



# Microcontroller basics

Embedded systems for  
mortals

# Topics

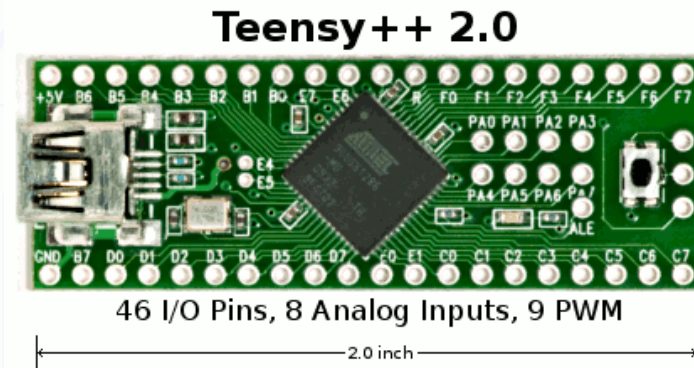
- Working with Arduino IDE
- Basic electric components
- Your first program
- Digital input and output

# Arduino IDE with Teensy MCU

- Download Arduino IDE from (version 1.6.7):  
<http://arduino.cc/en/Main/Software>
- Download Teensyduino from (version 1.27):  
[http://pjrc.com/teensy/td\\_download.html](http://pjrc.com/teensy/td_download.html)

# About Teensy

- Microcontroller used in these excersises is Teensy++ 2.0
- 16 MHz 8-bit Atmel Processor (AT90USB1286)



# Basic Components

## Passive components

Resistor



Capacitor



Push button



Inductor

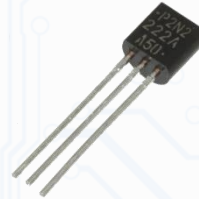


## Active components

LED



Transistor

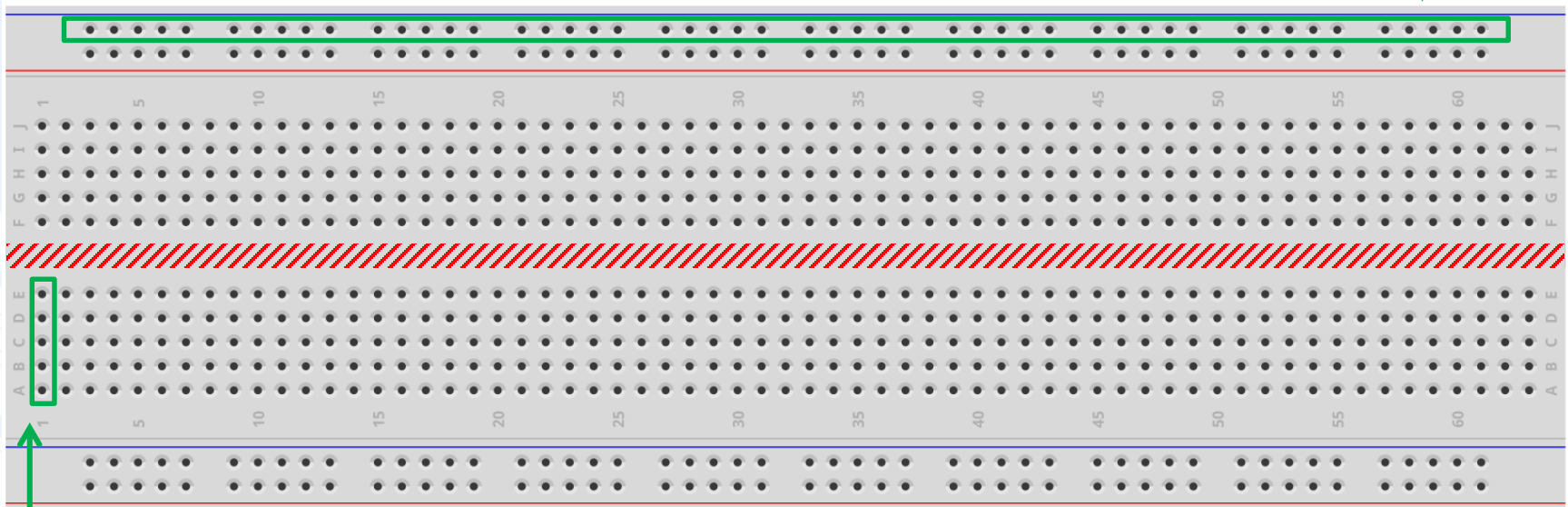


Integrated circuit (IC)



# Breadboard

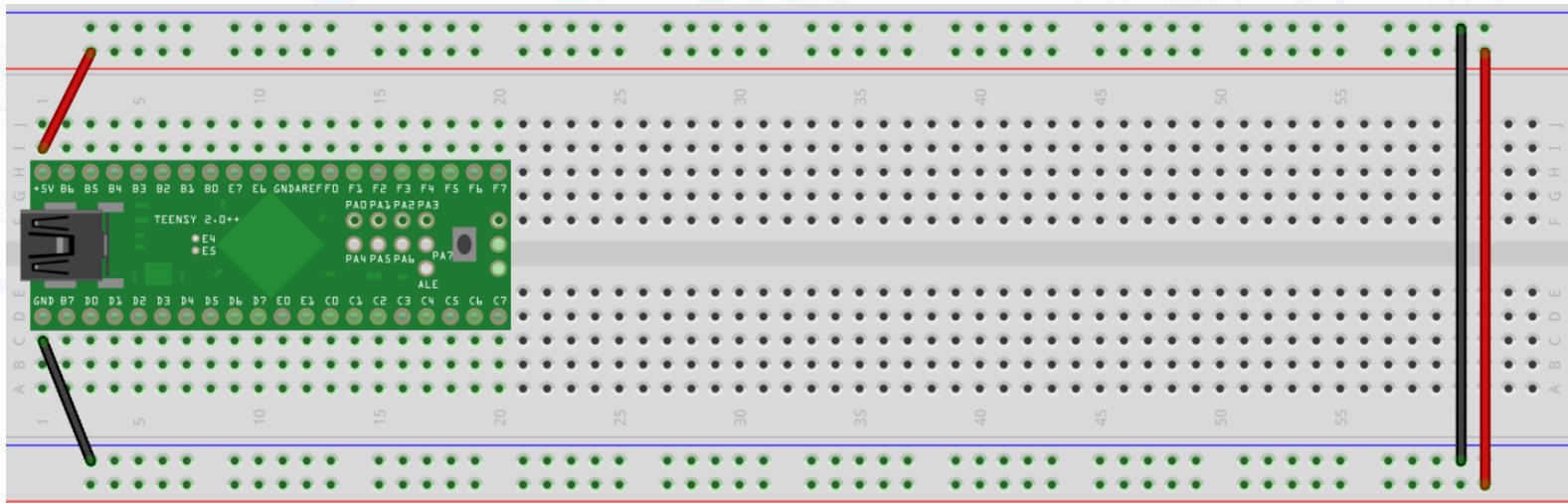
Terminals with blue and red lines are called power busses and are connected together horizontally.



Terminals in the middle are connected together vertically. The gap in the middle separates the two sides.

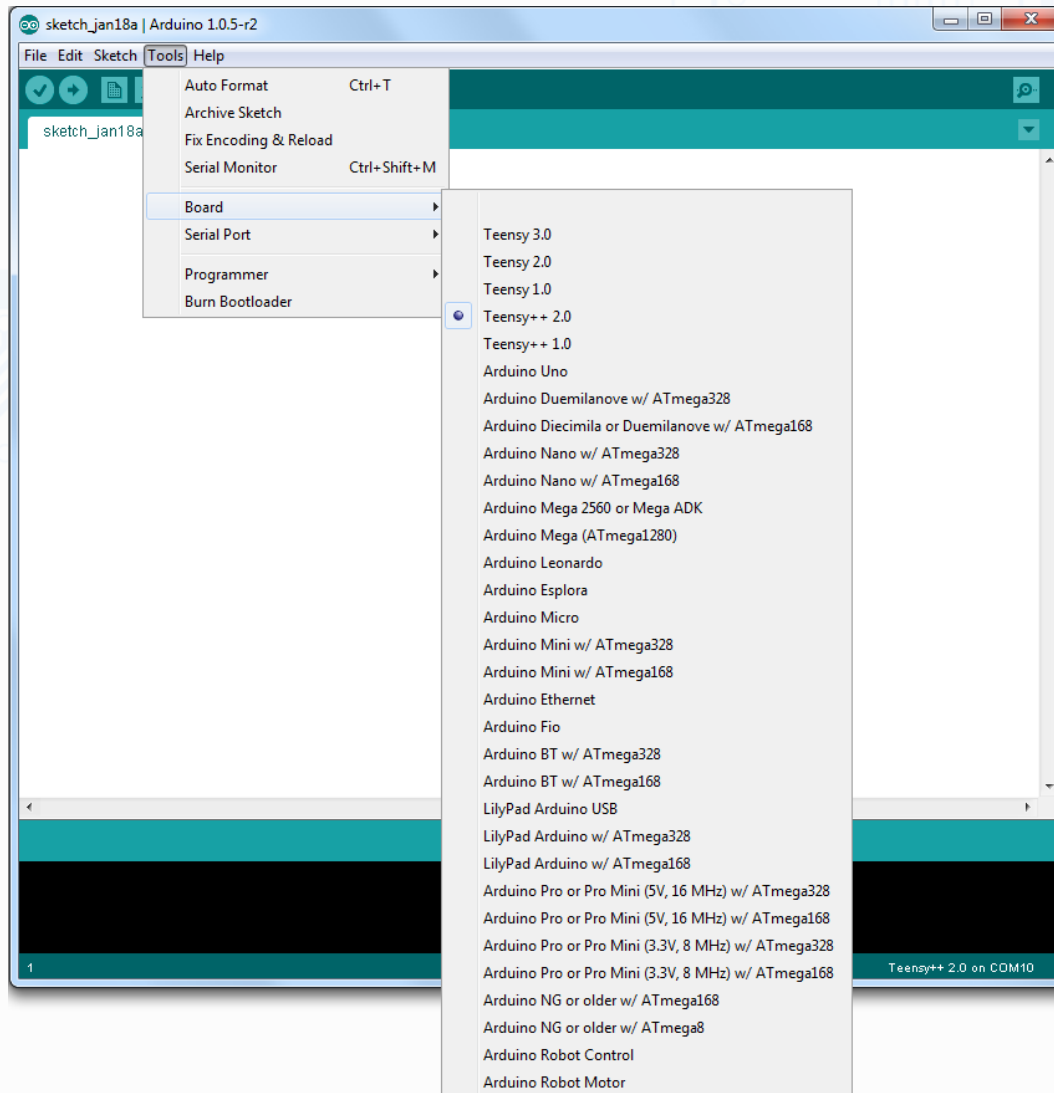
# Breadboard

Example: How to connect Teensy++ 2.0



fritzing

# Setting up Arduino IDE





# Writing your first program

## Basic blinking LED

```
byte ledPin = 6; //Variable to store the pin number
                //(the built in LED on Teensy++2.0 is connected to pin 6)

void setup()
{
  pinMode(ledPin, OUTPUT); //set ledPin as output
  digitalWrite(ledPin, LOW); //Set the default state of the pin to GND(0 Volts)
}

void loop()
{
  digitalWrite(ledPin, HIGH); //LED ON
  delay(100); //Wait 100ms (=0,1s)
  digitalWrite(ledPin, LOW); //LED OFF
  delay(1000); //Wait 1000ms (=1s)
}
```

# Uploading the program

1. Click Verify
2. Click Upload
3. Plug in and reset Teensy

```
Example1 | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Example1
int ledPin = 6; //Variable to store the pin number
// (the built in LED is connected to pin 6)
void setup()
{
  pinMode(ledPin, OUTPUT); //set LedPin as output
  digitalWrite(ledPin, LOW); //Set the default state of the pin to GND(0 Volts)
}
void loop()
{
  digitalWrite(ledPin, HIGH); //LED ON
  delay(100); //Wait 100ms (=0,1s)
  digitalWrite(ledPin, LOW); //LED OFF
  delay(1000); //Wait 1000ms (=1s)
}
Done uploading.
Estimated memory use: 22 bytes (of a 8 192 byte maximum)
Please press the RESET BUTTON on your Teensy to upload your sketch. Auto-reboot only works if
the Teensy is running a previous sketch.
17 Teensy++ 2.0 on COM3
```



# Arduino C – Basic functions

```
void setup()  
{
```

Setup function is run once, when the microcontroller boots up or resets.

```
void loop()  
{
```

After setup function the processor moves to run code inside the loop function. Code inside loop function will be run over and over until the microcontroller is shut down.

```
pinMode(var1, var2)
```

pinMode functions sets the mode of given pin. Var1 is the number of the pin and var2 is the mode (**INPUT**, **INPUT\_PULLUP**, **OUTPUT**)

```
digitalWrite(var1, var2)
```

digitalWrite changes the status of the pin. Var1 is the number of the pin and var2 is the status (**LOW**, **HIGH**).

## IMPORTANT TO NOTICE:

- It's required to have both setup() and loop() functions in the code
- Depending whether the pin is set as an **OUTPUT** or **INPUT** the actual effect of digitalWrite() is different



# Example 2 - Code

Reading digital input (Using push button)

```
byte ledPin = 6;
byte buttonPin = 12;

void setup()
{
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  pinMode(buttonPin, INPUT); //set buttonPin as input
  digitalWrite(buttonPin, HIGH); //Set the default state of the pin to HIGH(+5V)
}

void loop()
{
  if(digitalRead(buttonPin) == LOW)
  {
    digitalWrite(ledPin, HIGH); //LED ON
  }
  else
  {
    digitalWrite(ledPin, LOW); //LED OFF
  }
}
```

$$P=U*I \quad M \quad U=R*I$$

- Microcontrollers typically operate on low voltages (0-5V) → You must be careful when connecting devices
- Know the electrical limits of the microcontroller: Teensy can handle max 5V/40mA per pin
- Always double check the wiring! If you see smoke it's already too late!

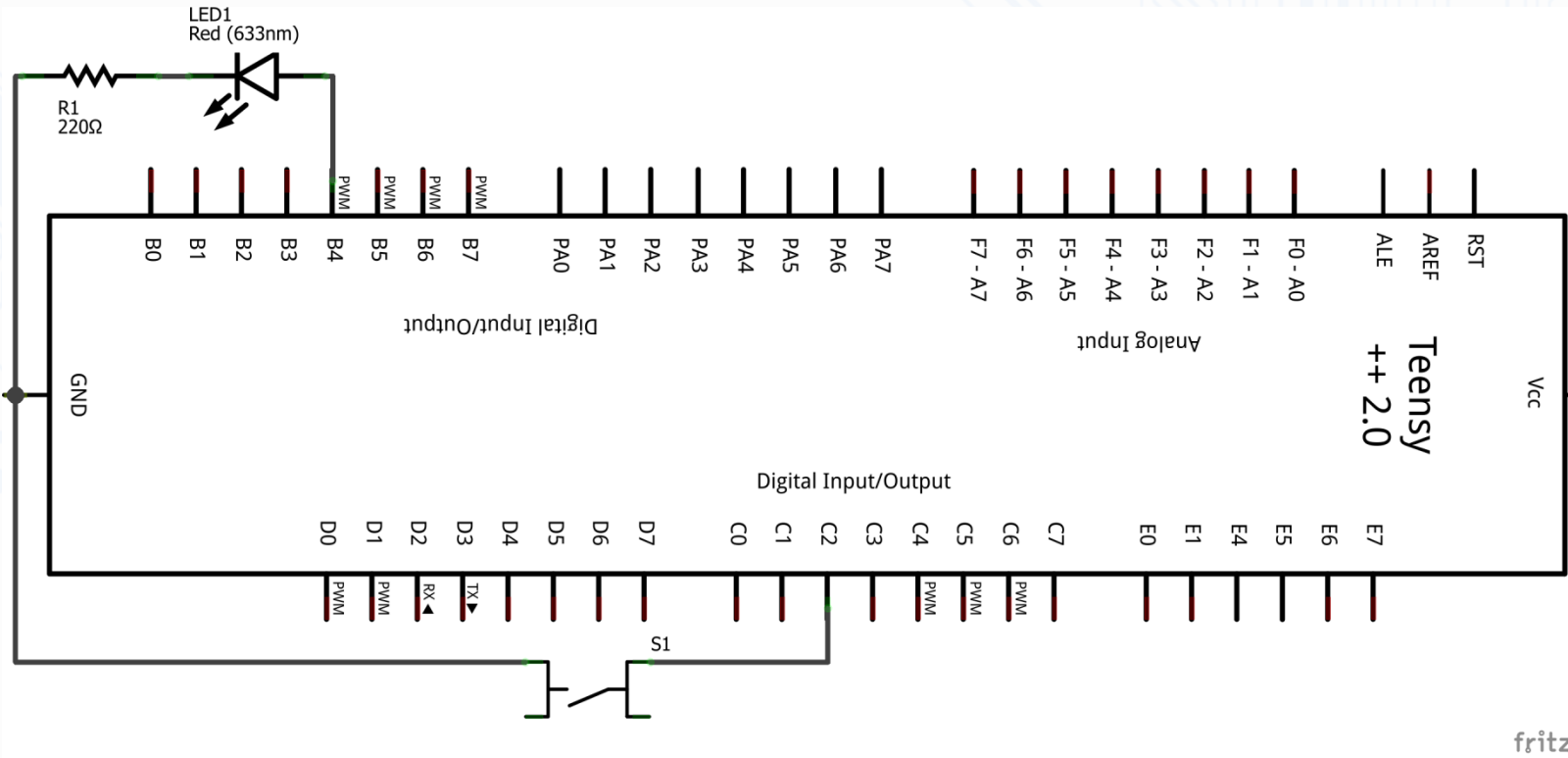
$$P=U*I$$

M

$$U=R*I$$

- To prevent overloading a pin or a component with excessive current you need to use a resistor
- Example: Using an LED – Calculating the required resistor size
  - Operation voltage for LED: 5V
  - Recommended current 23mA
- $U = R * I \rightarrow R = \frac{U}{I} = \frac{5V}{0,023A} \approx 220\Omega$

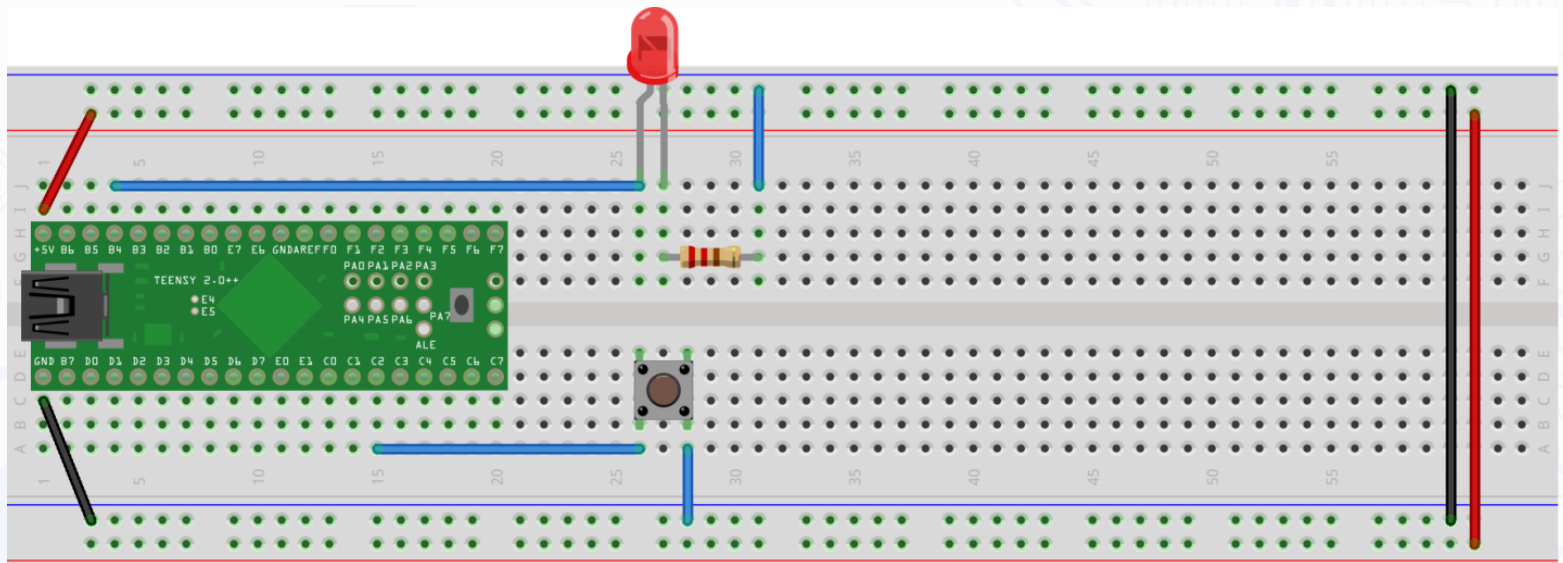
# Example 3 - Schematic



fritzing



# Example 3 - Breadboard Setup



fritzing

# Example 3 - Code

Controlling external LED with push button

```
byte ledPin = 24; //new ledPin value
byte buttonPin = 12;

void setup()
{
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  pinMode(buttonPin, INPUT);
  digitalWrite(buttonPin, HIGH);
}

void loop()
{
  if(digitalRead(buttonPin) == LOW)
  {
    digitalWrite(ledPin, HIGH); //LED ON
  }
  else
  {
    digitalWrite(ledPin, LOW); //LED OFF
  }
}
```

# You have just entered the Matrix

