

Ch. 2 IoT Development Boards

Sec 1 – Arduino Board Intro

COMPSCI 147

Internet-of-Things; Software and Systems

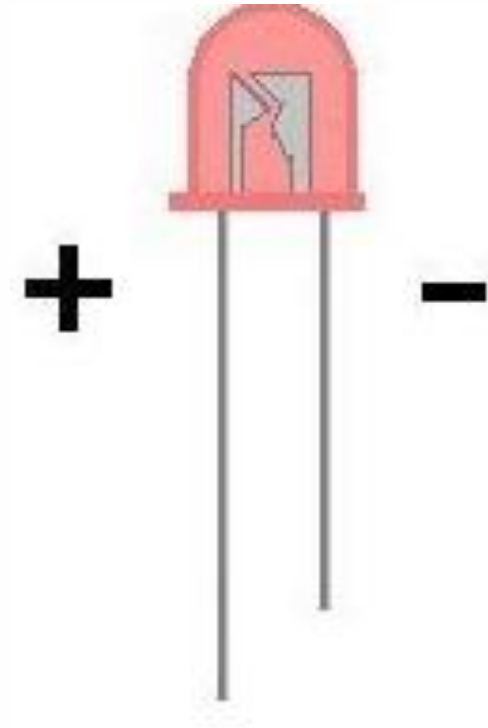


Step 1: Let's construct a simple circuit

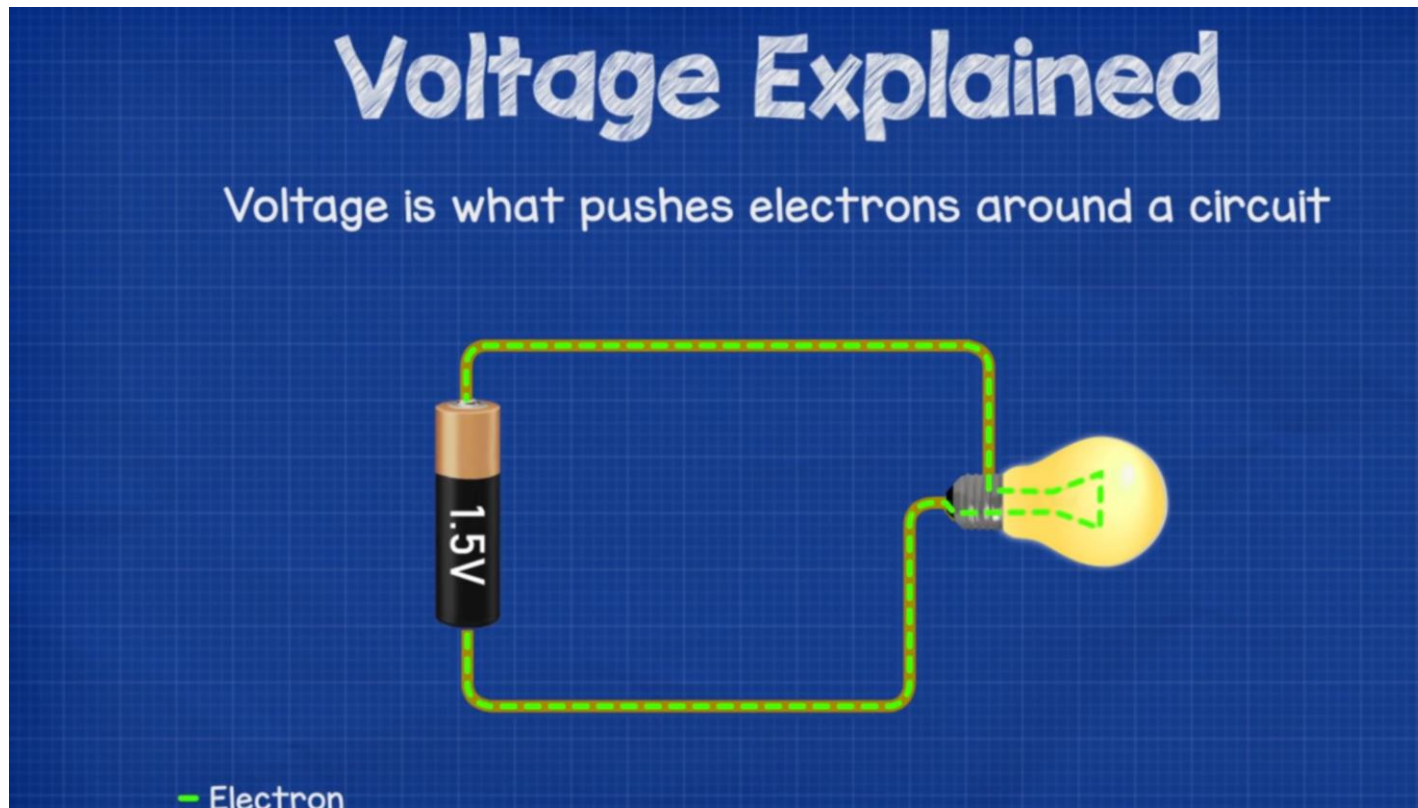
- Make an LED glow

Step 1: Let's construct a simple circuit

- Make an LED glow
- Need to flow current (flow of electrons) through the LED
- Voltage is what pushes electrons around a circuit.
- Without voltage, electrons move randomly in any direction.



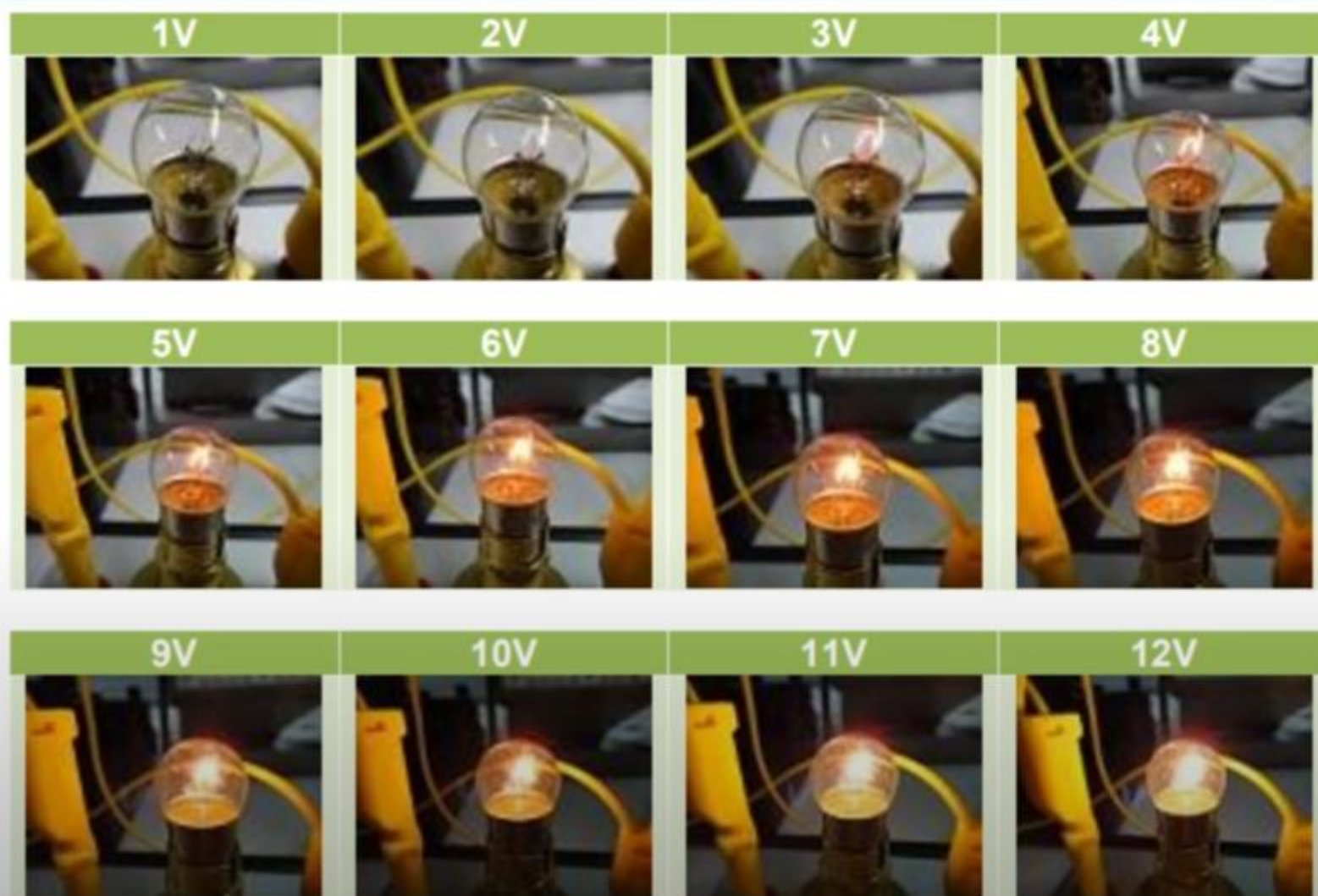
Step 1: Let's construct a simple circuit



https://www.youtube.com/watch?v=w82aSjLuD_8

Voltage Explained - What is Voltage? Basic electricity potential difference

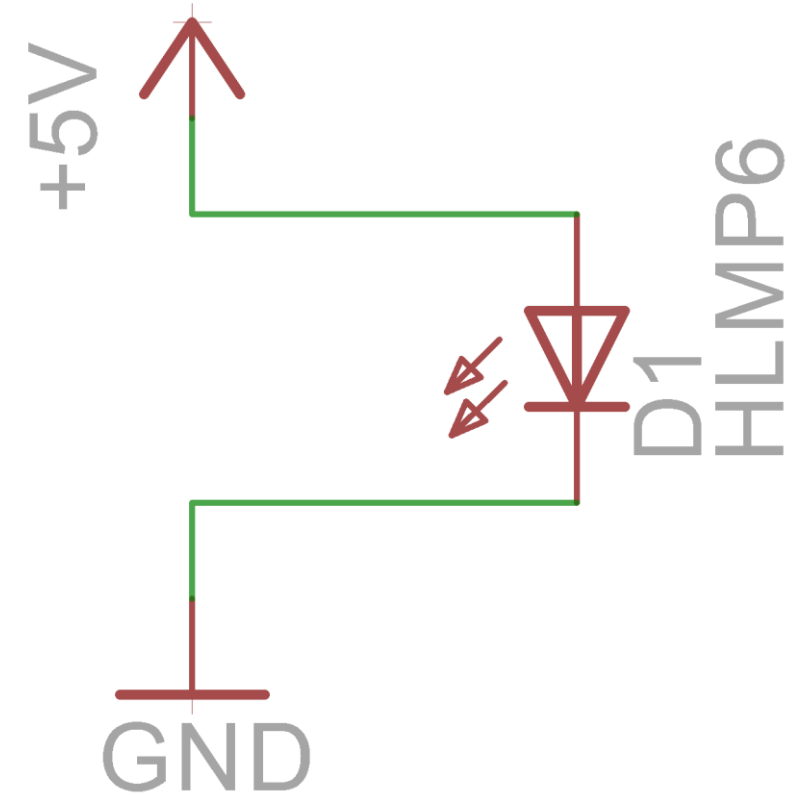
Step 1: Let's construct a simple circuit



As voltage decreases
so does the brightness
because there is less
pressure to force electrons
so less light is produced

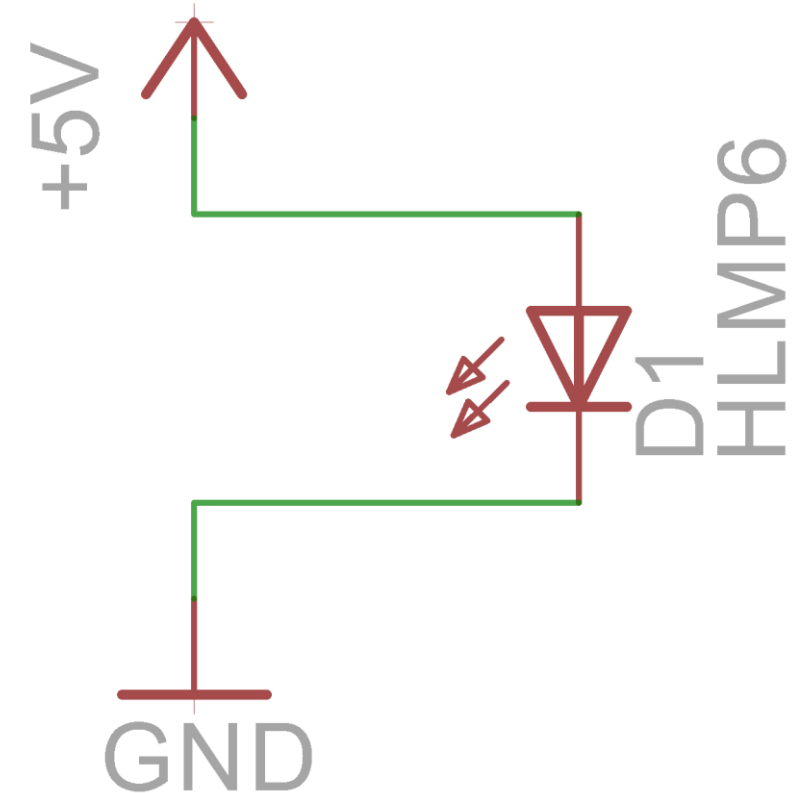
Step 1: Let's construct a simple circuit

- USB works with 5V (DC)
- What will happen if we plug in LED with a 5V power supply ?

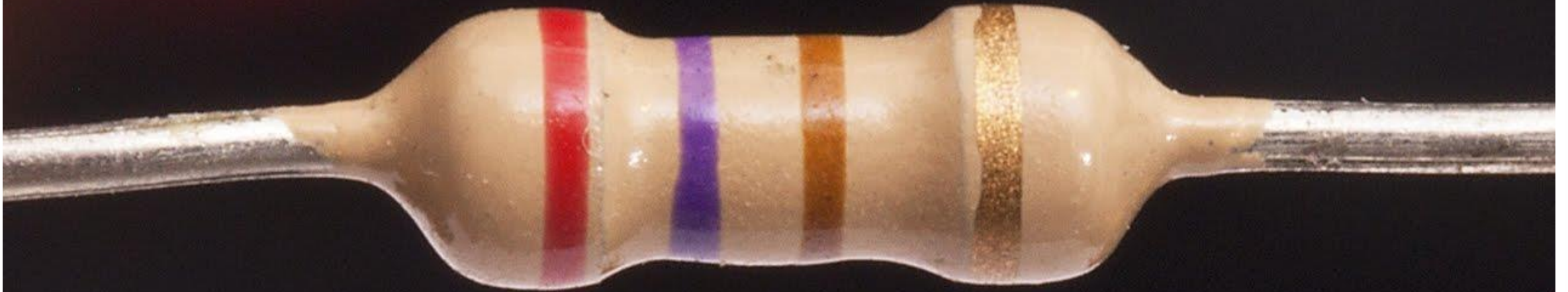


Step 1: Let's construct a simple circuit

- USB works with 5V (DC)
- What will happen if we plug in LED with a 5V power supply ?



RESISTANCE

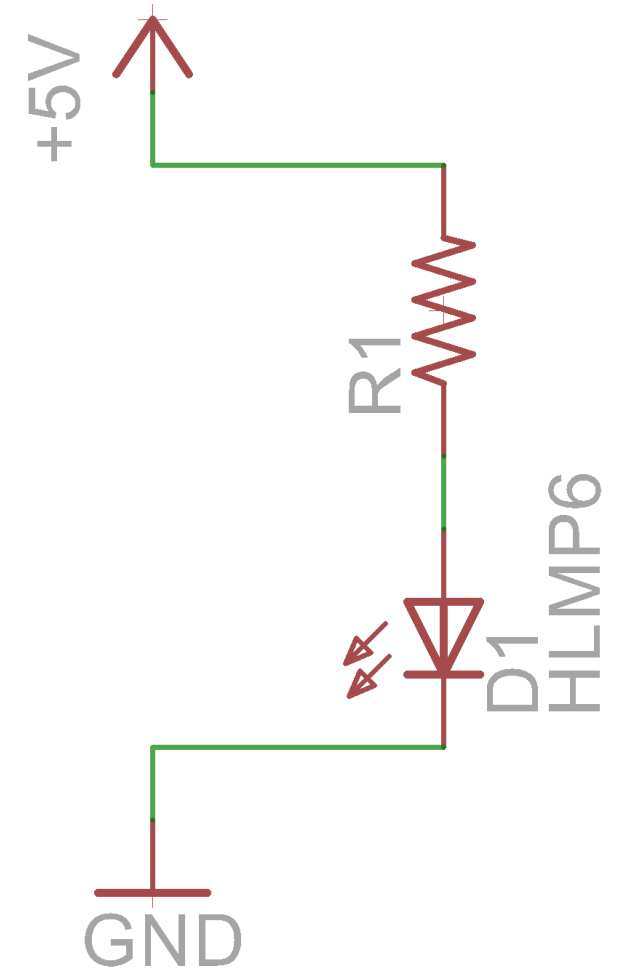


Step 1: Let's construct a simple circuit

- USB works with 5V (DC)
- What will happen if we plug in LED with a 5V power supply ?

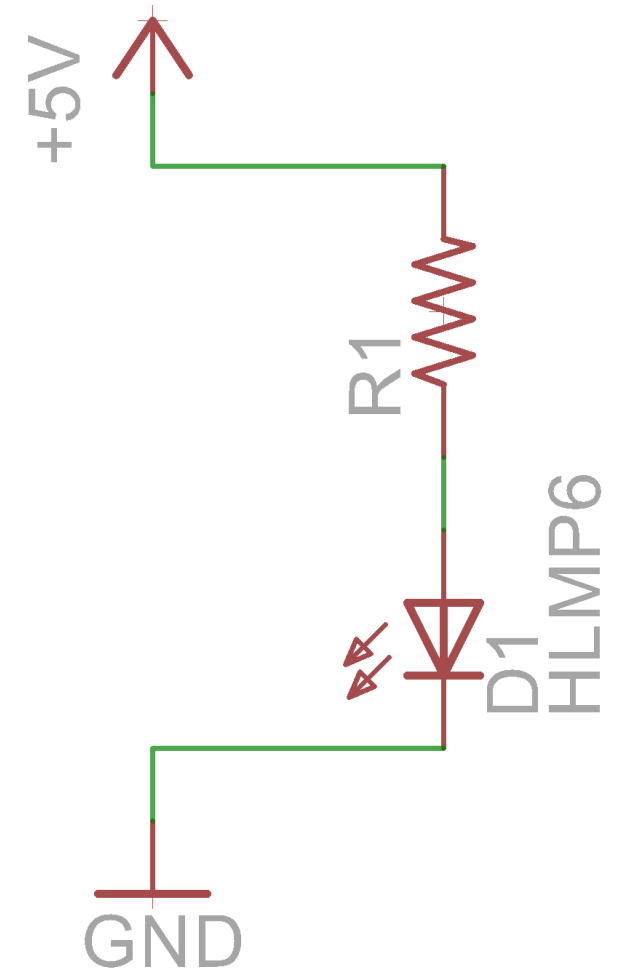


- Meet **resistance**



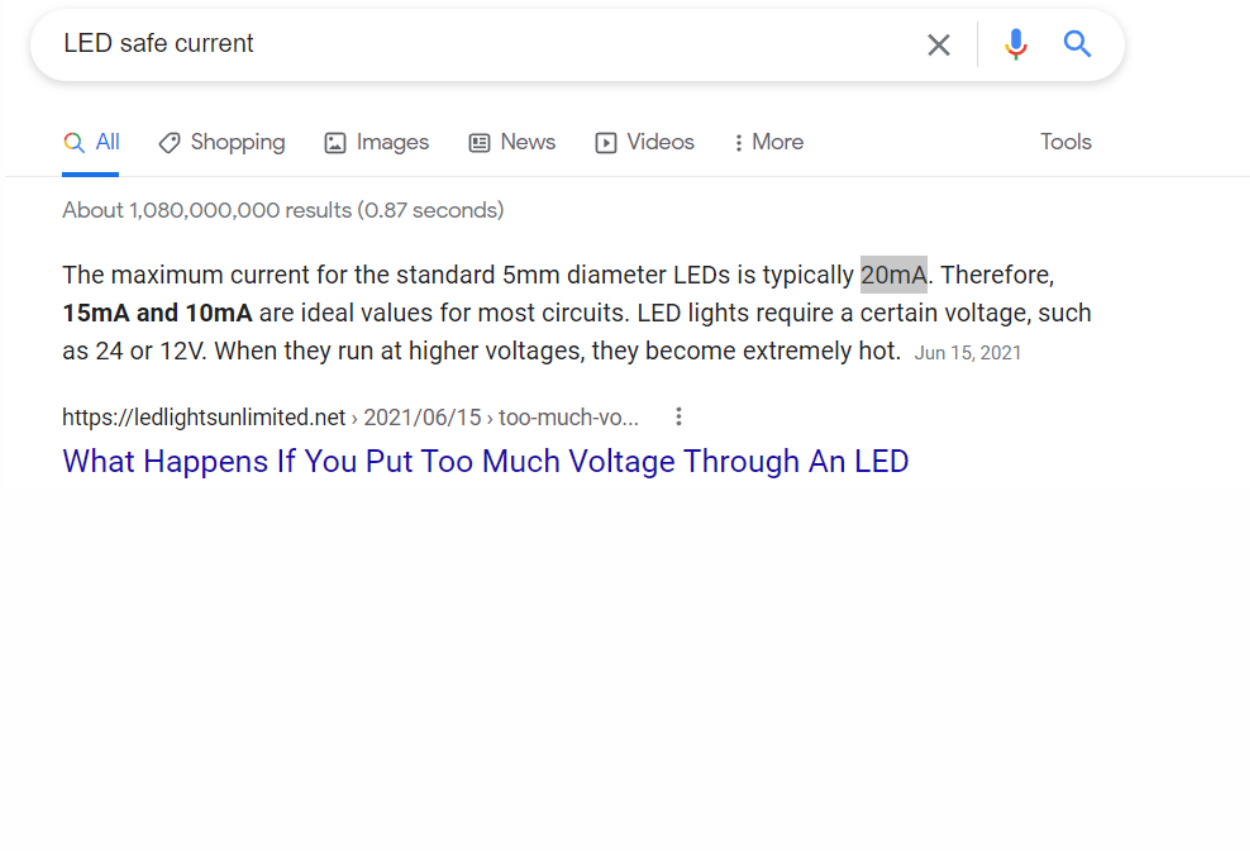
Step 1: Let's construct a simple circuit

- What should be the value of resistor ?



Step 1: Let's construct a simple circuit

- What should be the value of resistor ?



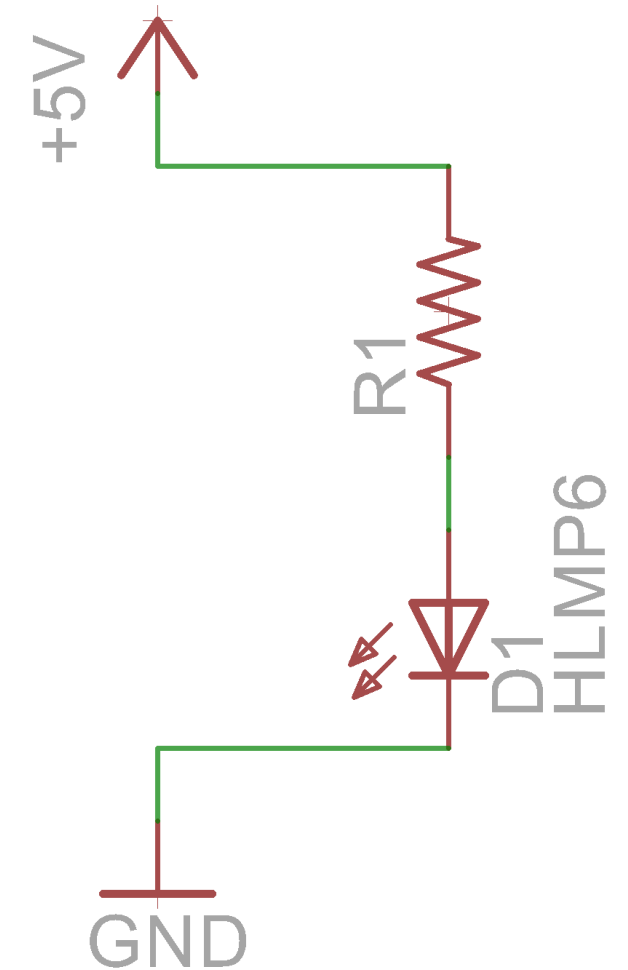
A screenshot of a Google search interface. The search bar contains the text "LED safe current". Below the search bar, the search results are displayed. The first result is from "ledlightsunlimited.net" dated "2021/06/15" with the title "What Happens If You Put Too Much Voltage Through An LED". The snippet of the result states: "The maximum current for the standard 5mm diameter LEDs is typically 20mA. Therefore, 15mA and 10mA are ideal values for most circuits. LED lights require a certain voltage, such as 24 or 12V. When they run at higher voltages, they become extremely hot. Jun 15, 2021".

LED safe current

About 1,080,000,000 results (0.87 seconds)

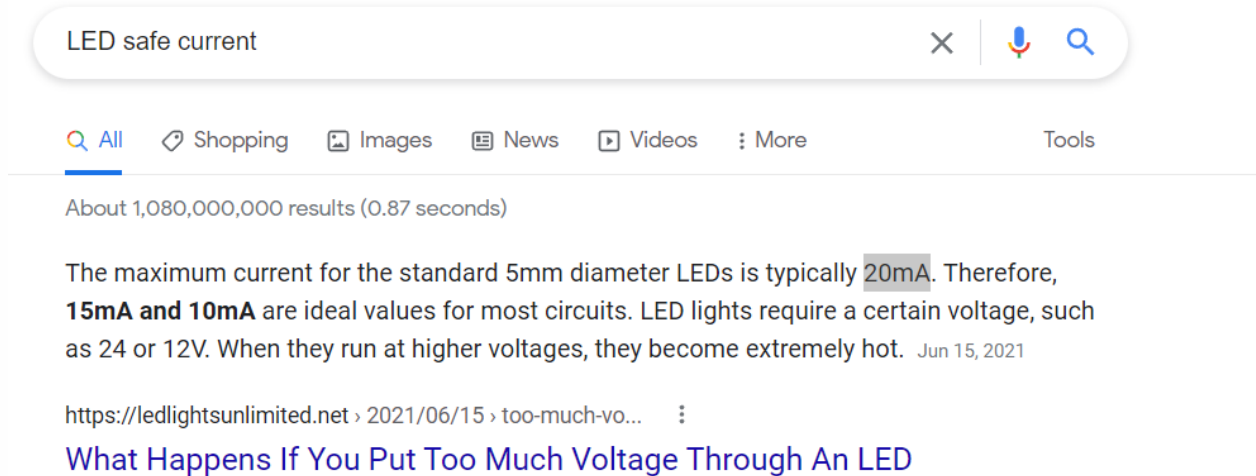
The maximum current for the standard 5mm diameter LEDs is typically 20mA. Therefore, 15mA and 10mA are ideal values for most circuits. LED lights require a certain voltage, such as 24 or 12V. When they run at higher voltages, they become extremely hot. Jun 15, 2021

<https://ledlightsunlimited.net> › 2021/06/15 › too-much-vo...
What Happens If You Put Too Much Voltage Through An LED



Step 1: Let's construct a simple circuit

- What should be the value of resistor ?

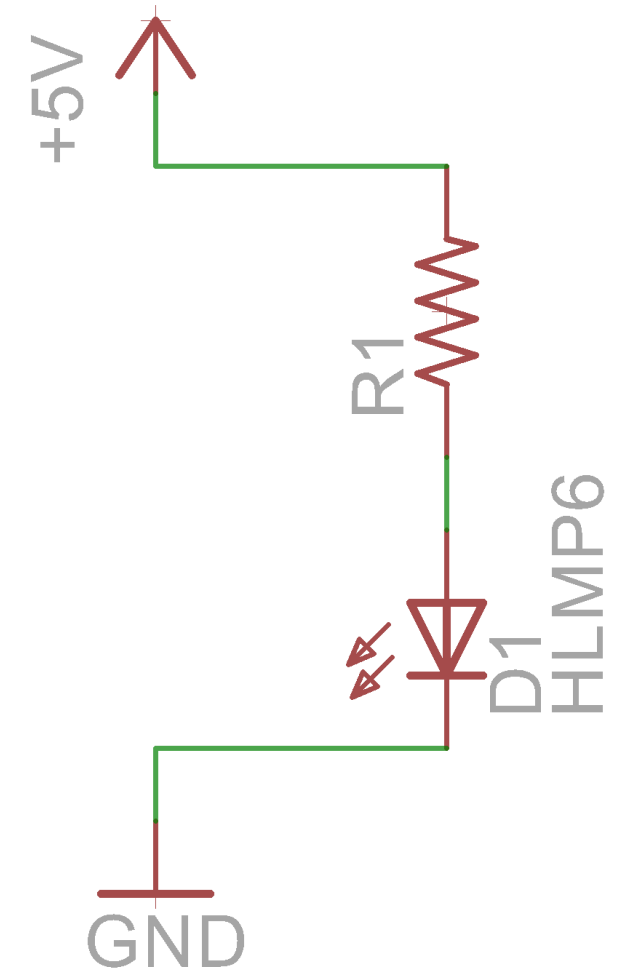


LED safe current

Search results for "LED safe current":

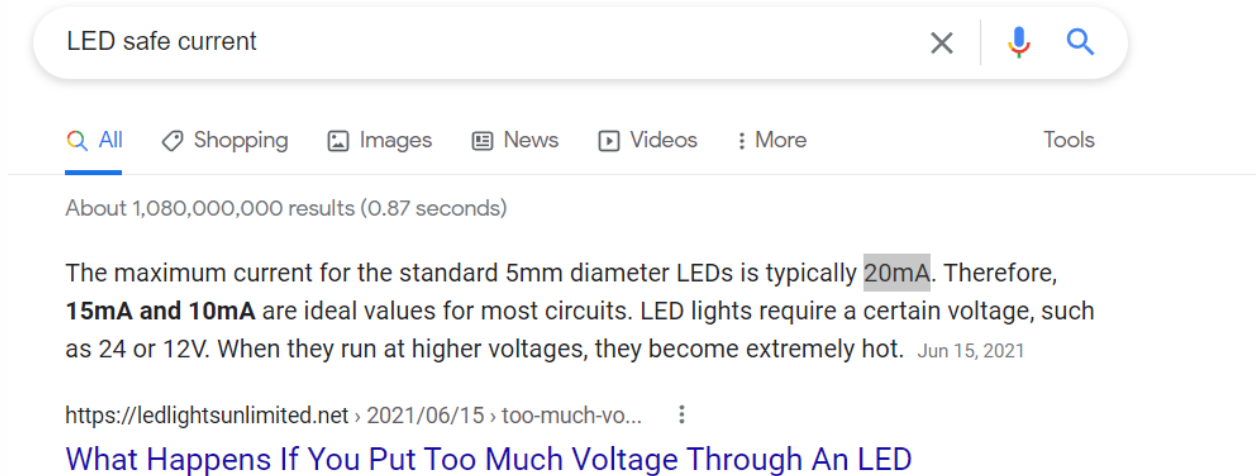
- Search filters: All, Shopping, Images, News, Videos, More, Tools
- Results: About 1,080,000,000 results (0.87 seconds)
- Text snippet: "The maximum current for the standard 5mm diameter LEDs is typically 20mA. Therefore, 15mA and 10mA are ideal values for most circuits. LED lights require a certain voltage, such as 24 or 12V. When they run at higher voltages, they become extremely hot. Jun 15, 2021"
- Source: <https://ledlightsunlimited.net> > 2021/06/15 > too-much-vo...
[What Happens If You Put Too Much Voltage Through An LED](#)

- $V=IR$ (Ohm's Law)



Step 1: Let's construct a simple circuit

- What should be the value of resistor ?

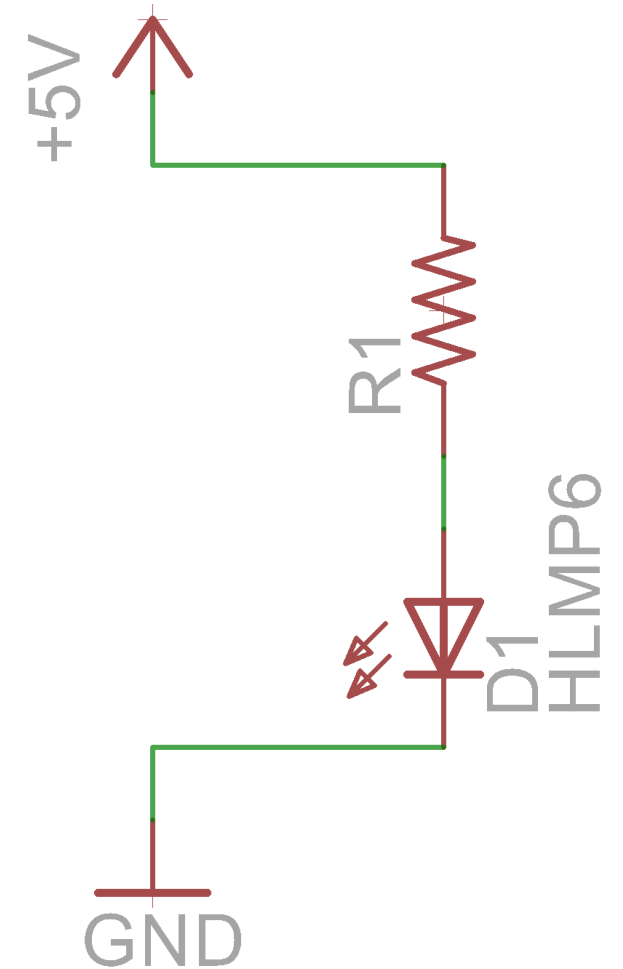


LED safe current

Search results for "LED safe current":

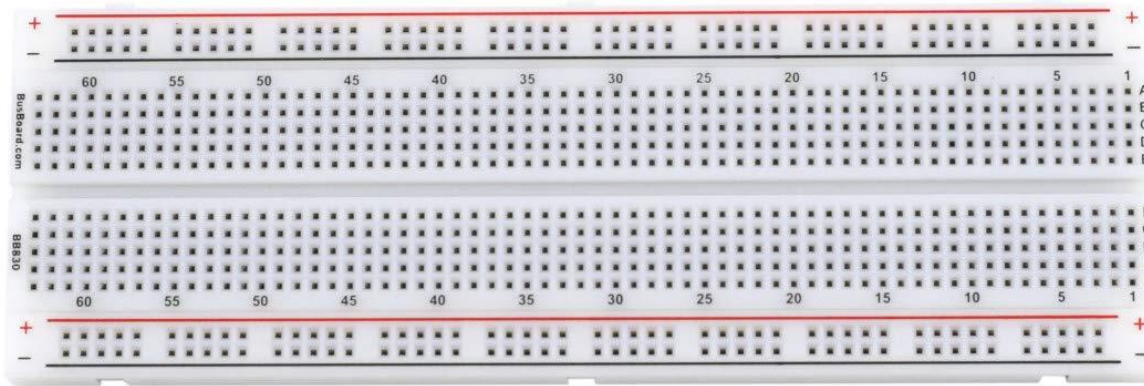
- Search filters: All, Shopping, Images, News, Videos, More, Tools
- Results: About 1,080,000,000 results (0.87 seconds)
- Text snippet: "The maximum current for the standard 5mm diameter LEDs is typically 20mA. Therefore, 15mA and 10mA are ideal values for most circuits. LED lights require a certain voltage, such as 24 or 12V. When they run at higher voltages, they become extremely hot. Jun 15, 2021"
- Source: <https://ledlightsunlimited.net> > 2021/06/15 > too-much-vo...
[What Happens If You Put Too Much Voltage Through An LED](#)

- $V=IR$ (Ohm's Law)
- $R = 5V / 0.020A$
~250 Ohm

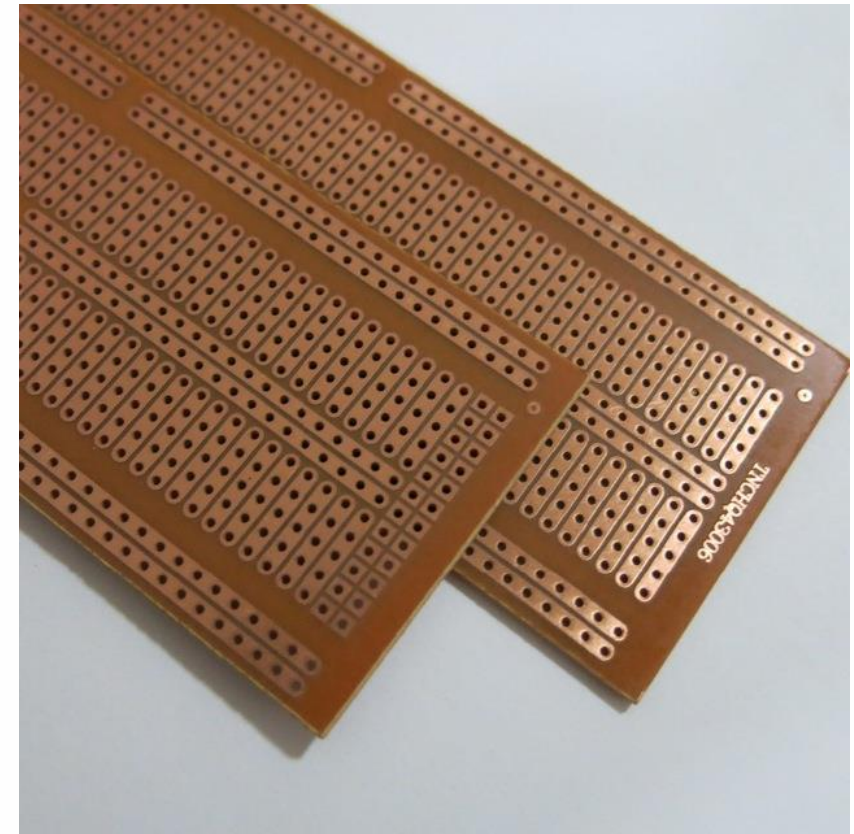


Step 1: Let's construct a simple circuit

- Prototype mechanisms



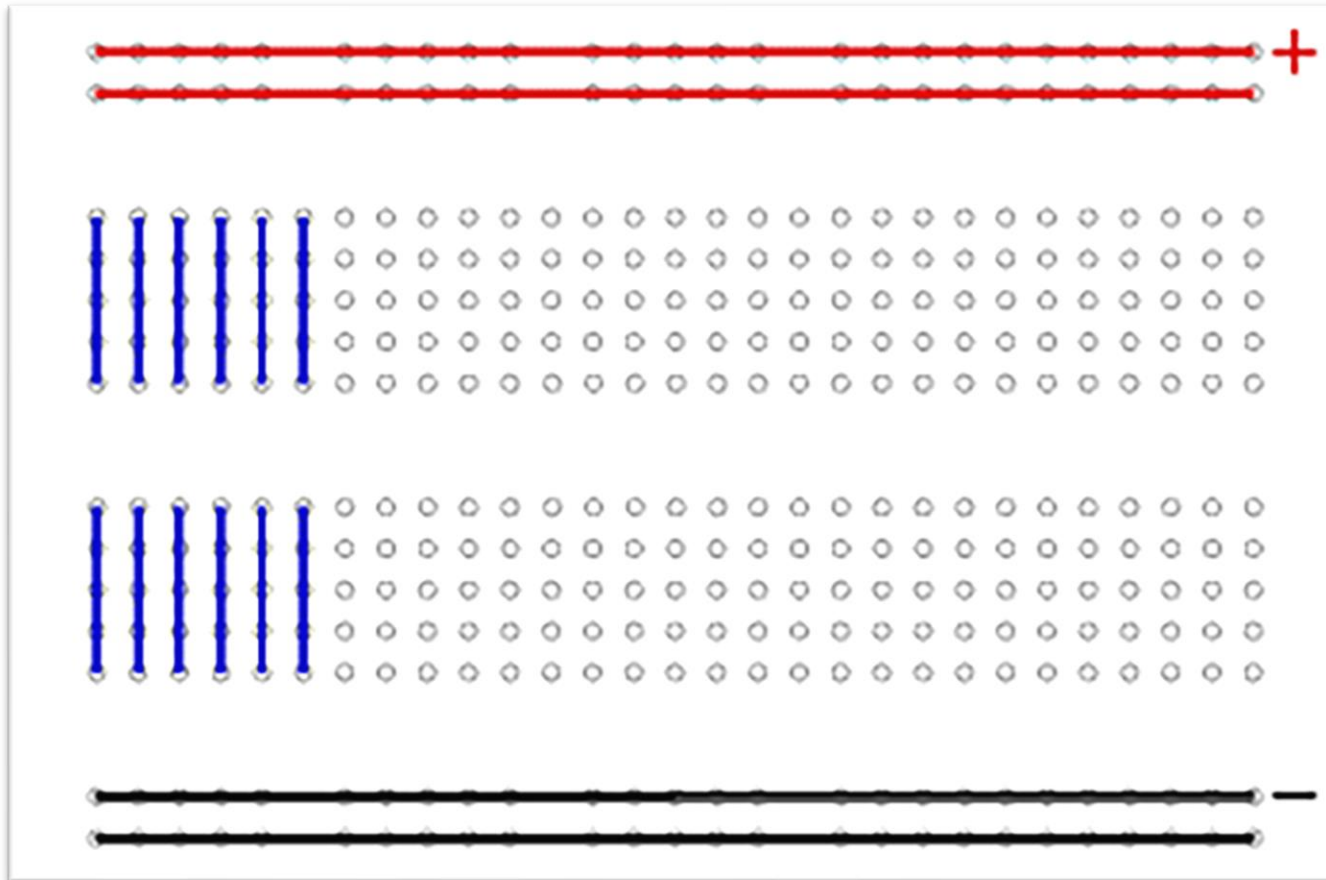
Breadboard



Vero/Strip board

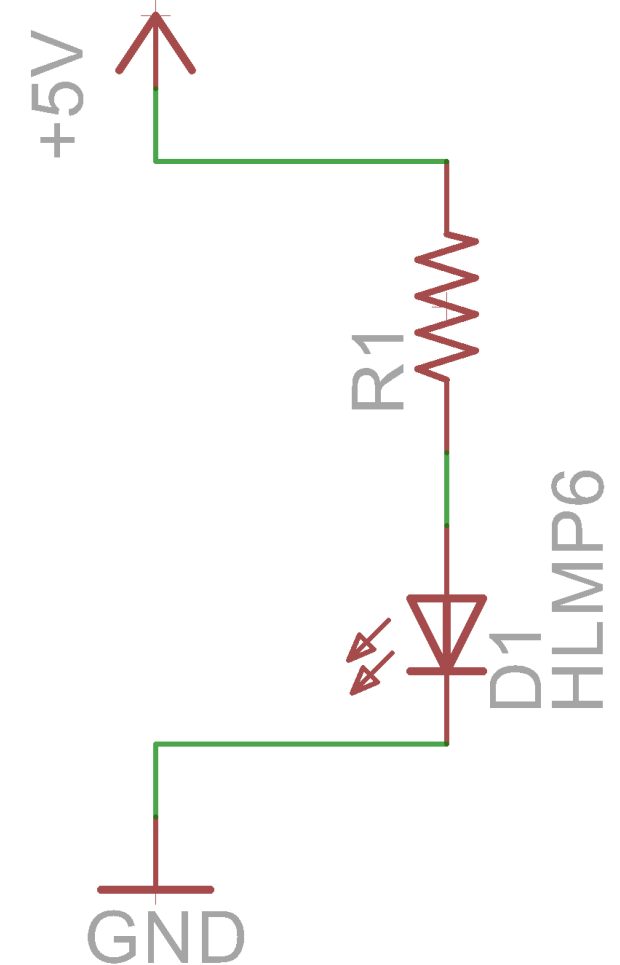
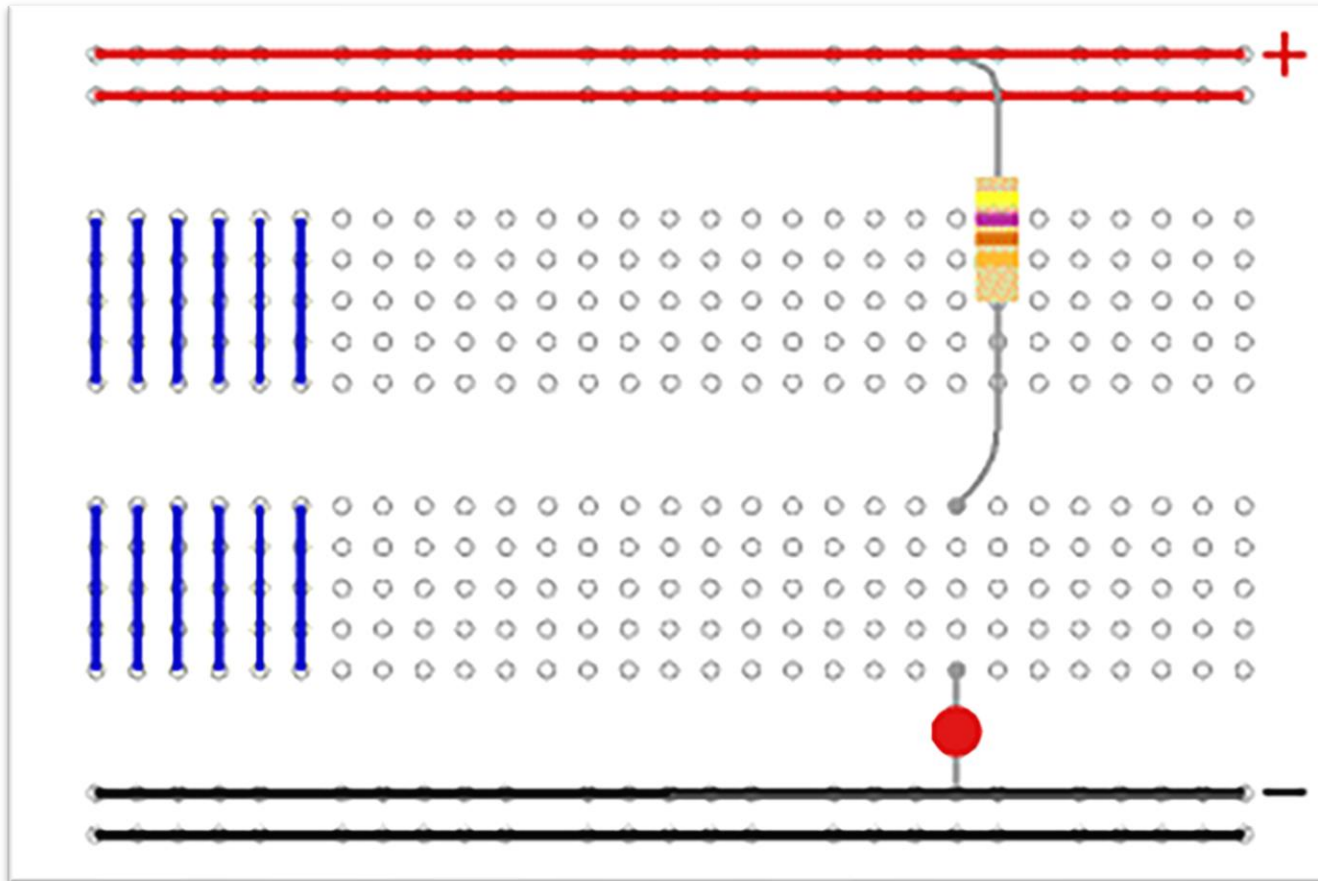
Step 1: Let's construct a simple circuit

- Bread board connections



Step 1: Let's construct a simple circuit

- Bread board connections



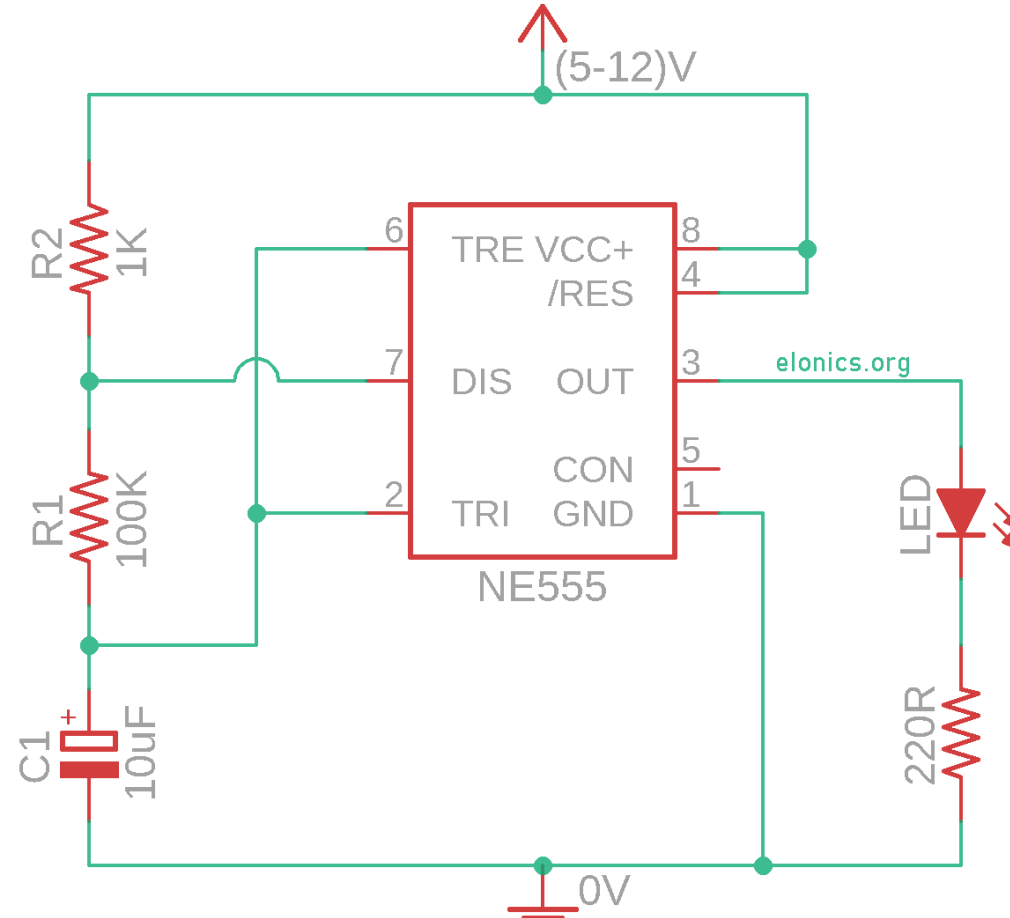
Step 2: Make the light blink for every 1 second

- Goal:



Step 2: Make the light blink for every 1 second

- Goal:

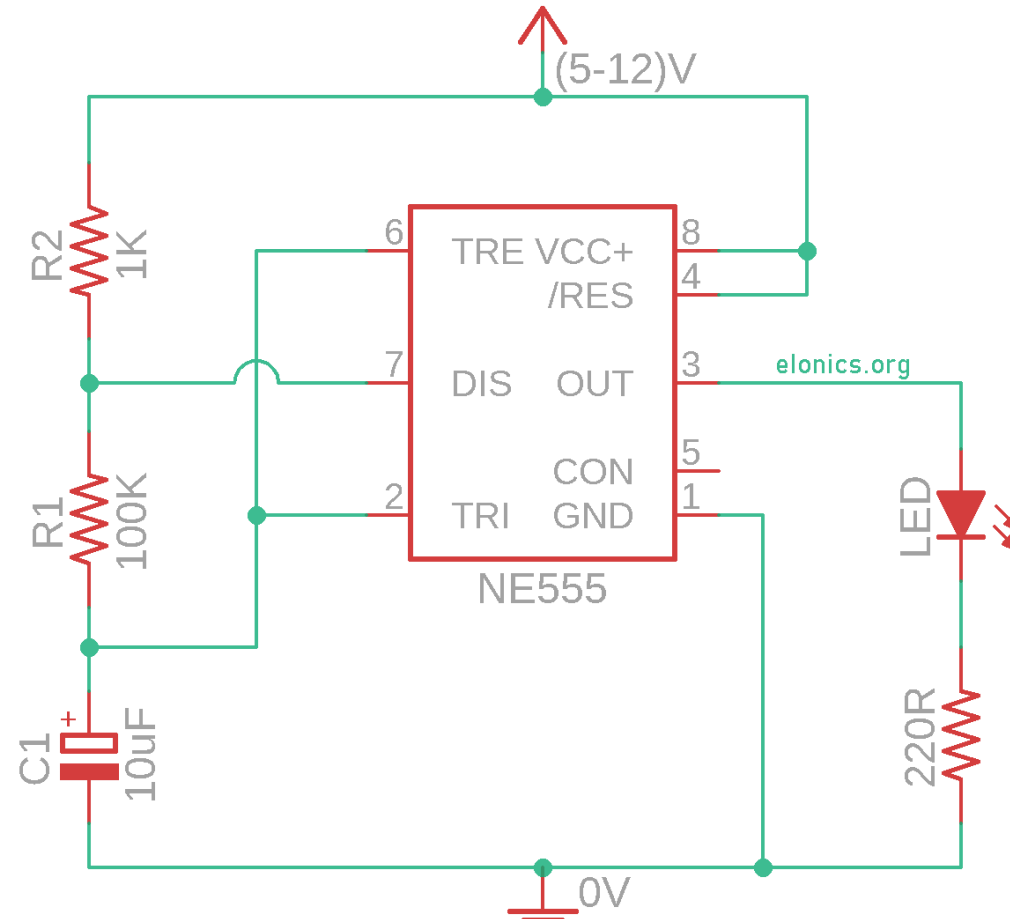


FLASHING/BLINKING LED SCHEMATIC

<https://elonics.org/adjustable-led-flasher-using-555-timer/>

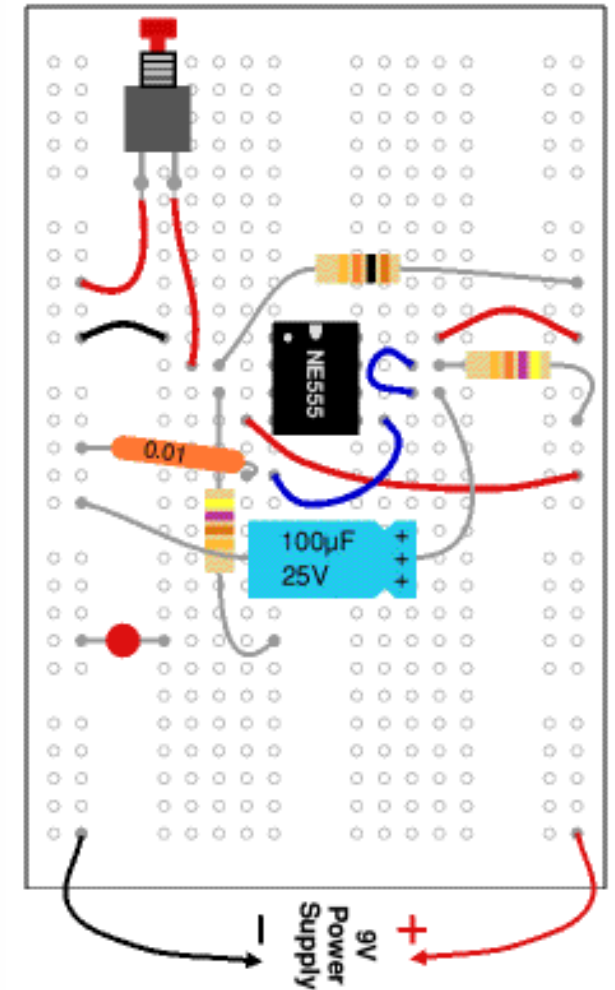
Step 2: Make the light blink for every 1 second

- Goal:



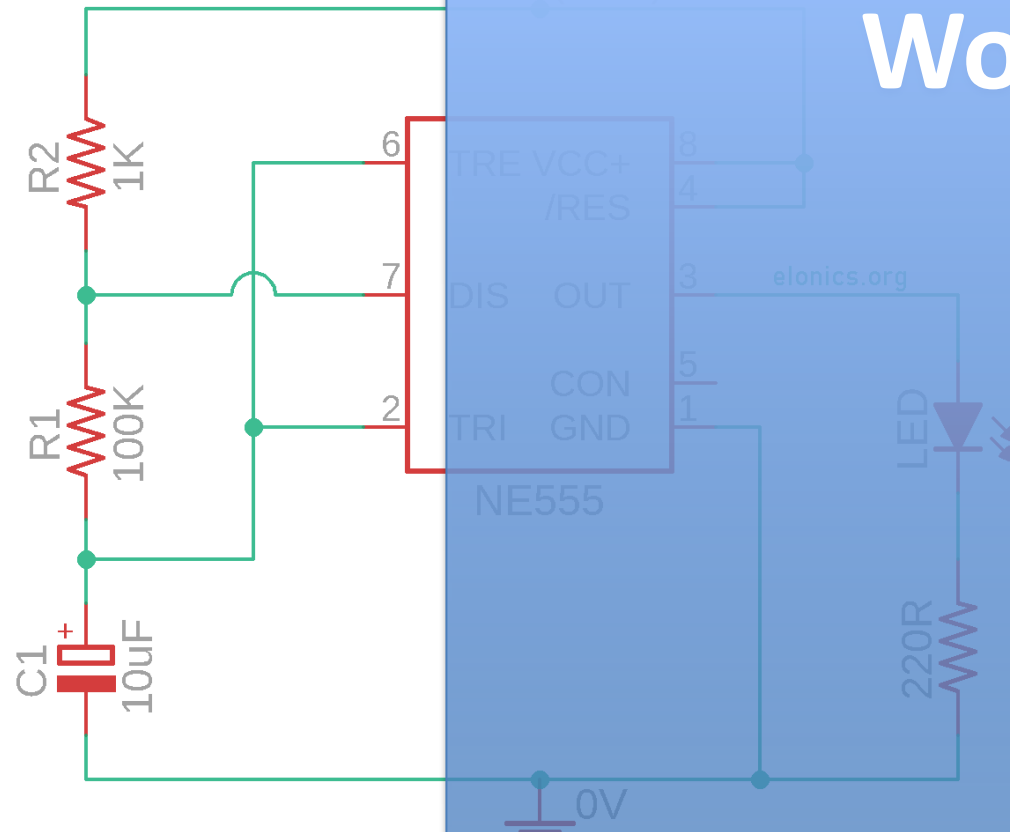
FLASHING/BLINKING LED SCHEMATIC

<https://elonics.org/adjustable-led-flasher-using-555-timer/>



Step 2: Make the light blink for every 1 second

- Goal:



Wouldn't it be nice..?

```
void loop()
{
  led(ON);
  sleep(1);
  led(OFF);
  sleep(1);
}
```

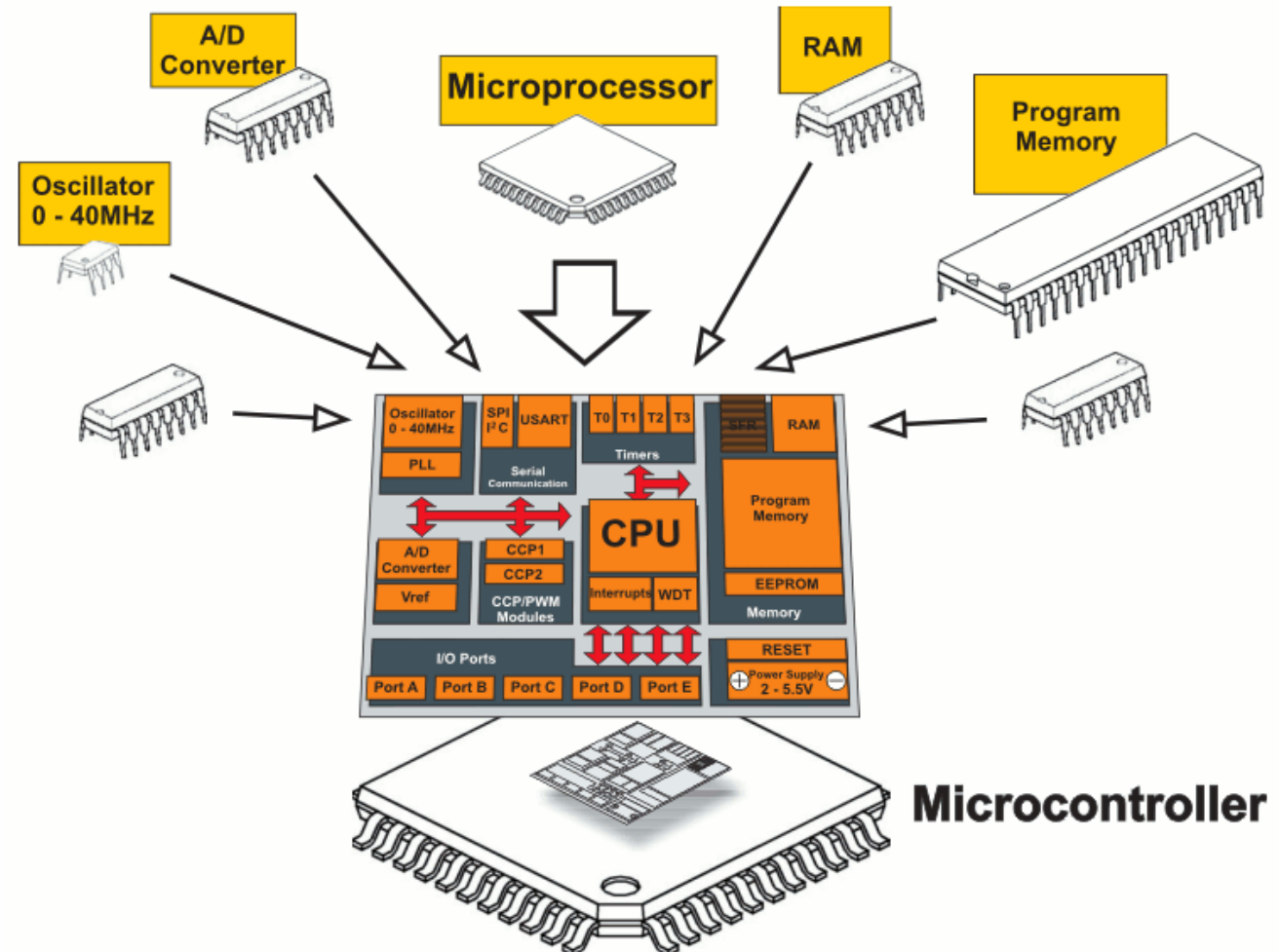
FLASHING/BLINKING LED SCHEMATIC

<https://elonics.org/adjustable-led-flasher-using-555-timer/>



WHAT IS A MICROCONTROLLER

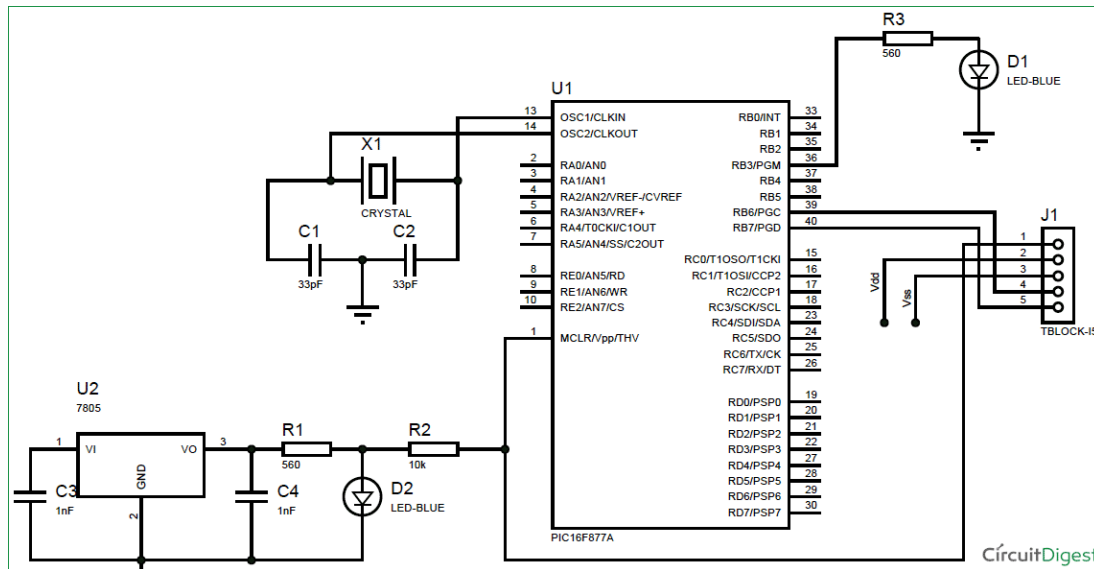
- A small computer on a single chip with
 1. Processor
 2. Memory
 3. input/output
- Typically, "**embedded**" inside some device that they control
- A microcontroller is often small and low cost



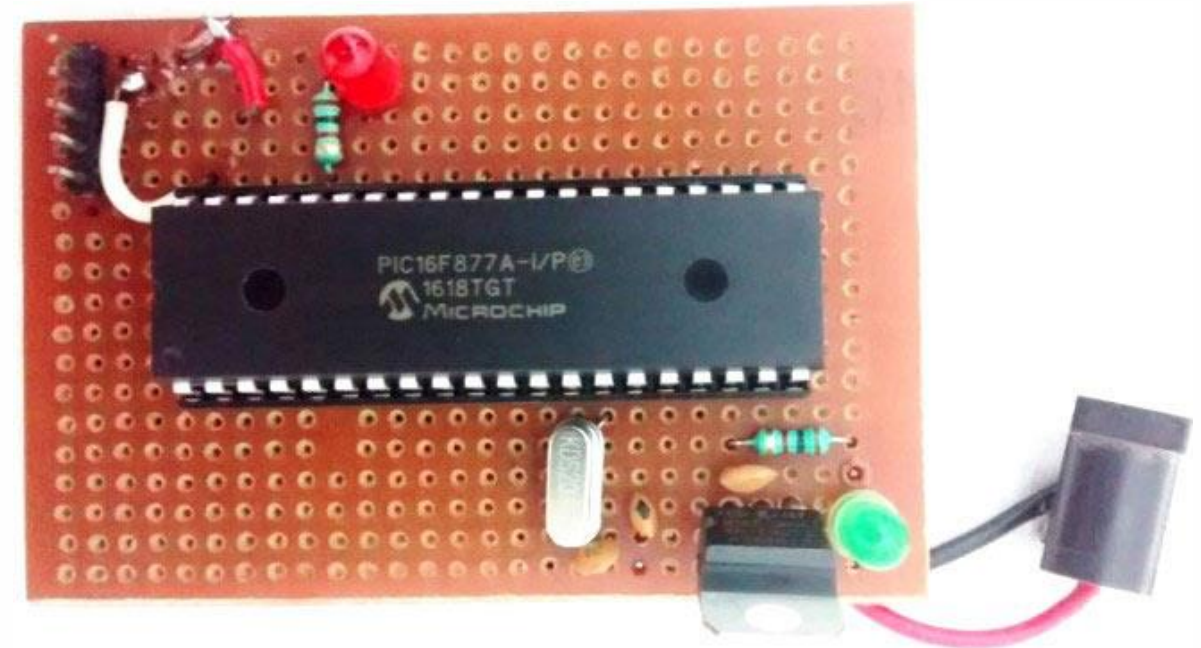
STEP 3: BLINKING LED WITH A MICROCONTROLLER

STEP 3: BLINKING LED WITH A MICROCONTROLLER

- Construct circuit

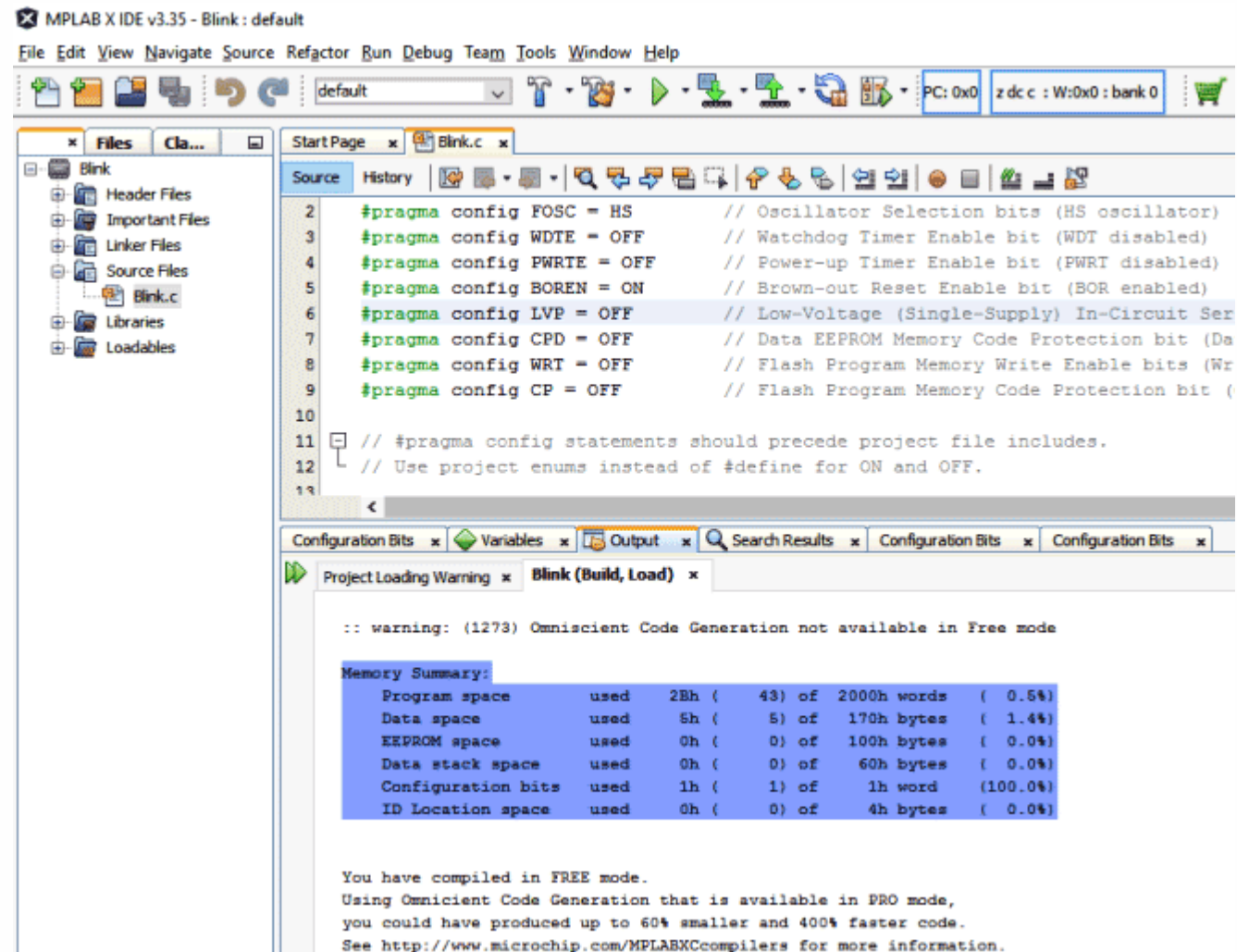


<https://circuitdigest.com/microcontroller-projects/led-blinking-with-pic-microcontroller>



STEP 3: BLINKING LED WITH A MICROCONTROLLER

- Construct circuit
- Compile program



MPLAB X IDE v3.35 - Blink: default

File Edit View Navigate Source Refactor Run Debug Team Tools Window Help

default PC: 0x0 zdc c : W:0x0 : bank 0

Files Cla... StartPage x Blink.c x

Source History

```
2 #pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)
3 #pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)
4 #pragma config PWRT = OFF // Power-up Timer Enable bit (PWRT disabled)
5 #pragma config BOREN = ON // Brown-out Reset Enable bit (BOR enabled)
6 #pragma config LVP = OFF // Low-Voltage (Single-Supply) In-Circuit Ser
7 #pragma config CPD = OFF // Data EEPROM Memory Code Protection bit (Da
8 #pragma config WRT = OFF // Flash Program Memory Write Enable bits (Wr
9 #pragma config CP = OFF // Flash Program Memory Code Protection bit (
10
11 // #pragma config statements should precede project file includes.
12 // Use project enums instead of #define for ON and OFF.
13
```

Configuration Bits x Variables x Output x Search Results x Configuration Bits x Configuration Bits x

ProjectLoading Warning x Blink (Build, Load) x

:: warning: (1273) Omniscient Code Generation not available in Free mode

Memory Summary:

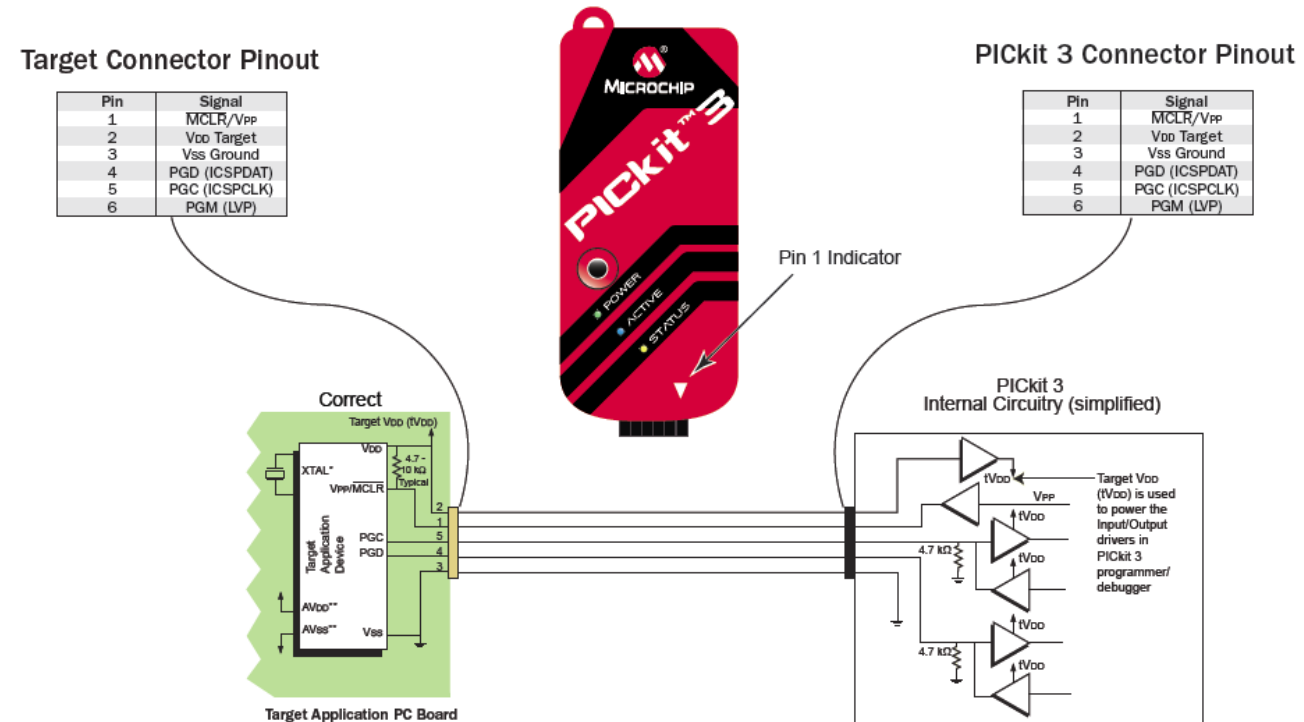
Program space	used	2Bh (43) of 2000h words (0.5%)
Data space	used	5h (5) of 170h bytes (1.4%)
EEPROM space	used	0h (0) of 100h bytes (0.0%)
Data stack space	used	0h (0) of 60h bytes (0.0%)
Configuration bits	used	1h (1) of 1h word (100.0%)
ID Location space	used	0h (0) of 4h bytes (0.0%)

You have compiled in FREE mode.
Using Omniscient Code Generation that is available in PRO mode,
you could have produced up to 60% smaller and 400% faster code.
See <http://www.microchip.com/MPLABXCcompilers> for more information.

STEP 3: BLINKING LED WITH A MICROCONTROLLER

- Construct circuit
- Compile program
- Flash the program

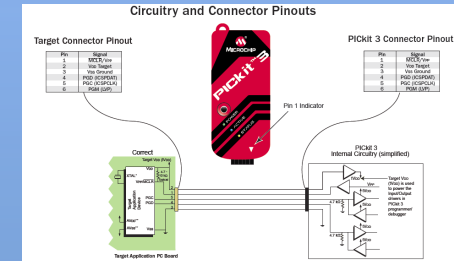
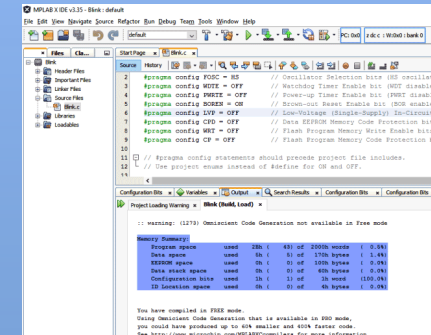
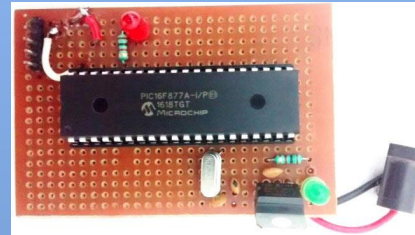
Circuitry and Connector Pinouts



STEP 3: BLINKING LED WITH A MICROCONTROLLER

- Construct circuit
- Compile program
- Flash the program

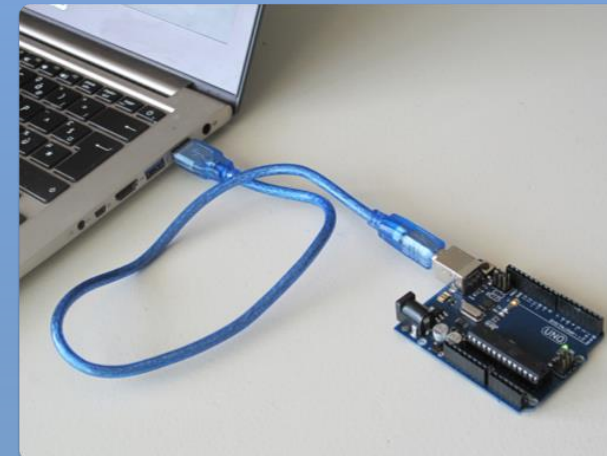
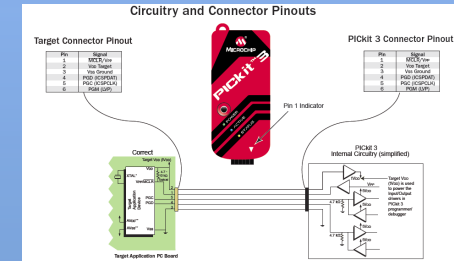
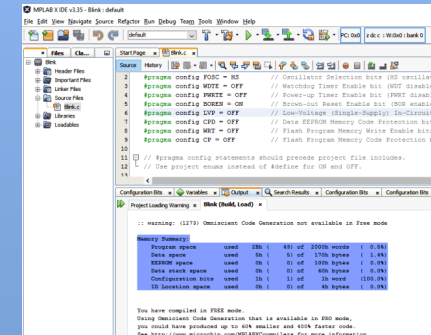
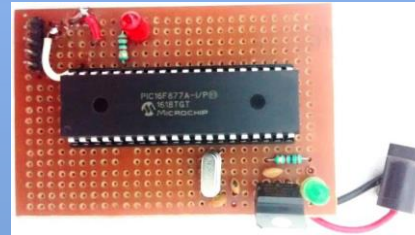
Wouldn't it be nice..?



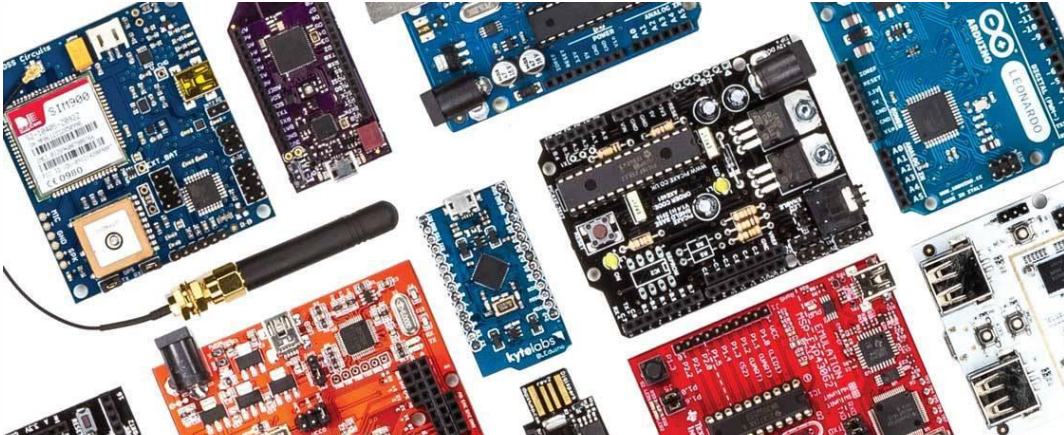
STEP 3: BLINKING LED WITH A MICROCONTROLLER

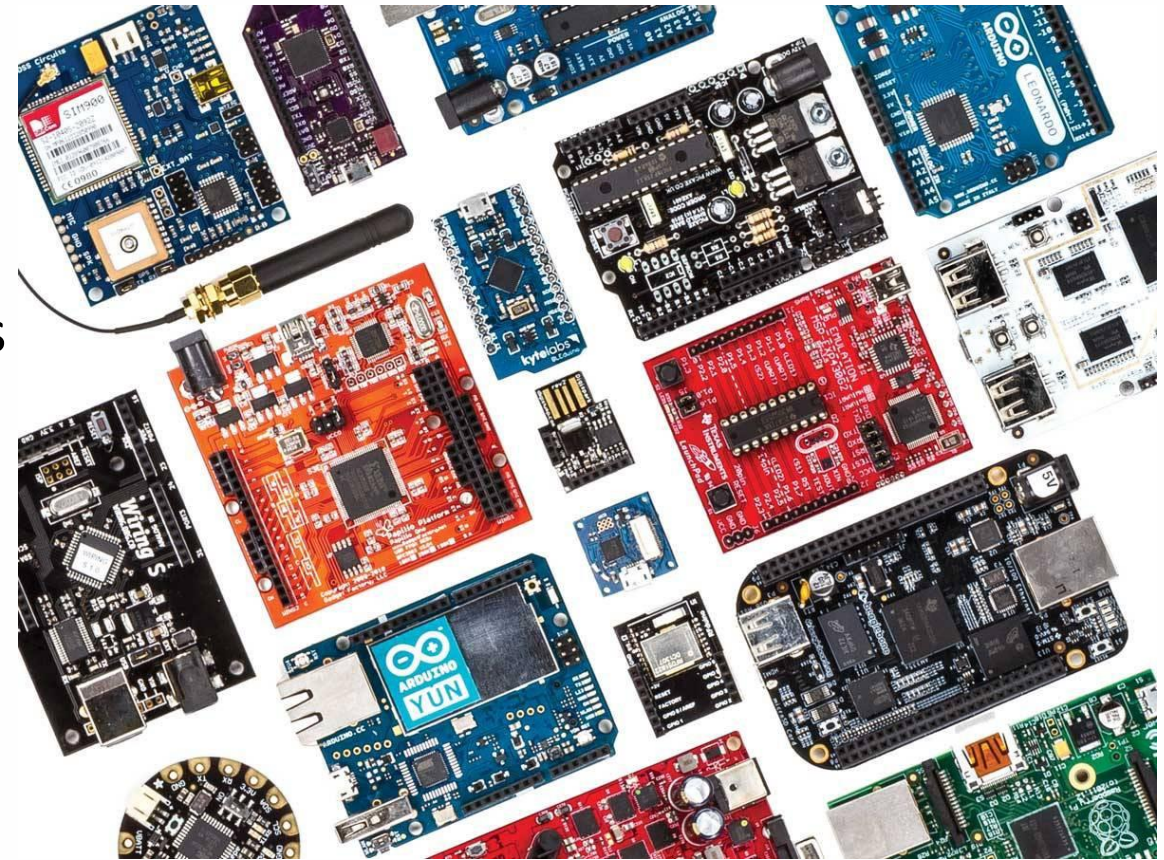
- Construct circuit
- Compile program
- Flash the program

Wouldn't it be nice..?



WHAT IS A DEVELOPMENT BOARD

- A printed circuit board designed to facilitate work with a particular microcontroller.
 - Typical components include:
 - a) power circuit
 - b) programming interface
 - c) basic input; usually buttons and LEDs
 - d) I/O pins
- 
- A collage of various microcontroller development boards, including Arduino Uno, Raspberry Pi, and others, illustrating the components of a microcontroller board. The boards are shown in different colors (blue, red, black, purple) and are populated with various electronic components like chips, capacitors, and connectors. Some boards have labels like 'Arduino Uno', 'Raspberry Pi', and 'kyteblo'. A black cable with a gold connector is also visible.



WHAT IS THE ARDUINO?

One of the most popular development boards in the world

The word “Arduino” can mean 3 things

A physical piece
of hardware



A programming
environment

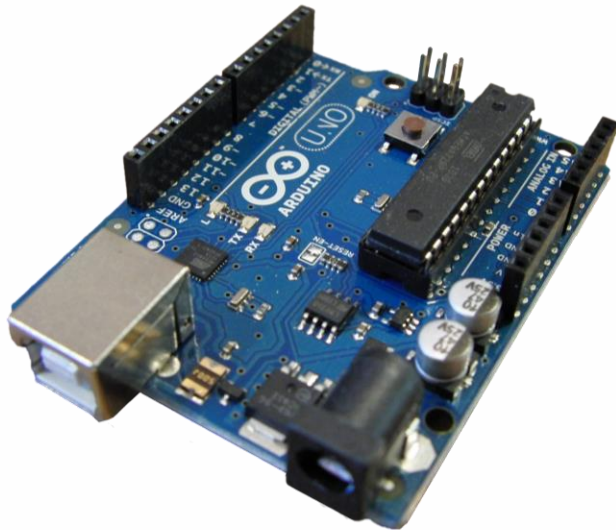


A community
& philosophy



ARDUINO

- **Open-source electronic prototyping platform** based on flexible, easy-to-use hardware and software intended to make the application of interactive objects or environments more accessible.
- **Tons of information on Arduino**
- Wide variety of **Arduino boards**, development **platforms**, **software**, **applications**, etc.,...
- The basic board is called Arduino Uno



- Arduino can
 - **sense the environment** by receiving input from a variety of sensors
 - **affect its surroundings** by controlling lights, motors, and other actuators.
- It is not meant for high-performance processing

WHAT IS ARDUINO USED FOR?

- a) Physical Computing projects / research
- b) Interactive Installations
- c) Rapid prototyping
- d) When you wish to move beyond the traditional Mouse, Keyboard and Monitor to develop novel and custom interactions in your project work.

ARDUINO BOARD VERSIONS

- There is a wide variety of different Arduino boards (18+).
- Among the most populars:



LilyPad Arduino

- Wearables
- Flexible
- Fabric applications



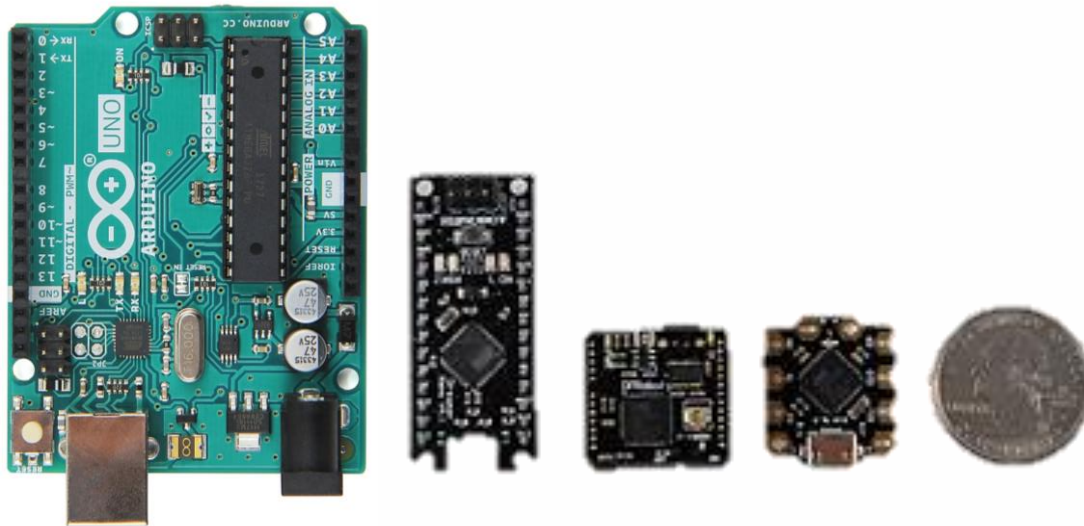
Arduino Due

- ARM Cortex-M3 -> Faster!
- 32 bits vs 8 bit
- Larger scale applications



ARDUINO VERSIONS

- There is a wide variety of different Arduino boards (18+).
- Among the most populars:



Arduino Beetle

- Current smallest form factor
- Arduino Uno based board
- Bluetooth 4.0
- MicroUSB

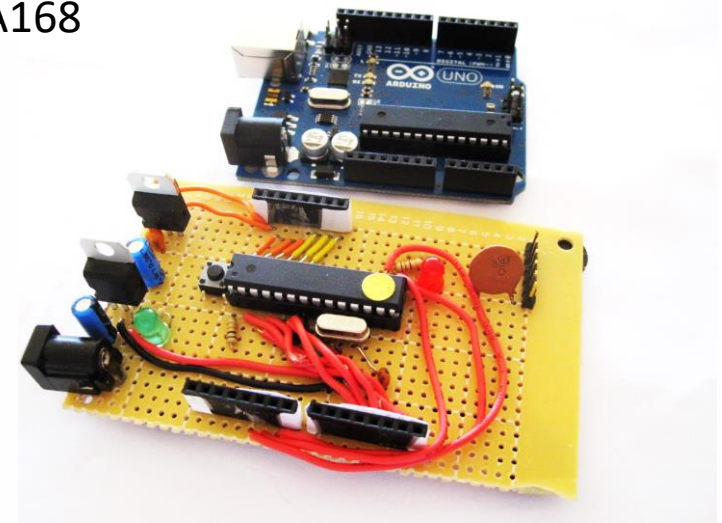
left to right: Arduino UNO, Nano, Nova and Beetle

SOME PROPERTIES OF ARDUINO

- **Inexpensive** (compared to other platforms)
 - The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50.
- It **can communicate with a computer** via serial connection over USB and via shield with Ethernet, GSM, Zigbee, Bluetooth, etc.
- It can be **powered from USB or standalone DC power** (batteries also).
- It **can run standalone** from a computer (chip is programmable) and it has memory (a small amount).
- **Cross-platform** - The Arduino IDE software runs on Windows, Macintosh OSX, and Linux operating systems.
- It can work with both Digital and Analog electronic signals. Sensors and Actuators.

SOME PROPERTIES OF ARDUINO (CONT.)

- **Simple, clear programming environment**
 - Arduino programming environment is easy for beginners, yet flexible enough for advanced users.
- It is **Open Source**, in terms of both Hardware and Software.
 - Software can be extended by experienced programmers.
 - Arduino can be expanded through C++ libraries
 - Arduino is based on the AVR C programming language. Programming in AVR C offers more advance features, but more difficult to learn.
 - Extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168
 - Plans published under a Creative Commons license
 - Designers can make their own version of the module, extending it and improving it.
 - Even relatively inexperienced users can build the breadboard version of the module.



WHAT CAN ARDUINO DO?

Sensors (to sense stuff)

- Push buttons, touch pads, tilt switches.
- Variable resistors (eg. volume knob / sliders)
- Photoresistors (sensing light levels)
- Thermistors (temperature)
- Ultrasound (proximity range finder)
- Etc....

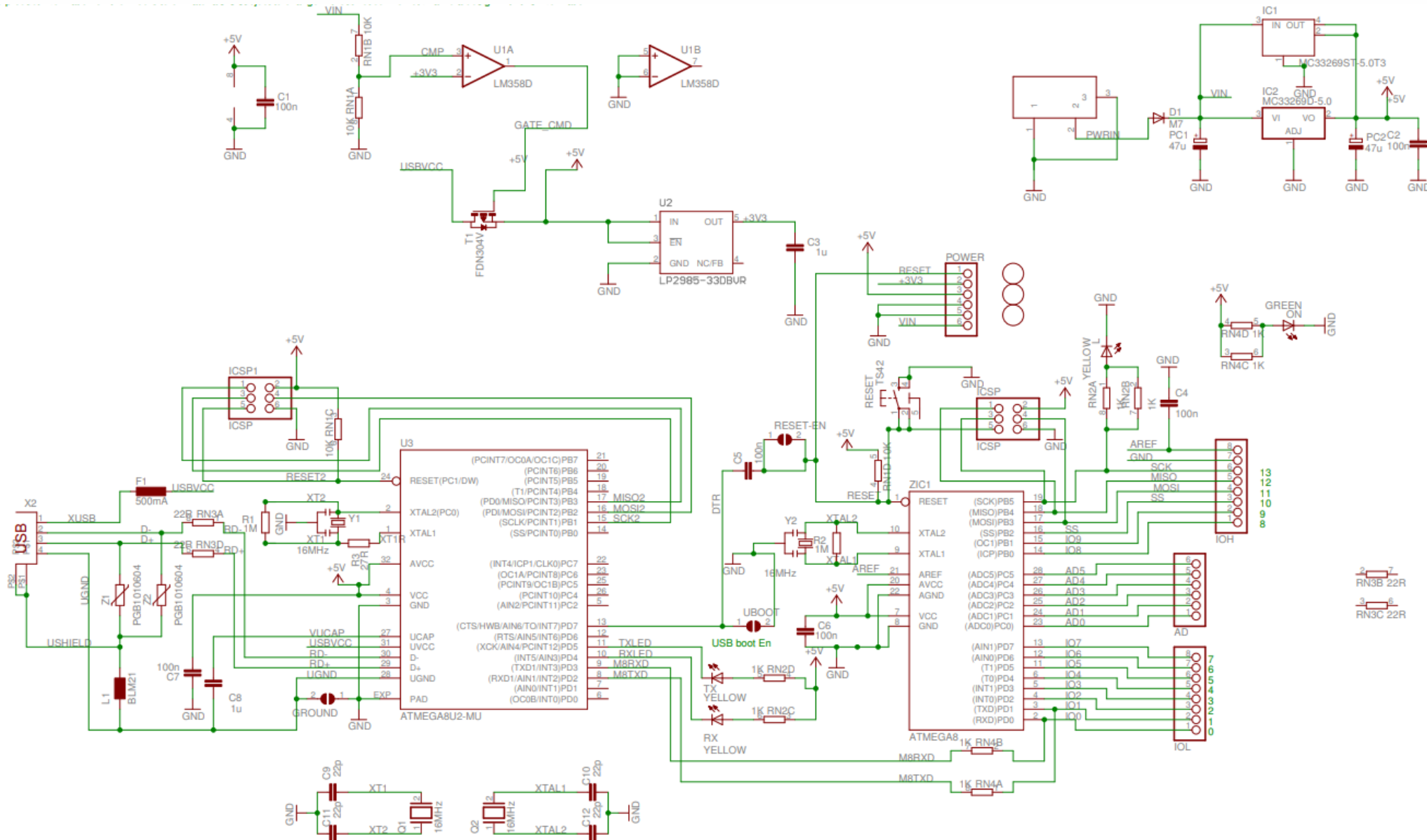
Actuators (to do stuff)

- Lights, LED's
- Motors
- Speakers
- Displays (LCD)
- Etc...

REFERENCES

- Harold Timmis, “**Practical Arduino Engineering**”, Apress, 2011.
 - Free access at [Springer](#)
- Michael Margolis, “**Arduino Cookbook**”, O'Reilly Media, 2011.
 - Free access [here](#)
- Language References (standard instructions)
 - Free access at [Arduino.cc](#)

WHAT IS IT MADE UP OF



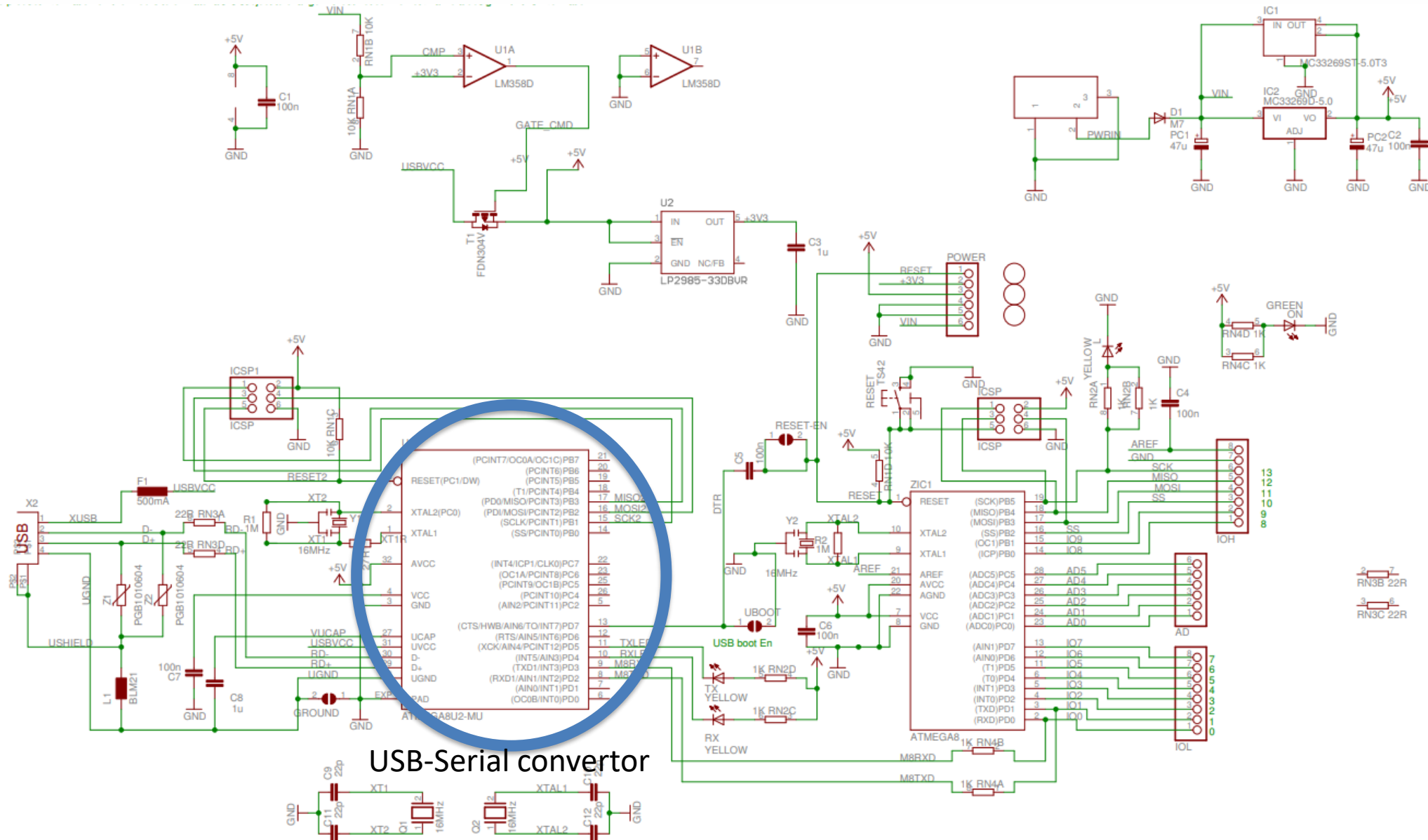
The diagram shows a complex PCB layout for an ATmega8 microcontroller. Key components include:

- Microcontroller:** ATMEGA8U2-MU (U2), ATMEGA8 (U3).
- Power Regulation:** LM358D (U1A, U1B), LP2985-33DBVR (U4), MC33269ST-5.0T3 (IC2).
- USB Interface:** ICSP1, ICSP, USB boot En, USB boot En.
- Reset Circuit:** RESET-EN, RESET, RESET2.
- Other Components:** Various capacitors (C1-C12), resistors (R1-R10), LEDs (TX LED, RX LED, GREEN ON), and connectors (X1, X2, X3, X4).

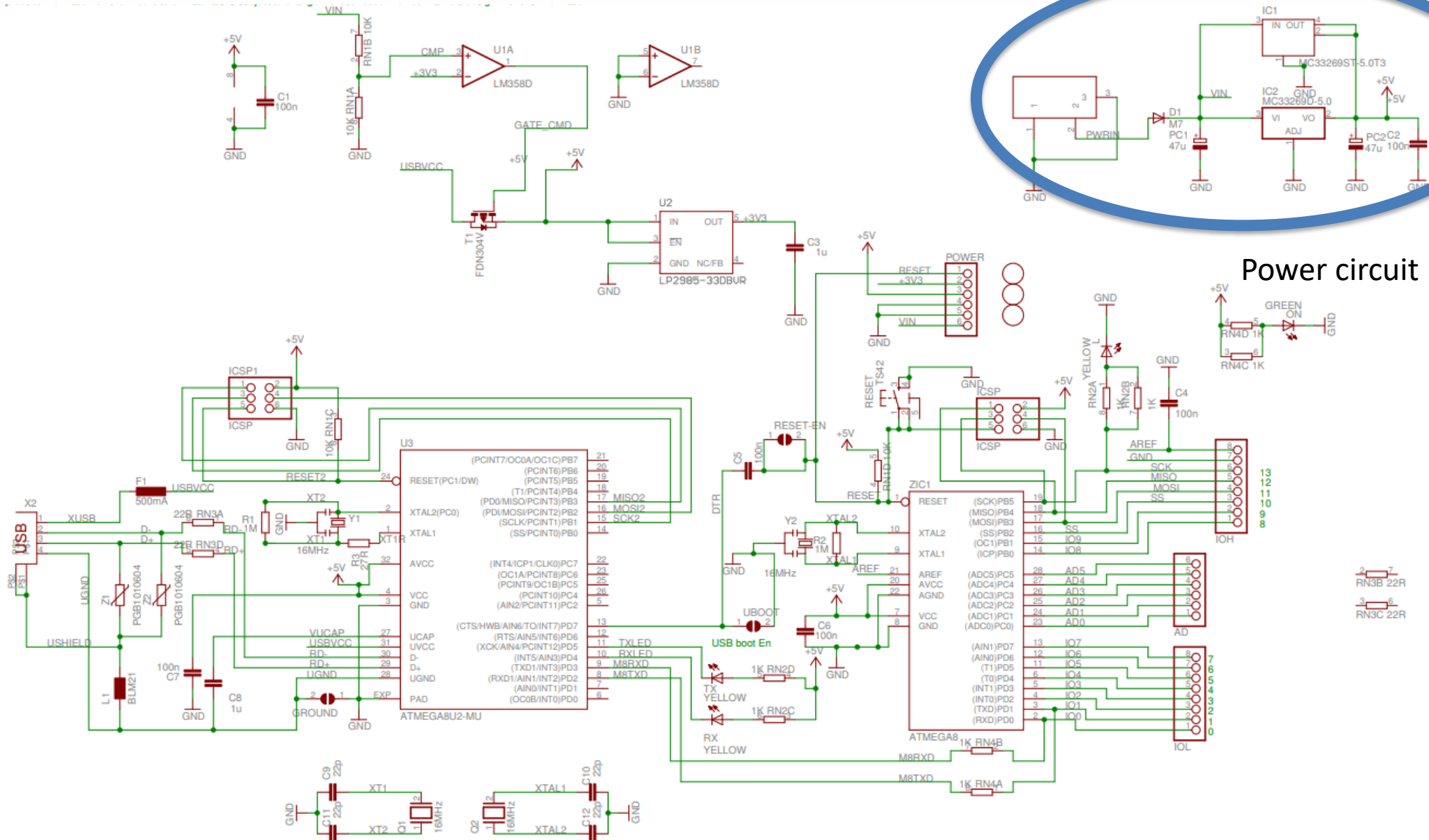
The layout is color-coded: green for power and ground, red for signal, and blue for USB-related components. A large blue oval highlights the microcontroller and its immediate connections.

Micro-controller: ATMEGA8

WHAT IS IT MADE UP OF



WHAT IS IT MADE UP OF



General Purpose Input/Output (GPIO) Pins

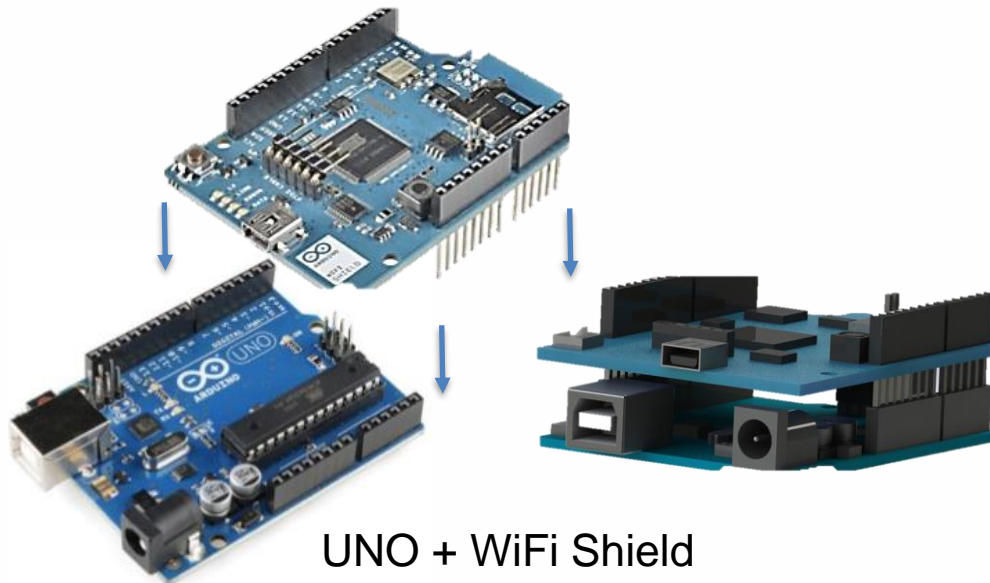
HARDWARE

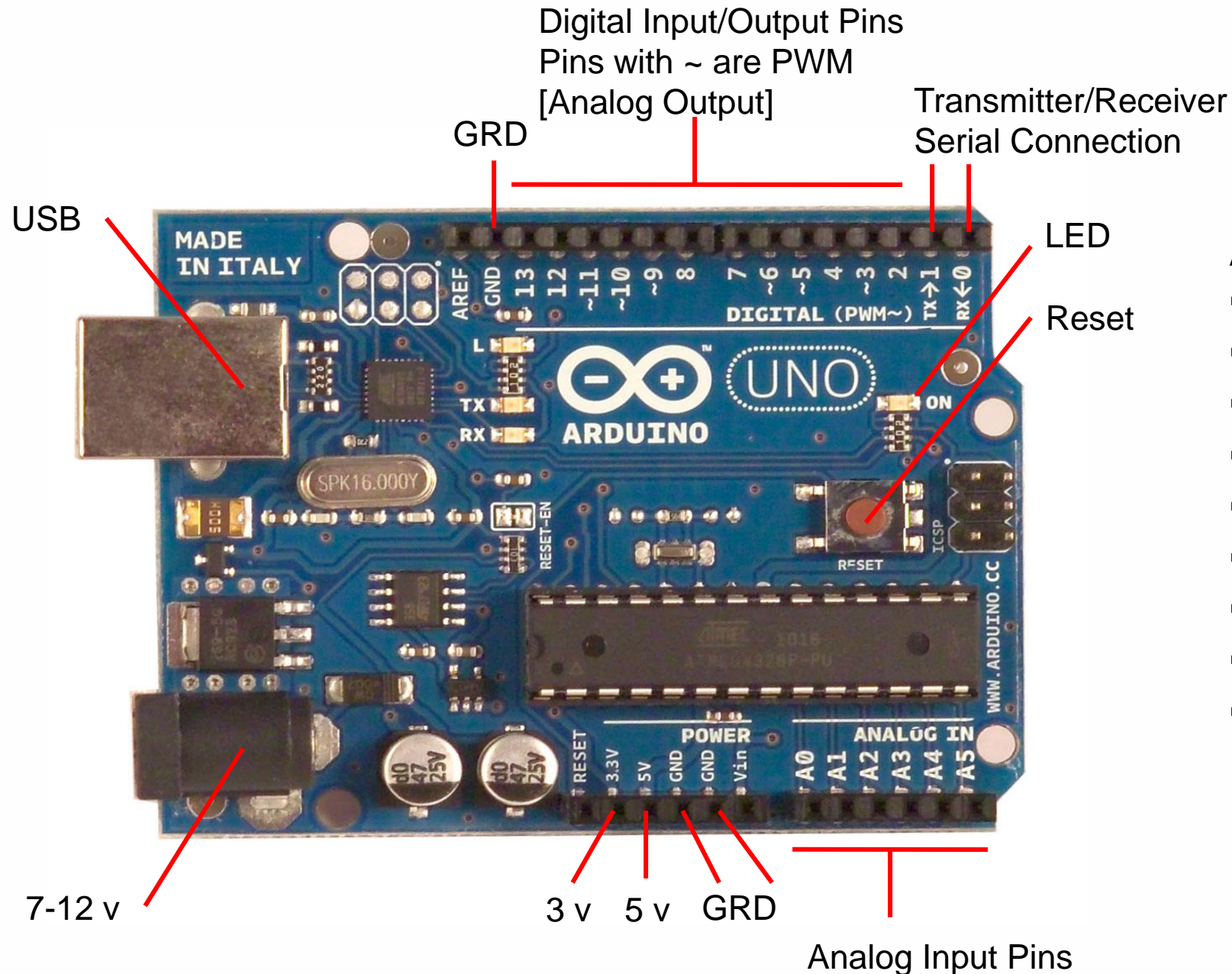
- The basic Arduino board consists of an Atmel 8-bit AVR processor with complementary **components** to facilitate **programming** and **incorporation** into other circuits.
 - Other processors are also used in some Arduino boards, like **ARM** processors.
- Arduino's microcontroller is **pre-programmed with a boot loader**
 - Simplifies uploading of programs to the on-chip flash memory
 - Allows the use of an ordinary computer as the programmer.
- There are many Arduino-compatible, Arduino-pin-compatible and Arduino-derived boards.
 - Some are functionally equivalent to an Arduino and may be used interchangeably.
 - The **Sparkfun Photon RedBoard** is one of them.



SHIELDS

- Connectors in Arduino are exposed in a standard way, allowing the CPU board to be connected to a variety of **interchangeable add-on modules** known as **shields**.
 - Some shields communicate with the Arduino board **directly over various pins**, but many shields are individually **addressable via an [I²C serial bus](#)**, allowing many shields to be stacked and used in parallel.
- Arduino boards make use of shields—printed circuit expansion boards that plug into the normally supplied Arduino pin-headers. Shields can provide **easy sensor/actuator interface, motor controls, GPS, ethernet, LCD display, etc...**



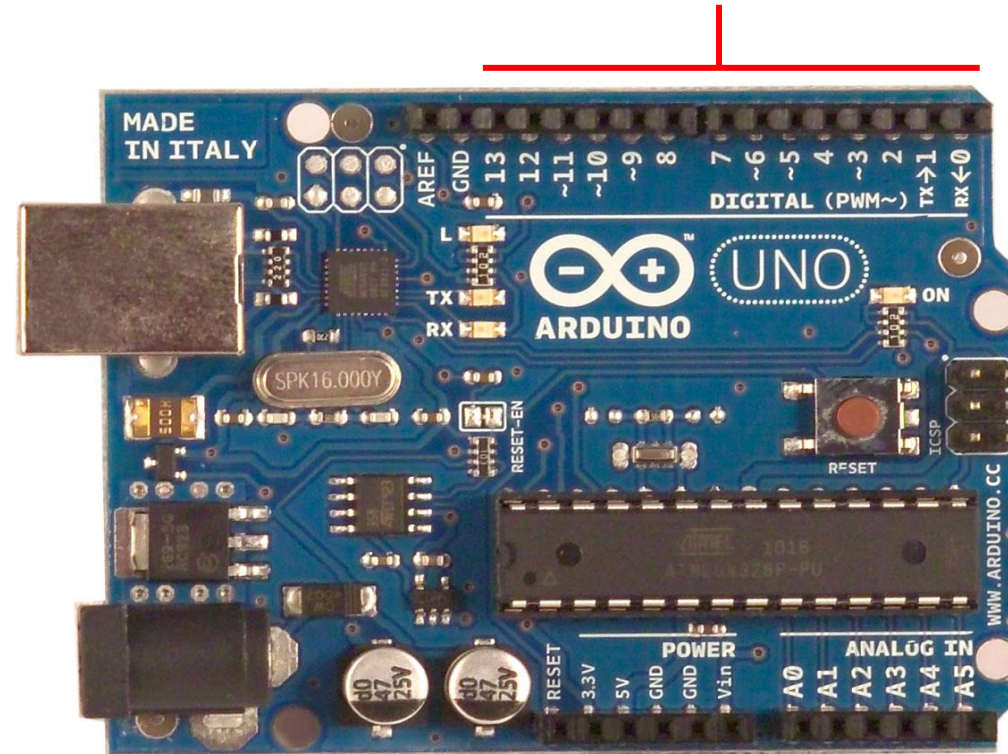
**Arduino UNO Specs:**

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14
- (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA

PINS

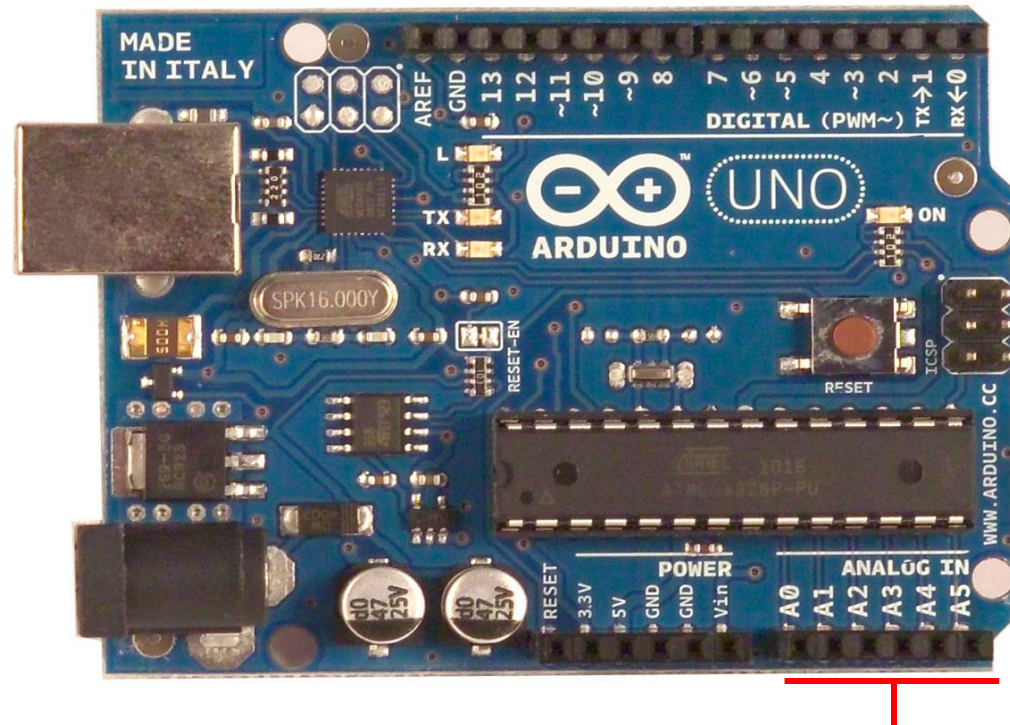
- **14 Digital** IO pins (pins 0–13)
 - These can be **inputs or outputs**, which is specified by the sketch you create in the IDE.

Digital Input/Output Pins



PINS

- 6 **Analog In** pins (pins 0–5)
 - **Dedicated** analogue input pins
 - Convert analog values (i.e., **voltage readings** from a sensor) into a number between **0 and 1023**.

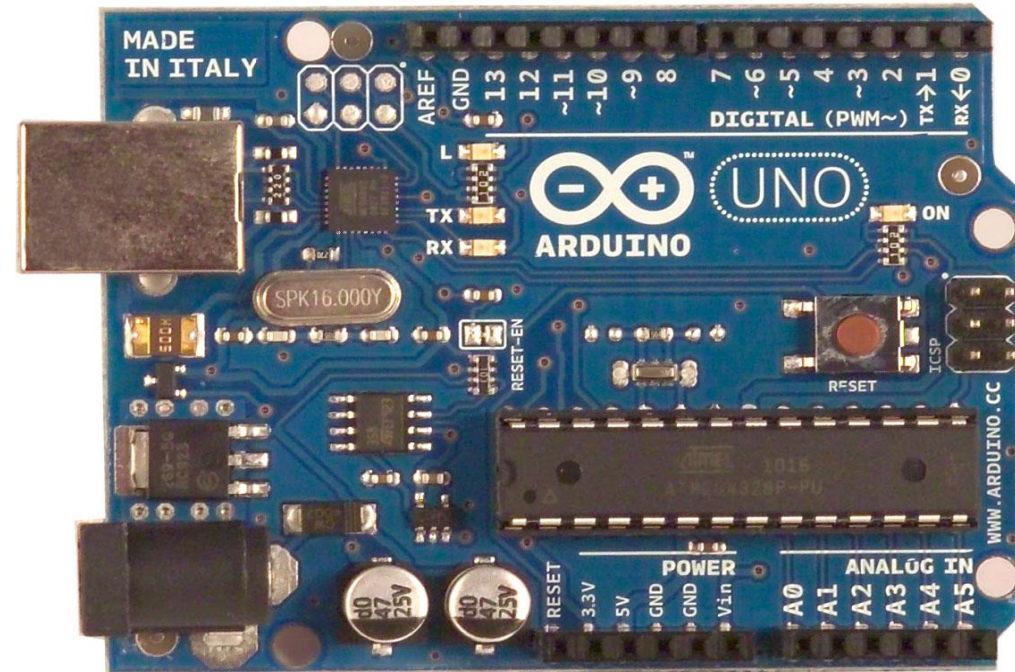


Analog Input Pins

PINS

- 6 **Analog Out** pins (pins 3, 5, 6, 9, 10, and 11)
 - 6 of the **digital pins** that can be **reprogrammed** for analog output using the **sketch** you create in the IDE.

Pins with ~ are PWM [Analog Output]

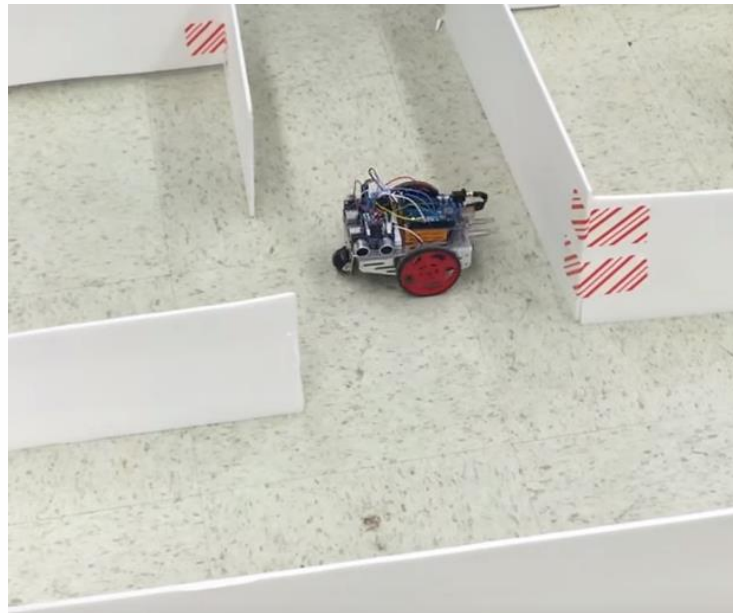


WHAT CAN YOU DO WITH AN ARDUINO?

- Your limit is only your creativity!



Color Classifier



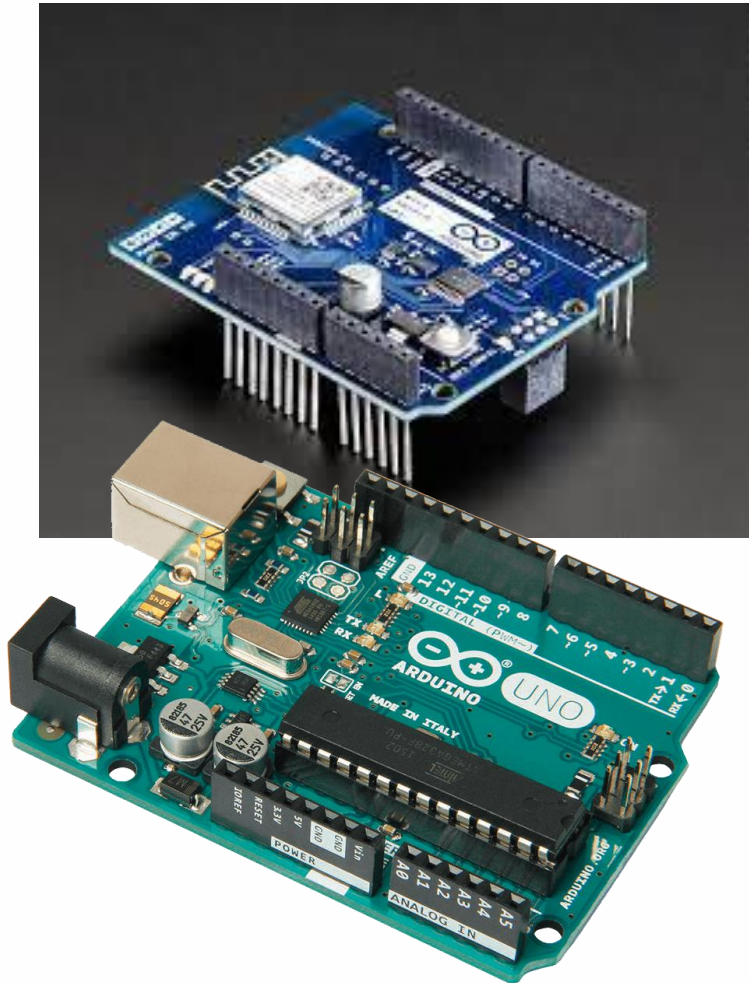
Maze Solver Robot



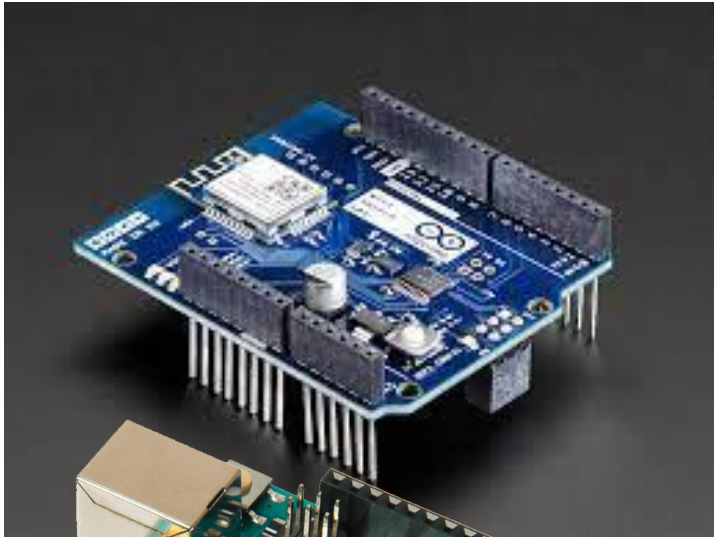
Wall-E Robot

How to use Arduino to talk to the internet?

How to use Arduino to talk to the internet?



How to use Arduino to talk to the internet?

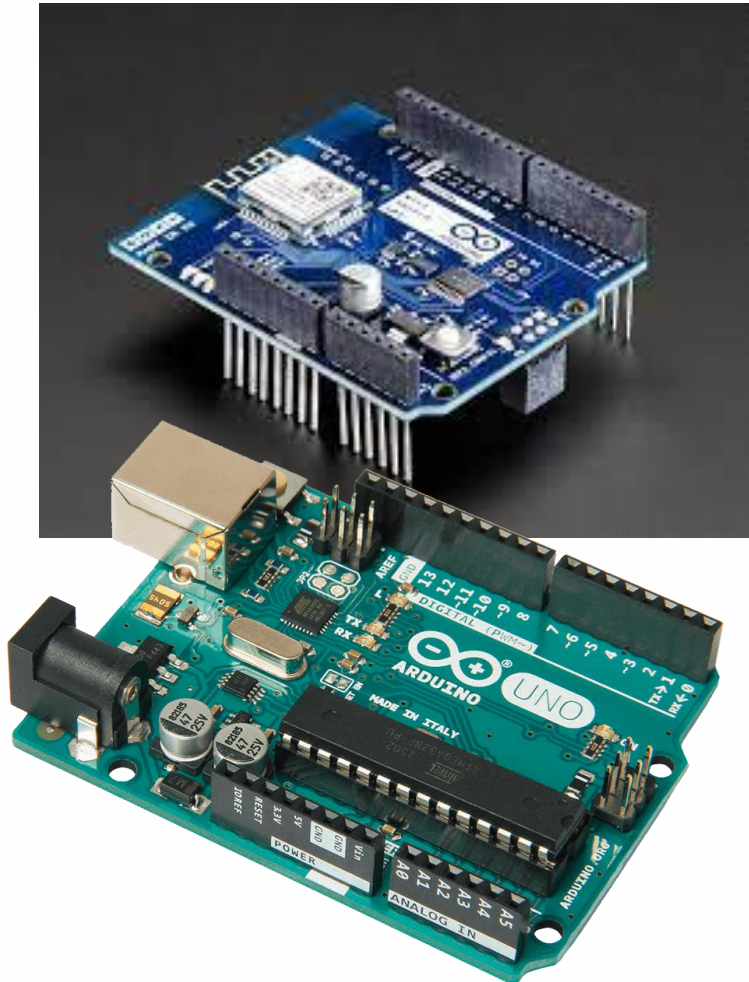


50\$

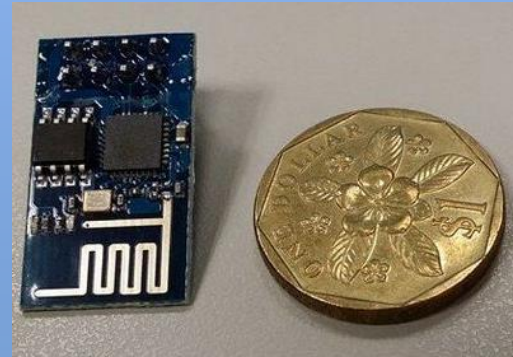


25\$

How to use Arduino to talk to the internet?



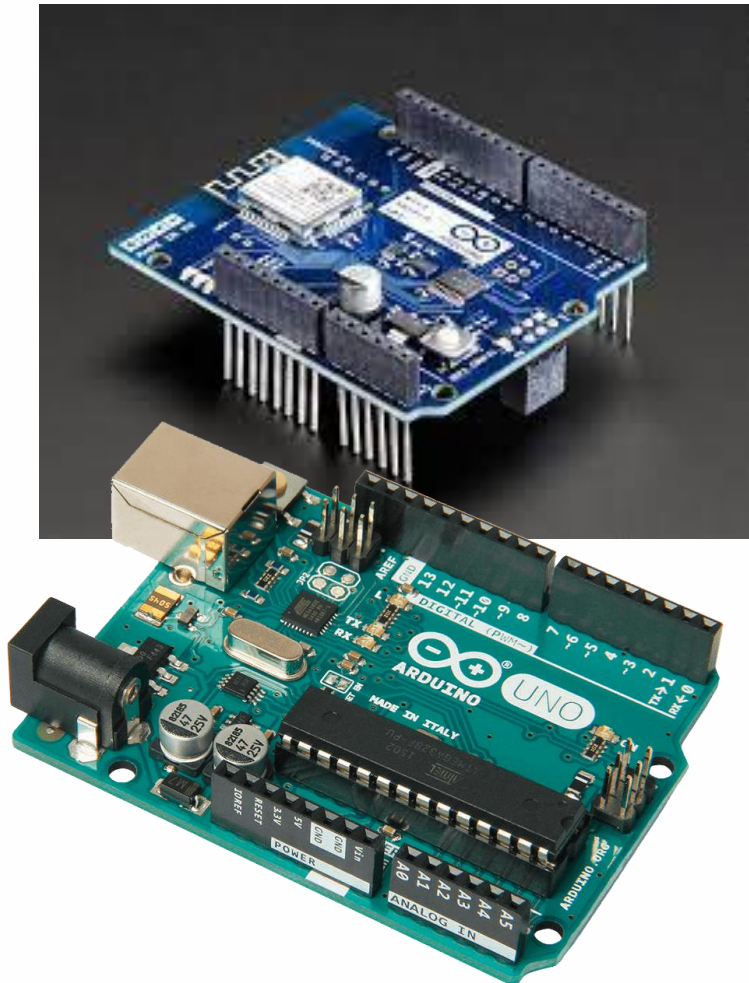
Wouldn't it be nice..?



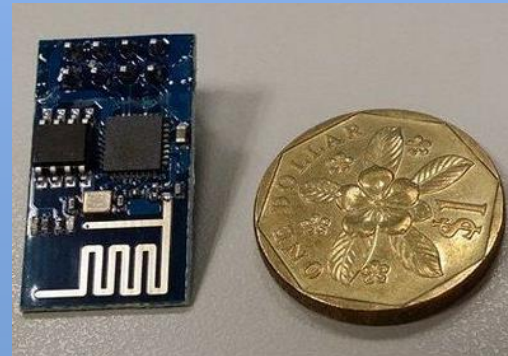
Smaller



How to use Arduino to talk to the internet?



Wouldn't it be nice..?



ESP8266 ESP-01 ESP01 ESP 01 Serial Wireless WIFI Module
Receiver Board For Arduino Raspberry Pi 3 MODULE

★★★★★ 4.8 25 Reviews 97 orders

US \$1.01 ~~US \$1.83~~ -45%

Smaller

Cheaper

