Faculty of Engineering

Intelligent Systems Engineering

Embedded Systems

0.0 Introduction

What is a Microcontroller?

- A small computer on a single chip
  - containing a processor, memory, and input/output
- Typically "**embedded**" inside some device that they control
- A microcontroller is often small and low cost.

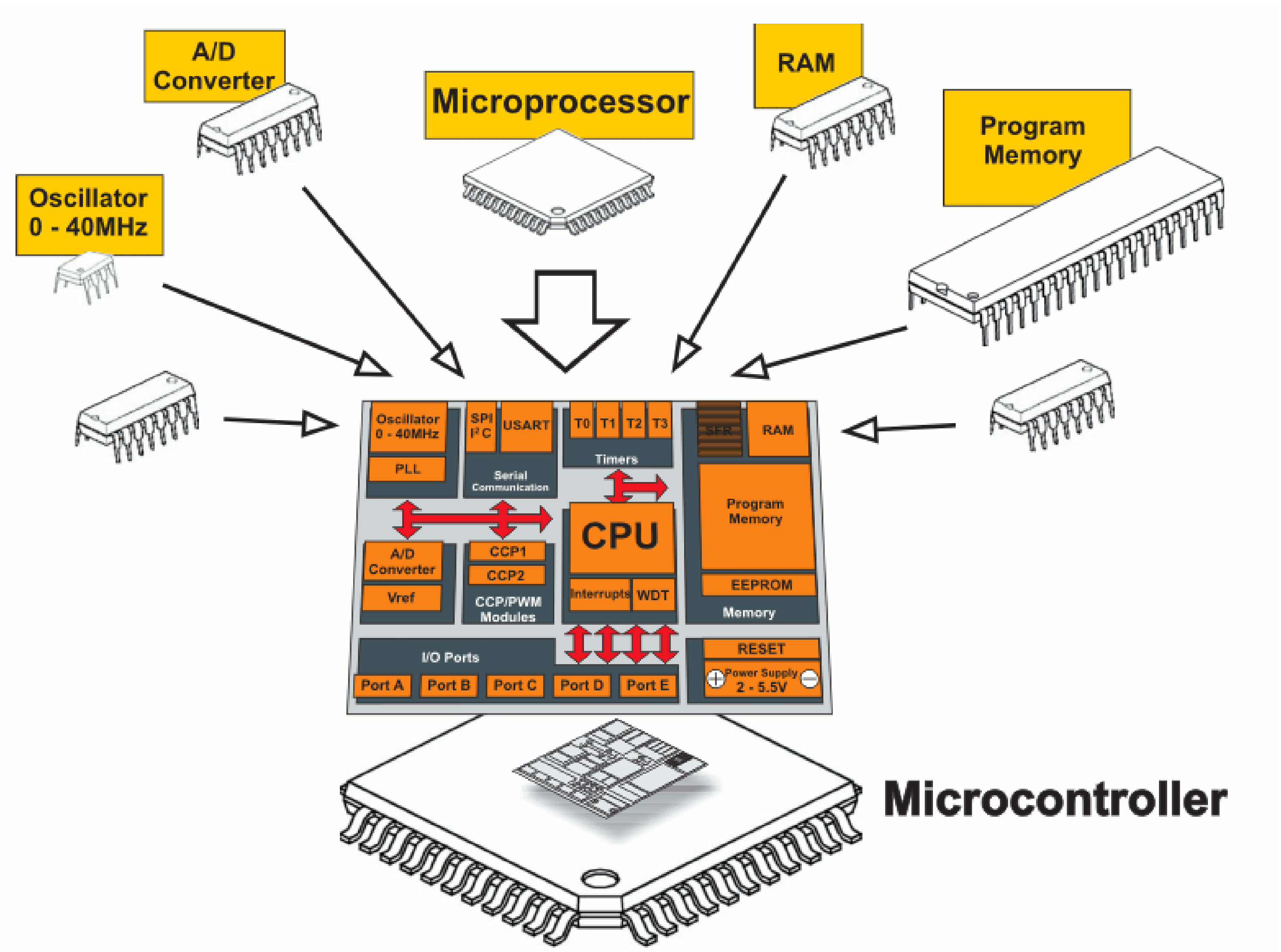
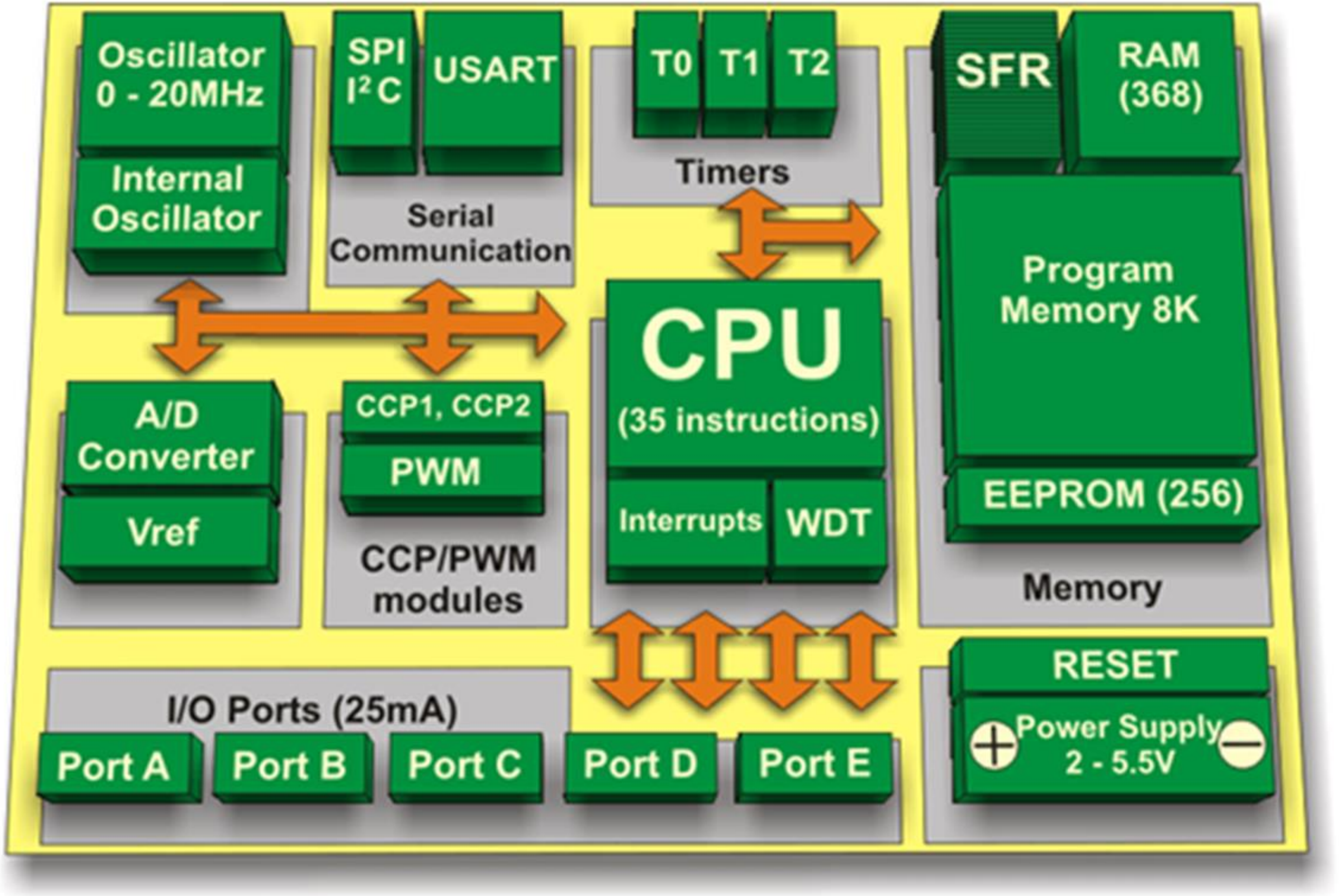
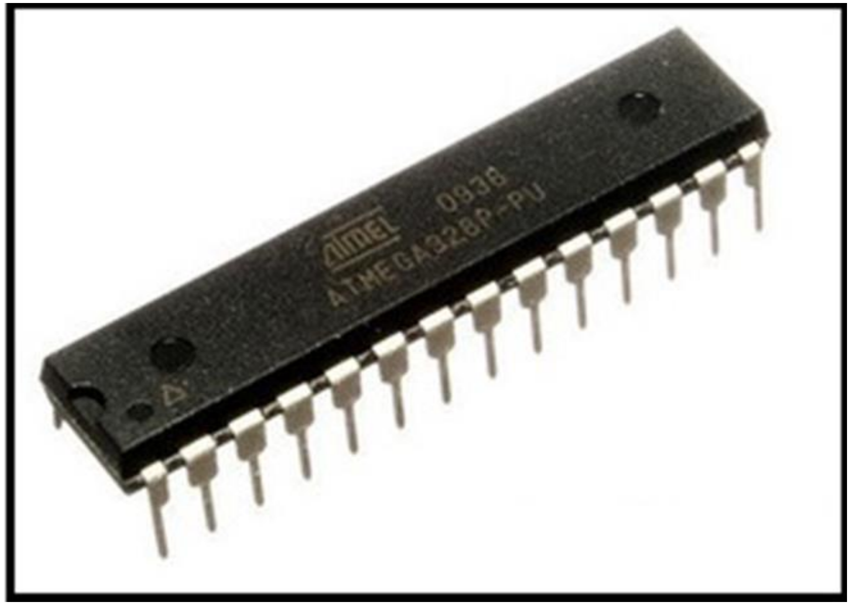


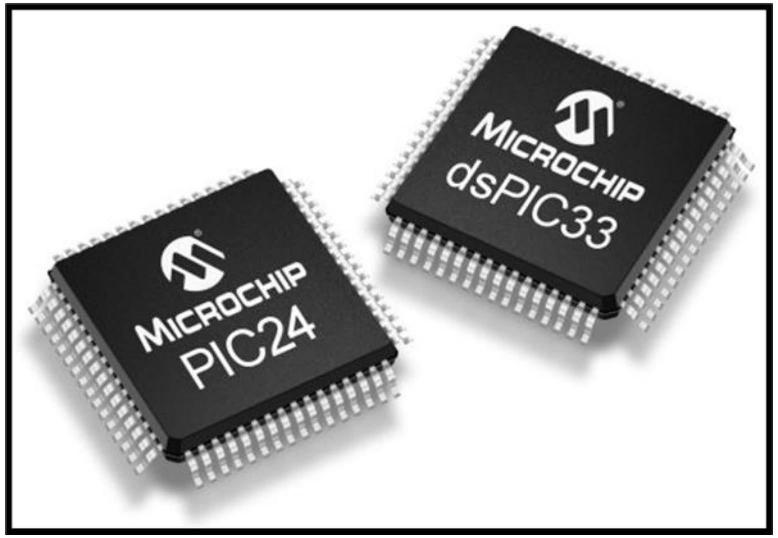
Fig. 0-1 Microcontroller versus Microprocessor

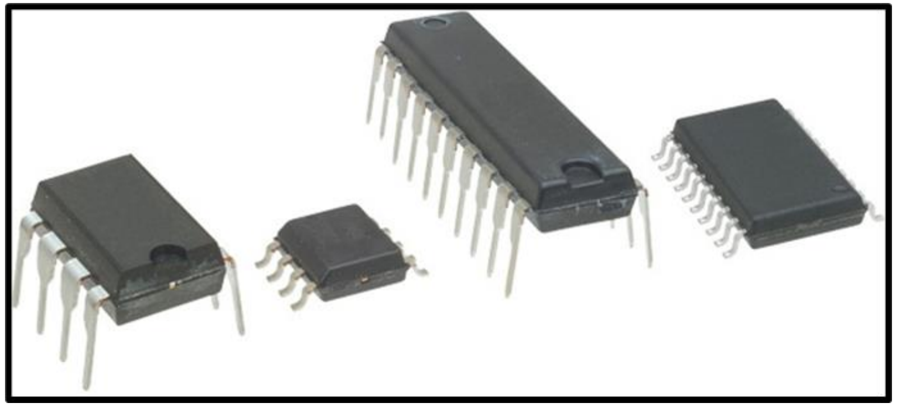
Microcontroller Internal Structure

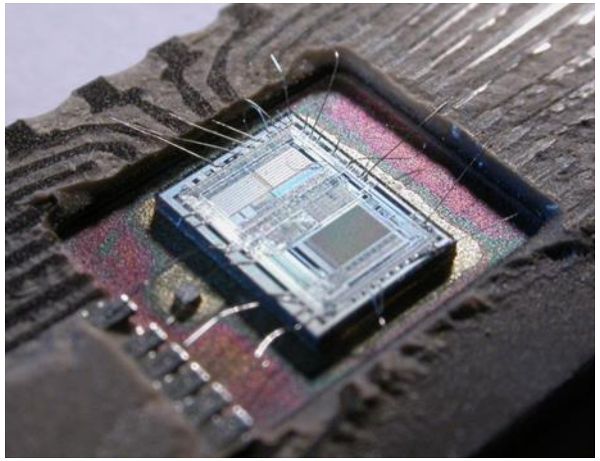


Examples of Microcontrollers?

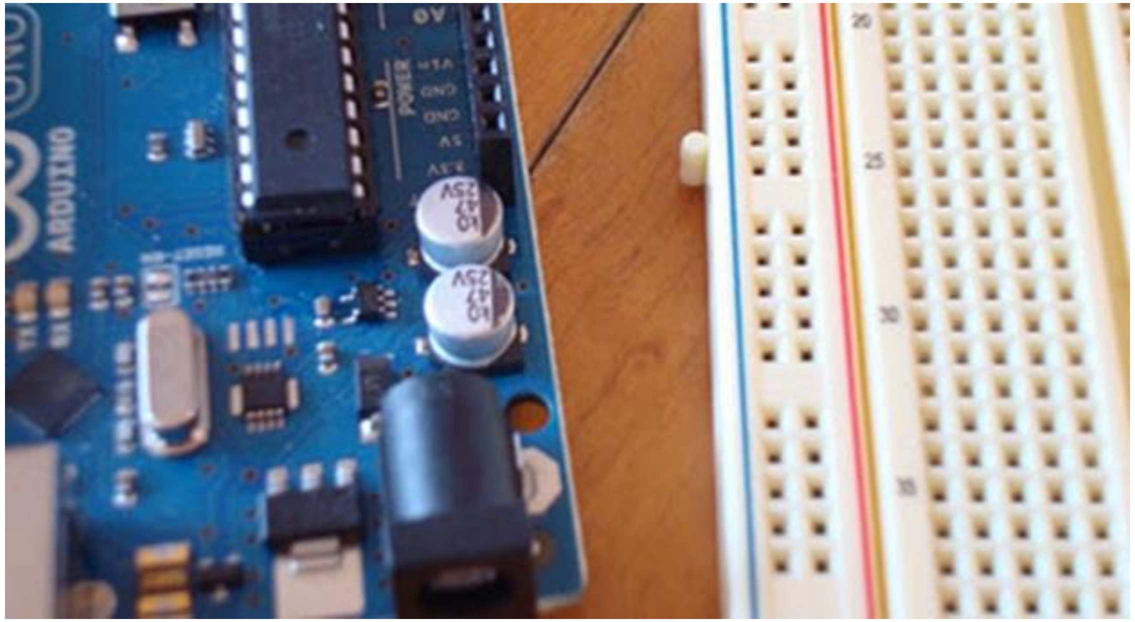








What is a Development Board?

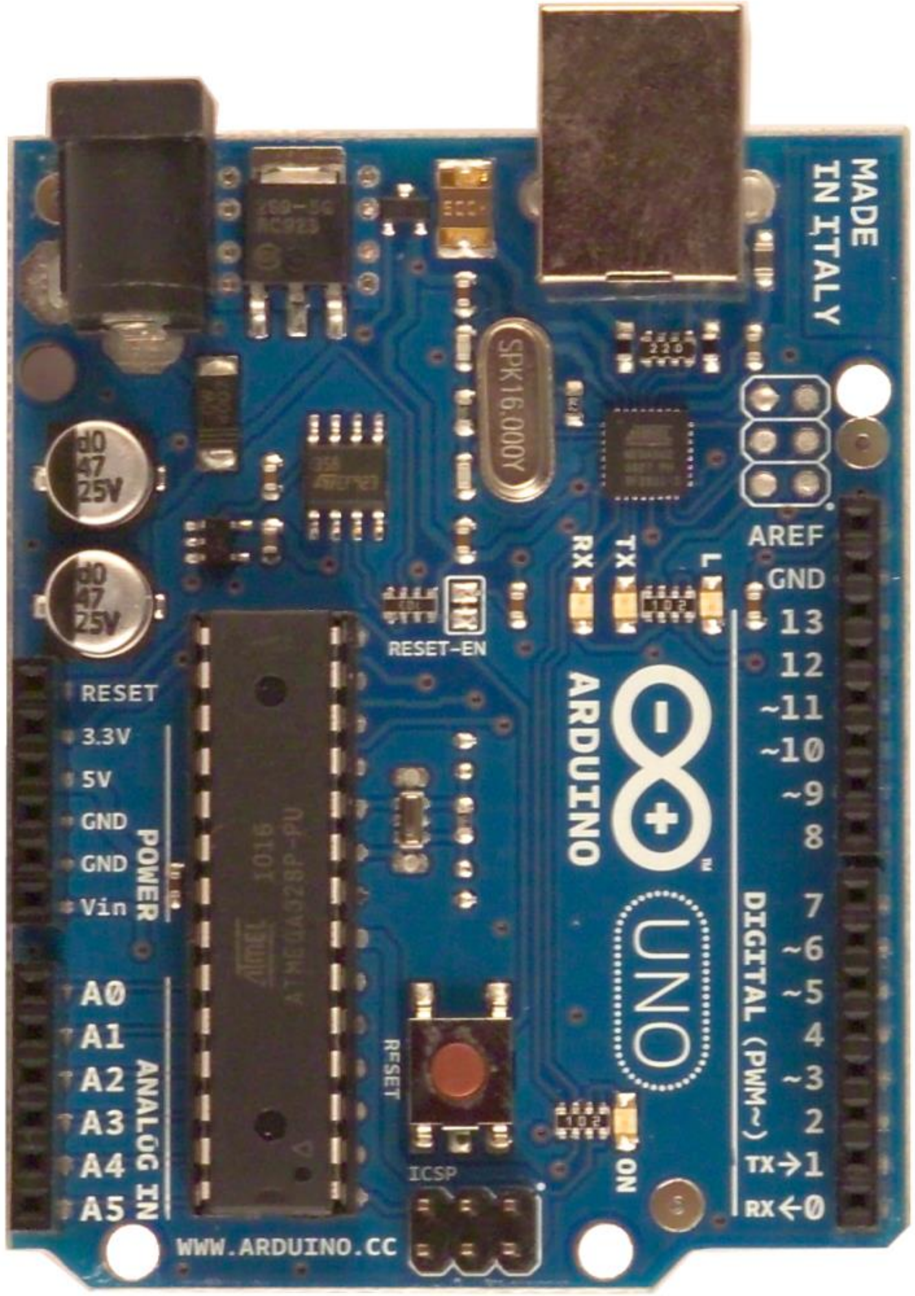


- A printed circuit board designed to facilitate work with a particular microcontroller.

- Typical components include:
  - power circuit
  - programming interface
  - basic input; usually buttons and LEDs
  - I/O pins

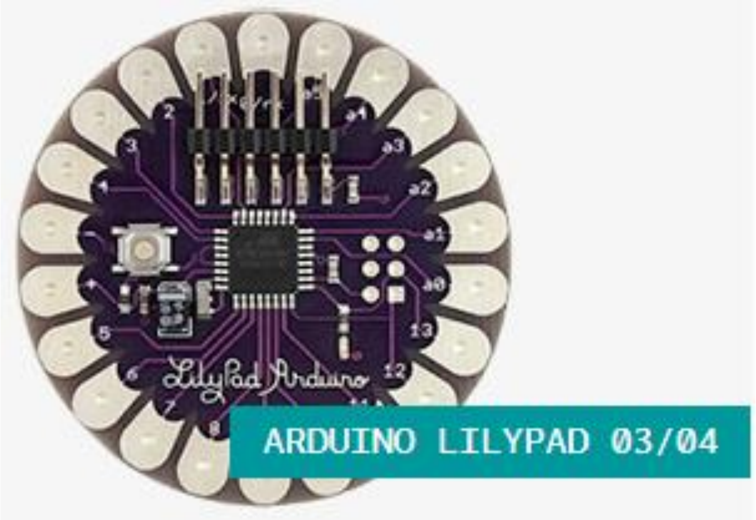
Why Arduino?

- Easy to use platform
- Ease of programming
- Simple USB interface
- Modifiable IDE
- Many shields available

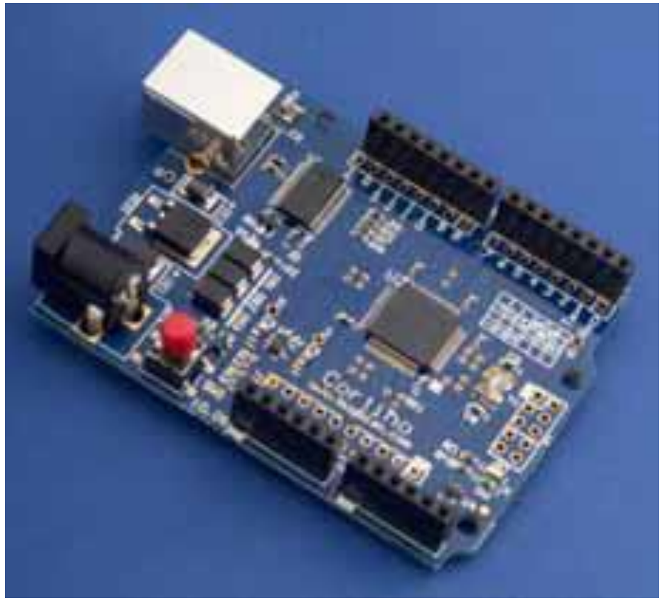


Flavors of Arduino:

- **Many Flavors of Arduino**
  - *Arduino Uno*
  - Arduino Leonardo
  - Arduino LilyPad
  - Arduino Mega
  - Arduino Nano
  - Arduino Mini
  - Arduino Mini Pro
  - Arduino BT



Arduino-like Systems:



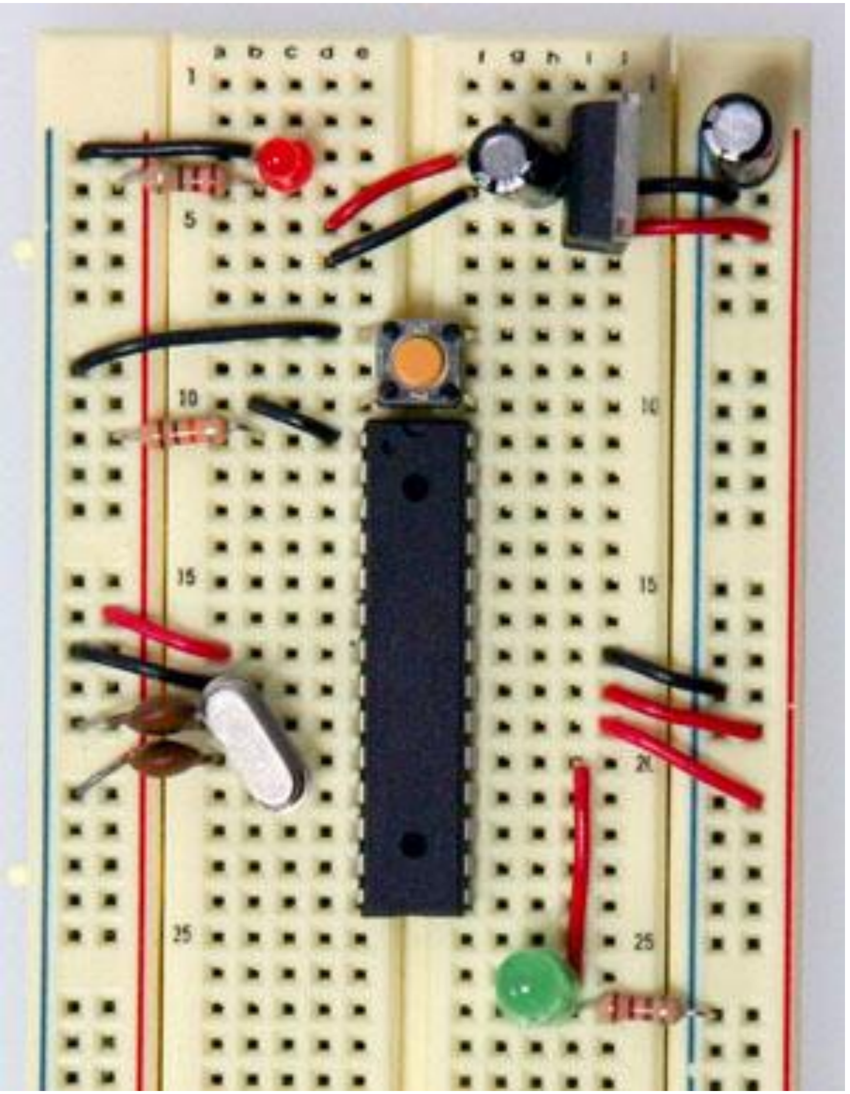
- Cortino (ARM)
- Xduino (ARM)
- LeafLabs Maple (ARM)
- BeagleBoard (Linux)
- Wiring Board (Arduino predecessor)
- ChipKit Uno32/Max32

Arduino Add-ons (Shields):

- TFT Touch Screen
- Data logger
- Motor/Servo shield
- Ethernet shield
- Audio wave shield
- Cellular/GSM shield
- WiFi shield
- Proto-shield
- ...many more

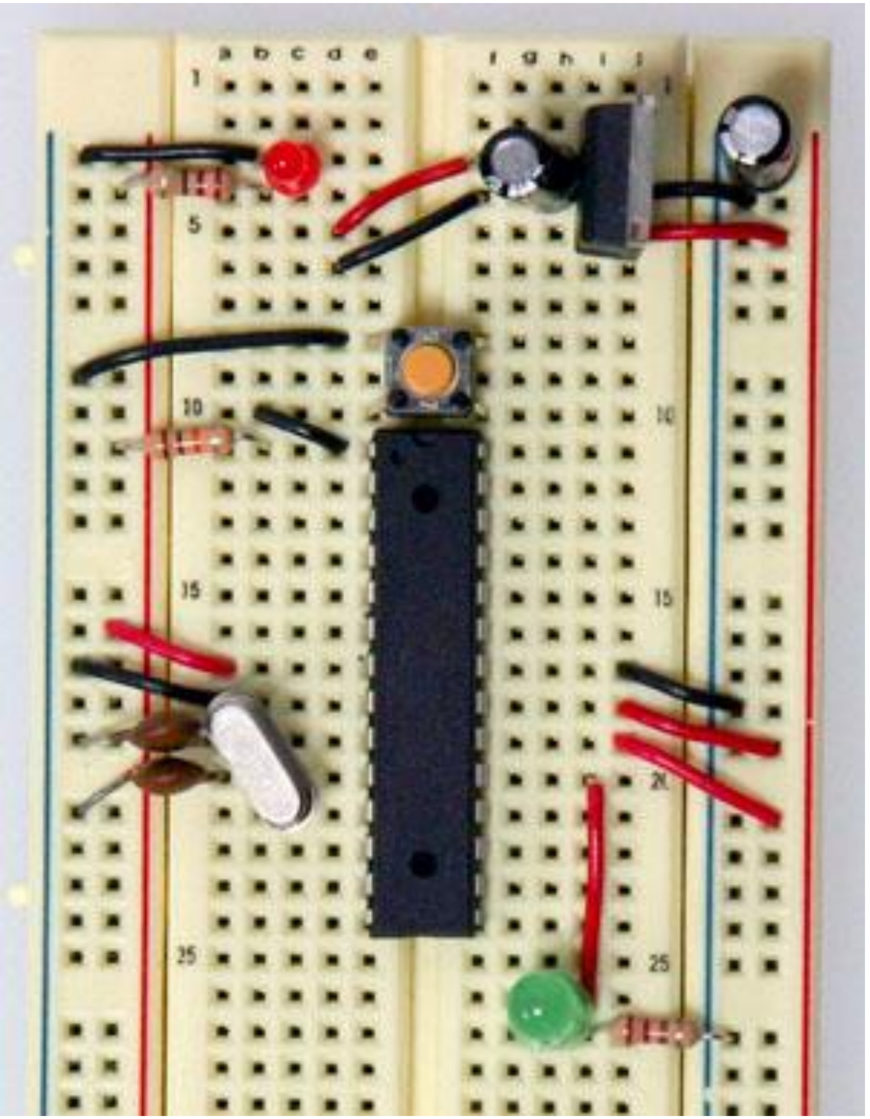
Components of the Arduino:

- ATMega168/328
- 16MHz crystal/filtering capacitors
- Onboard power regulators
- FTDI USB <-> Serial Chip
- Hardware

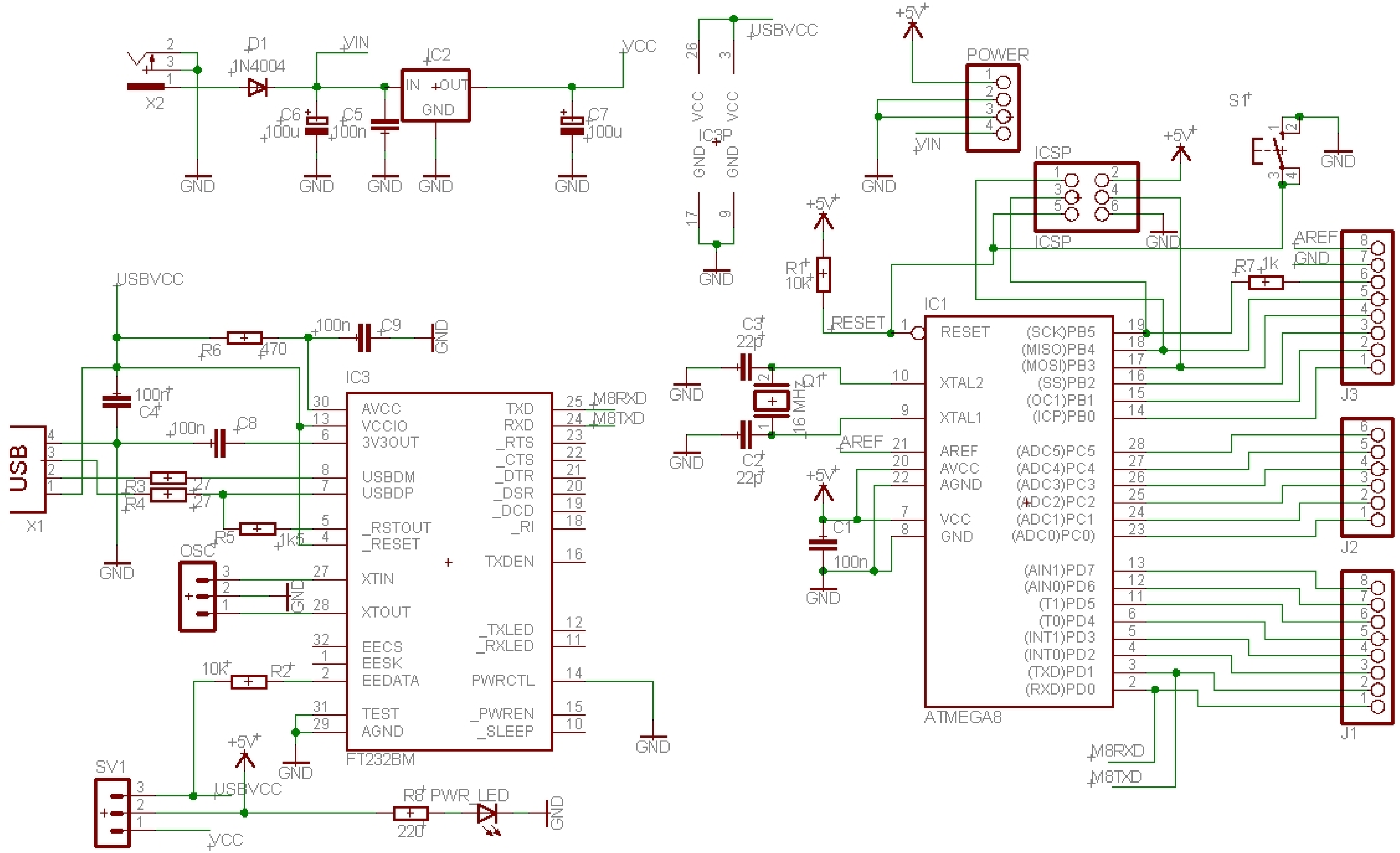


Where to Get an Arduino Board:

- Purchase from online vendor (available worldwide)
  - Sparkfun
  - Adafruit
  - DFRobot
- … or build your own
  - PC board
  - Solderless breadboard



Schematic



X2
D1
1N4004
VIN
IC2
IN
OUT
GND
VCC
C6
100u
C5
100n
C7
100u
GND
USBVCC
26
3
VCC
VCC
IC3P
GND
GND
17
9
+5V
POWER
1
2
3
4
VIN
S1
ICSP
+5V
ICSP
GND
AREF
GND
R7
1k
J3
R1
10k
RESET
IC1
C3
22p
Q1
16 MHz
C2
22p
C1
100n
RESET
XTAL2
XTAL1
AREF
AVCC
AGND
VCC
GND
(SCK)PB5
(MISO)PB4
(MOSI)PB3
(SS)PB2
(OC1)PB1
(ICP)PB0
(ADC5)PC5
(ADC4)PC4
(ADC3)PC3
(ADC2)PC2
(ADC1)PC1
(ADC0)PC0
(AIN1)PD7
(AIN0)PD6
(T1)PD5
(T0)PD4
(INT1)PD3
(INT0)PD2
(TXD)PD1
(RXD)PD0
ATMEGA8
J2
J1
M8RXD
M8TXD
USBVCC
R6
470
C9
100n
C4
100n
C8
100n
USB
X1
R3
27
R4
27
R5
1k5
OSC
IC3
AVCC
VCCIO
3V3OUT
USBDM
USBDP
_RSTOUT
_RESET
XTIN
XTOUT
EECS
EESK
EEDATA
TEST
AGND
TXD
RXD
_RTS
_CTS
_DTR
_DSR
_DCD
_RI
TXDEN
_TXLED
_RXLED
PWRCTL
_PWREN
_SLEEP
FT232BM
M8RXD
M8TXD
R2
10k
SV1
+5V
USBVCC
VCC
R8
220
PWR_LED
GND

Necessary parts for any circuit

- ATMega168/328
- The ‘brain’ of the Arduino
- Program is loaded onto the chip
- Runs main loop until power is removed
- That’s it! All other parts are optional!

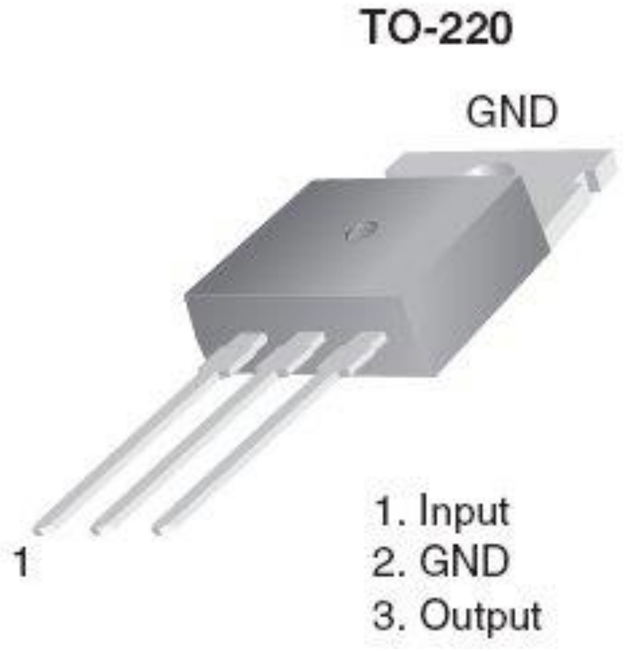
(C) HVWTech, 2008

Optional parts: Timing

- 16Mhz Crystal
- The ‘heartbeat’ of the ATMega chip
- Speed of crystal determines chip speed
- Possible to over/underclock depending on application
- ATMega series has onboard oscillator; less precise

Power Supply

- 5 Volt and 3.3 Volt Regulators
- Filtering capacitors
- Automatic switching between external and USB Power
- Leave it out if you have a filtered 5 Volt power supply

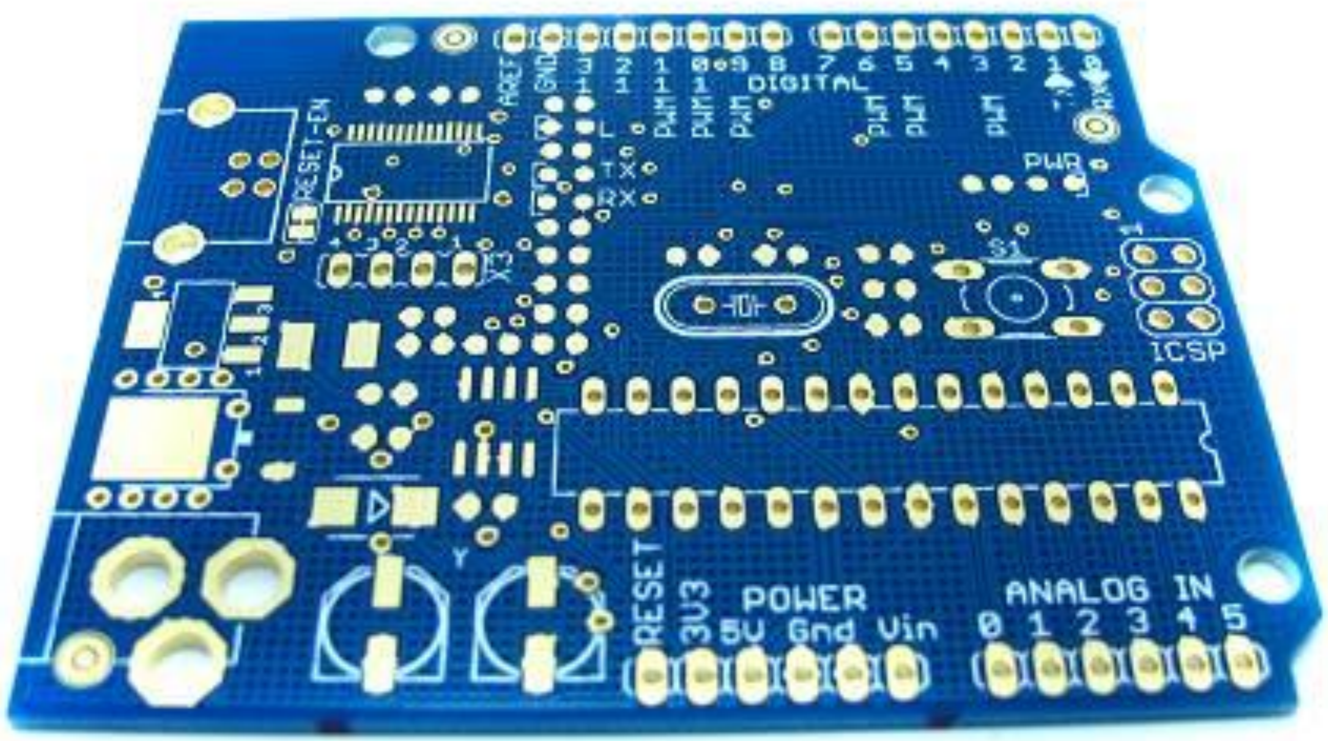


FTDI USB Chip

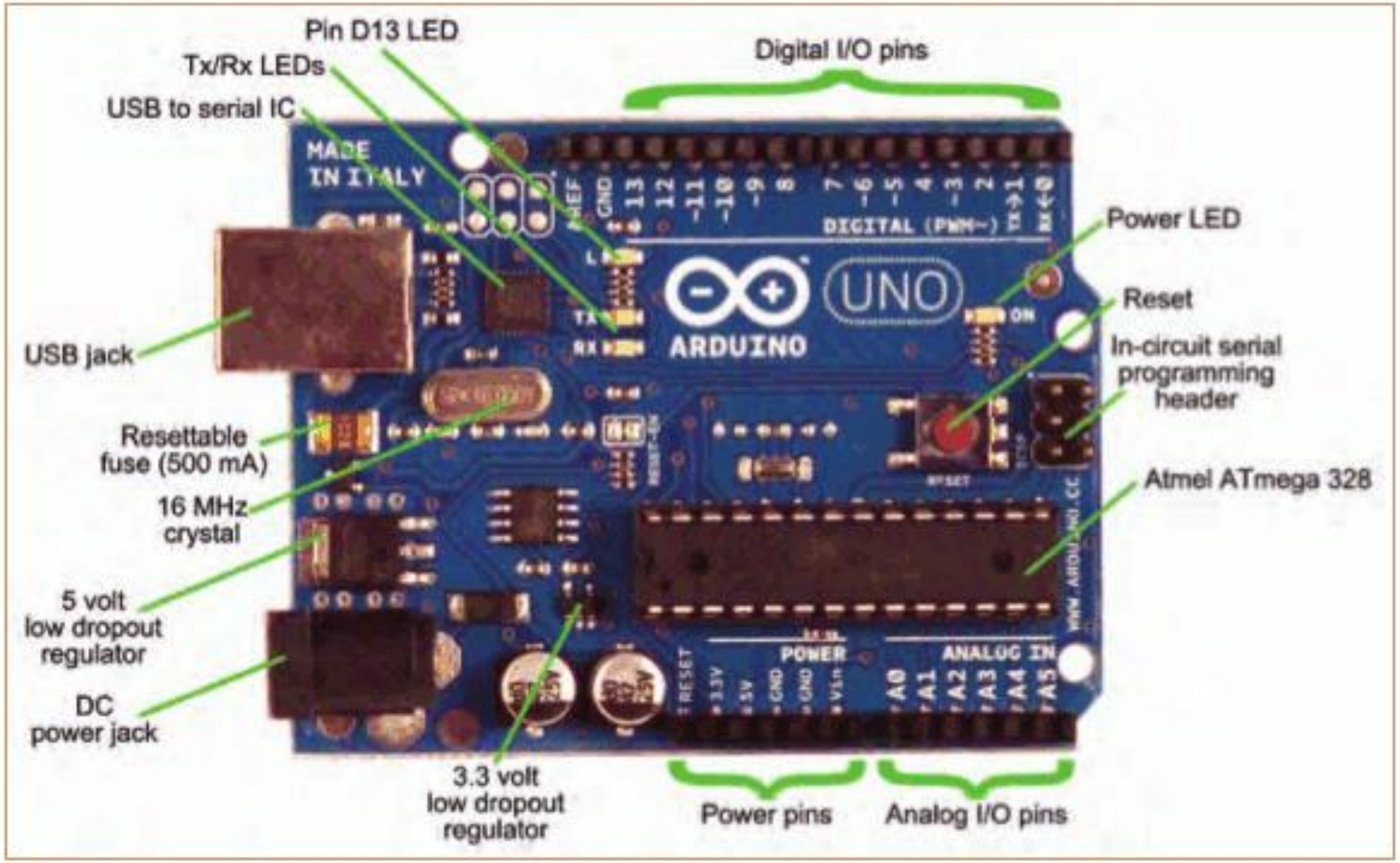
- Allows your Arduino to communicate with your computer over a simple USB link
- Abstraction
- Only necessary for communicating with USB

Hardware

- Circuit Board
- Headers
- USB port
- Sockets
- All optional, use them if you need them

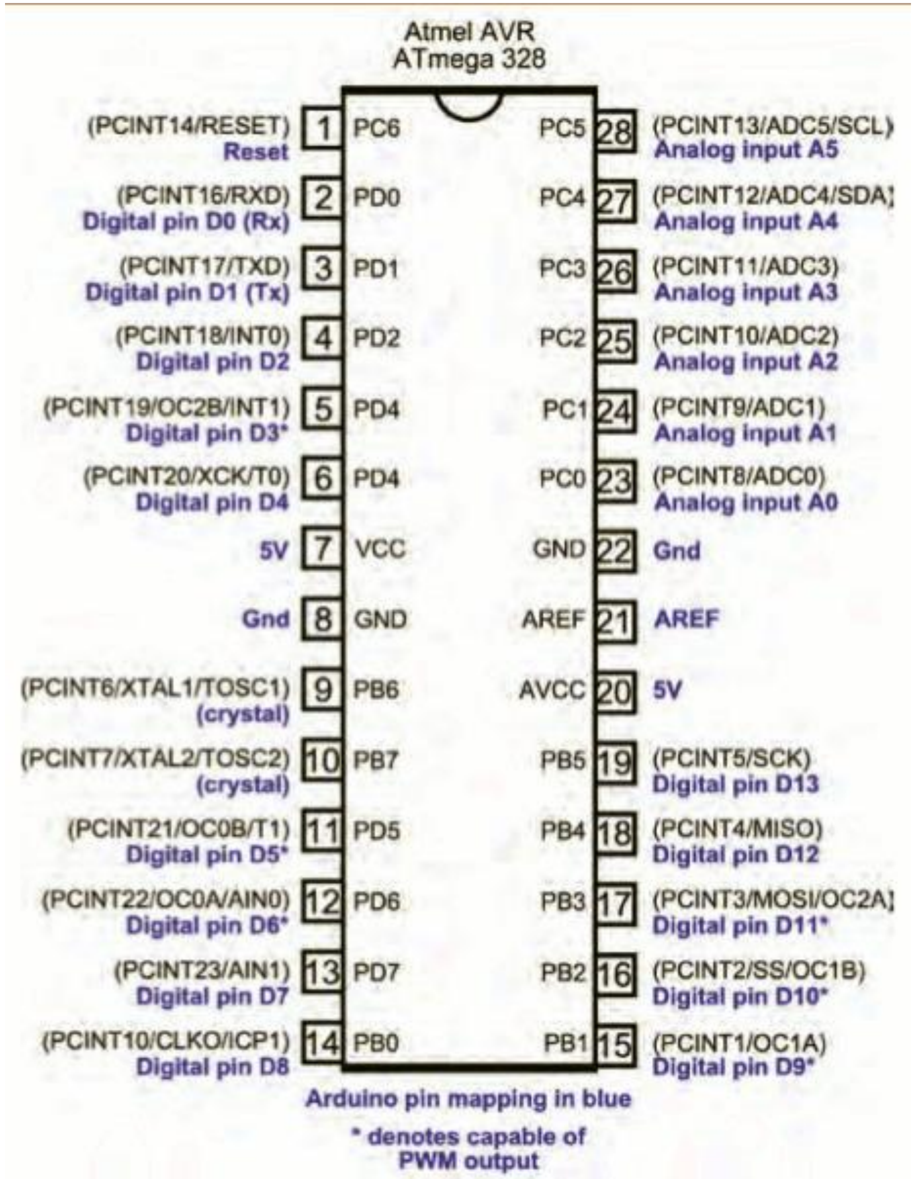


The Arduino Development Board

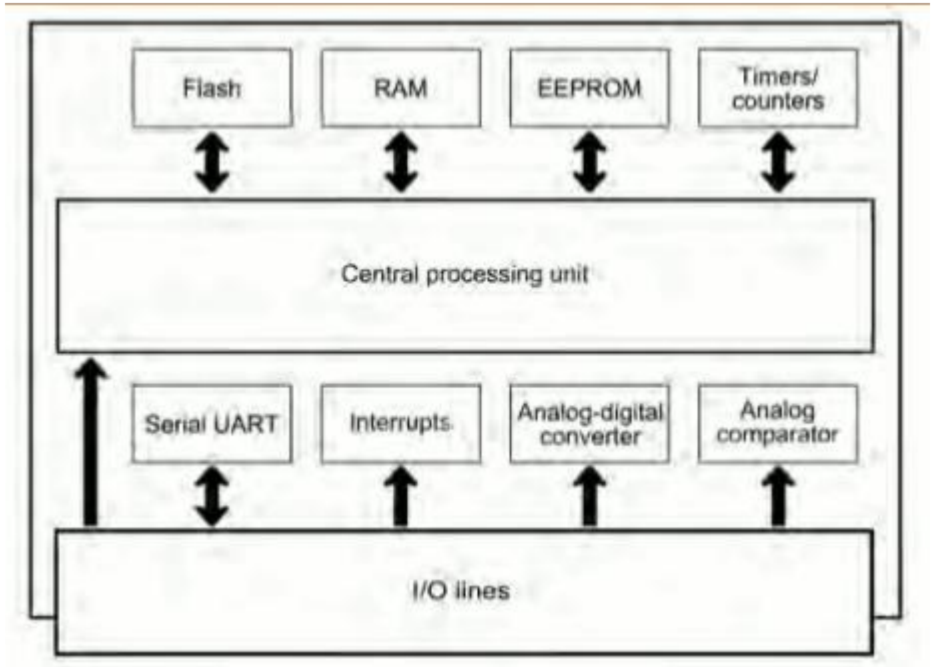


- Input voltage: 7-12 V (USB, DC plug, or Vin)
- Max output current per pin: 40 mA

The Arduino Microcontroller: Atmel ARV Atmega 328



Atmel AVR
ATmega 328
(PCINT14/RESET) Reset 1 PC6
(PCINT16/RXD) Digital pin D0 (Rx) 2 PD0
(PCINT17/TXD) Digital pin D1 (Tx) 3 PD1
(PCINT18/INT0) Digital pin D2 4 PD2
(PCINT19/OC2B/INT1) Digital pin D3* 5 PD4
(PCINT20/XCK/T0) Digital pin D4 6 PD4
5V 7 VCC
Gnd 8 GND
(PCINT6/XTAL1/TOSC1) (crystal) 9 PB6
(PCINT7/XTAL2/TOSC2) (crystal) 10 PB7
(PCINT21/OC0B/T1) Digital pin D5* 11 PD5
(PCINT22/OC0A/AIN0) Digital pin D6* 12 PD6
(PCINT23/AIN1) Digital pin D7 13 PD7
(PCINT10/CLKO/ICP1) Digital pin D8 14 PB0
PC5 28 (PCINT13/ADC5/SCL) Analog input A5
PC4 27 (PCINT12/ADC4/SDA) Analog input A4
PC3 26 (PCINT11/ADC3) Analog input A3
PC2 25 (PCINT10/ADC2) Analog input A2
PC1 24 (PCINT9/ADC1) Analog input A1
PC0 23 (PCINT8/ADC0) Analog input A0
GND 22 Gnd
AREF 21 AREF
AVCC 20 5V
PB5 19 (PCINT5/SCK) Digital pin D13
PB4 18 (PCINT4/MISO) Digital pin D12
PB3 17 (PCINT3/MOSI/OC2A) Digital pin D11*
PB2 16 (PCINT2/SS/OC1B) Digital pin D10*
PB1 15 (PCINT1/OC1A) Digital pin D9*
Arduino pin mapping in blue
* denotes capable of PWM output

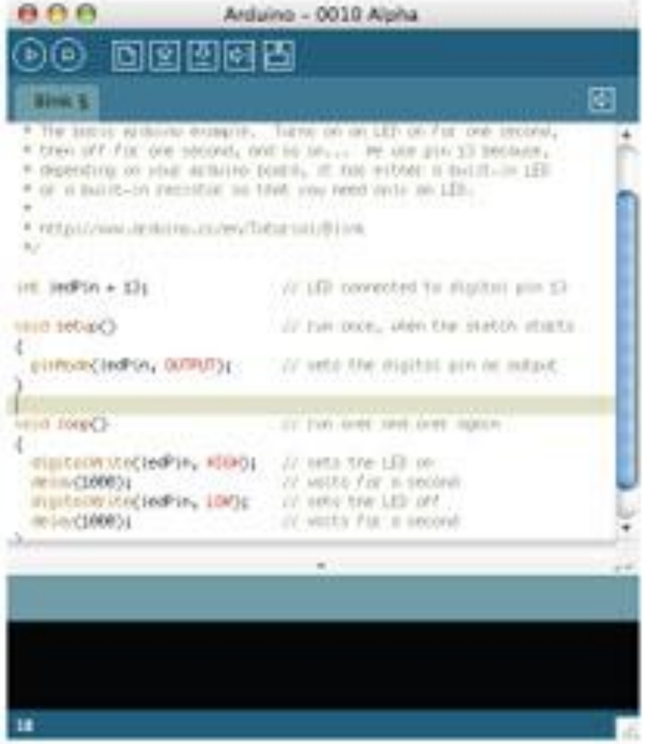


How to work with Arduino?

The word "Arduino" can mean 3 things

A physical piece of hardware

A programming environment



```
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

A community & philosophy

:: About the Arduino Playground ::

Welcome to the Arduino Playground, a wiki where all the users of Arduino can contribute and benefit from their collective research.

This is the place to post and share your own code, circuit diagrams, tutorials, DIY instructions, tips and tricks, and after all the hard work, to show off your projects!

Arduino Playground is a **work in progress**. We can use all the help you can give, so please read the Participate section and get your fingers typing!

:: RoadMap: What Needs to be Done? ::

There is a lot to do...most of the pages are just stubs, simple placeholders waiting for you to fill them up. However there is a small roadmap of things I personally think should be developed first (again, this is a wiki so you are more than welcome to get in

Two methods of using the Arduino

- Working with the Arduino to solve a problem
- Temporary solution
- One time only

①

- Prototyping with the Arduino to solve a need
- Preparing a product for market
- Mass production

②

Phases of Circuit Design

Arduino Approach

- Idea
- Sketch
- Read datasheets
- Gather materials
- Prototype
- Testing
- Optional ---------------->

Stand-alone approach

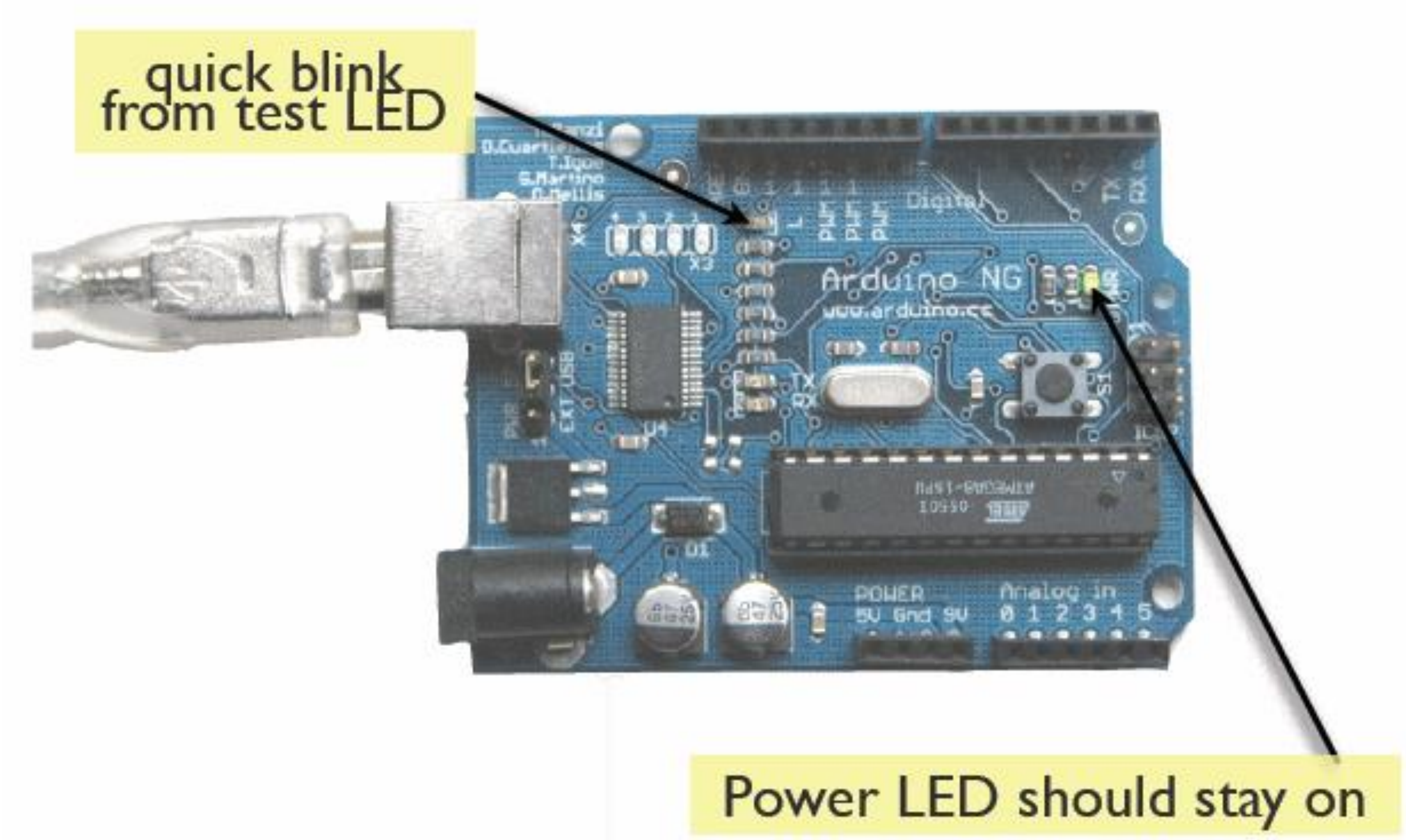
- Idea
- Sketch
- Read datasheets
- Gather materials
- Prototype
- Testing
- Refining
- Testing
- Release to manufacture

Getting Started

- Check out: http://arduino.cc/en/Guide/HomePage

1. **Download & install the Arduino environment (Integrated Development Environment = “IDE”)**
2. **Connect the board to your computer via the USB cable**
3. **If needed, install the drivers (not needed in lab)**
4. **Launch the Arduino IDE**
5. **Select your board**
6. **Select your serial port**
7. **Open the blink example**
8. **Upload the program**

Try It: Connect the USB Cable



quick blink from test LED

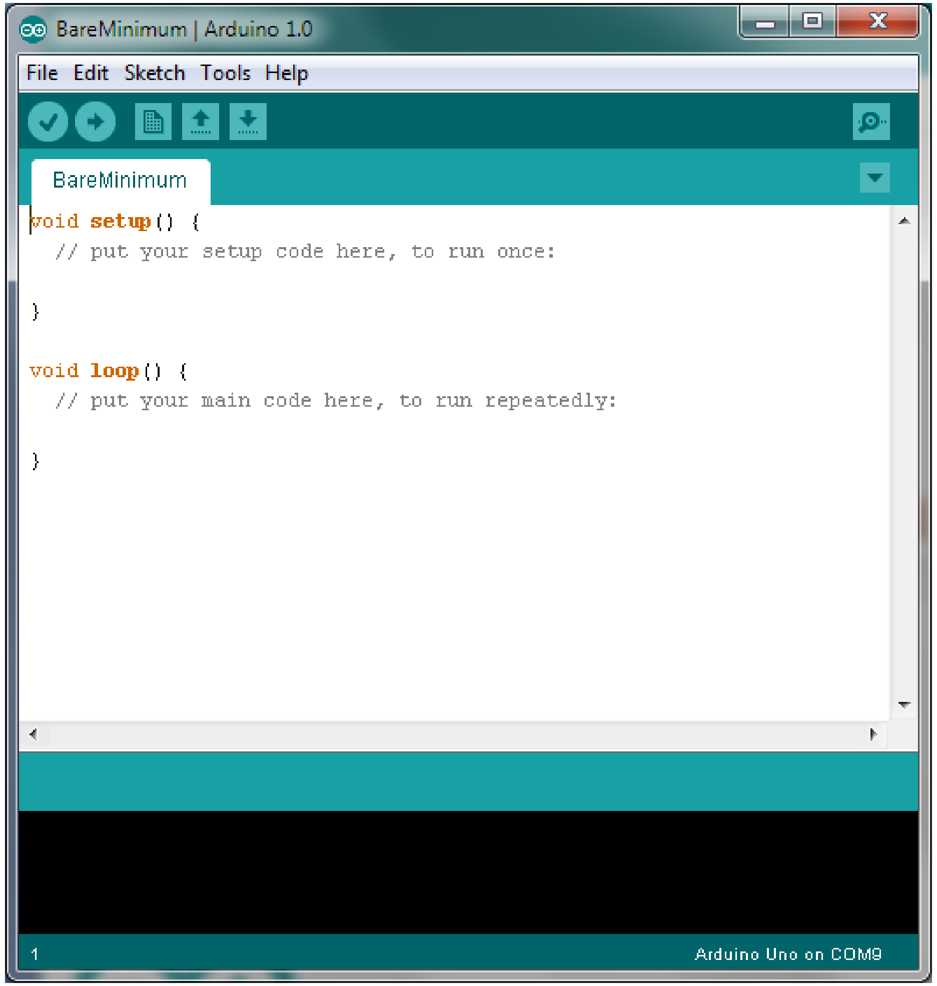
Power LED should stay on

Elements of the Arduino IDE:

- Text editor
  - ➢ syntax and keyword Coloring
  - ➢ Automatic indentation
  - ➢ Programming shortcuts

- Compiler

- Hardware Interface
  - ➢ Uploading programs
  - ➢ Communicating with Arduino via USB



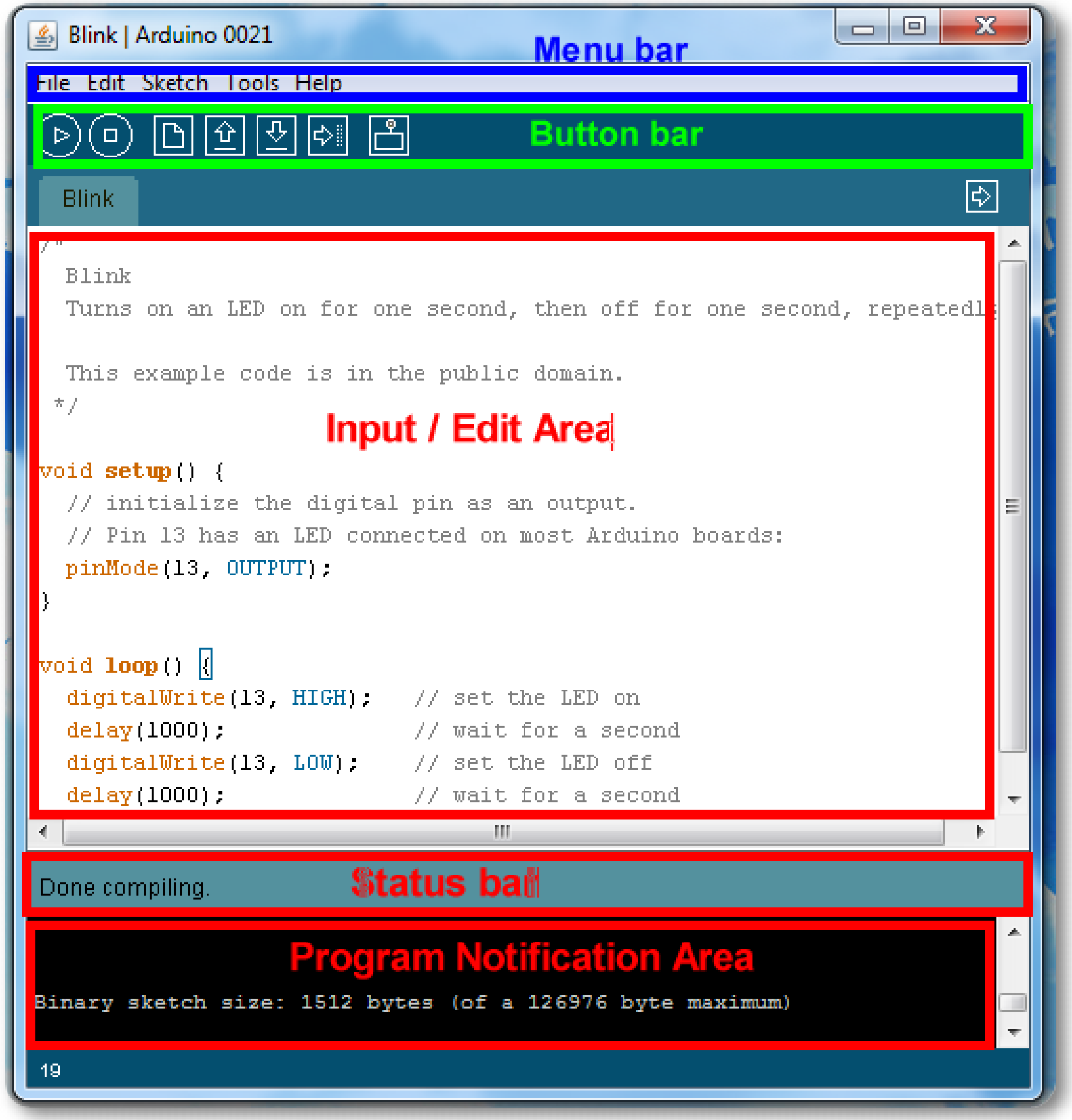
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Elements of the Arduino IDE:

- Text editor
  - syntax and keyword Coloring
  - Automatic indentation
  - Programming shortcuts

- Compiler

- Hardware Interface
  - Uploading programs
  - Communicating with Arduino via USB



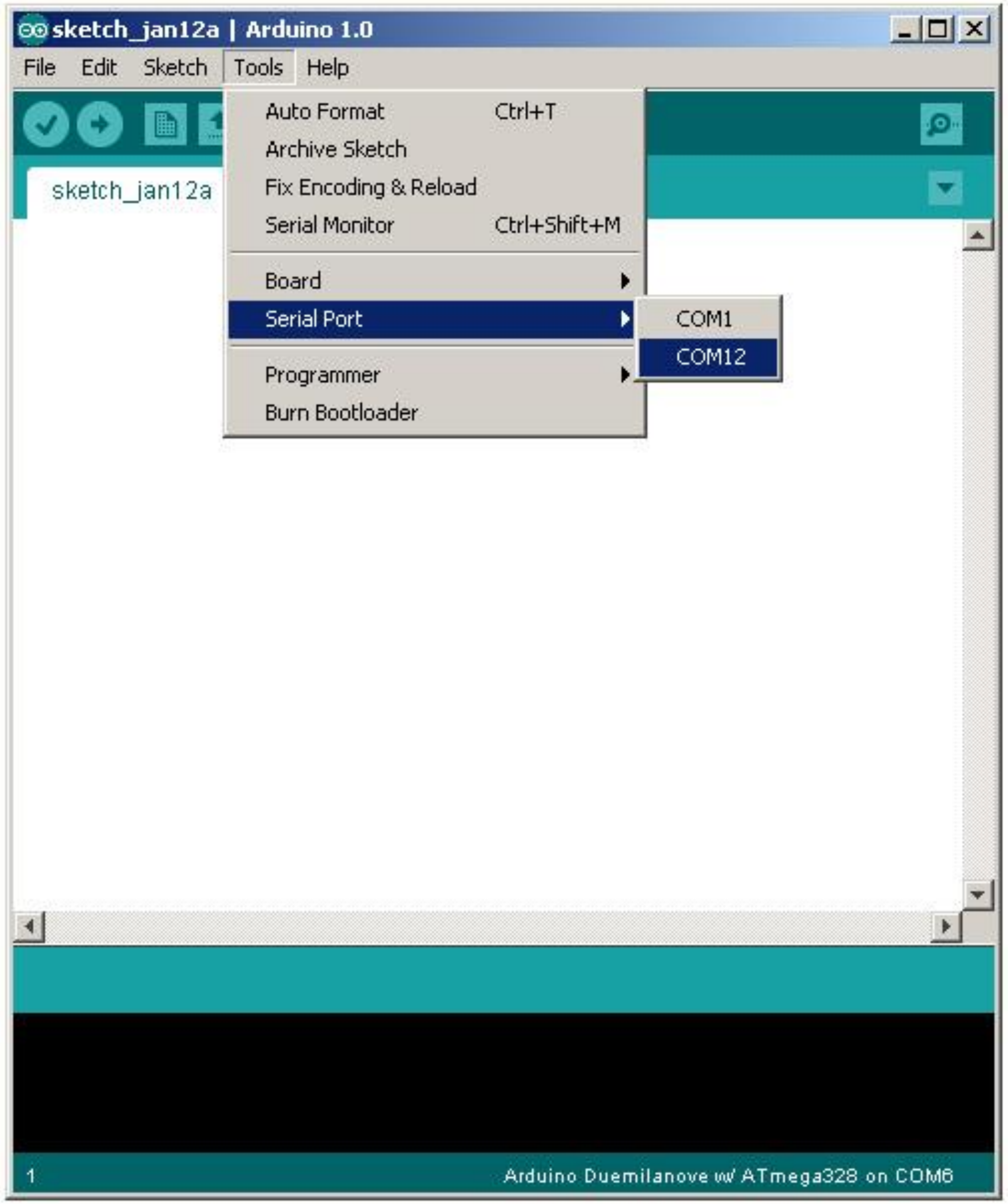
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedl

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
```

```
Done compiling.
```

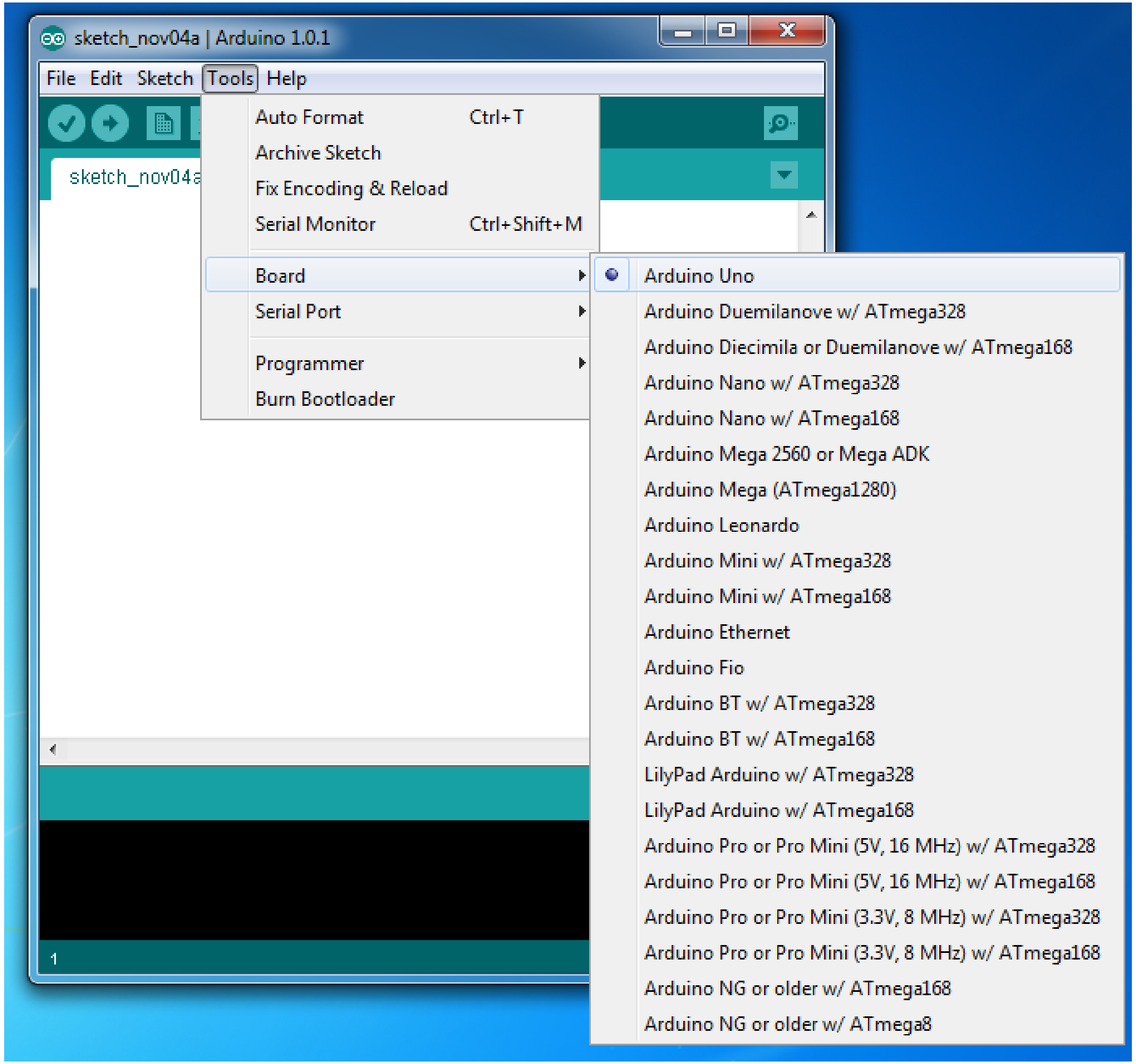
```
Binary sketch size: 1512 bytes (of a 126976 byte maximum)
```

See: http://arduino.cc/en/Guide/Environment for more information

Select Serial Port and Board:

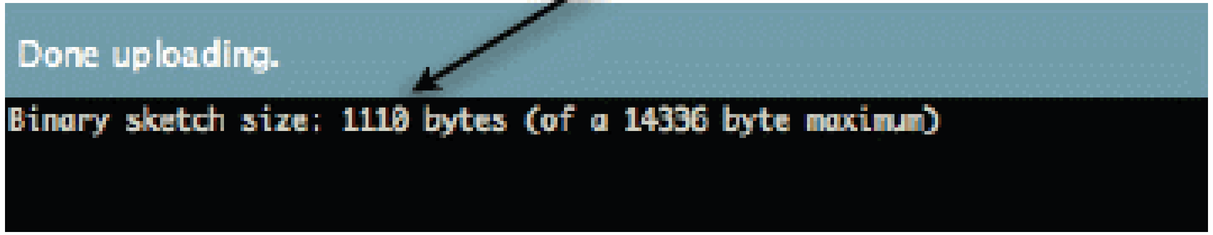


Arduino Duemilanove w/ ATmega328 on COM6


sketch_nov04a | Arduino 1.0.1
File
Edit
Sketch
Tools
Help
sketch_nov04a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
● Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
1

Status Messages:

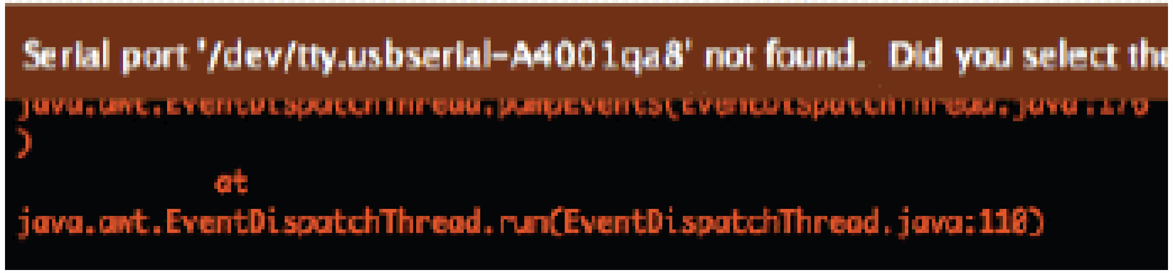
Uploading worked



```
Done uploading.
Binary sketch size: 1110 bytes (of a 14336 byte maximum)
```

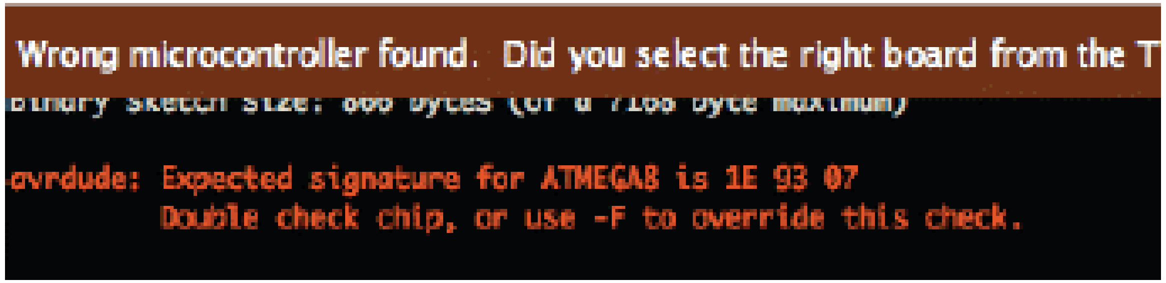
Size depends on
complexity of your sketch

Wrong serial port selected



```
Serial port '/dev/tty.usbserial-A4001qa8' not found.  Did you select the
java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:170
)
            at
java.awt.EventDispatchThread.run(EventDispatchThread.java:110)
```

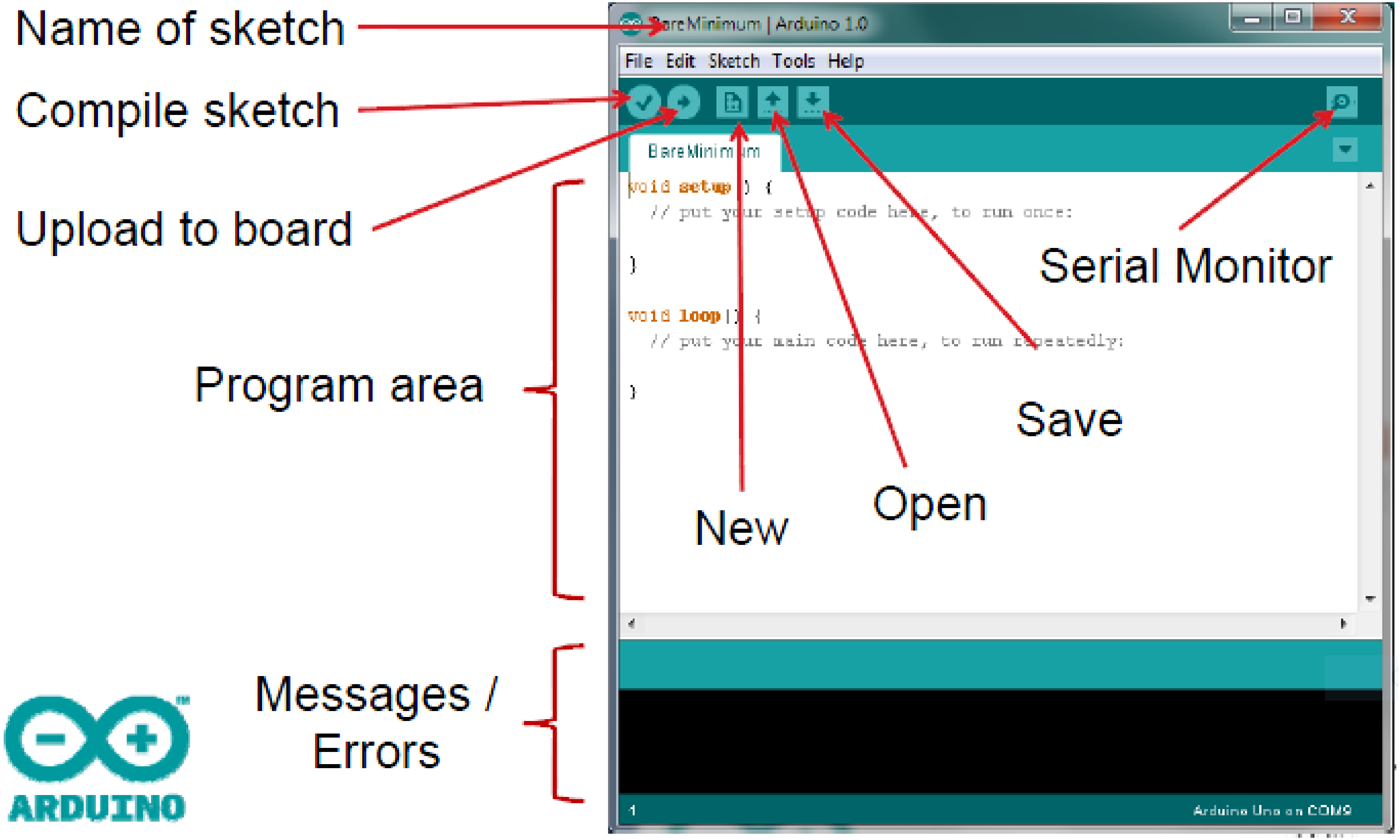
Wrong board selected



```
Wrong microcontroller found.  Did you select the right board from the T
Binary sketch size: 666 bytes (of a 7168 byte maximum)

avrdude: Expected signature for ATMEGA8 is 1E 93 07
         Double check chip, or use -F to override this check.
```

nerdy cryptic error messages

Using the Arduino IDE:



Name of sketch

Compile sketch

Upload to board

Program area

Messages / Errors

New

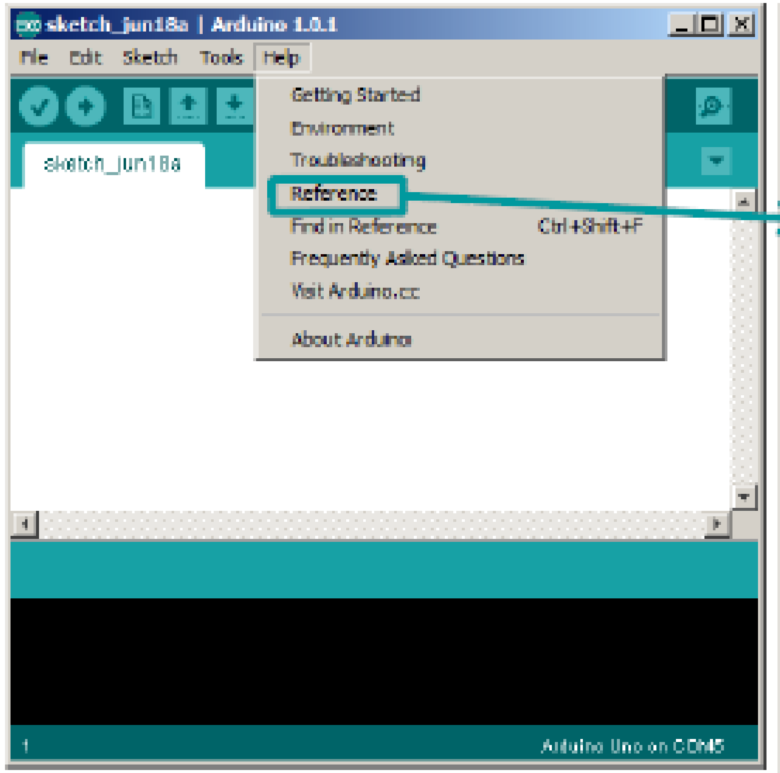
Open

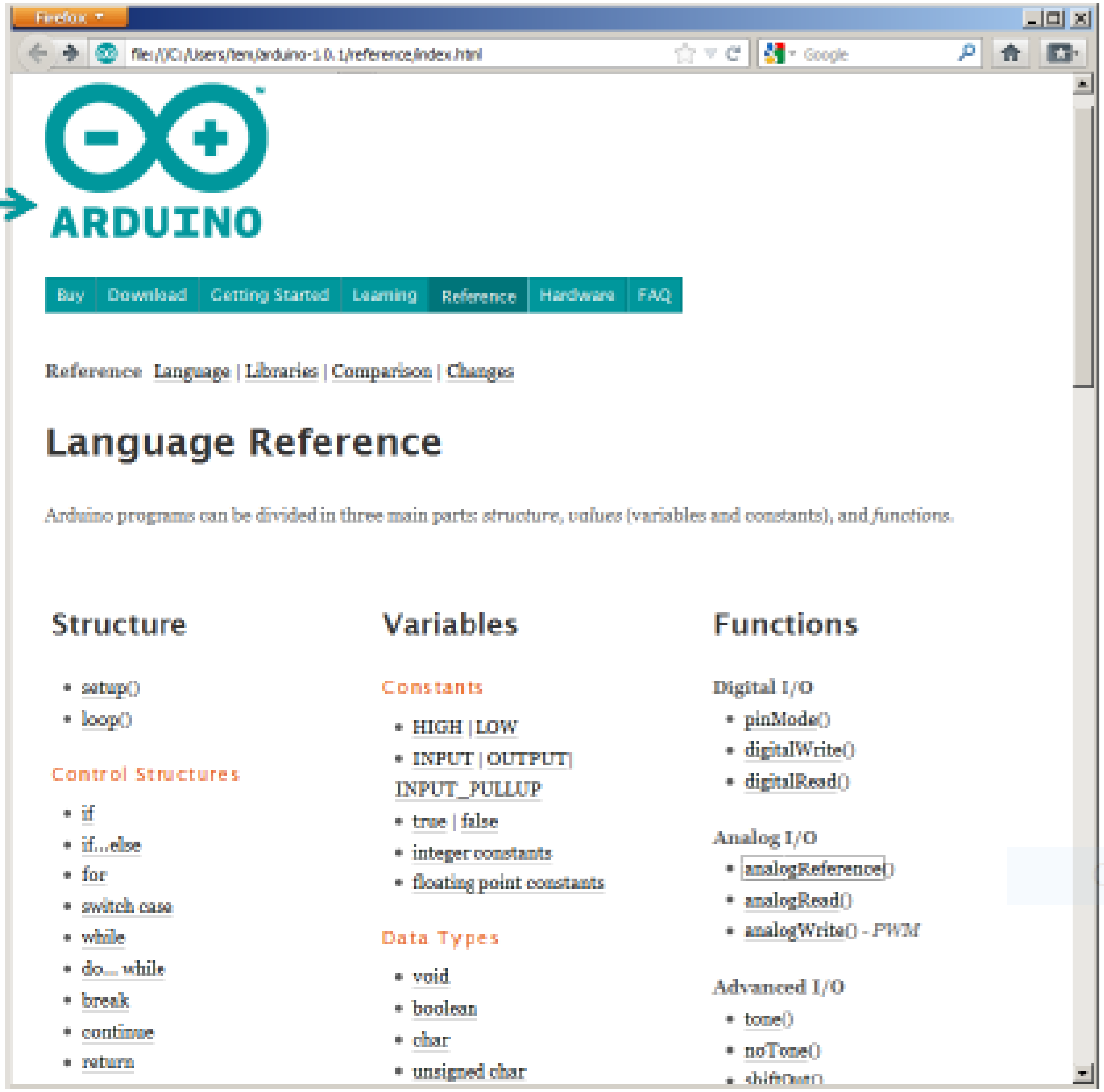
Save

Serial Monitor

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino Reference:





Reference Language | Libraries | Comparison | Changes

Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT| INPUT_PULLUP
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

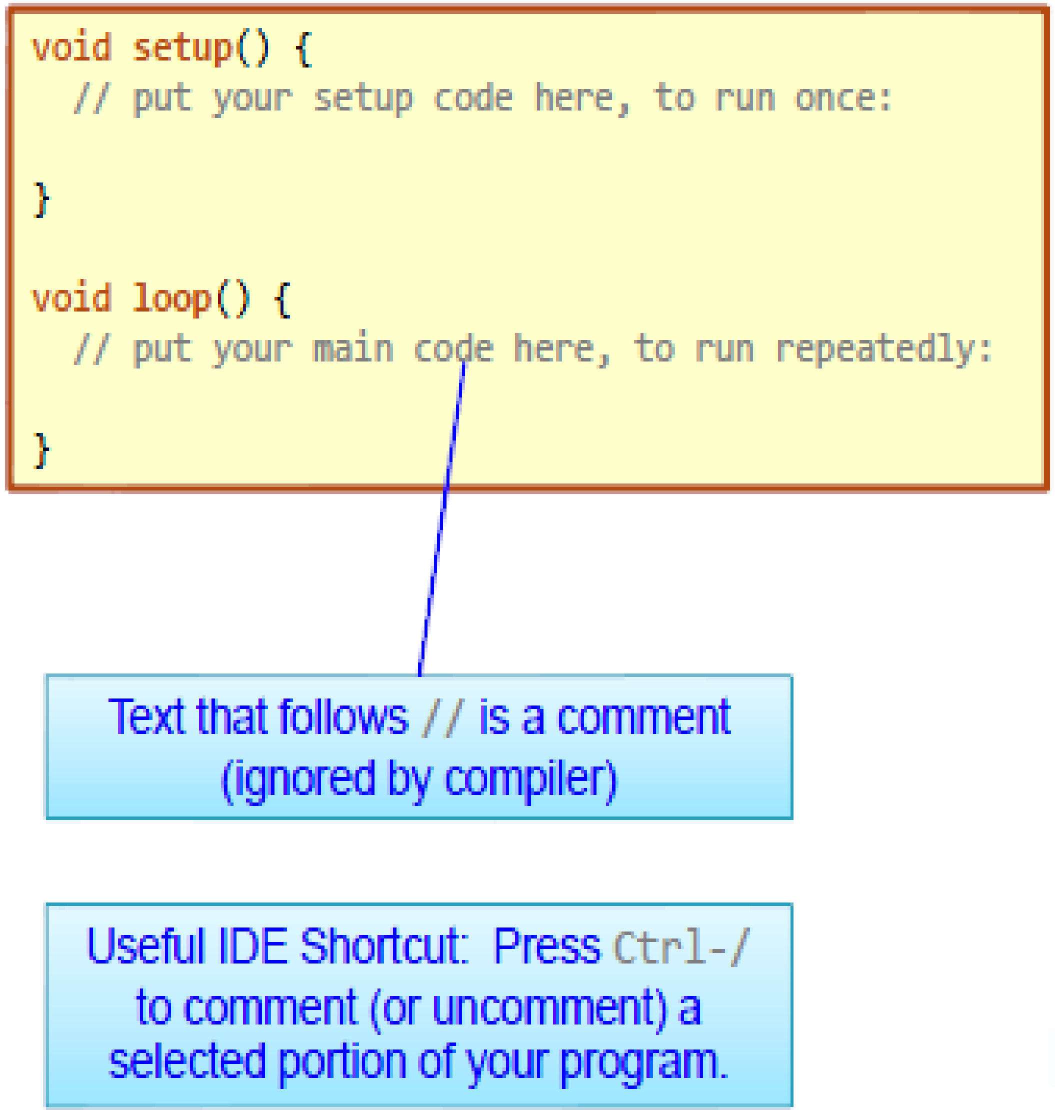
Advanced I/O

- tone()
- noTone()
- shiftOut()

Arduino Reference is installed locally or available online at http://arduino.cc/

Arduino Sketch Structure:

- `void setup()`
  - Will be executed only when the program begins (or reset button is pressed)
- `void loop()`
  - Will be executed repeatedly



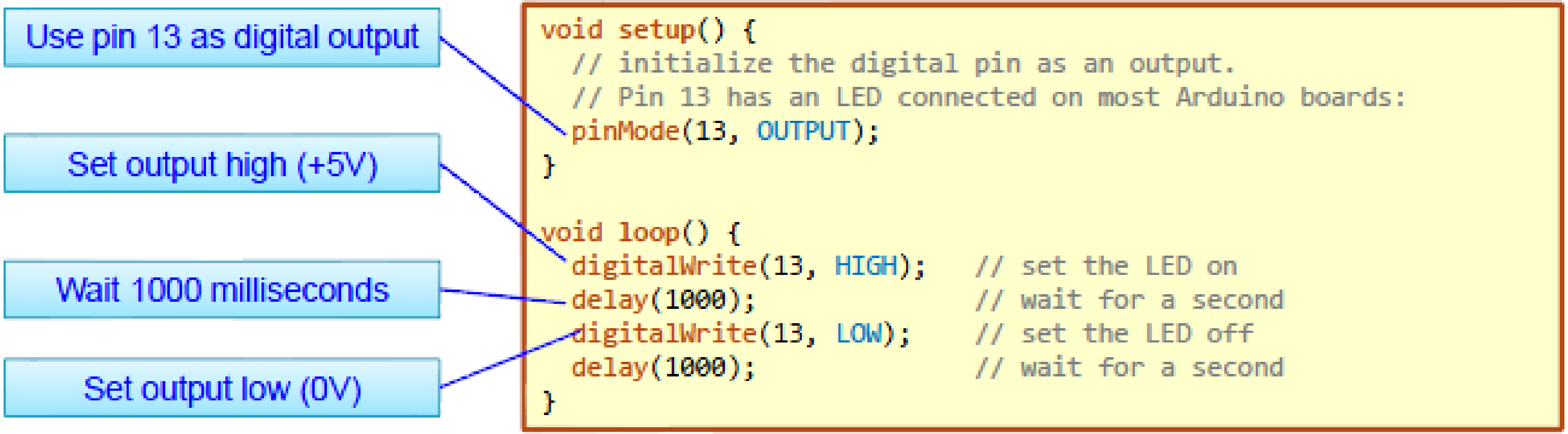
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Text that follows // is a comment (ignored by compiler)

Useful IDE Shortcut: Press Ctrl-/ to comment (or uncomment) a selected portion of your program.

Uploading and Running the Blink Sketch:

- Load the “Blink” example (File→Examples→Basics→Blink)



```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

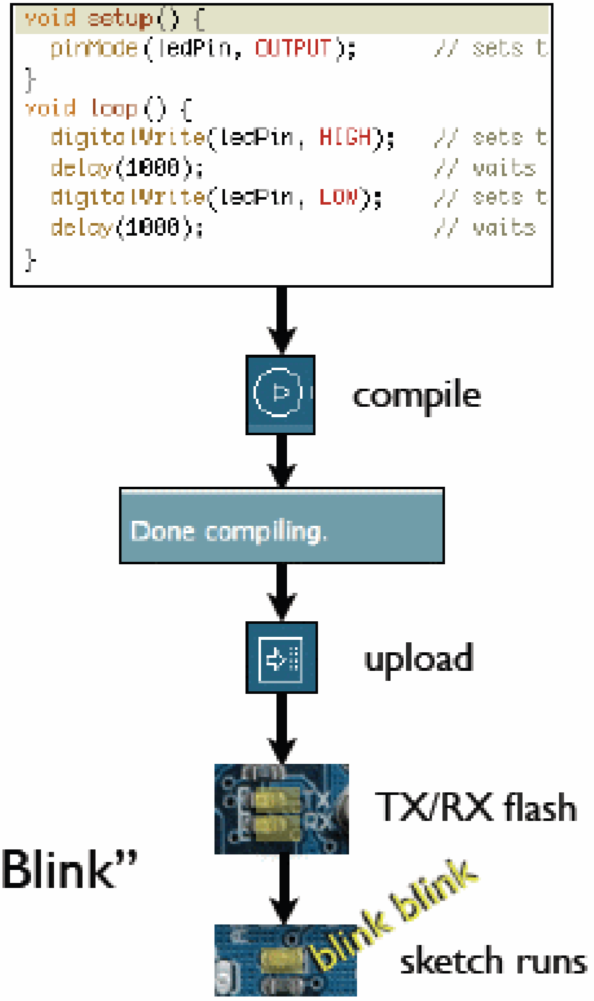
- Compile, then upload the program
- Congratulations! you are now blinkers!

Using Arduino

- Write your sketch
- Press Compile button
  (to check for errors)
- Press Upload button to program
  Arduino board with your sketch

Try it out with the “Blink” sketch!

Load “File/Sketchbook/Examples/Digital/Blink”



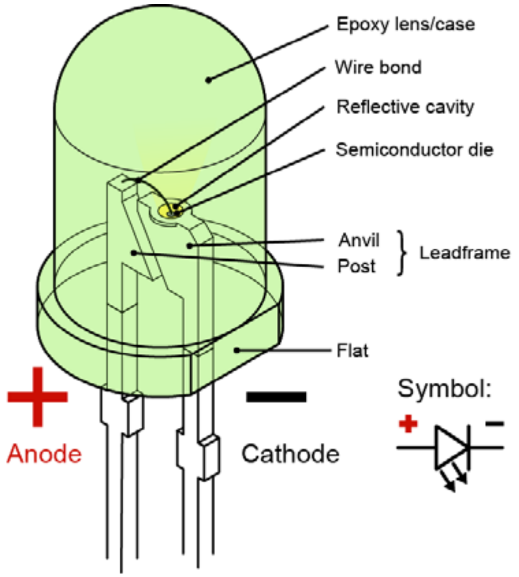
```
void setup() {
  pinMode(ledPin, OUTPUT);       // sets t
}
void loop() {
  digitalWrite(ledPin, HIGH);    // sets t
  delay(1000);                   // waits
  digitalWrite(ledPin, LOW);     // sets t
  delay(1000);                   // waits
}
```

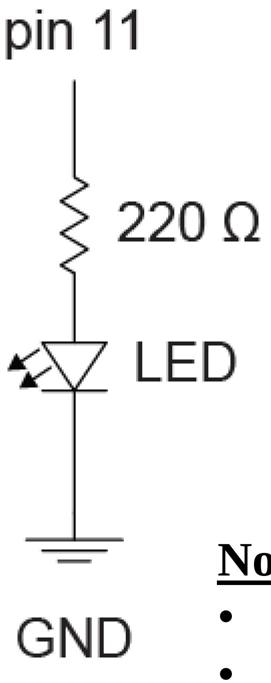
Done compiling.

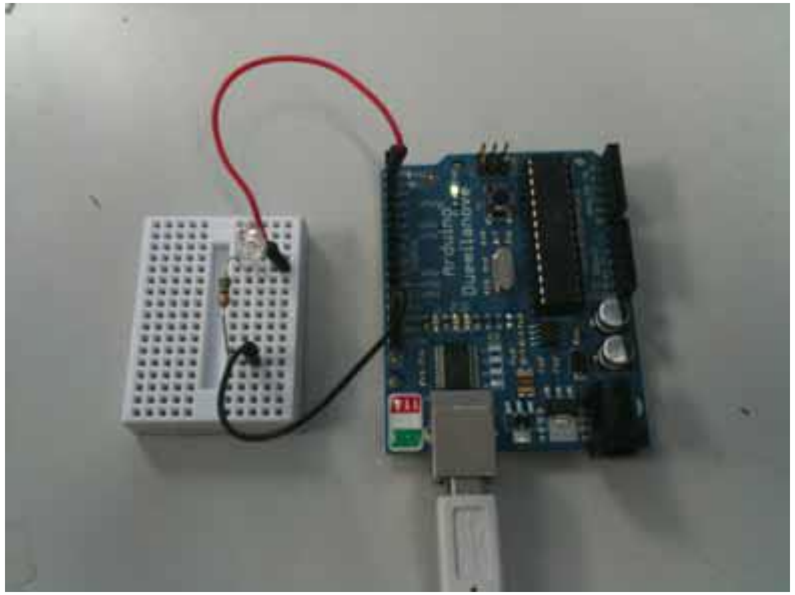
www.todbot.com/blog/bionicarduino

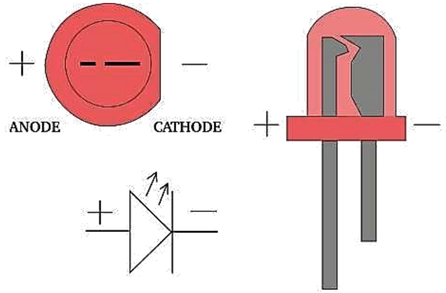
Creating and Saving a Sketch:

- Now connect your own LED







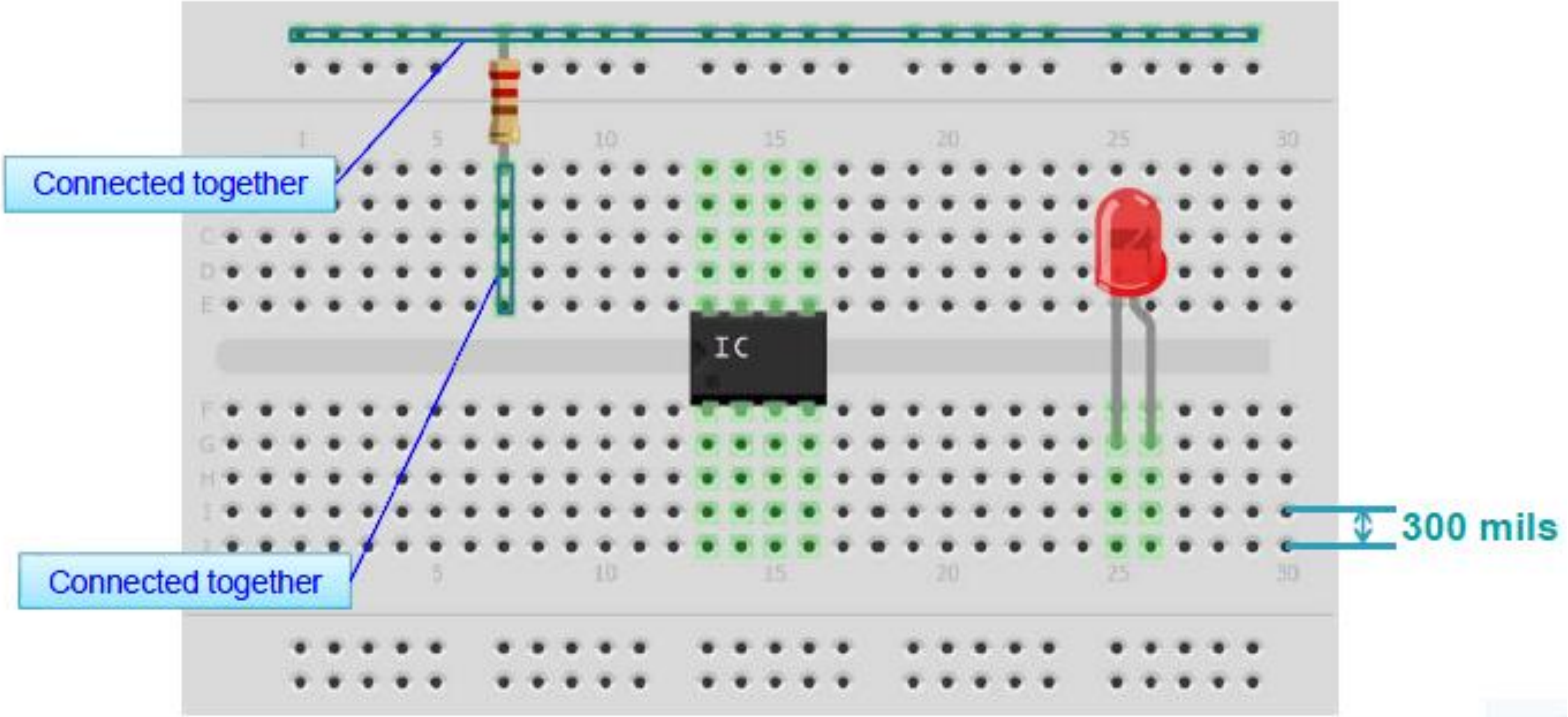


Notes:

- Resistor is needed to limit current
- Resistor and LED may be interchanged (but polarity of LED is important)
- Pin 13 is special: has built-in resistor and LED
- Change program and upload

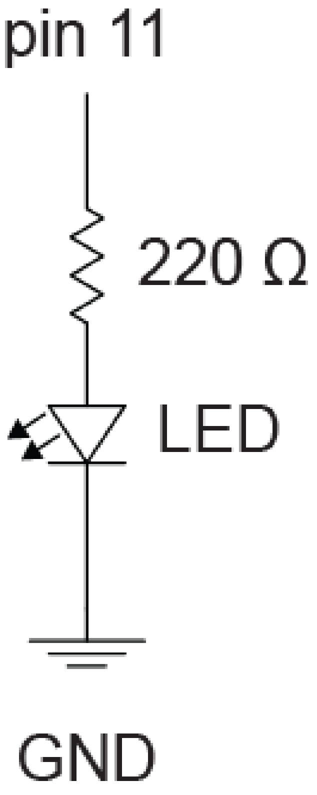
Creating and Saving a Sketch:

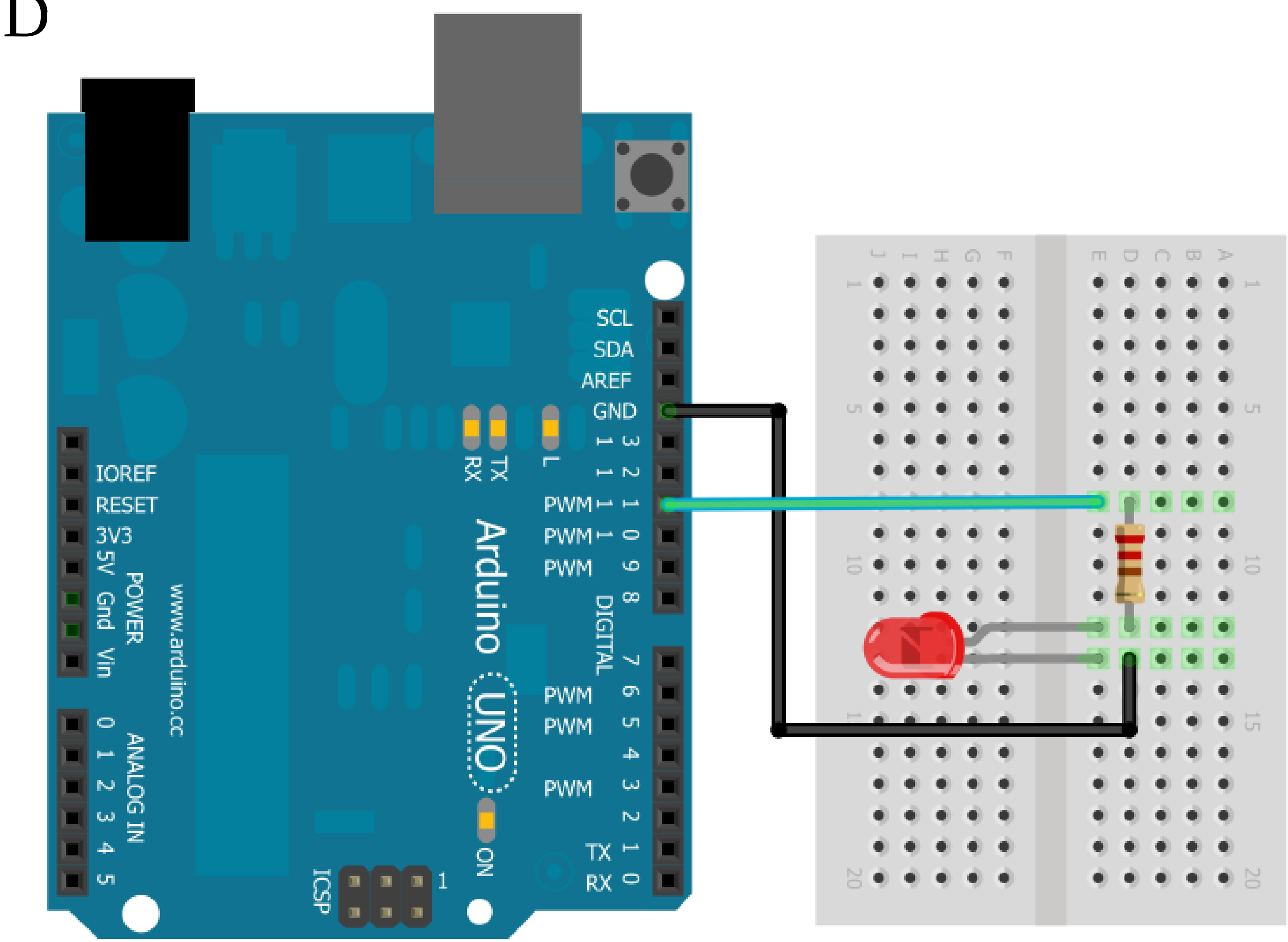
- Using the Breadboard



Creating and Saving a Sketch:

- Now connect your own LED





Assignment # 1:

- Change the blink rate
  - ➢ how fast can the LED blink (before you can no longer perceive the blinking?)

- How would you make the LED dimmer?
  - ➢ (...without changing the resistor?)