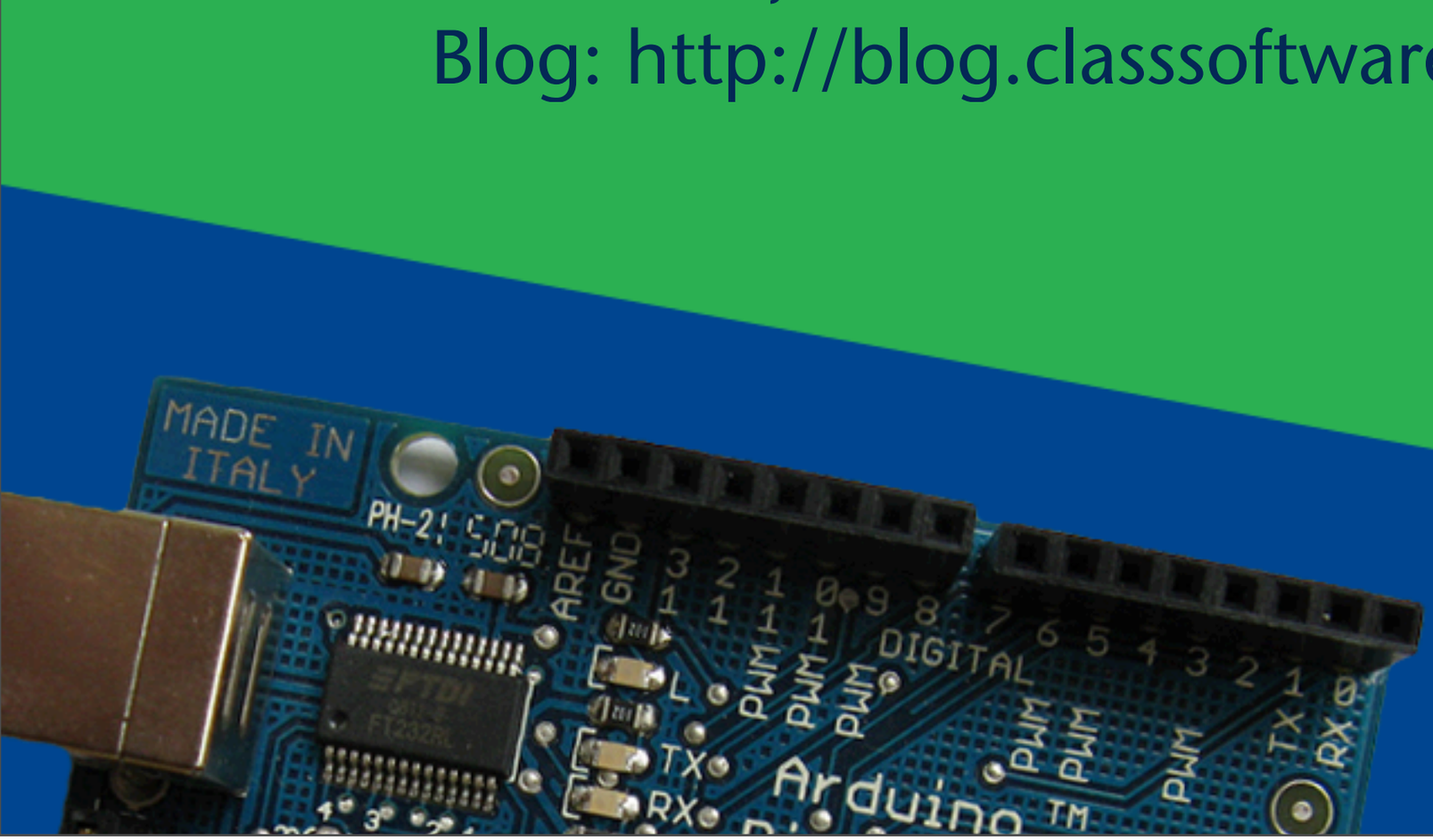
Arduino Platform Part I

Justin Mclean
Class Software

Email: justin@classsoftware.com
Twitter: @justinmclean
Blog: http://blog.classsoftware.com



asdevs

Who am I?

- Director of Class Software for almost 15 years
- Developing and creating web applications for 15 years
- Programming for 25 years
- Adobe certified developer and trainer in Flex and ColdFusion
- Adobe Community Champion
- Based in Sydney Australia

Arduino

Hardware and software overview

Arduino

- Open source hardware and software platform
- Free software
- Easy to program
- Low cost hardware
- Several physical form factors

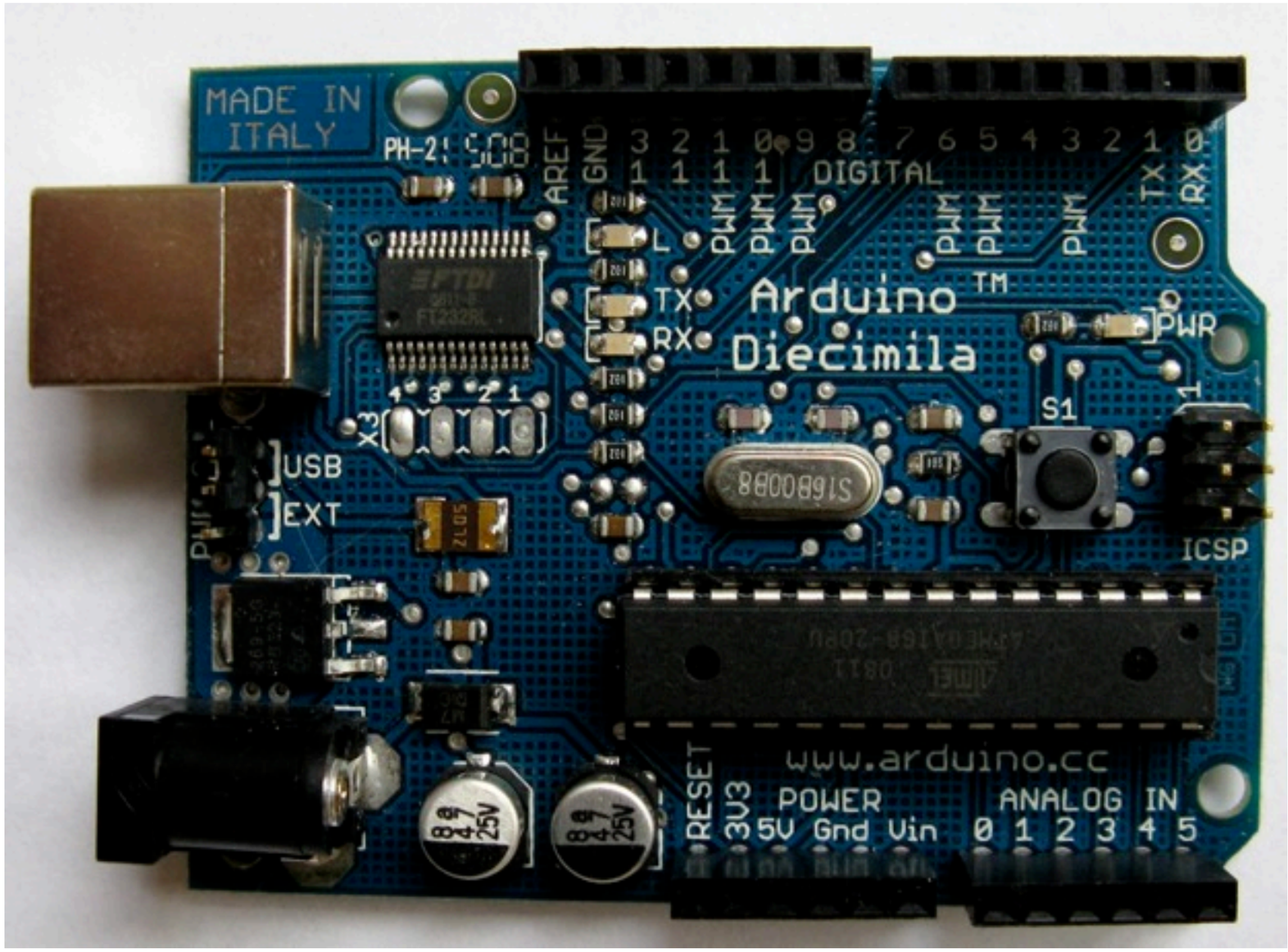
Hardware

- ATmega micro-controller from Atmel
- Arduino Duemilanove and Uno
- Arduino Pro and Pro mini
- Lillypad (wearable)
- Funnel IO
- Mega
- Many others

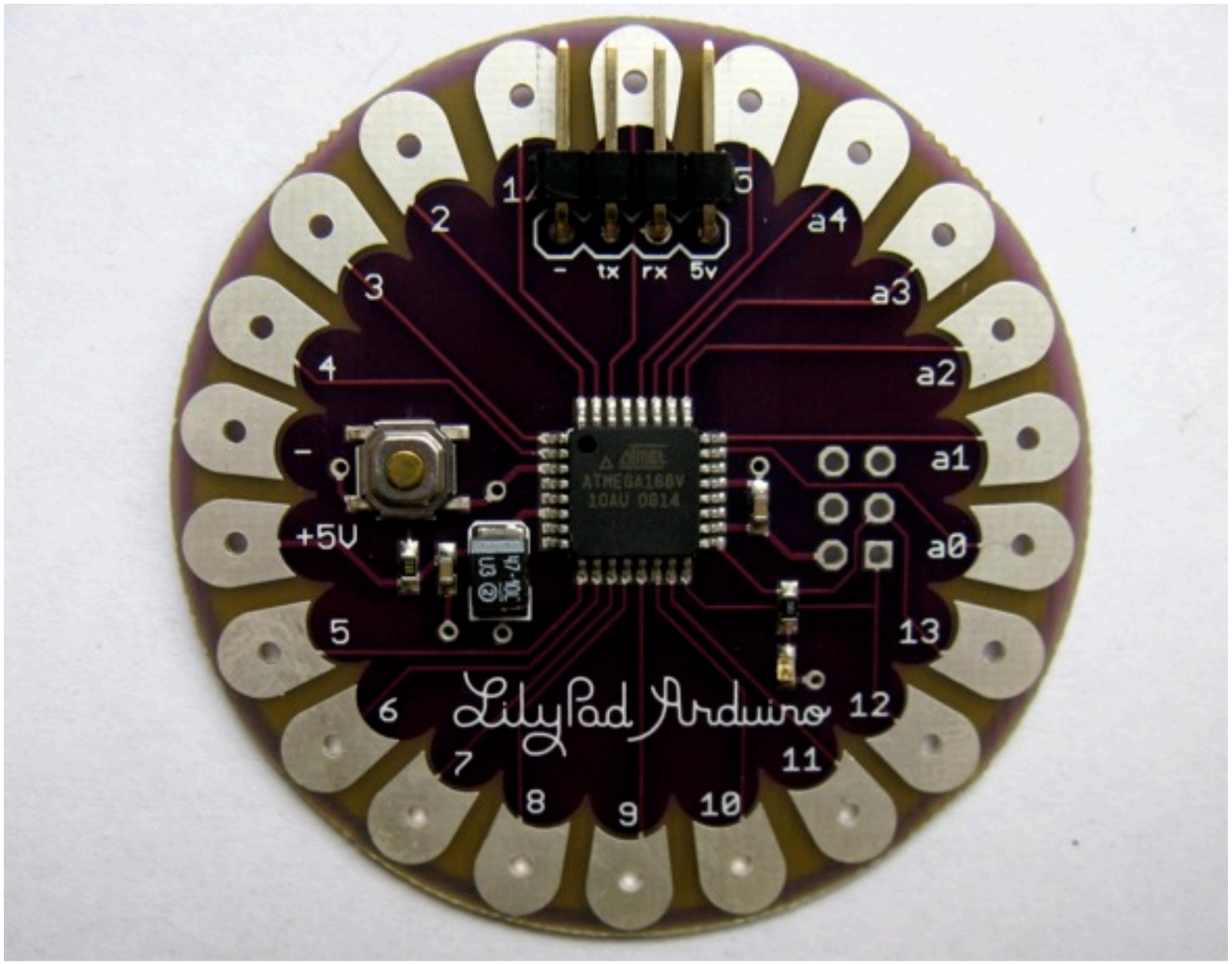
Arduino Board

- Connects via USB
- Power from USB or plug
- Digital inputs/outputs
- Analogue inputs
- PWM outputs - pulse width modulation
- Reset button

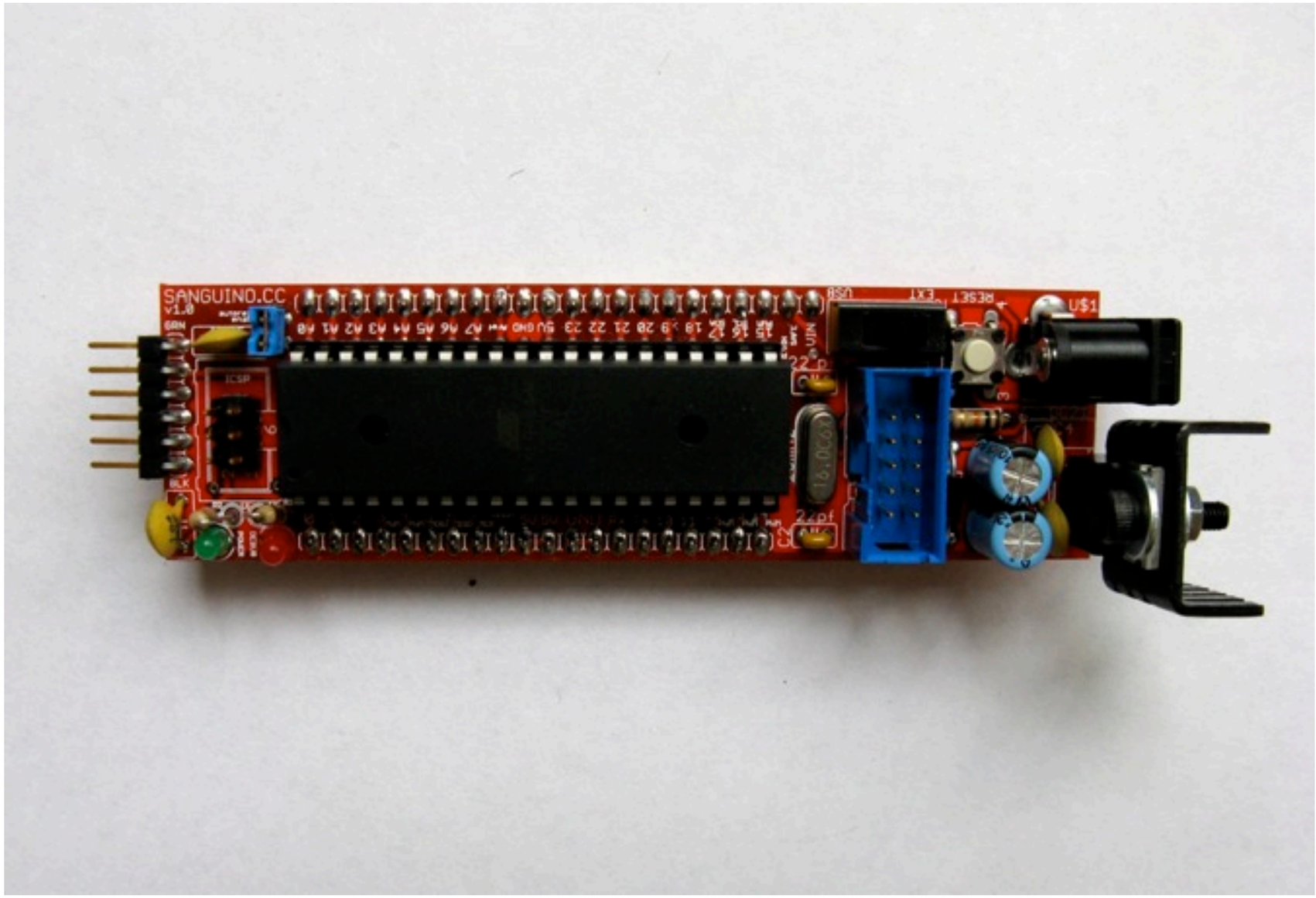
Arduino Boards



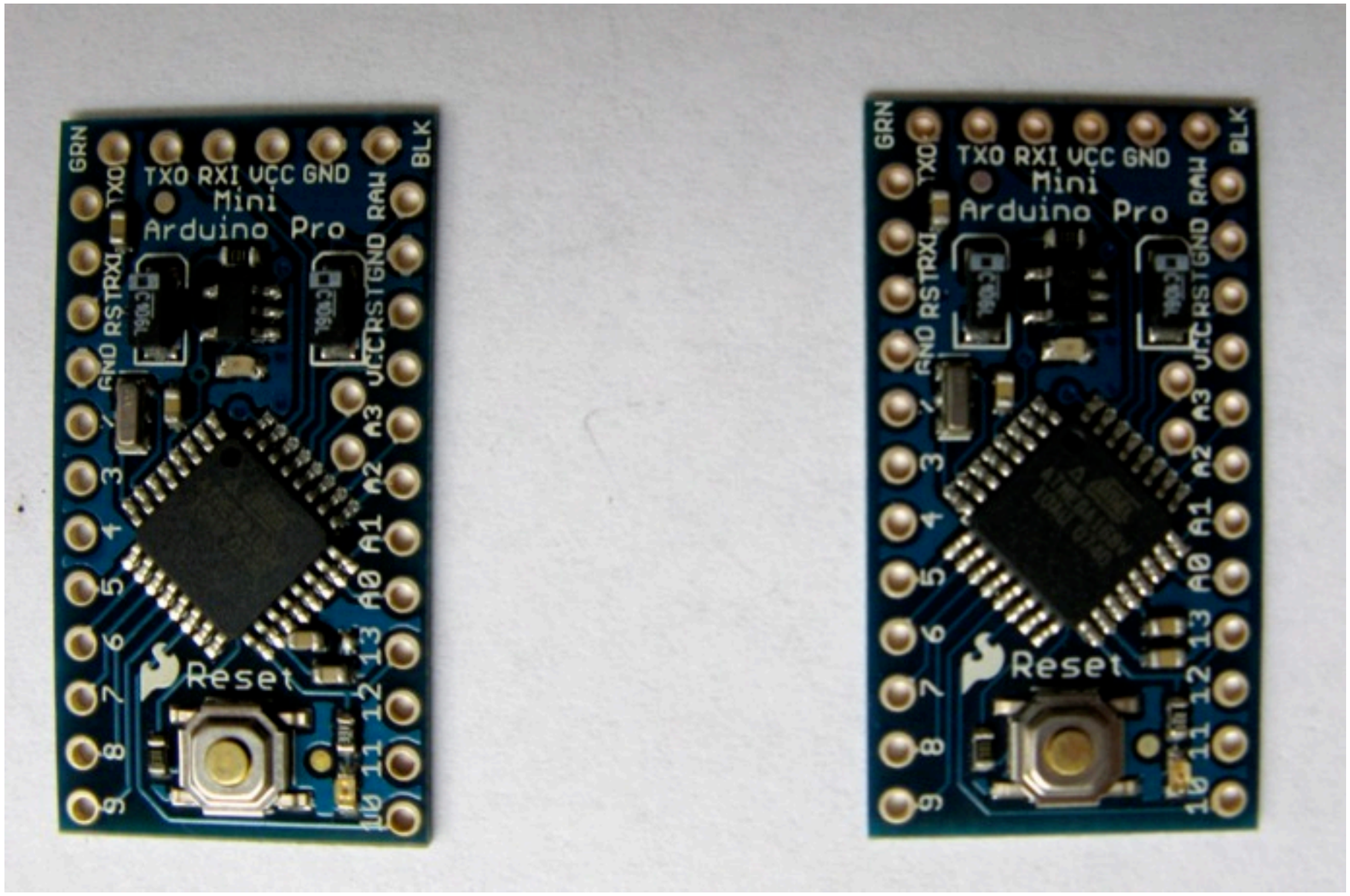
Arduino Boards



Arduino Boards



Arduino Boards



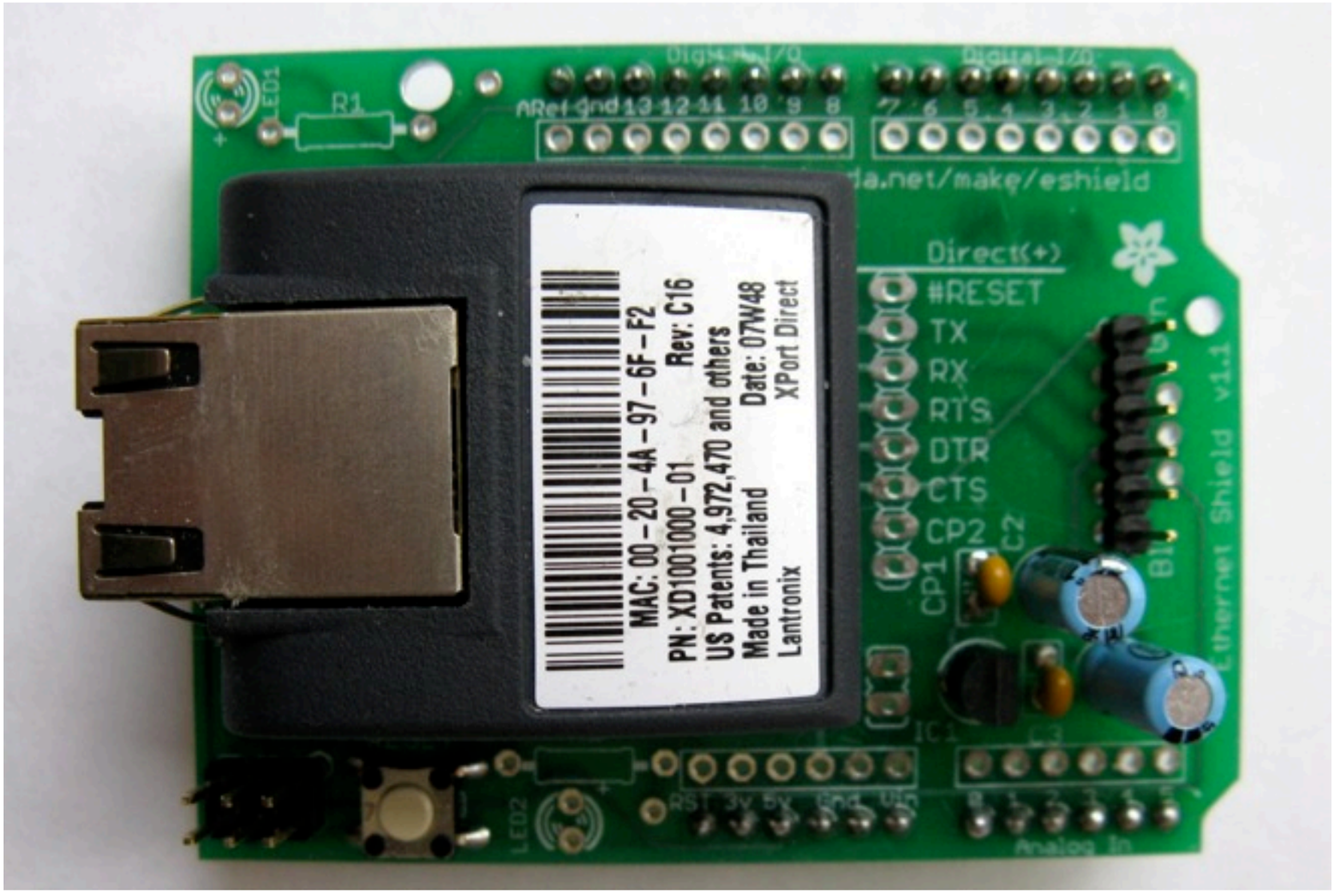
ATmega328

- High performance low power RISC
- 16 Mzh up to 16 mips
- 32K Flash (2K used for bootloader), 1K EEPROM, 2K SRAM
- SPI and 2 wire serial interfaces
- External interrupts, timers, pulse width modulation
- Harvard architecture (modified)

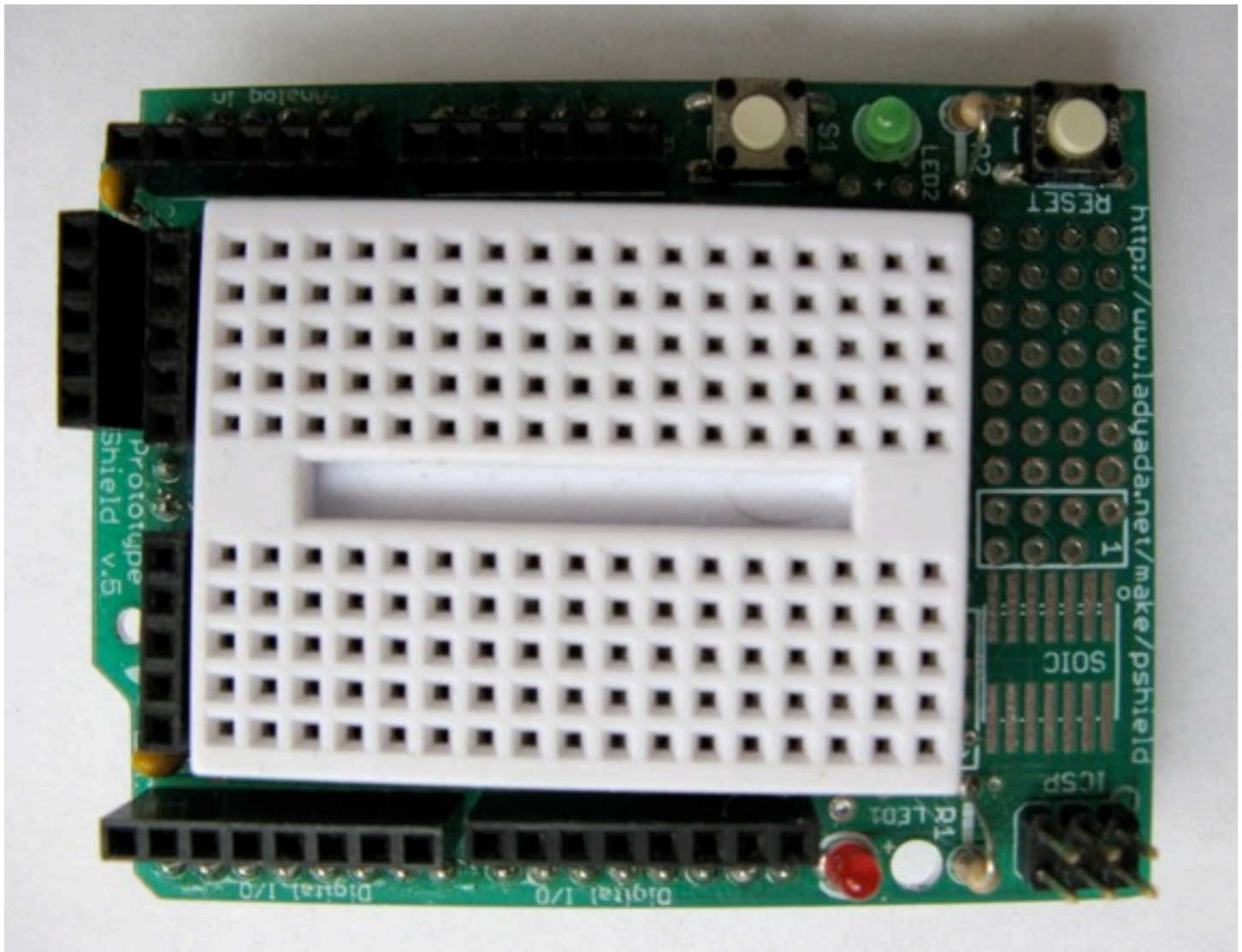
Shields

- Plug on top of Arduino
- Many available
- Can make your own
- Can be stacked

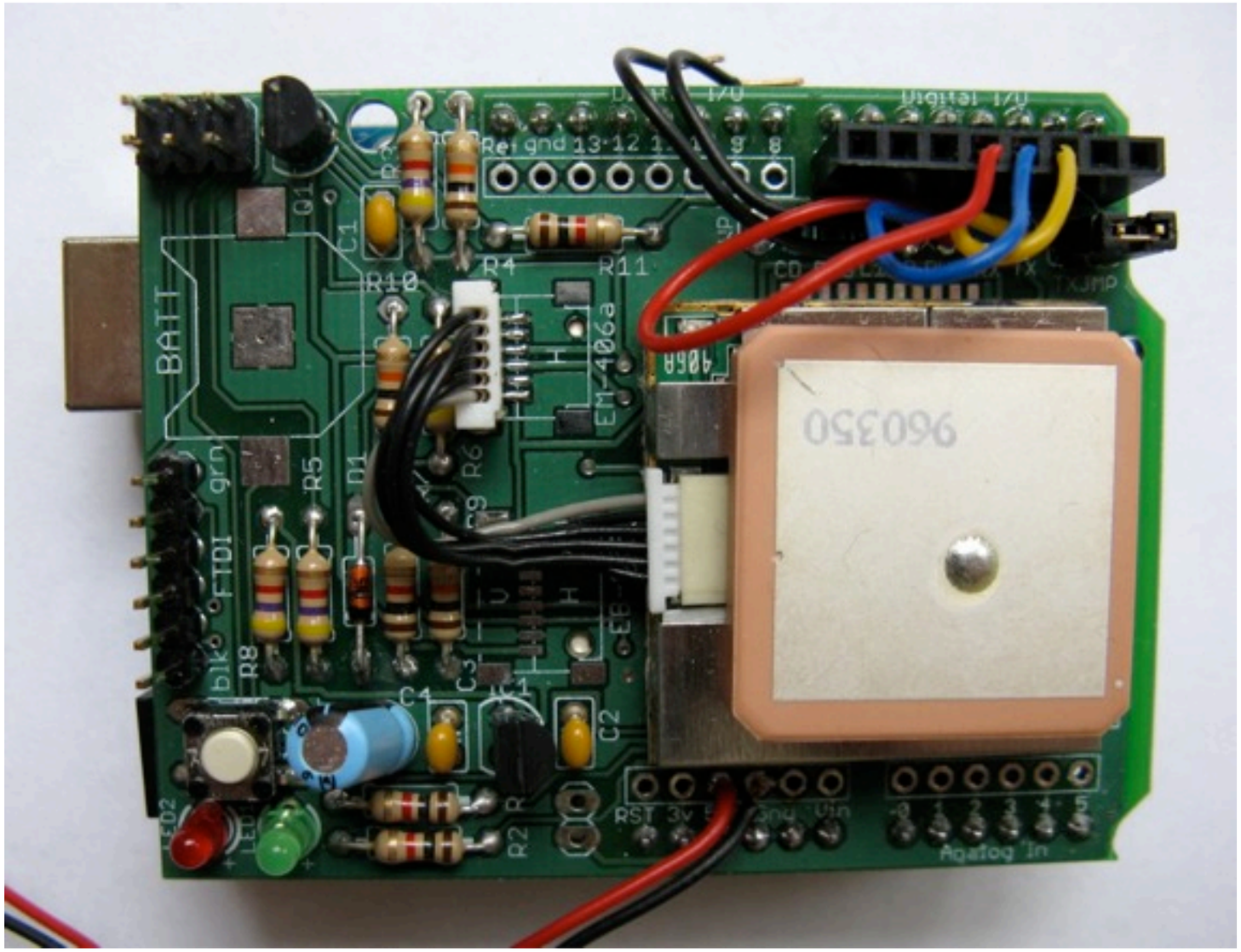
Arduino Shields



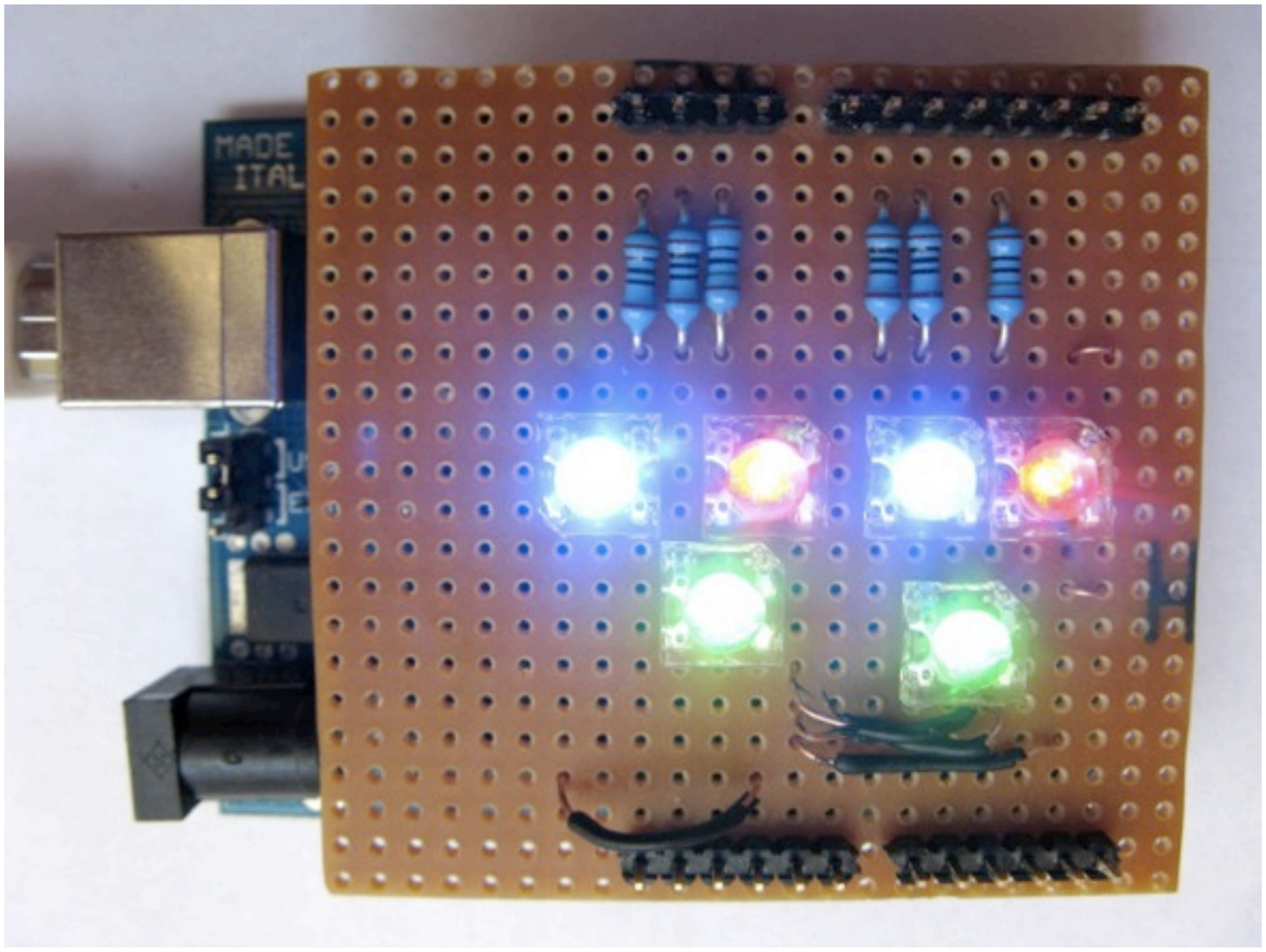
Arduino Shields



Arduino Shields



Arduino Shields



IDE

- IDE open source and cross platform
- Projects are called sketches
- Many open source sketches and libraries available

Arduino IDE

SuperFluxRGB

```
int redLed[] = {3,9};
int greenLed[] = {5,10};
int blueLed[] = {6,11};

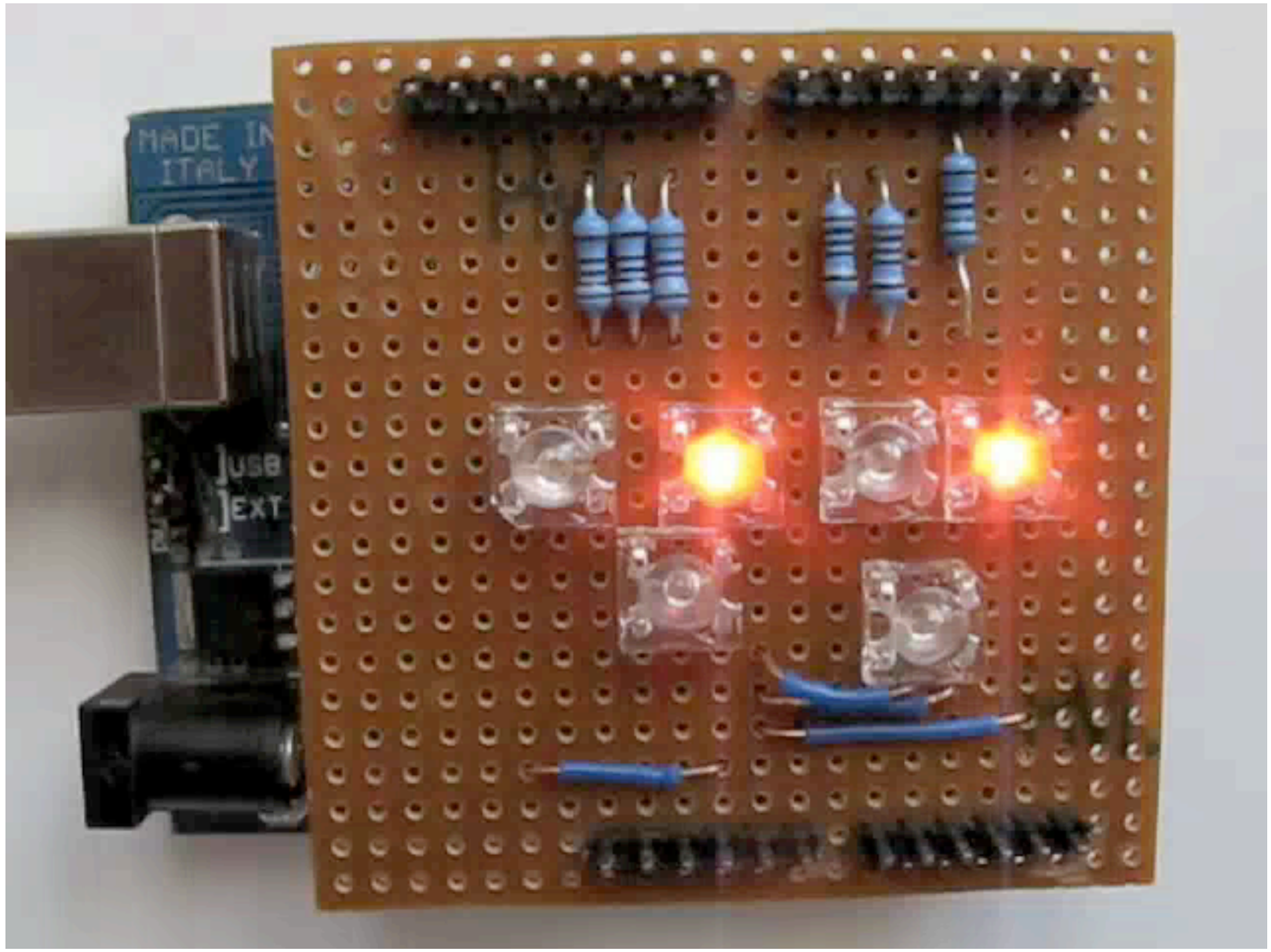
float redFactor = 1.0;
float greenFactor = 76.0/160.0;
float blueFactor = 76.0/85.0;

void setLedColour(int led, int red, int green, int blue) {
  int redMod = int(red*redFactor);
  int greenMod = int(green*greenFactor);
  int blueMod = int(blue*blueFactor);

  Serial.print(redMod, DEC);
  Serial.print(' ');
```

Led Shield Demo

Led Shield Demo

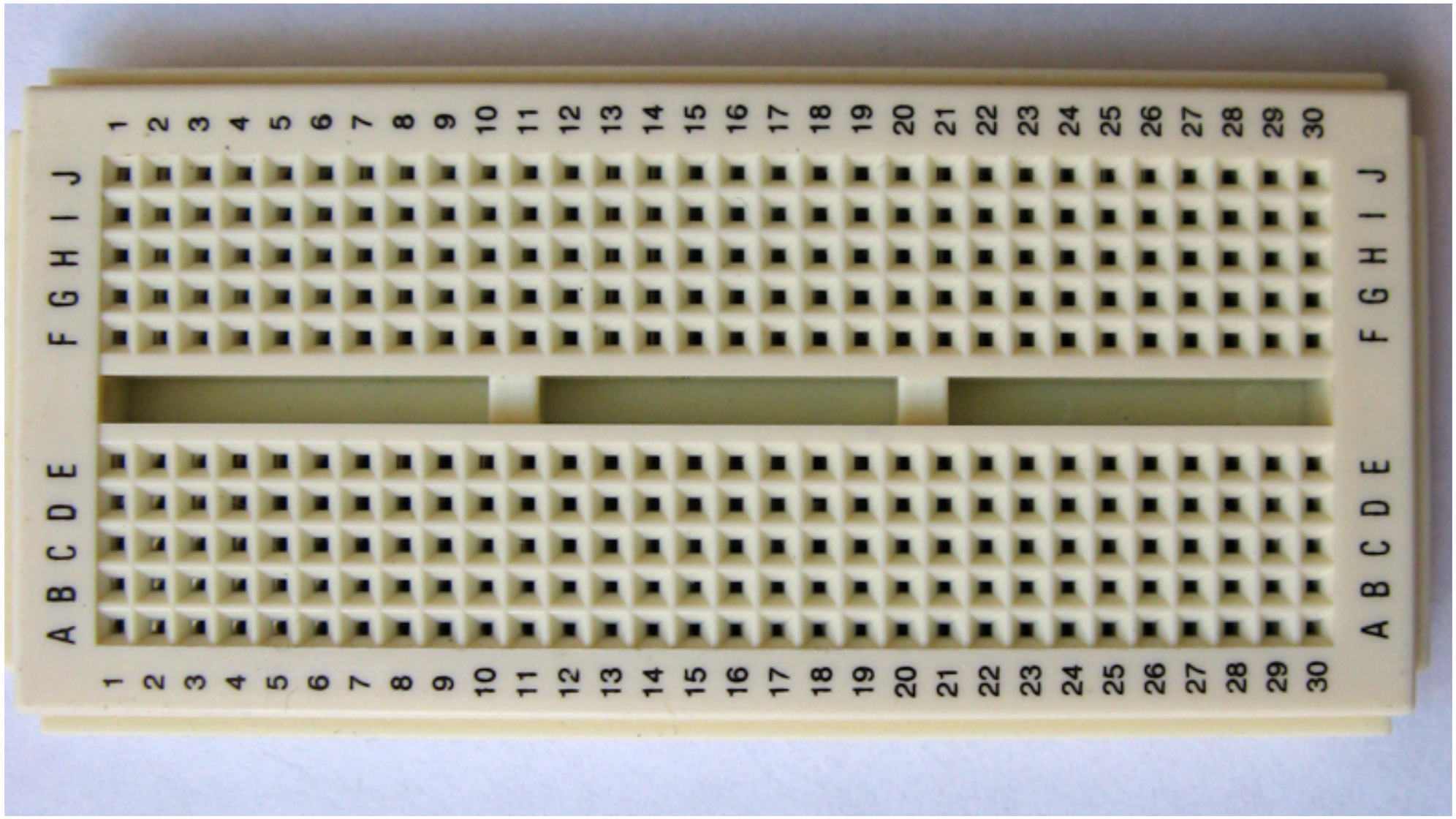


Programming

- C/C++ language based on wiring
- GCC under the hood
- Write code and compile in IDE
- Upload compiled code via USB
- Can monitor serial port
- Uploaded program is in non volatile memory

Prototyping

- Breadboards
- Serial port


A B C D E F G H I J
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Breadboard

Digital Outputs

Turning on a LED

Digital Inputs/Outputs

- Digital pins on Arduino are dual purpose
- Digital logic and voltage on = 5V off = 0V
- Can be set to be input or output via pinMode

Variables

- Data types include boolean, char, byte, int, long , float, double, string and array.
- int 16 bits, long 32 bits, float 32 bits
- Strings are nul terminated ‘\0’
- Declare by <datatype> <variable name>; eg int i;

Setup Function

- Used for initialisation
- Run when program loaded or board reset
- Best place to place calls to pinMode

LEDs

- Light emitting diodes
- Current will only flow in one direction
- Emits light with current applied
- Don’t connect directly to power source use in series with a resistor

Resistors

- Resistors limit current flowing through them
- Value and tolerance indicated by colour bands
- Resistor values for LEDs
- For RGB or LED digits you need multiple resistors.

Loop Function

- Place main code here
- Set digital output via digitalWrite
- Output 13 is connected to led on board

Debugging via Serial Port

- Use Serial.begin to set speed
- Use Serial.print or Serial.println to output
- Use serial monitor in IDE to view

Test Program

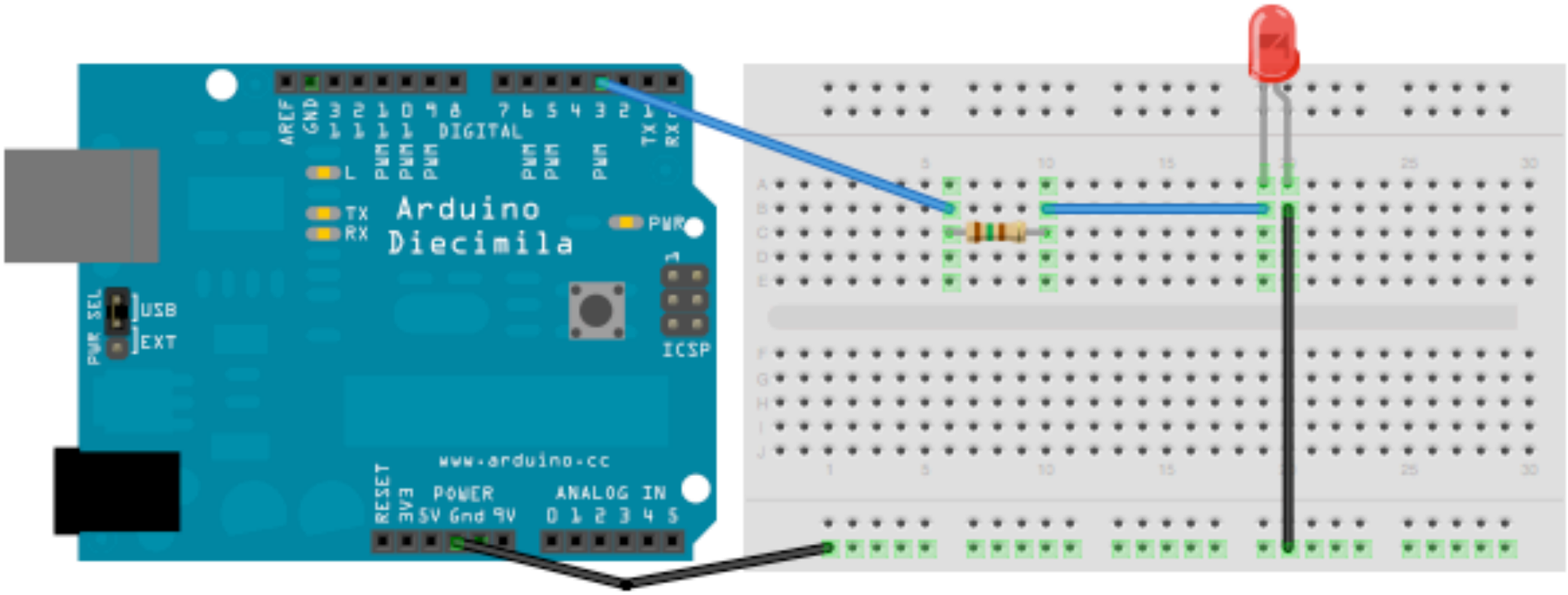
- Set output mode in setup function
- Turn on pin 13 LED in loop function
- Verify
- Upload

Breadboards

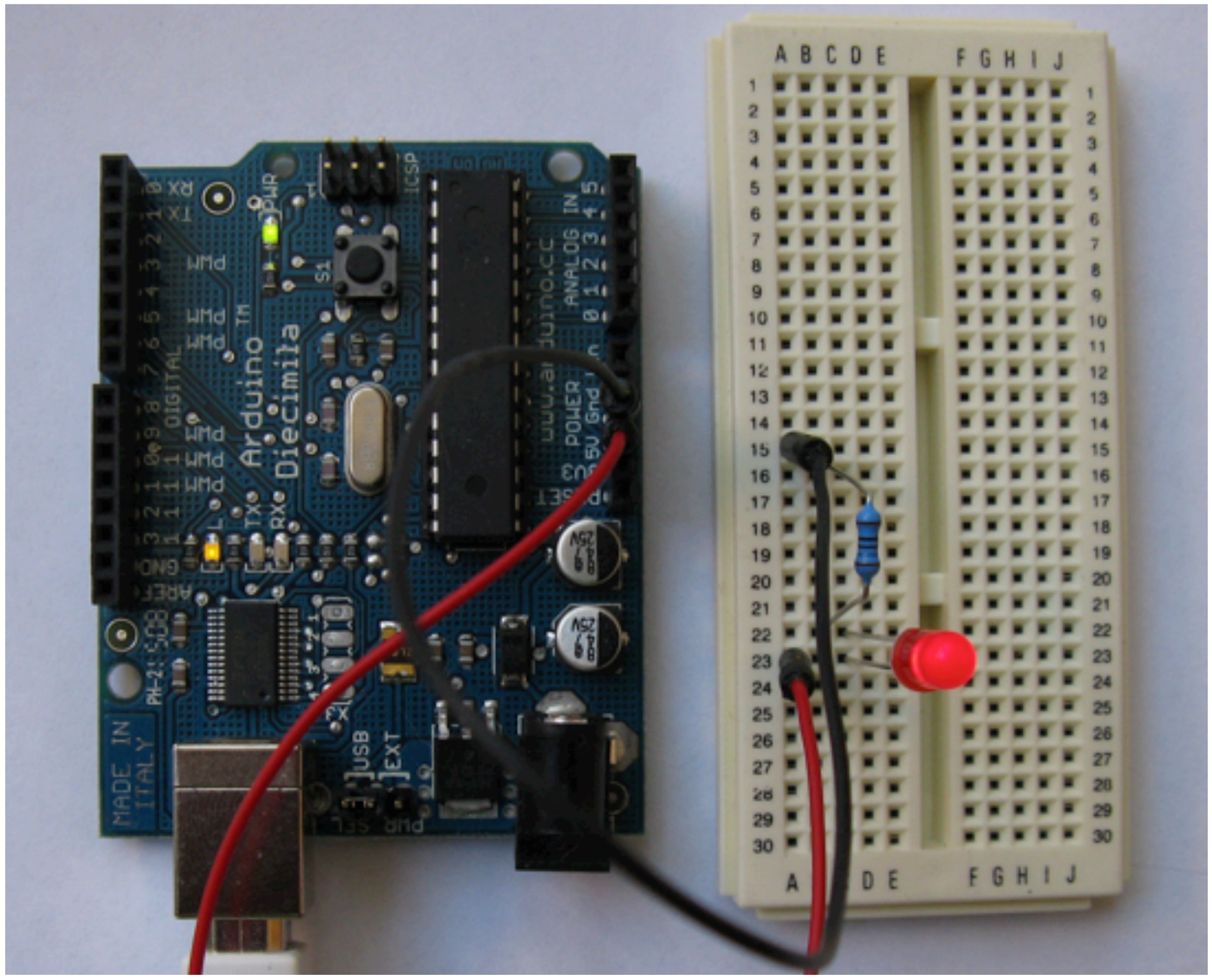
- Tracks under board
- Separated into 2 or 4 sections with optional power/ground sections
- Standard spacing (imperial) so most components can plug straight in

LED Circuit

- Add LED and resistor to breadboard
- Connect to Arduino
- Change pin no to 3


AREF GND 13 12 11 10 9 8 7 6 5 4 3 2 1 0
DIGITAL
PWM
L
TX
RX
Arduino
Diecimila
PWR
PWR SEL
USB
EXT
ICSP
www.arduino.cc
RESET 3V3 5V Gnd Vin
POWER
ANALOG IN
0 1 2 3 4 5

LED Circuit


A B C D E F G H I J
Arduino Diecimila
DIGITAL
ANALOG IN
POWER
MADE IN ITALY

LED Circuit

Blink

- Make led blink by calling delay

Analogue Inputs

Connecting sensors

Reading Inputs

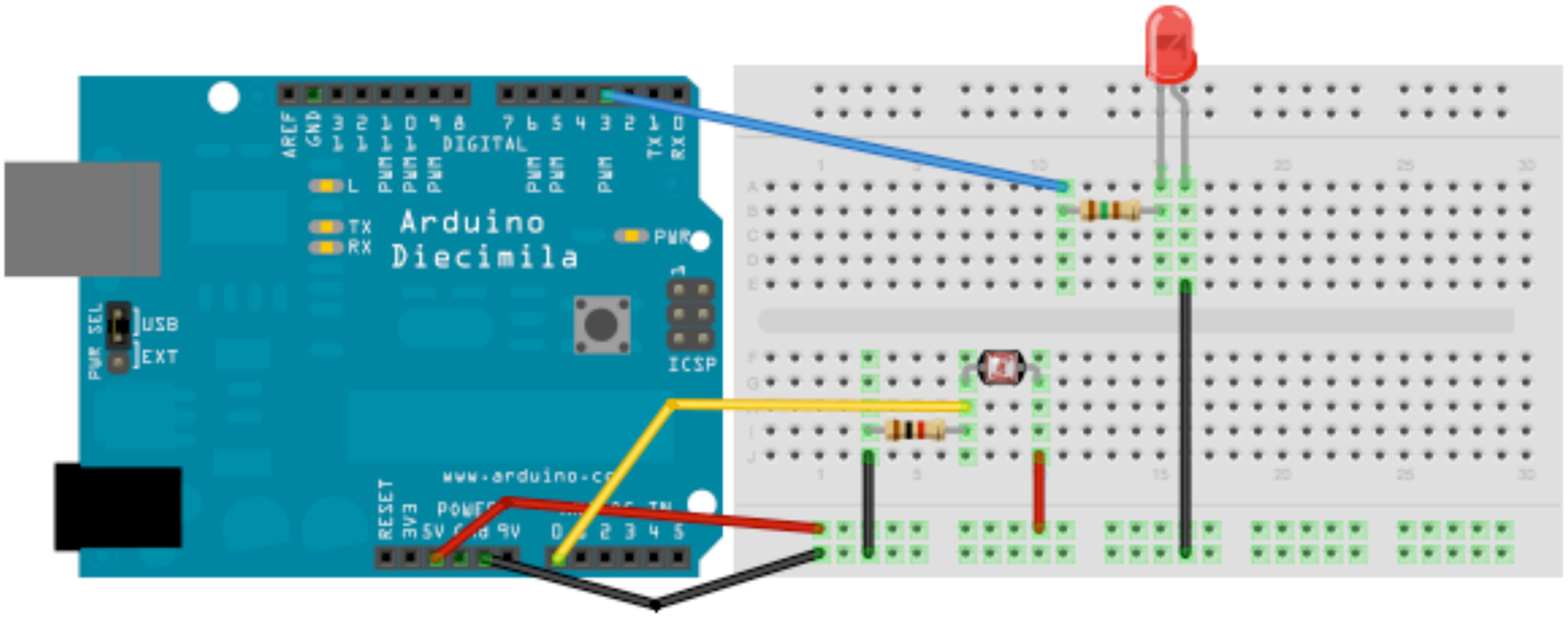
- Can read values via analogRead
- Result is in range 0 to 1023 (10 bits)
- 0V = 0 and 5V =1023

Analogue Input

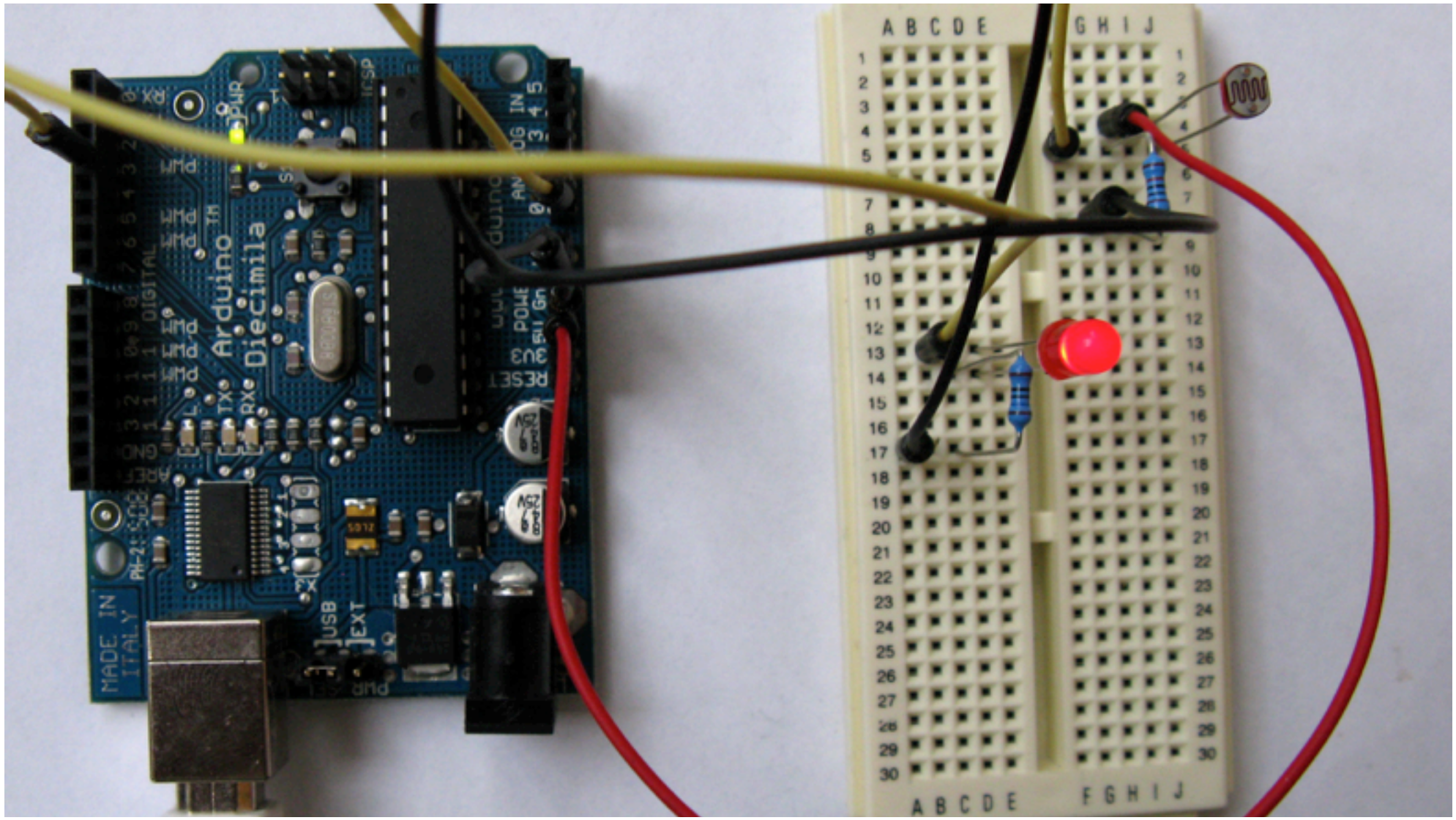
- Read potentiometer value
- Set led flash rate based on value

LDR

- Light dependant resistor (high resistance)
- Set flash rate based on value of LDR


AREF
GND
DIGITAL
TX
RX
L
TX
RX
Arduino
Diecimila
PWR
PWR SEL
USB
EXT
ICSP
RESET

LDR Circuit



LDR Circuit

Web Servers

Turning an Arduino into a web server

Web Servers

- Simpler than you think
- A web server:
  - Listens for connections
  - Parse requests
  - Send back status messages/resources requested

Ethernet Shields

- Many shields available
- Two main types DHCP/non DHCP
- Some config required in both cases
- Non DHCP you set IP and MAC address in code
- May have support for SD cards

Power Over Ethernet

- Power and data
- Power injection via
  - Split cable
  - Hub or injector
- Different standards/voltages
- May need power regulation

Ethernet Library

- Standard ethernet library
- Can code as client or server or both
- Create server like so:
  Server server(80);
- Bare bones server about 20 lines of code and 5K compiled

IP and MAC address

- Set IP address and MAC in your code
- Need to be careful with duplicates
- Set up like so:
  Server.begin(ip, mac);

HTTP Protocol

- HyperText Transfer Protocol
- Used by web servers to transfer web pages to be displayed in your web browser
- Connection (usually) on port 80

TCP Connections

- TCP three way connection handshake
- Client sends SYN with random number (A)
- Server replies with SYN-ACK containing A+1 and random number (B)
- Client replies with ACK containing B+1
- Luckily ethernet library does this for you!

Connection Code

- ```
  Client client = server.available();
  if (client) {
      while (client.connected()) {
          ....
      }
  }
  ```


  ```

HTTP Requests

- Start with request “GET index.html HTTP/1.1”
- Optional headers eg “host: www.domain.com” or “Accept-Language: en”
- Empty line
- Optional message body (POST and other requests)

HTTP Request Hack

- Standard GET request have request followed by blank line
- So just ignore what is requested and check for blank line

Request Code

- ```
  if (client.available() {
      char c = client.read();
      ...
  }
  ```
- ```
  if (c == '\n' || c == '\r') {
     blankline = true;
  }
  ```

HTTP Response

- Send back status line
- Send back content type
- Send (HTML or XML) content

Response Code

- Send on good request
```
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
```

- Send on bad request
```
client.println("HTTP/1.1 400 Bad Request");
client.println("Content-Type: text/html");
client.println();
```

Close Connections

- Clear up after you:
  client.flush();
  client.stop();

Resources

Finding out more information

Arduino Sites

- Ardunio (http://ardunio.cc)
- Tinker It! (http://tinker.it)
- Lady Ada (http://ladyada.net)
- Seeed Studio (http://www.seeedstudio.com)
- Modern Device (http://moderndevice.com)

Electronic Components Suppliers

- Spark fun (http://www.sparkfun.com)
- Electric Goldmine (http://www.goldmine-elec-products.com/)
- Digikey (http://www.digikey.com/)
- Farnell (http://ww.farnell.com/)

Other Sites

- Make magazine (http://makezine.com/)
- Evil Mad Scientist (http://evilmadscientist.com)
- NYC Resistor (http://nycresistor.com)