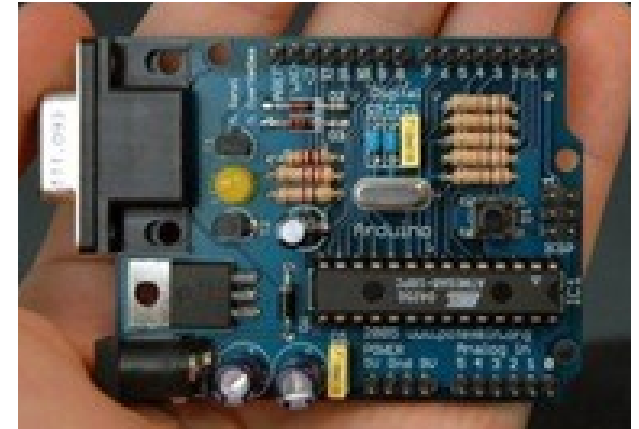


# Raspberry Pi-Teensy - **Teensy**

- 1) History/Basics
- 2) Why Arduino/Teensy?
- 3) Setting up the Arduino IDE
- 4) Sensors and Actuators
- 5) Communicating with the Raspberry Pi
  - a) Direct USB – Firmata
  - b) Direct Serial – roll your own
  - c) Direct I/O - primitive

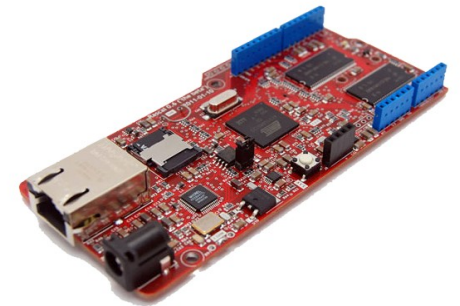
# History/Basics

- 1) Started in 2005 in Italy
- 2) **(Relatively) Simple, integrated development environment (IDE)**
- 3) Multiplatform (PC, Mac, Linux)
- 4) Open Source
- 5) [arduino.cc](http://arduino.cc)



# Why Arduino?

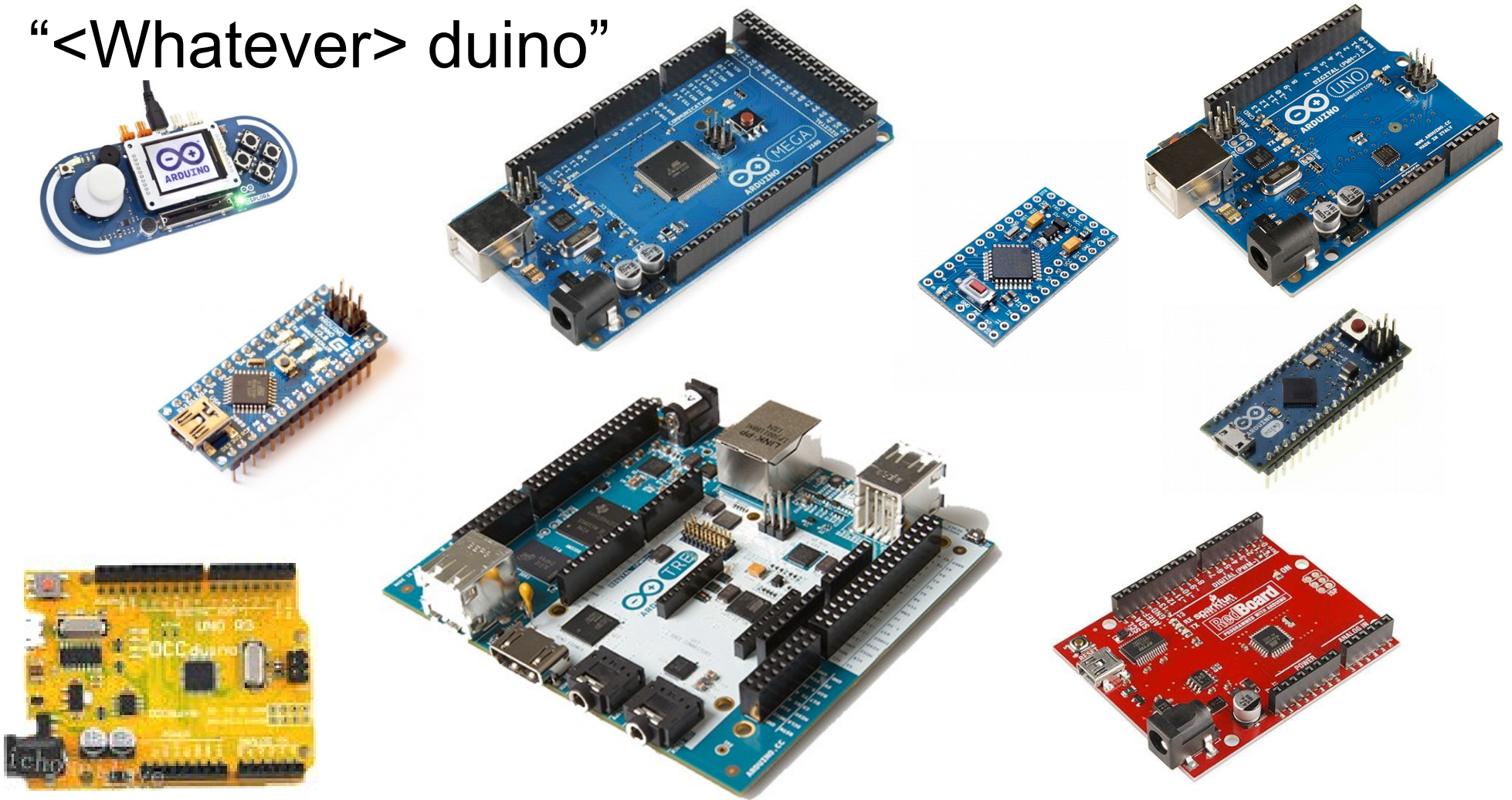
- 1) Cheap, expandable, **No Operating System**
- 2) Teach the basics of hardware/software, **discipline of small memory**; programmed in C/C++
- 3) Open Source – **huge base of users**
- 4) Gateway to Beaglebone, Raspberry Pi, RascalMicro
- 5) Build commercial products with ATMEL processors (really cheaply)



# 32 Flavors (and more) -

<http://arduino.cc/en/Main/Products>

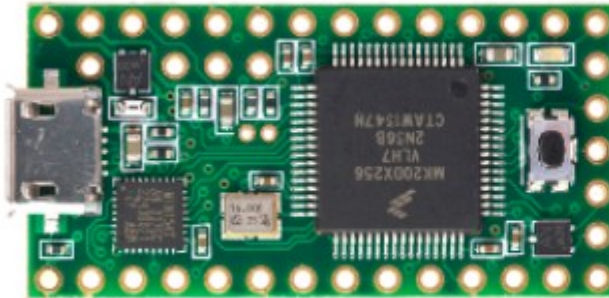
“<Whatever> duino”



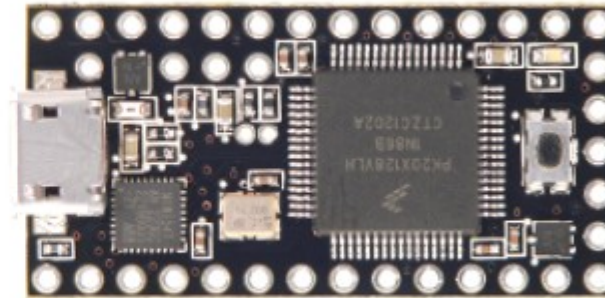


# An Alternative – The Teensy

**Teensy 3.1**



**Teensy 3.0**



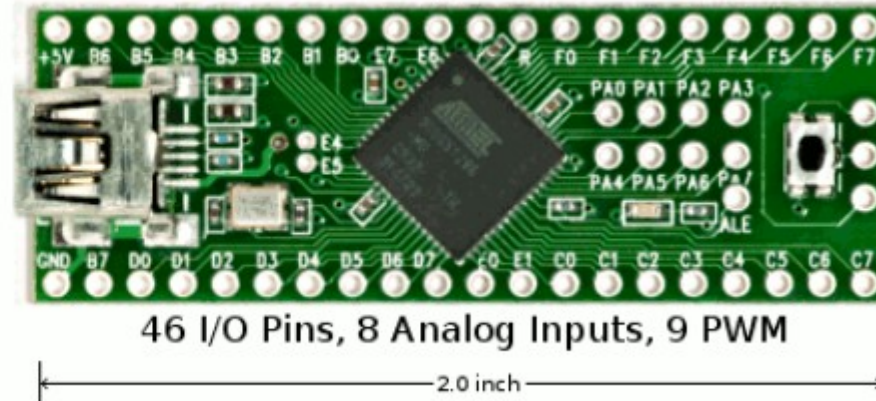
[Teensy 3.1 changes from Teensy 3.0](#)

**Teensy 2.0**



25 I/O, 12 Analog, 7 PWM

**Teensy++ 2.0**

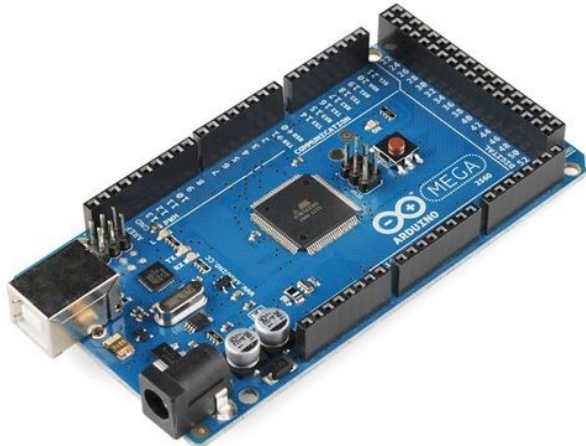


46 I/O Pins, 8 Analog Inputs, 9 PWM

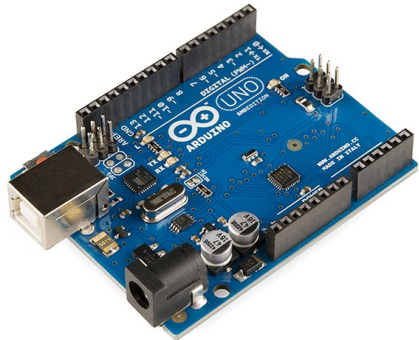
# Why Teensy?

- 1) Better, faster, cheaper – **It is NOT an Arduino (but can look and act like one – the important part)**
- 2) <https://www.pjrc.com/teensy/>
- 3) Cool features
  - a) More memory!
  - b) True analog output (12 bit - 4096)!
  - c) 5V tolerant (except analog input; only 3.3V)
  - d) Dual A/D!
  - e) Many I2C/SPI ports
- 4) Real-Time clock, keyboard emulation, capacitive touch
- 5) Note – Export restrictions (it's that good)

# A comparison...



Arduino  
MEGA  
\$50



Arduino  
UNO  
\$28



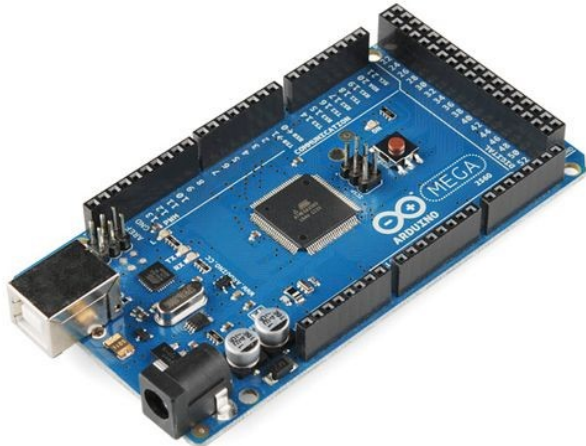
Beware  
cheap  
versions!



ATTiny85  
\$1.45



# The comparison extended...



Arduino  
MEGA  
\$50



Arduino  
UNO  
\$28



Teensy  
3.1  
\$19.80





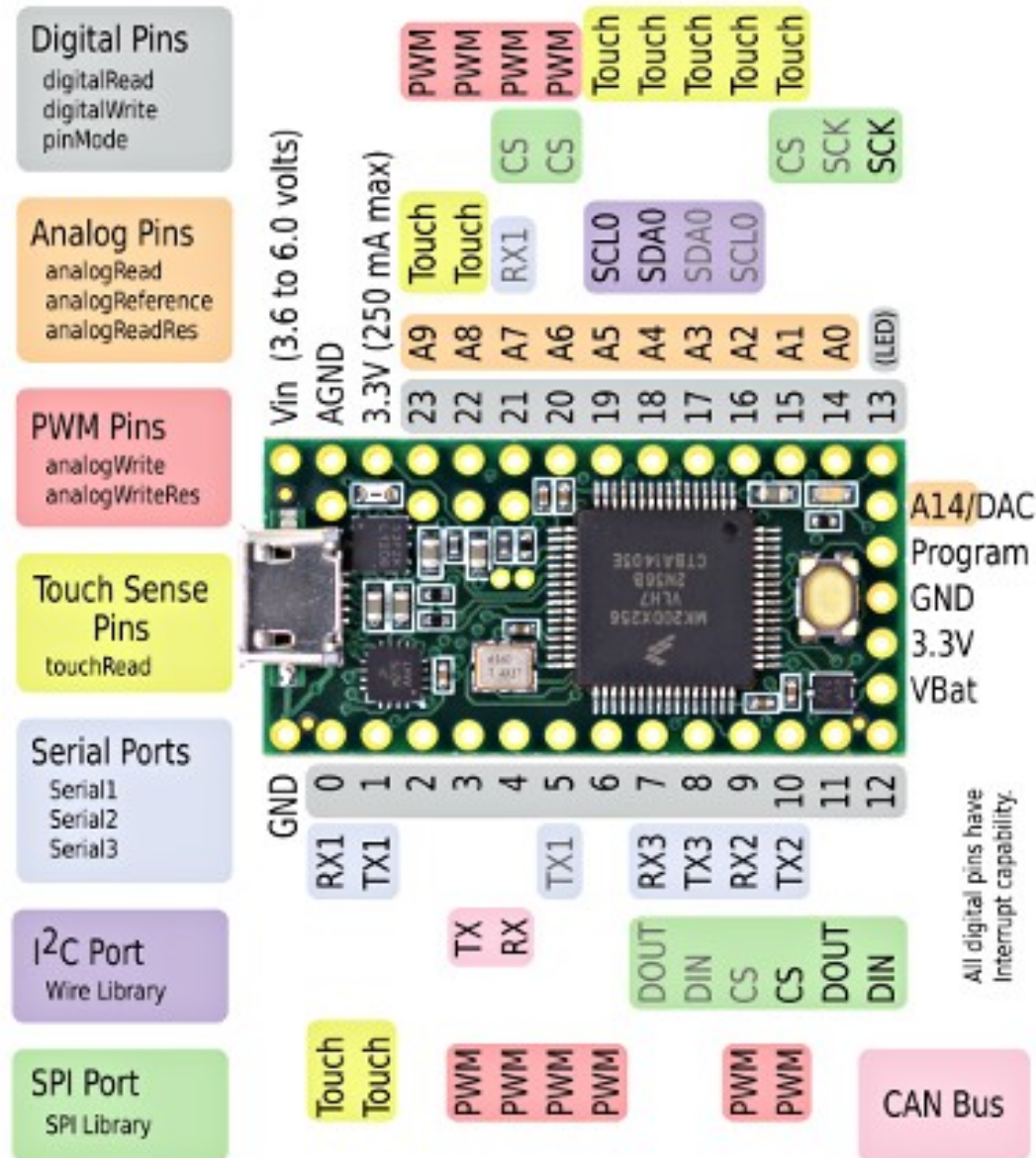
# The Teensy – Memory, Speed, I/O Details

## Key Features:

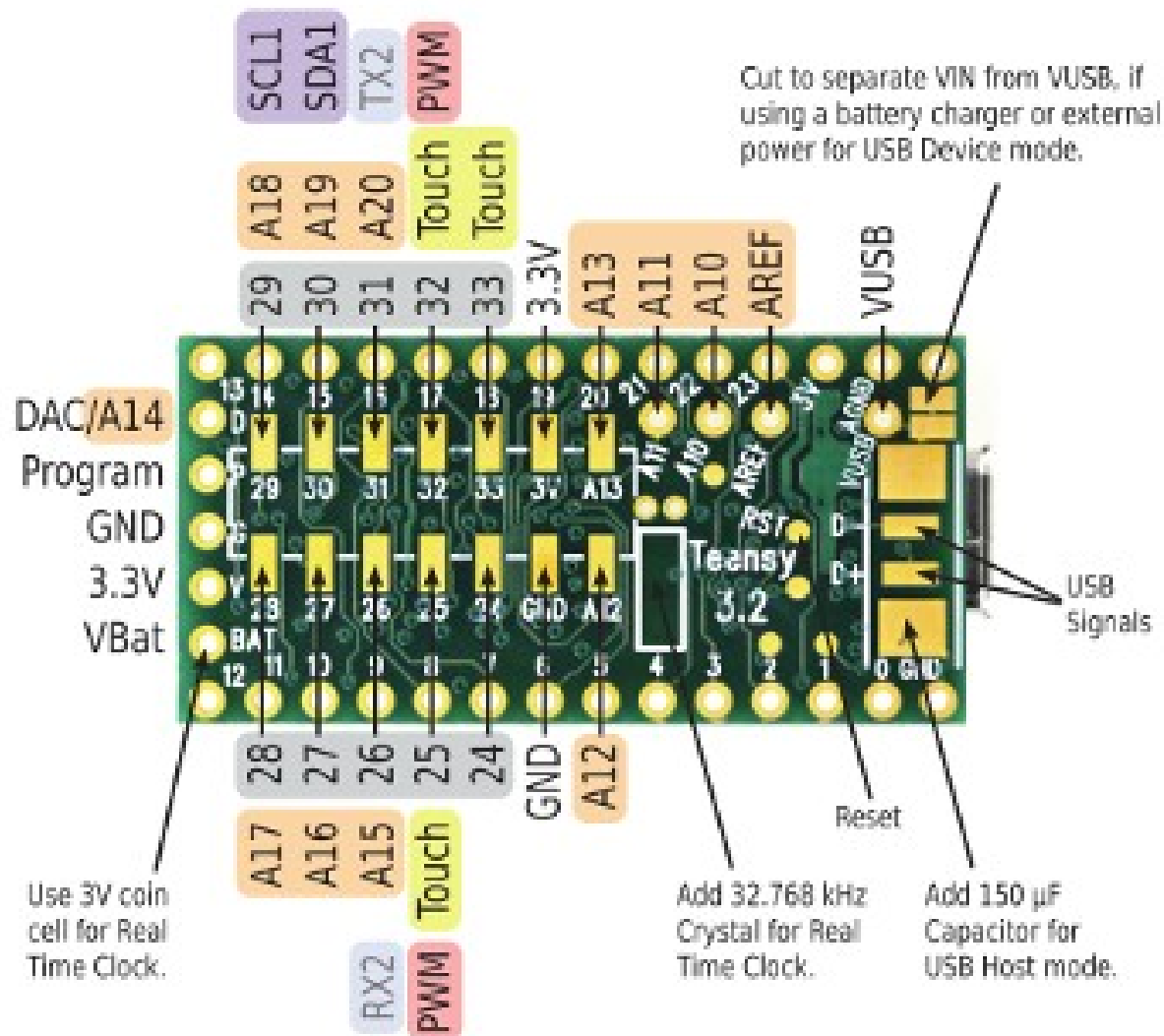
- USB can be any type of device
- AVR processor, 16 MHz
- Single pushbutton programming
- Easy to use Teensy Loader application
- Free software development tools
- Works with Mac OS X, Linux & Windows
- Tiny size, perfect for many projects
- Available with pins for solderless breadboard
- Very low cost & low cost shipping options

Specification	Teensy 2.0	Teensy++ 2.0	Teensy 3.0	Teensy 3.1
Processor	ATMEGA32U4 8 bit AVR 16 MHz	AT90USB1286 8 bit AVR 16 MHz	MK20DX128 32 bit ARM Cortex-M4 48 MHz	MK20DX256 32 bit ARM Cortex-M4 72 MHz
Flash Memory	32256	130048	131072	262144
RAM Memory	2560	8192	16384	65536
EEPROM	1024	4096	2048	2048
I/O	25, 5 Volt	46, 5 Volt	34, 3.3 Volt	34, 3.3V, 5V tol
Analog In	12	8	14	21
PWM	7	9	10	12
UART,I2C,SPI	1,1,1	1,1,1	3,1,1	3,2,1
Price	<a href="#">\$16.00</a>	<a href="#">\$24.00</a>	<a href="#">\$19.00</a>	<a href="#">\$19.80</a>

# The Teensy (3.1/3.2)

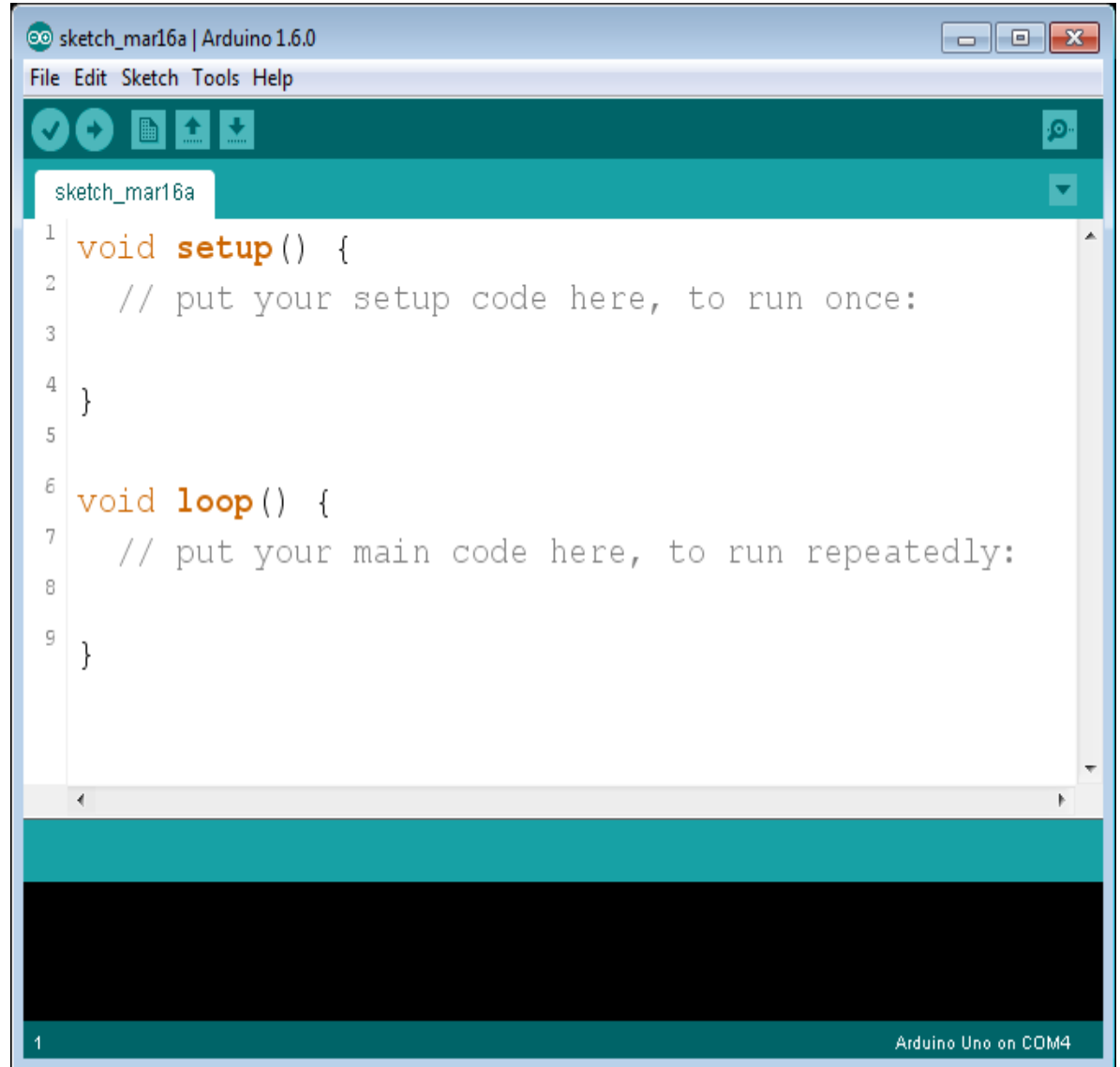


# The Teensy (3.1/3.2)



# The Standard Arduino IDE

- 1) Sketch area
- 2) Message area
- 3) Compile
- 4) Download
- 5) Serial Monitor
- 6) Sketch -  
examples
- 7) **Tools –**  
**configure which**  
**type, port**
- 8) Type, port in lower  
right hand corner
- 9) void setup() and  
void loop()





# The Arduino IDE + Teensy Add-On

<https://www.pjrc.com/teensy/teensyduino.html>

- 1) Teensyduino – add-on for Arduino IDE
- 2) Teensy Loader

A screenshot of the Arduino IDE interface. The title bar says 'Blink | Arduino 1.6.5'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The main text area shows the 'Blink' sketch code. The status bar at the bottom indicates 'Teensy 3.2 / 3.1, Serial, 96 MHz optimized (overclock), US English on COM100'.

```
1 /* LED Blink, Teensyduino Tutorial #1
2  http://www.pjrc.com/teensy/tutorial.html
3
4  This example code is in the public domain.
5  */
6
7  // Teensy 2.0 has the LED on pin 11
8  // Teensy++ 2.0 has the LED on pin 6
9  // Teensy 3.0 has the LED on pin 13
10 const int ledPin = 13;
11
12 // the setup() method runs once, when the sketch starts
13
14 void setup() {
15   // initialize the digital pin as an output.
16   pinMode(ledPin, OUTPUT);
17 }
18
19 // the loop() method runs over and over again,
20 // as long as the board has power
21
22 void loop() {
23   digitalWrite(ledPin, HIGH); // set the LED on
24   delay(1000);                // wait for a second
25   digitalWrite(ledPin, LOW);  // set the LED off
26   delay(1000);                // wait for a second
27 }
28
```

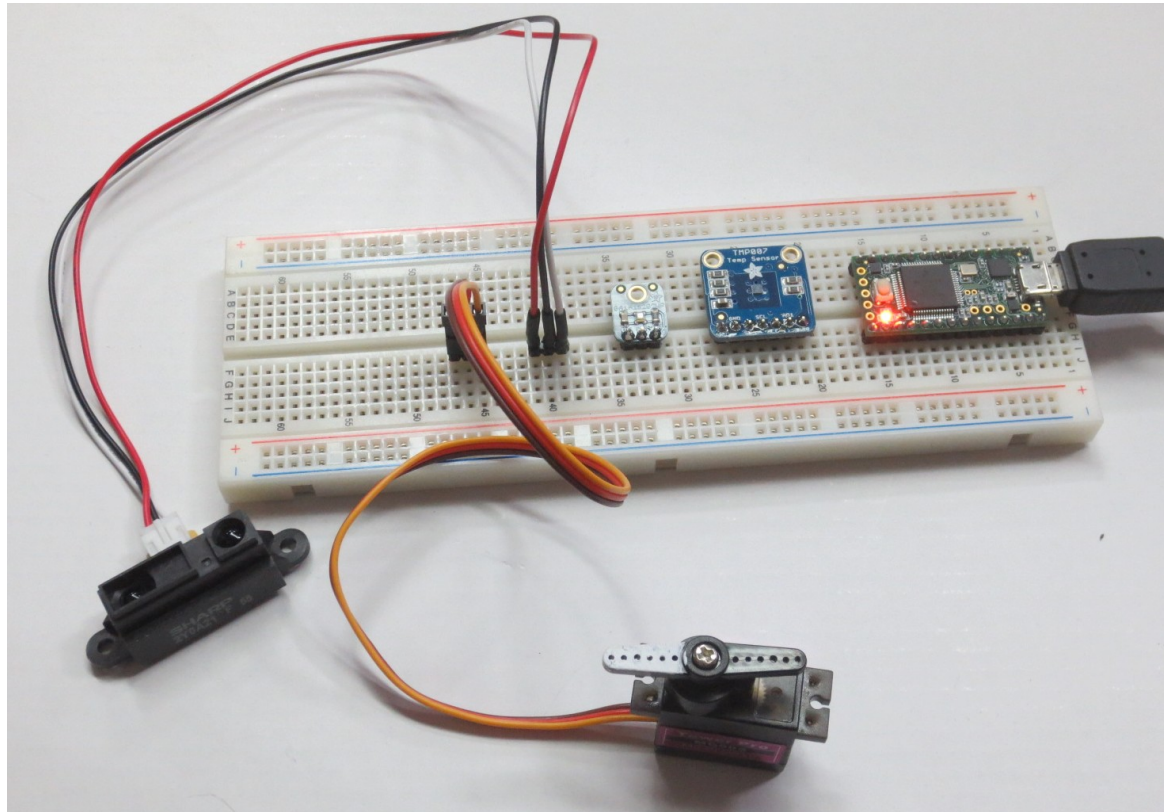
Done uploading.

Global variables use 3,508 bytes (5%) of dynamic memory, leaving 62,028 bytes for local variables. Maximum is 65,536 bytes.

10 Teensy 3.2 / 3.1, Serial, 96 MHz optimized (overclock), US English on COM100

# Teensy IDE – Sensors/Actuators

- 1) **Distance** (SHARP GP2Y0A21YK0F)
- 2) **Light sensor** – log scale (GA1A1S202WP)
- 3) **Temperature** – TMP007 (using I2C)
- 4) **Servo** (generic)



# Teensy IDE – Sensors/Actuators

Sample programs are at:

- 1) **Distance** (simple analog input; use `analogReadResolution(12);` )
- 2) **Light** (simple analog input; (simple analog input; use `analogReadResolution(12);` ))
- 3) **Temperature (I2C) – Requires Adafruit TMP007 library**
- 4) **Servo** (basic Servo program works; better libraries available)

**Have your pinout diagrams handy – not on the board!**

# Teensy to Raspberry PI

- 1) Via Firmata (USB) – via Python
- 2) Via Serial – very generic
- 3) Direct I/O pins – low level