

# Arduino For Total Newbies

## *w/ TV-B-Gone as example project*

Mitch Altman

Chief Scientist, **Cornfield Electronics**, San Francisco, CA

Inventor of **TV-B-Gone** universal remote controls

Co-founder of **3Ware** (successful Silicon Valley startup)

Pioneer of **VR** (in the mid-1980s)

Founding mentor at **HAX** (1<sup>st</sup> and biggest hardware accelerator)

Co-founder of **Noisebridge** (San Francisco hackerspace)

email: [mitch@CornfieldElectronics.com](mailto:mitch@CornfieldElectronics.com)

site: [www.CornfieldElectronics.com](http://www.CornfieldElectronics.com)

twitter: [@maltman23](https://twitter.com/maltman23)

flickr: [maltman23](https://www.flickr.com/photos/maltman23/)

WeChat: [mitchaltman](#)



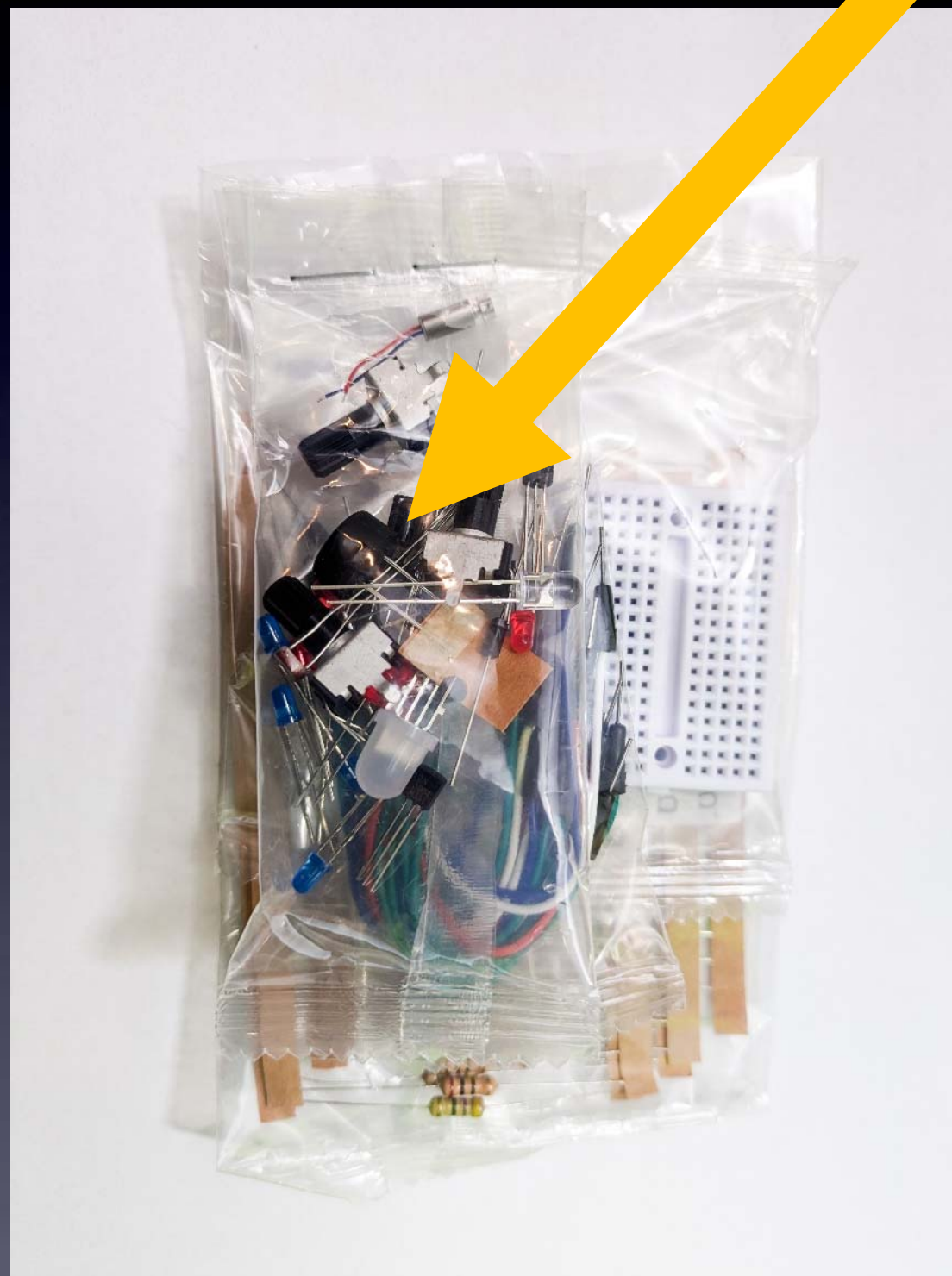
CORNFIELD ELECTRONICS

Bring all of this  
home with you!



U-Do-It-Duino kit

# Stuff!



Parts Pack

**DO NOT  
open this bag  
yet!**



USB-Serial  
cable



# Syllabus

- Intro
- Everything You Need to Know About Electronics
- How to solder / make your own Arduino
- How to Set Up and Use the Arduino Software
- How to Hack Arduino Programs (“Sketches”)
- How to Use Solderless Breadboards
- How to Read a Schematic
- Make a TV-B-Gone Remote Control with your Arduino Clone without soldering

(Don't bring these home)

# Tools





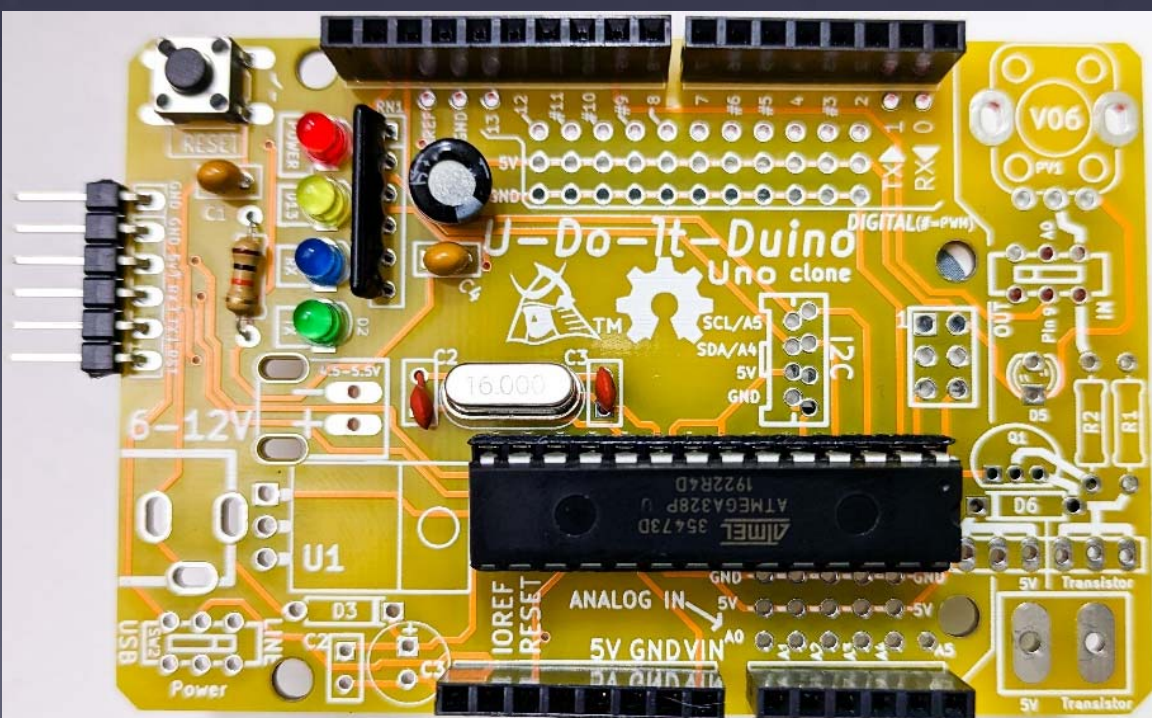
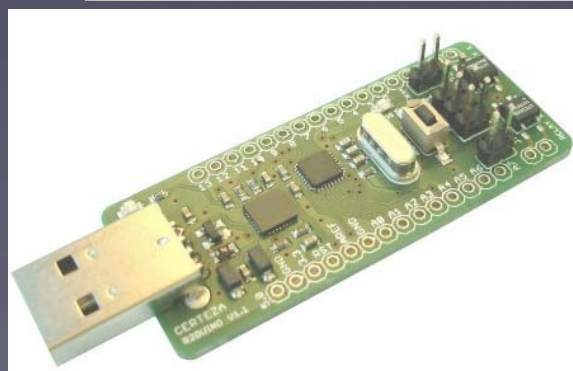
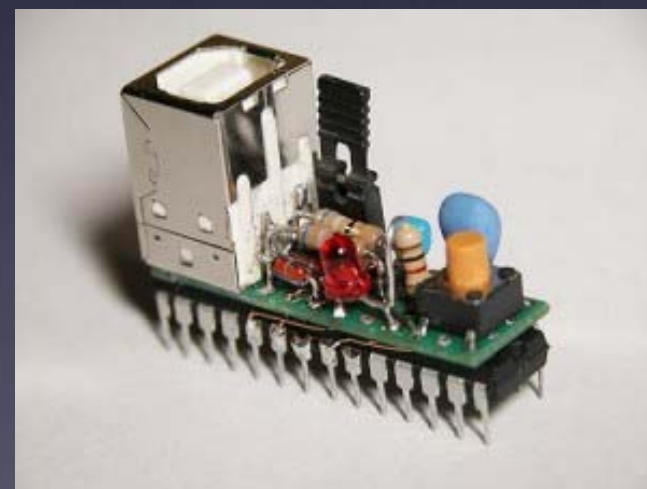
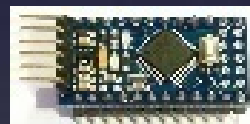
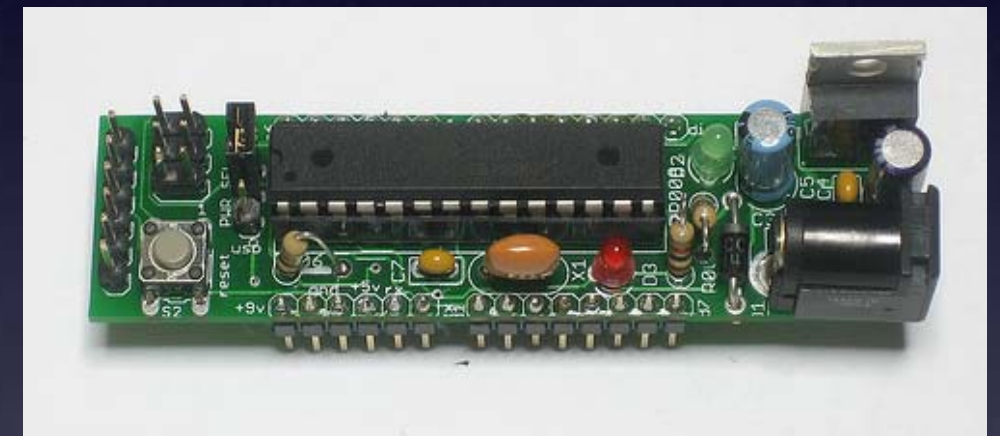
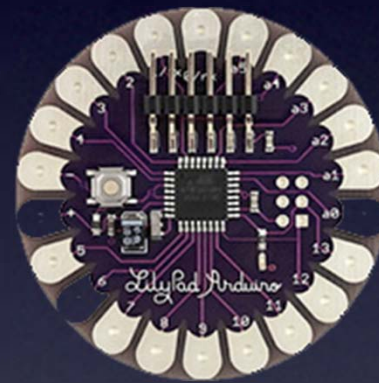
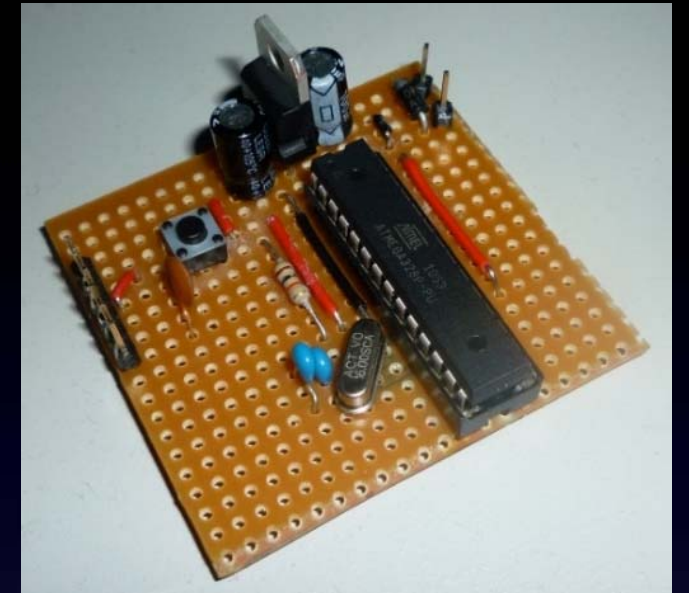
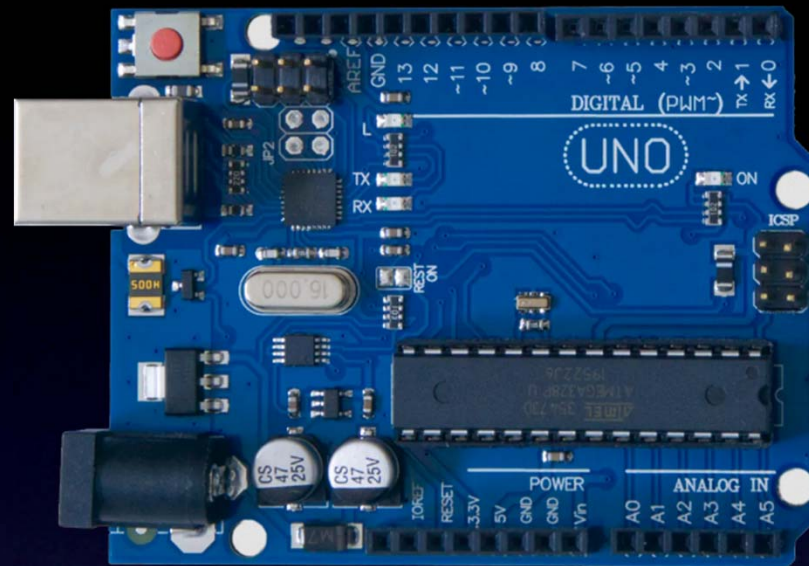
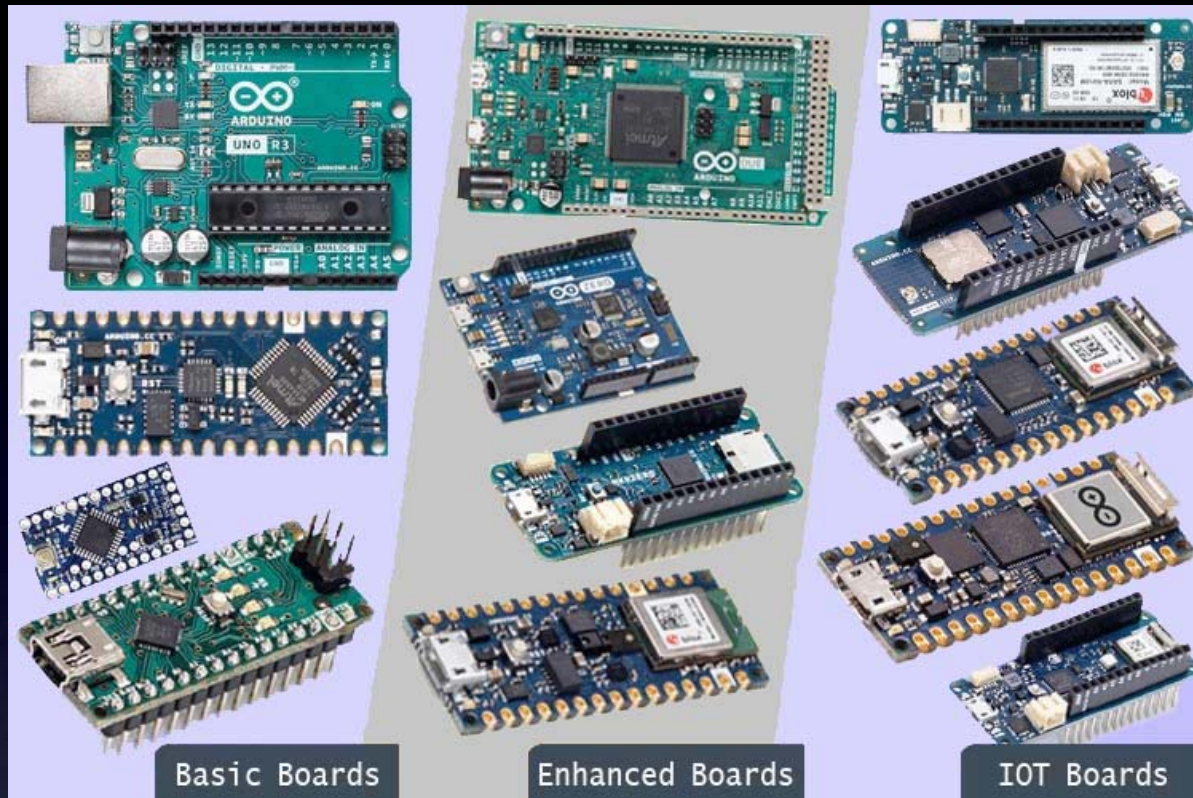
**I have these  
Toolkits  
for sale**

# Tools



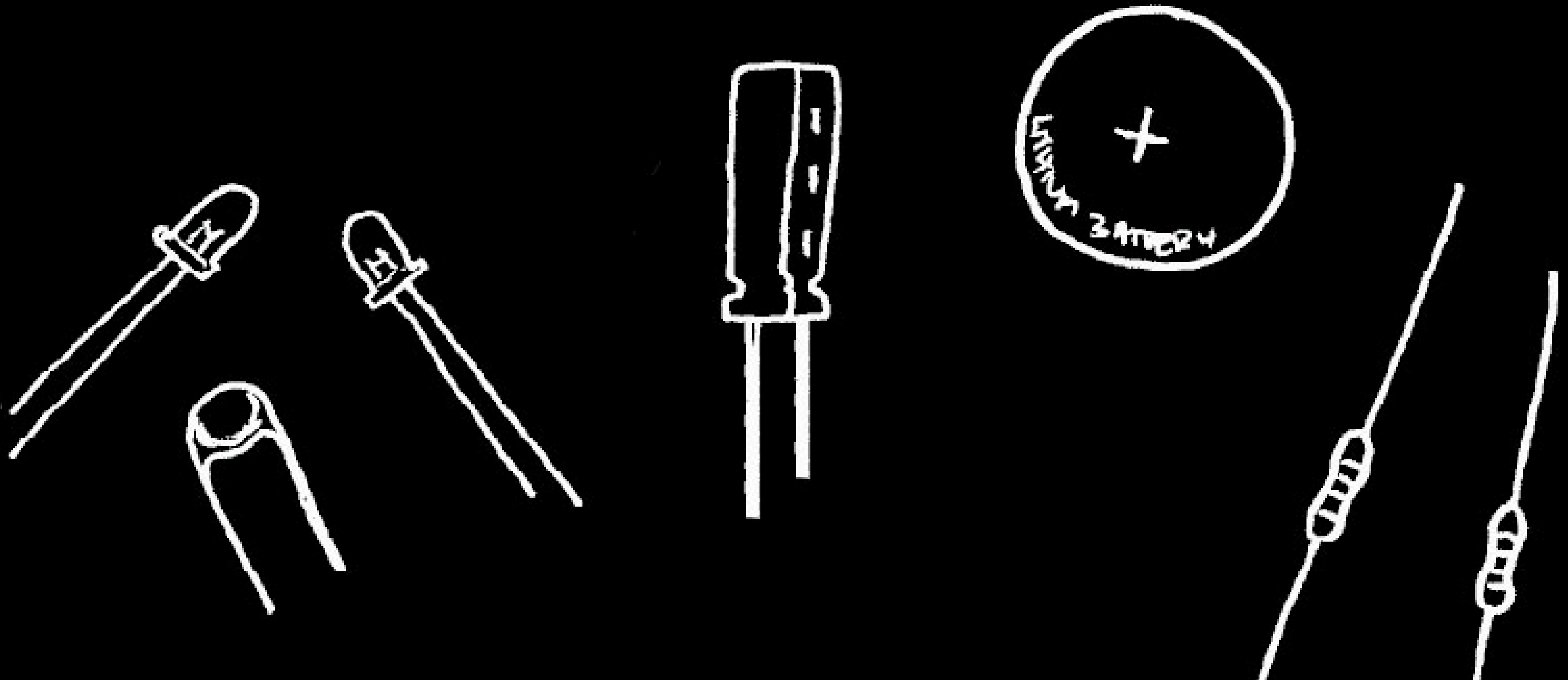


# Intro

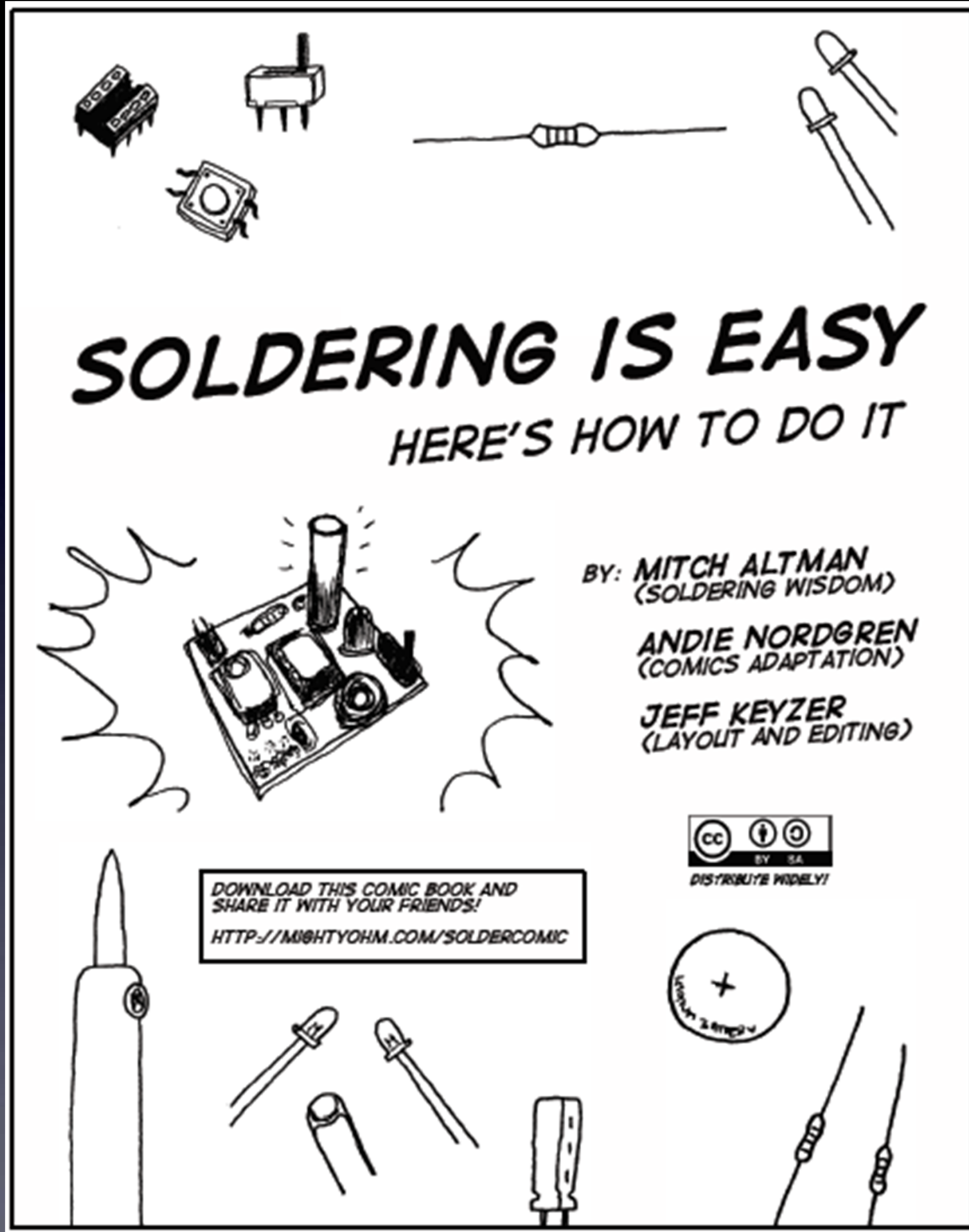




# *Everything You Need to Know About Electronics*



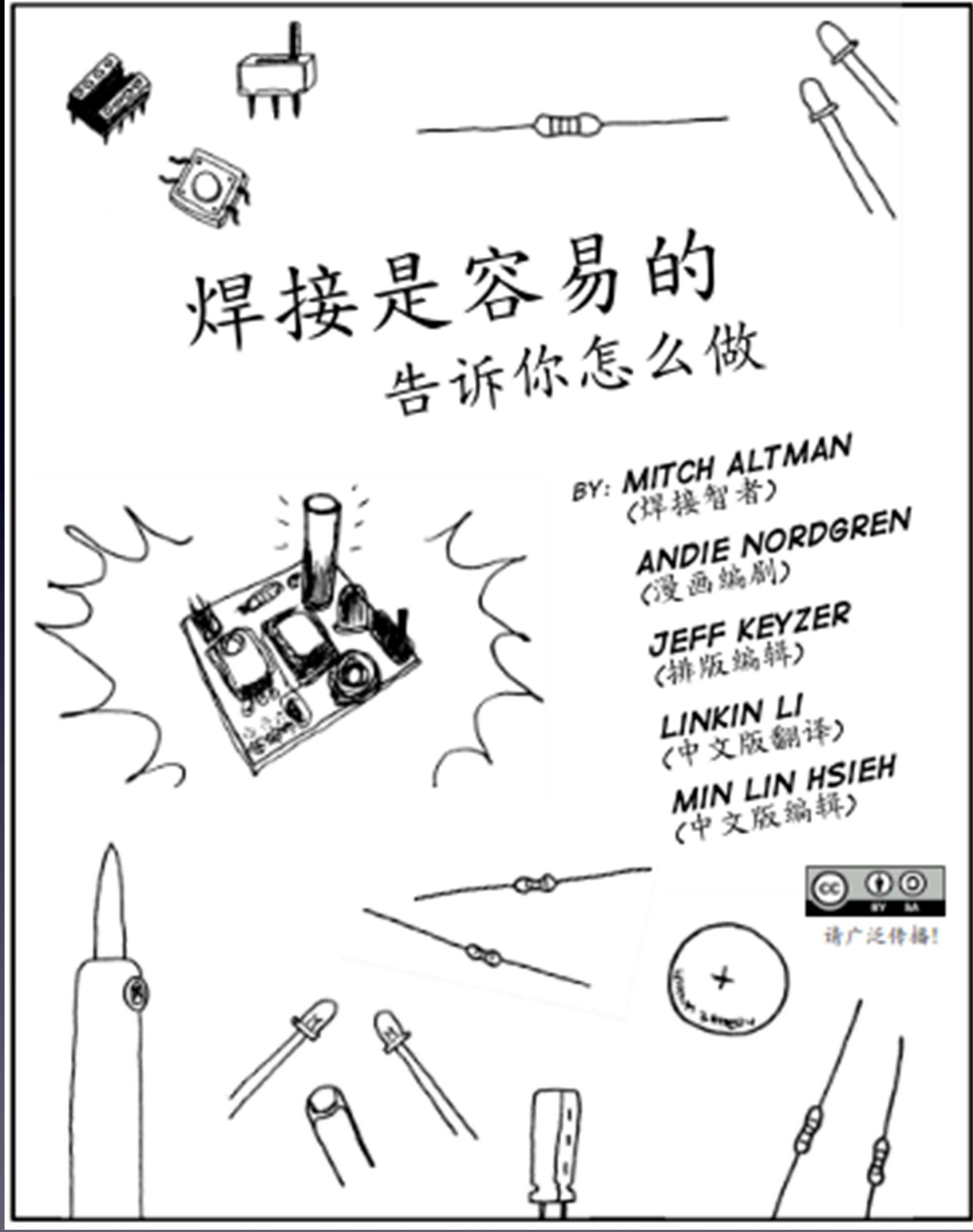
# Learn To Solder



download for free at:  
<http://mightyohm.com/soldercomic>

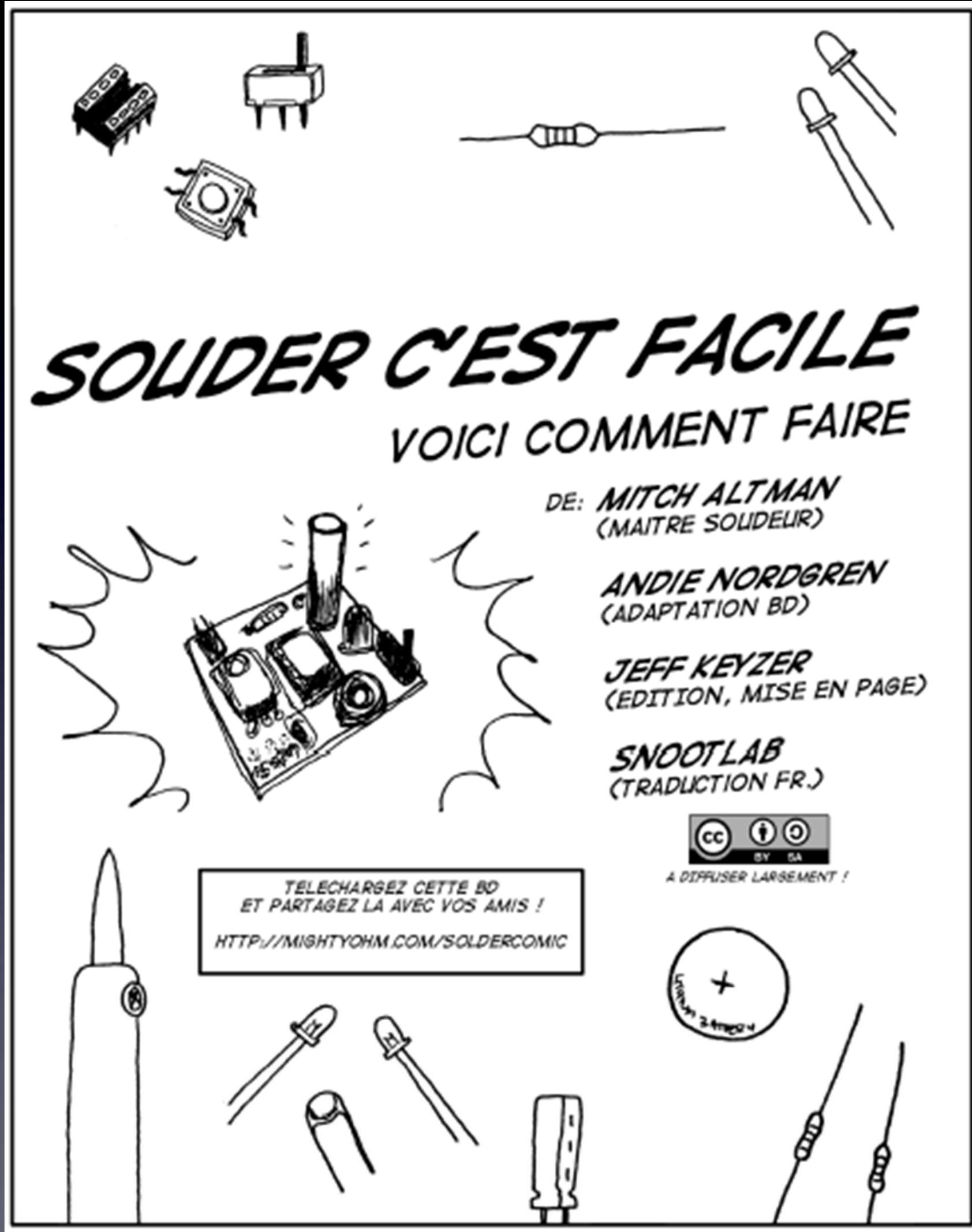


# Learn To Solder



Download in the language of your choice for free at:  
<http://mightyohm.com/soldercomic>

# Learn To Solder



Download in the language of your choice for free at:  
<http://mightyohm.com/soldercomic>



# Learn To Solder



Download in the language of your choice for free at:  
<http://mightyohm.com/soldercomic>

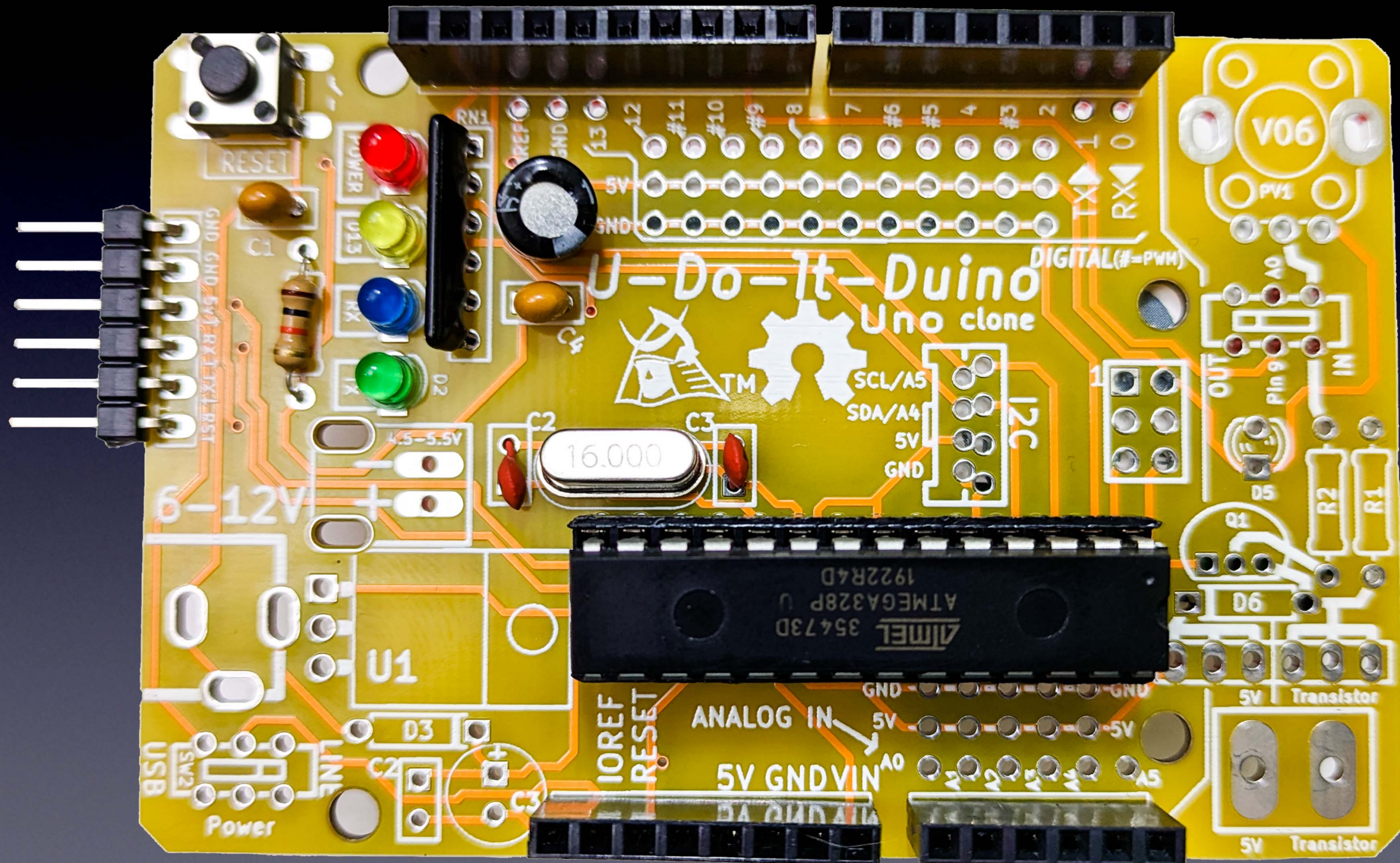
# Learn To Solder



Download in the language of your choice for free at:  
<http://mightyohm.com/soldercomic>



# Solder Your Arduino Clone

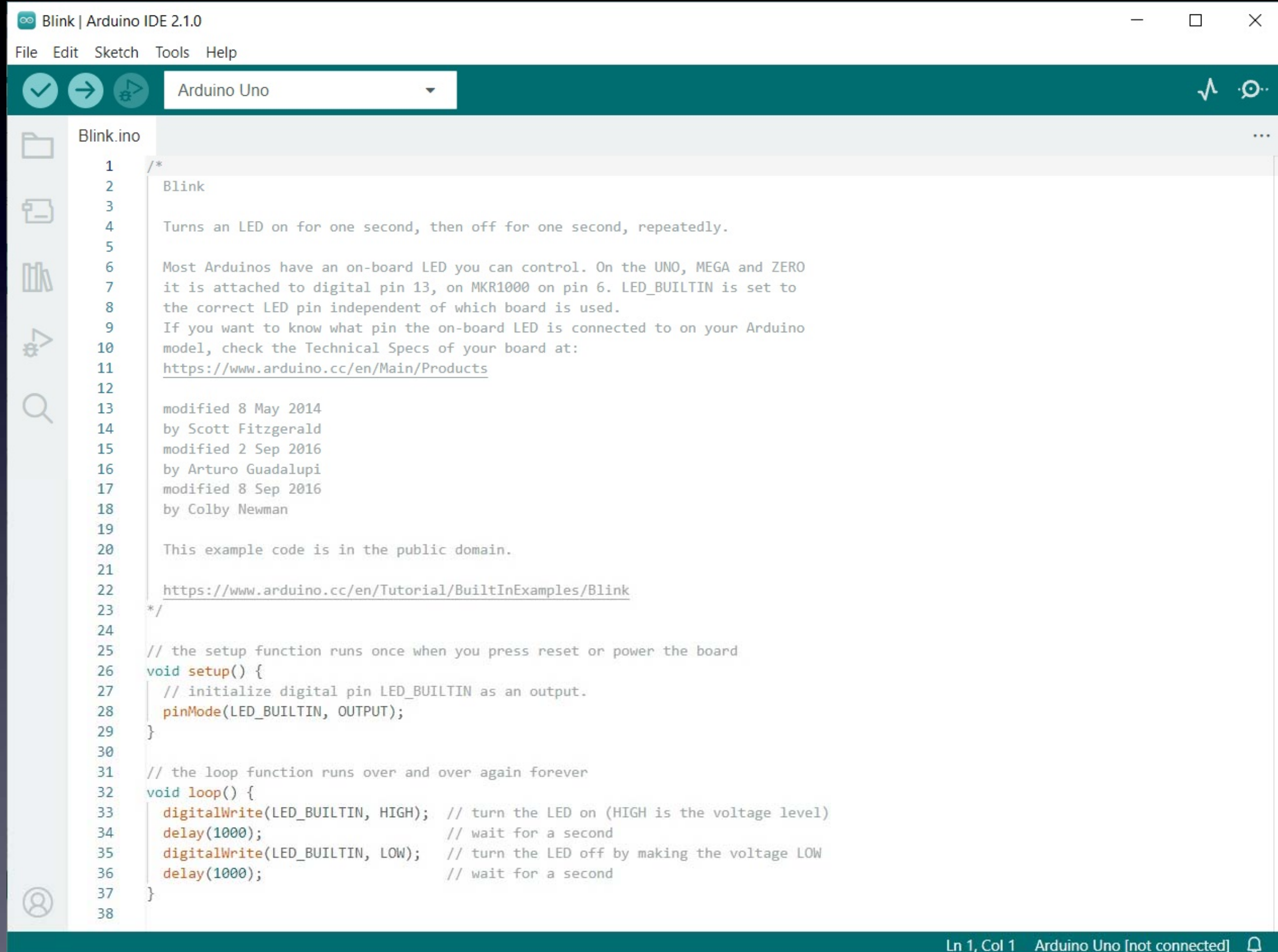




# How to Set Up and Use the Arduino Software



# How to Hack Arduino Programs (“Sketches”)



The screenshot shows the Arduino IDE 2.1.0 interface. The title bar reads "Blink | Arduino IDE 2.1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for checking, running, and uploading code, along with a dropdown menu currently set to "Arduino Uno". The left sidebar shows a file explorer with "Blink.ino" selected. The main editor area displays the code for the Blink sketch, which includes a multi-line comment describing the LED pin configuration and a C++ code block with `setup()` and `loop()` functions. The status bar at the bottom indicates "Ln 1, Col 1" and "Arduino Uno [not connected]".

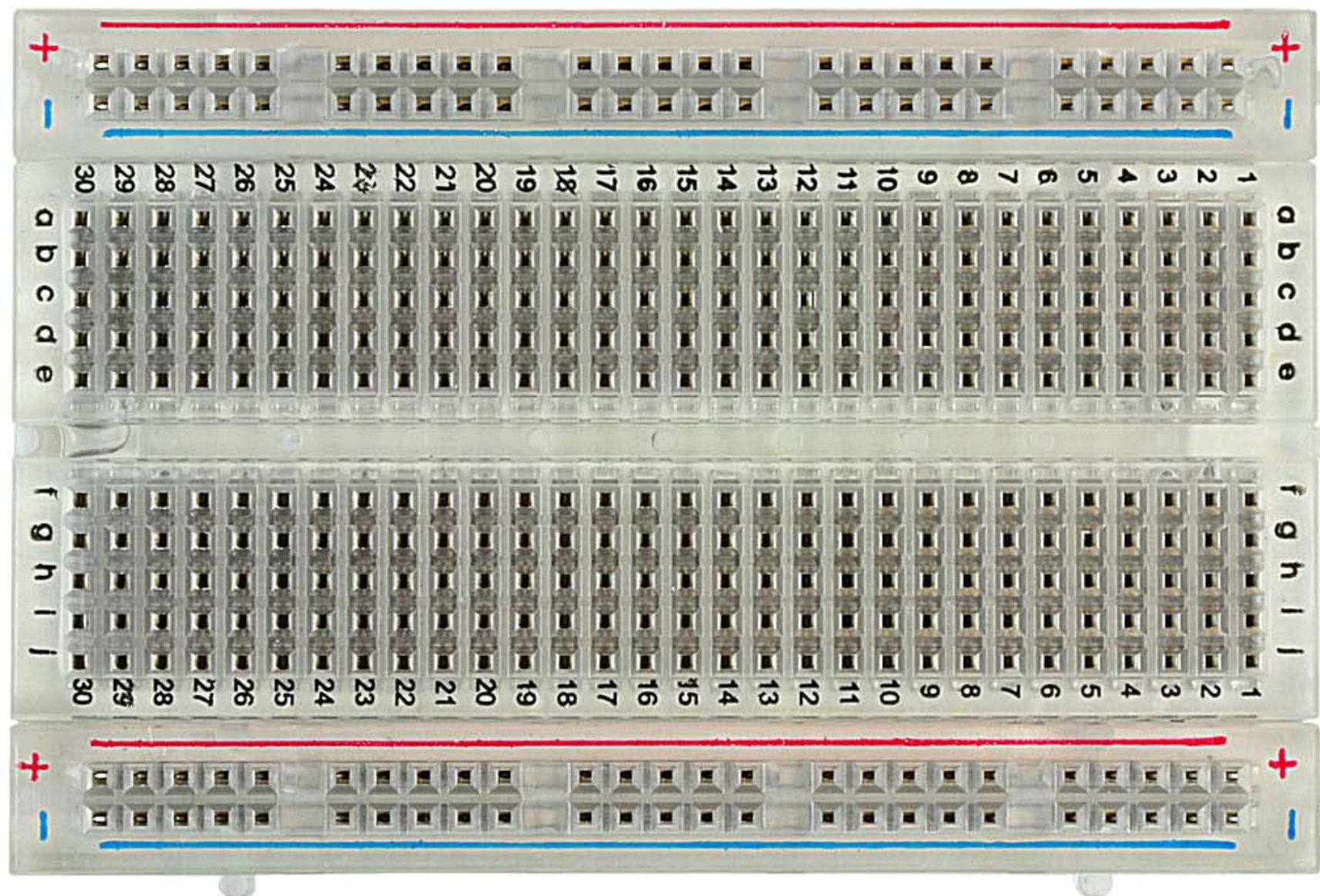
```
1  /*
2   Blink
3
4   Turns an LED on for one second, then off for one second, repeatedly.
5
6   Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
7   it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
8   the correct LED pin independent of which board is used.
9   If you want to know what pin the on-board LED is connected to on your Arduino
10  model, check the Technical Specs of your board at:
11  https://www.arduino.cc/en/Main/Products
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15  modified 2 Sep 2016
16  by Arturo Guadalupi
17  modified 8 Sep 2016
18  by Colby Newman
19
20  This example code is in the public domain.
21
22  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23  */
24
25  // the setup function runs once when you press reset or power the board
26  void setup() {
27    // initialize digital pin LED_BUILTIN as an output.
28    pinMode(LED_BUILTIN, OUTPUT);
29  }
30
31  // the loop function runs over and over again forever
32  void loop() {
33    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34    delay(1000); // wait for a second
35    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36    delay(1000); // wait for a second
37  }
38
```

Ln 1, Col 1    Arduino Uno [not connected]



# How to Use Solderless Breadboards

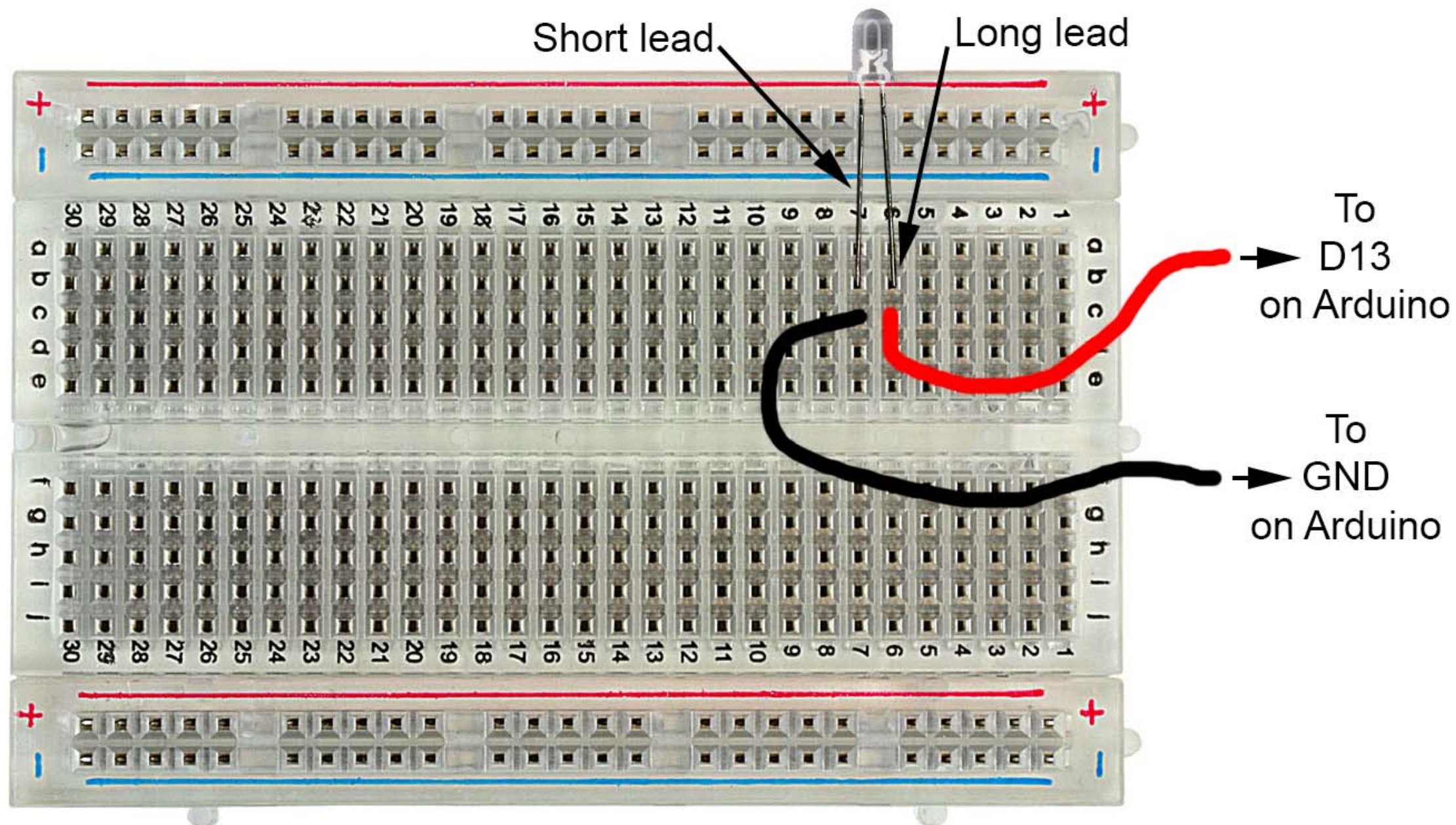
## Solderless Breadboard





# How to Use Solderless Breadboards

## Solderless Breadboard with LED and wires

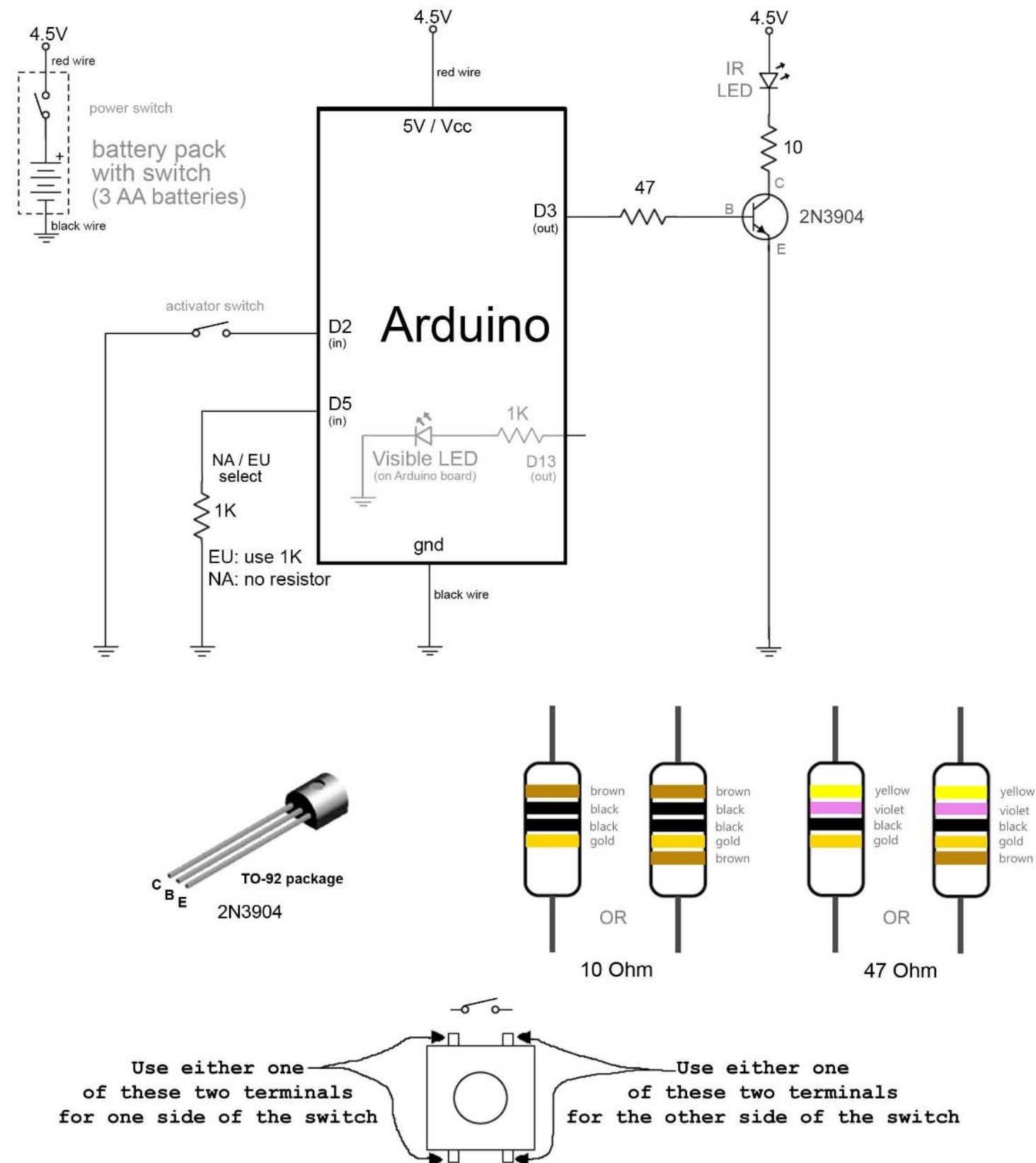


# How to Read a Schematic

## Arduino For Total Newbies

4-Sep-2015

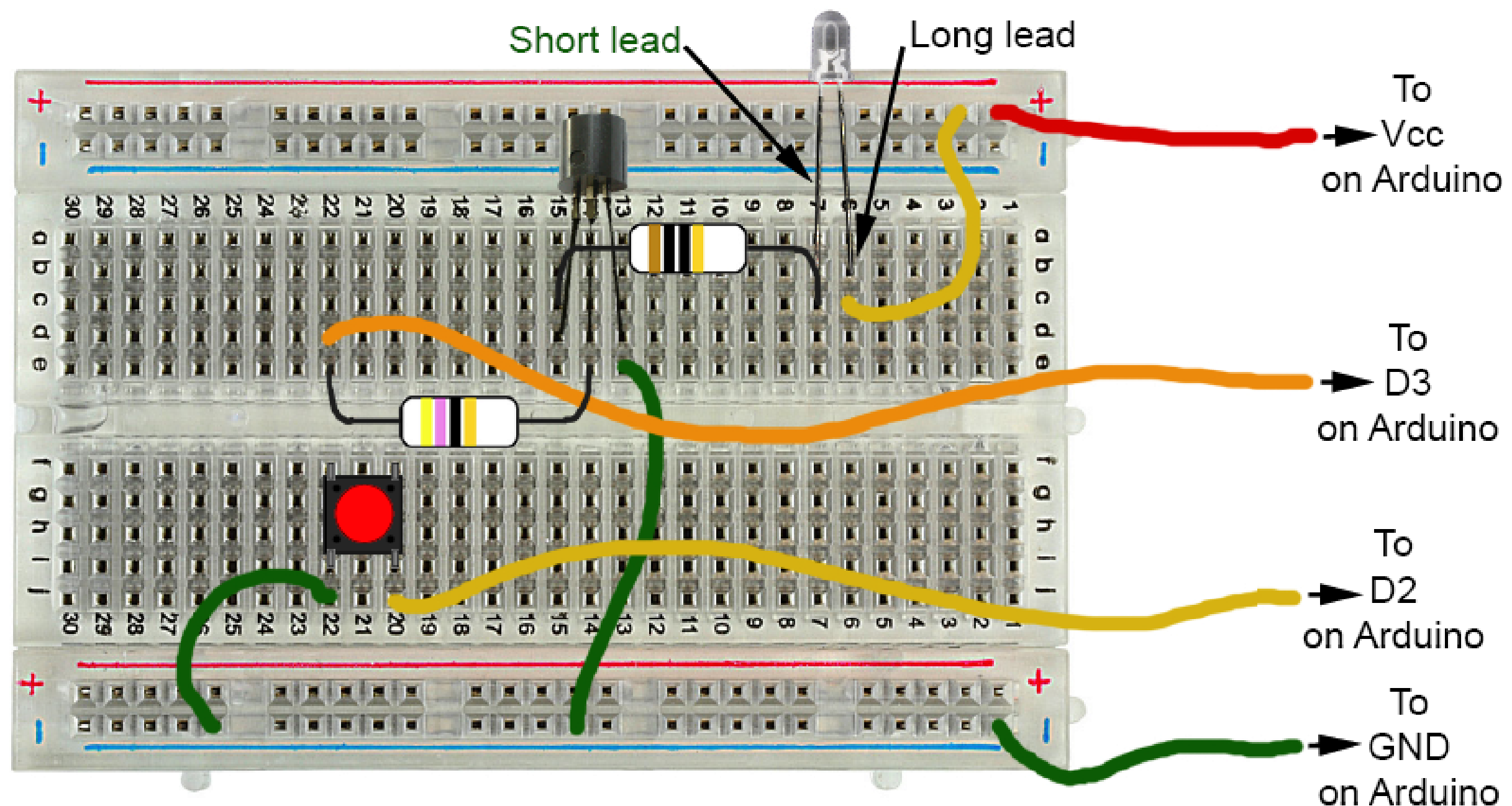
Mitch Altman (original TV-B-Gone hardware and firmware, modified TV-B-Gone Arduino design)  
Limore Fried (firmware modifications, kit design)  
Ken Shirriff (original modifications for Arduino)  
Johannes Schneemann (documentation)

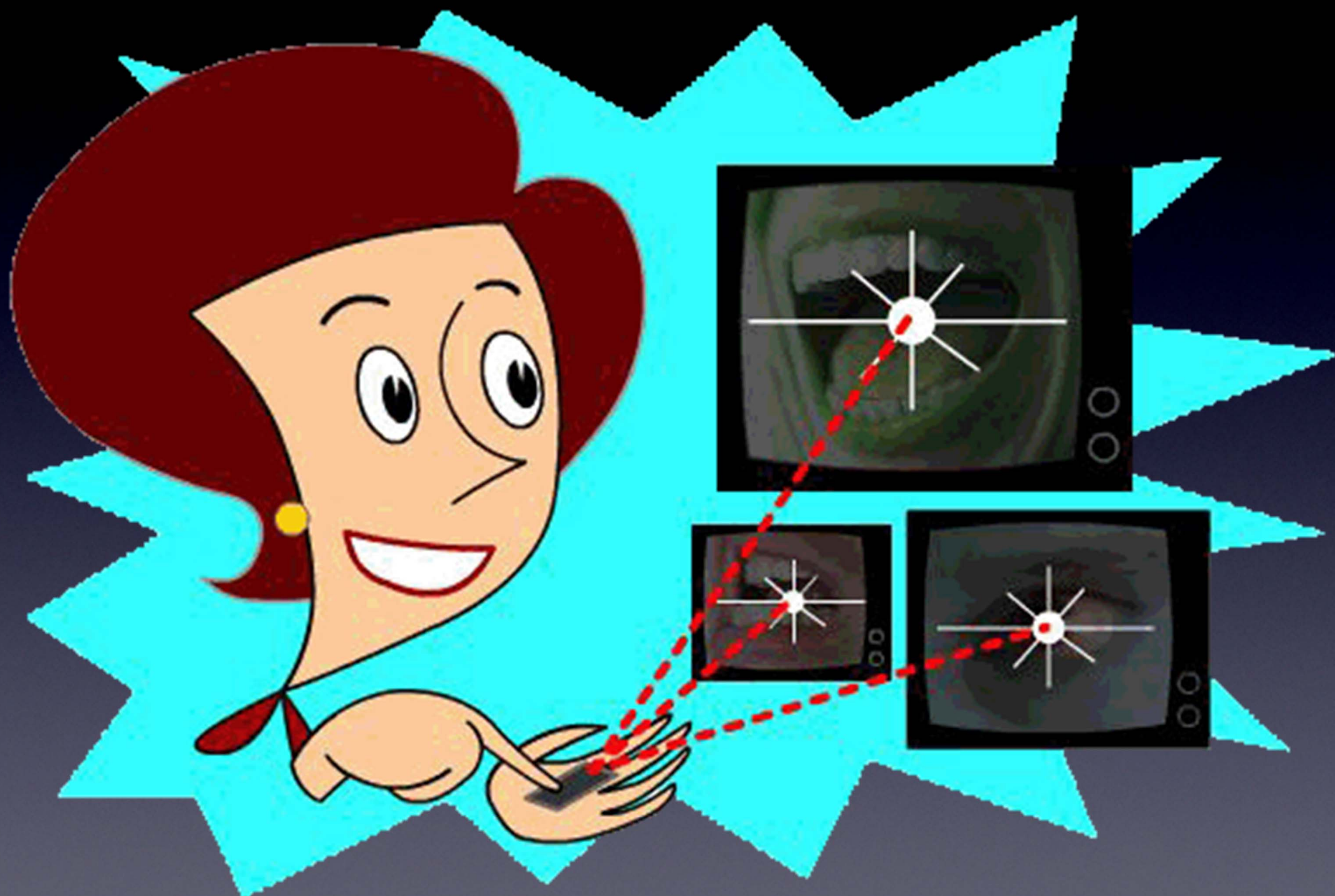




# Make a TV-B-Gone Remote Control with your Arduino Clone without soldering

## Solderless Breadboard with parts & wires for TV-B-Gone







Questions?

# Intro





# Intro



Arduino For Total Newbies Workshop at 30C3, Hamburg Germany

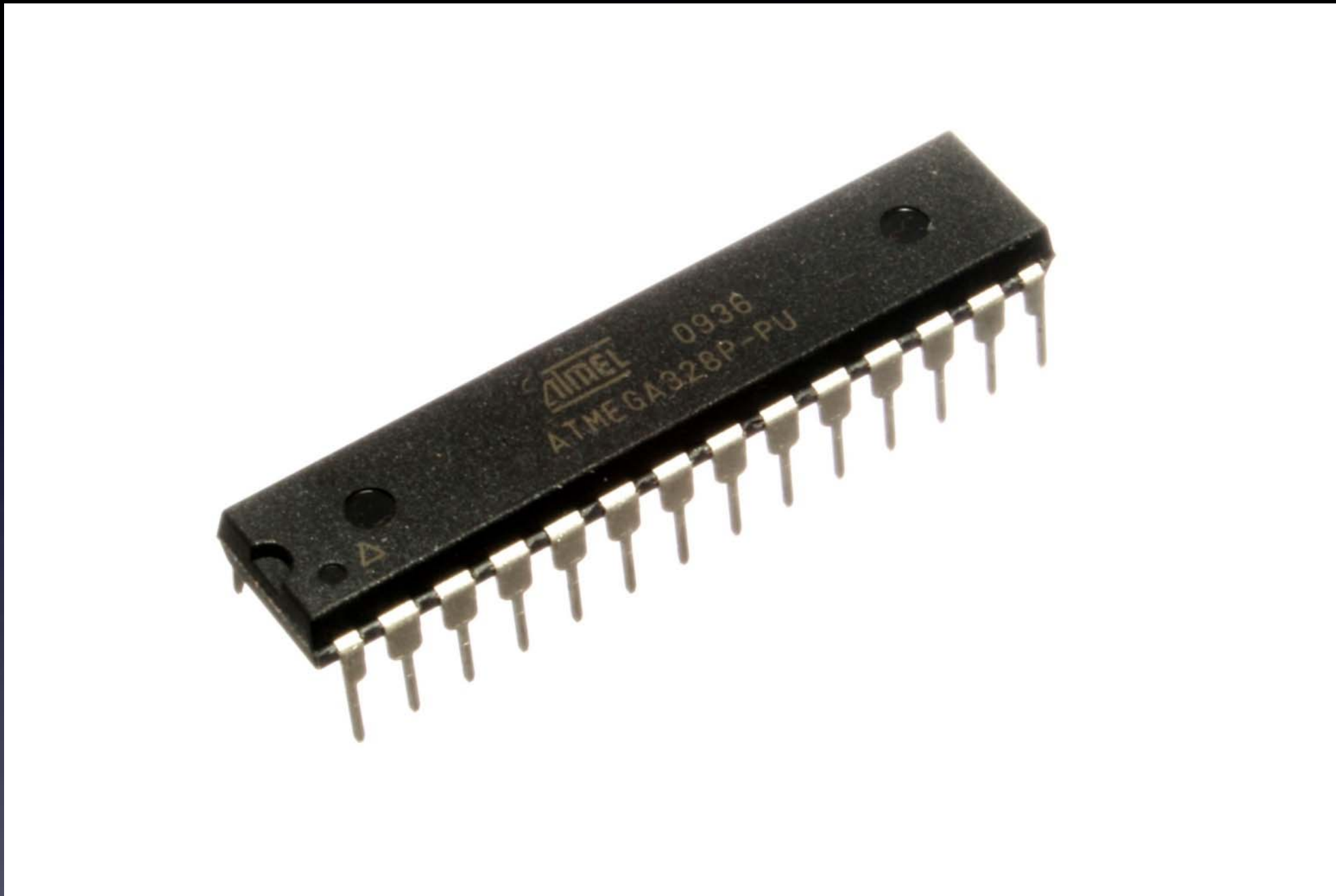


# Intro to Arduino



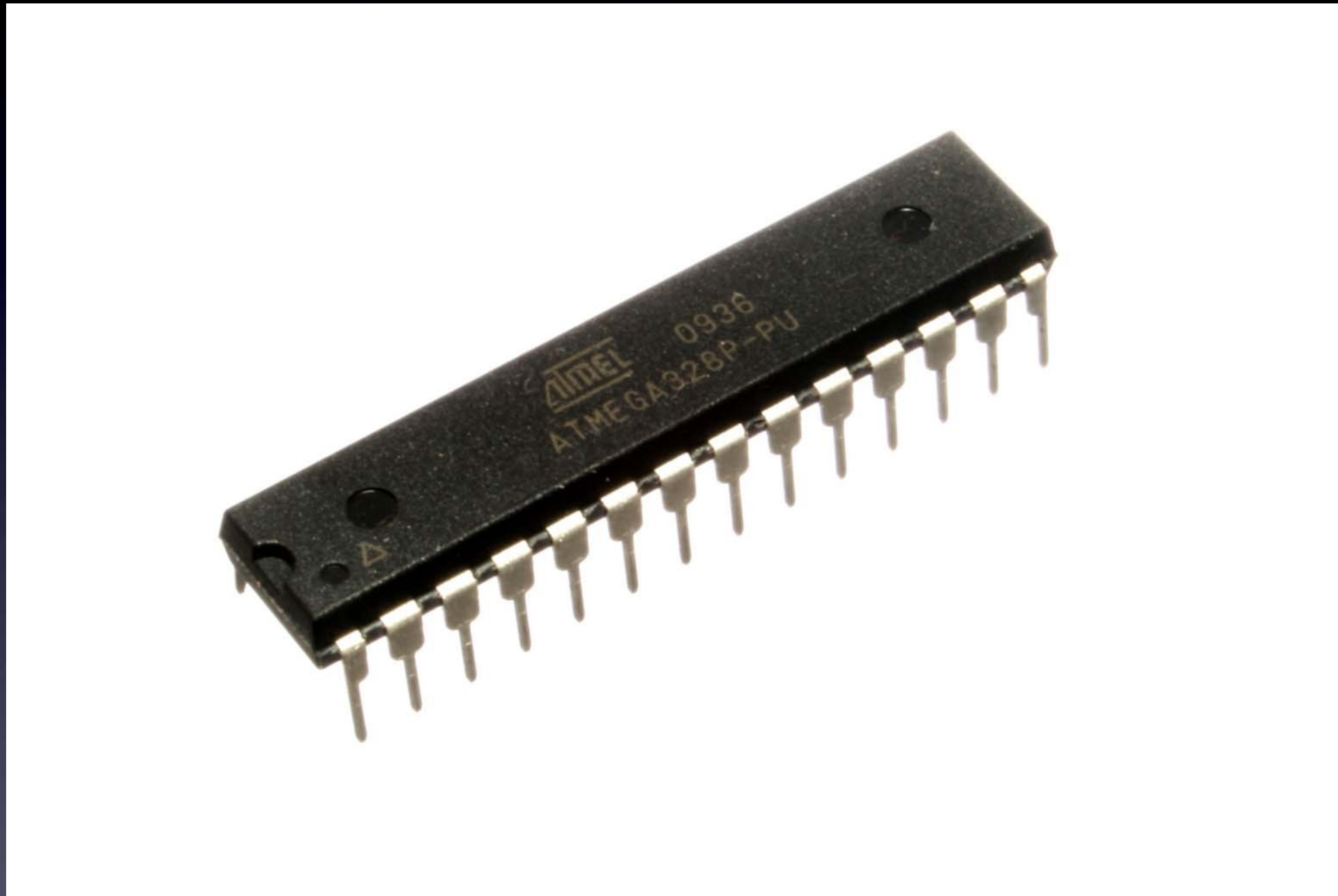


# Intro to Arduino: microcontrollers



A complete computer on a chip

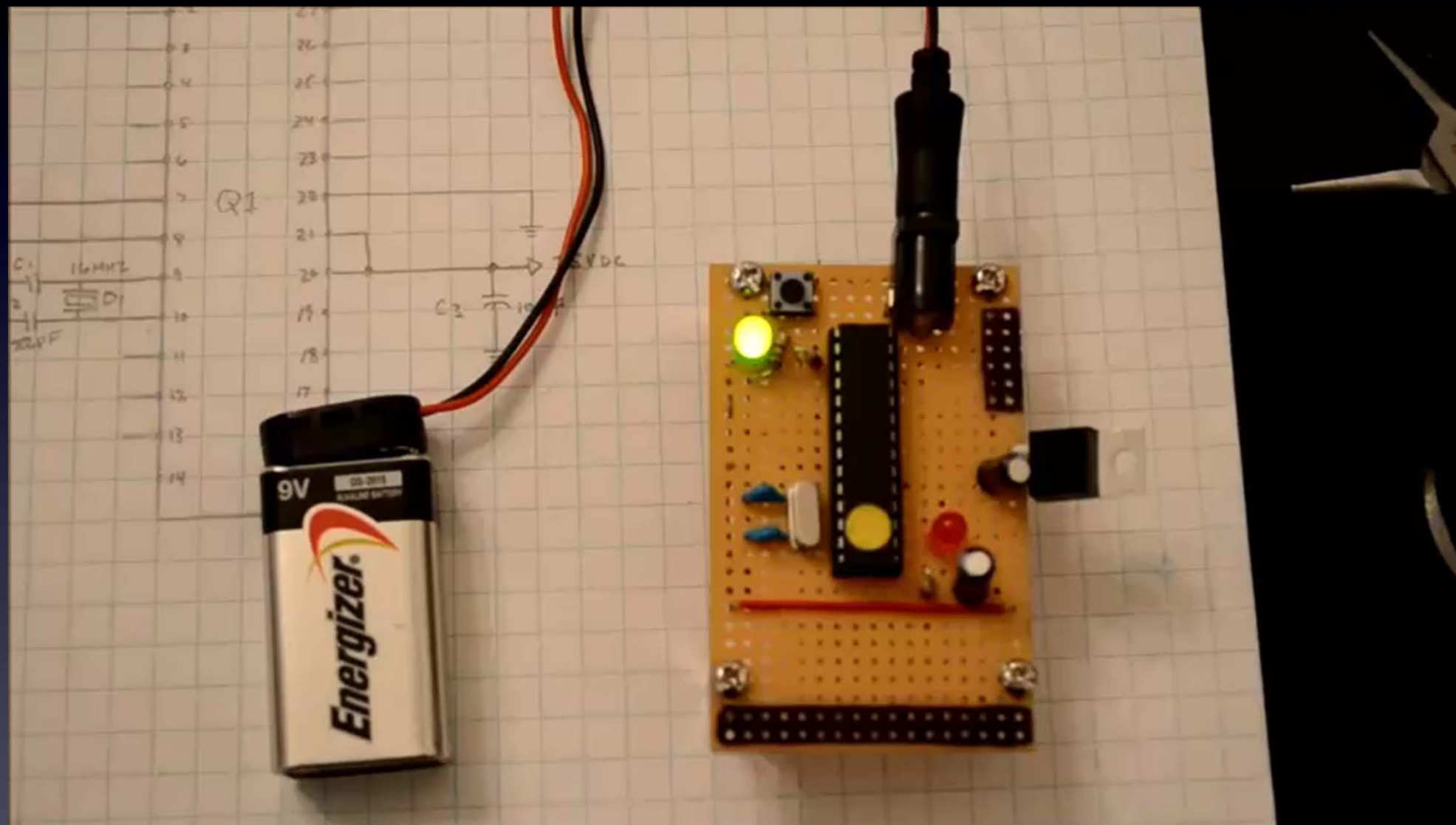
# Intro to Arduino: microcontrollers



**A complete computer on a chip:**  
they control parts connected to their pins

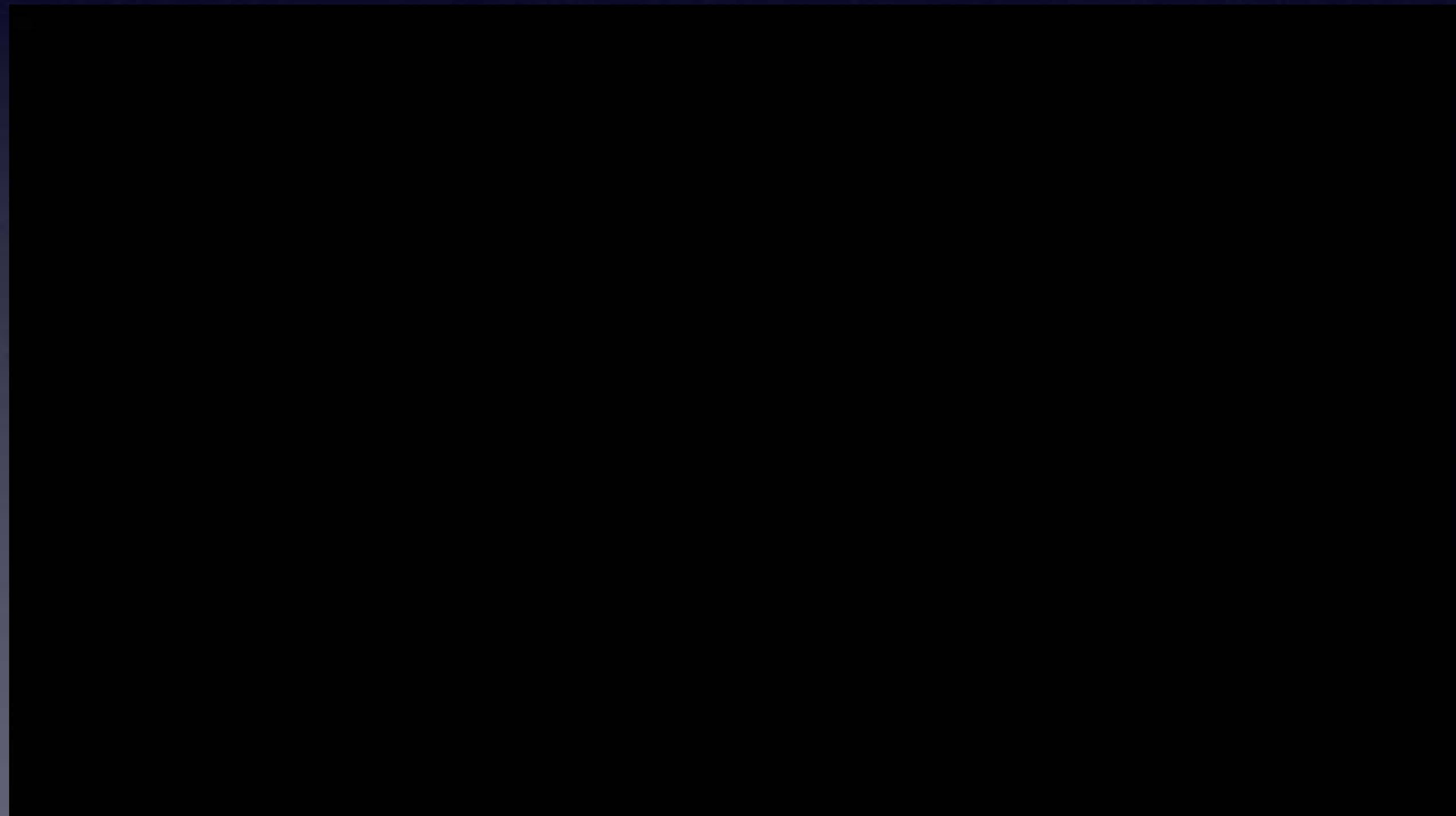


# Intro to Arduino: microcontrollers



Intro to Arduino:  
microcontrollers  
*-- one of Mitch's projects --*

*ArduTouch*  
*music*  
*synthesizer*  
*kit*

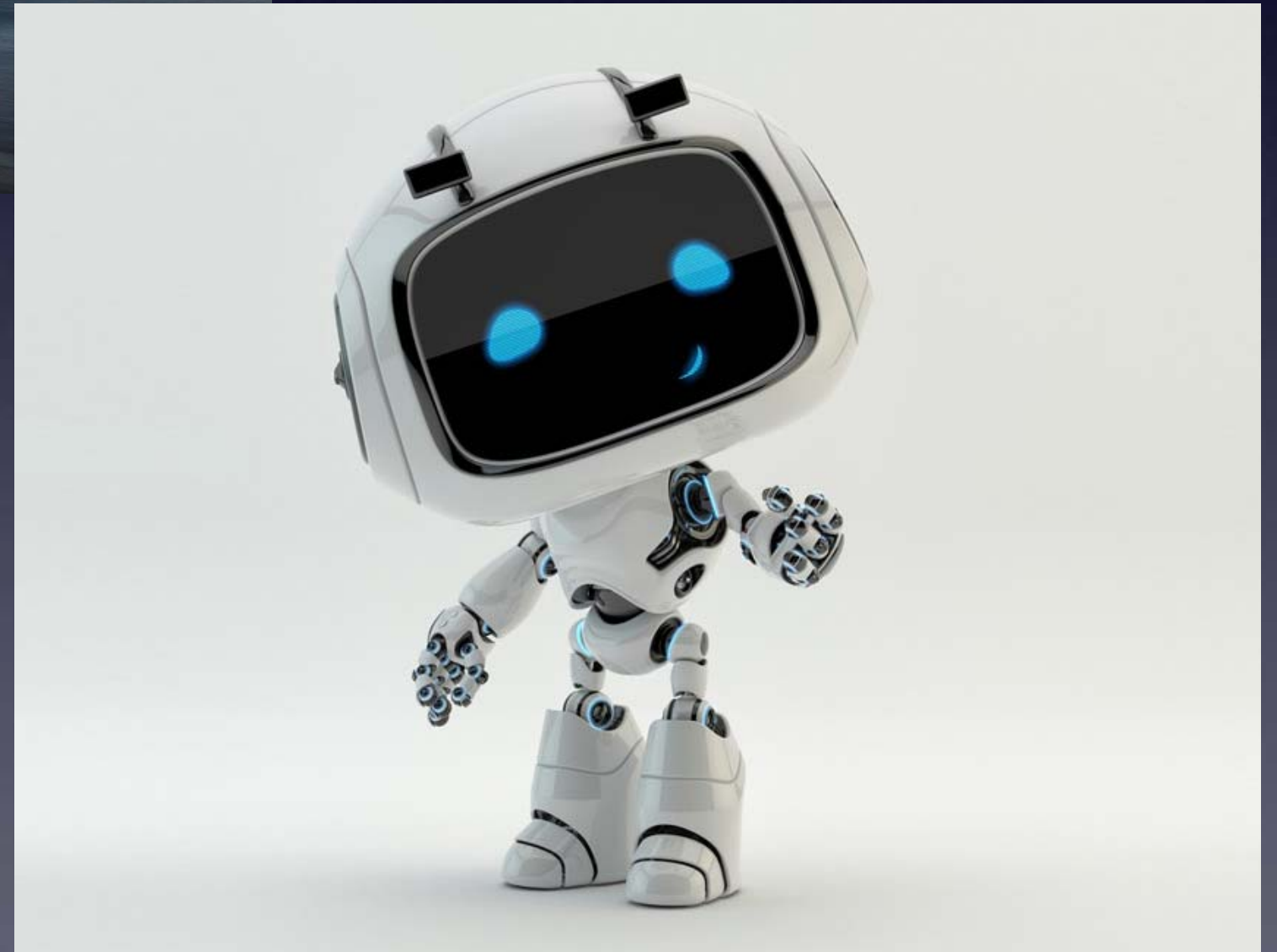




# Intro to Arduino: microcontrollers



# Intro to Arduino: microcontrollers





# Intro to Arduino: microcontrollers





# Intro to Arduino: microcontrollers





# Intro to Arduino: microcontrollers

-- *one of Mitch's projects* --

*TV-B-Gone*





# Intro to Arduino: microcontrollers

*-- one of Mitch's projects --*



***TV-B-Gone***  
*Just a remote control,  
but only one button:  
OFF !*



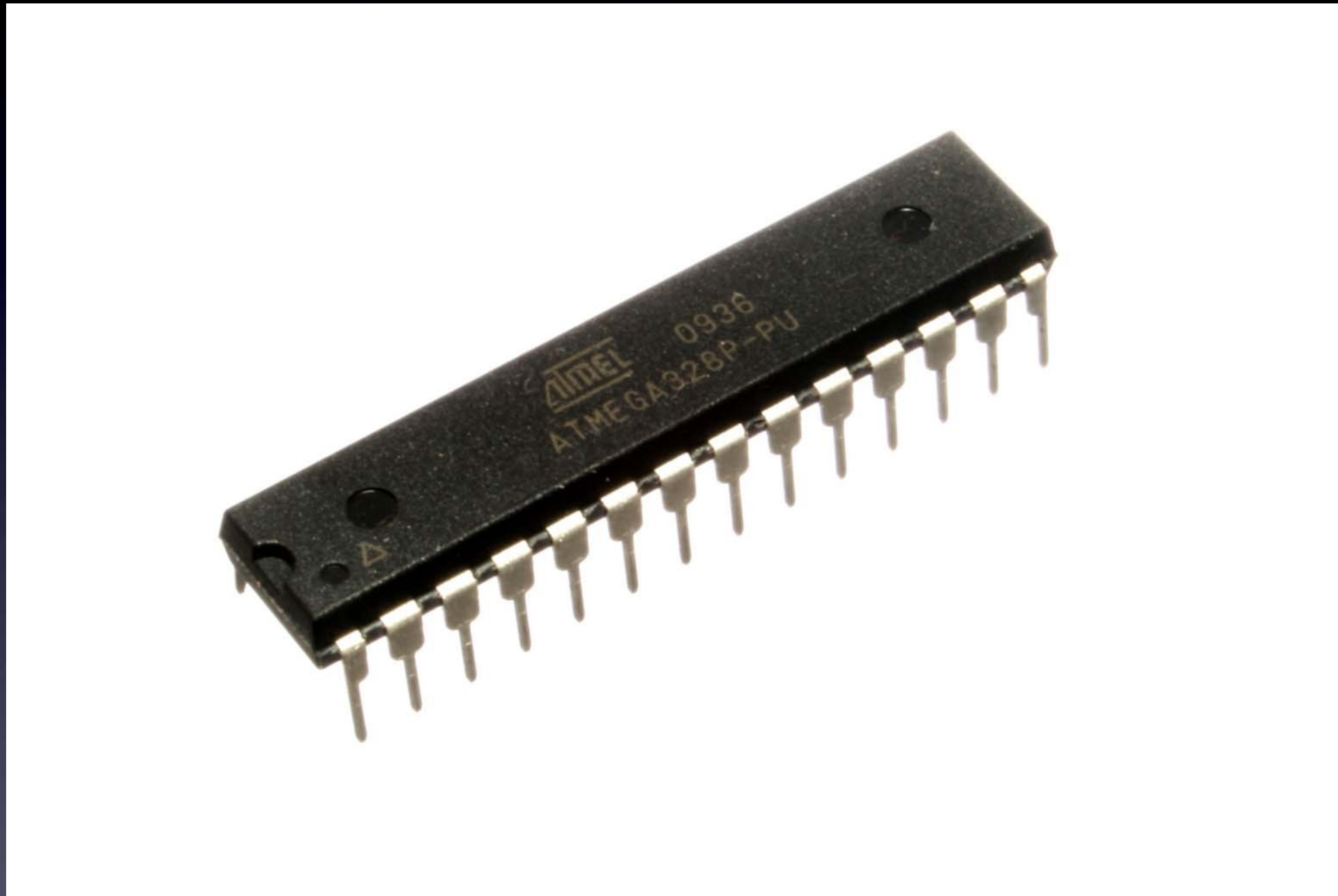
# Intro to Arduino: microcontrollers

*-- one of Mitch's projects --*



*TV-B-Gone*

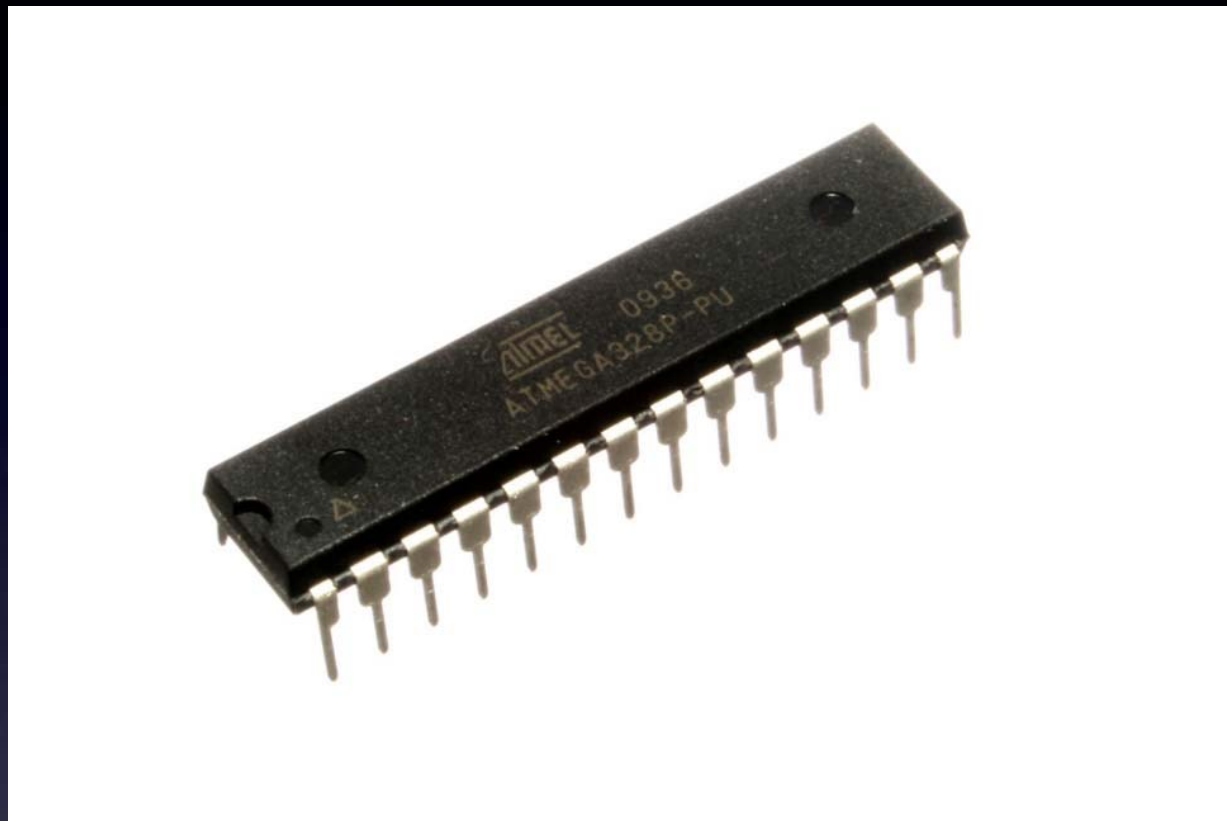
# Intro to Arduino: microcontrollers



**A complete computer on a chip:**  
they control parts connected to their pins



# Intro to Arduino: microcontrollers



*A complete computer on a chip:*

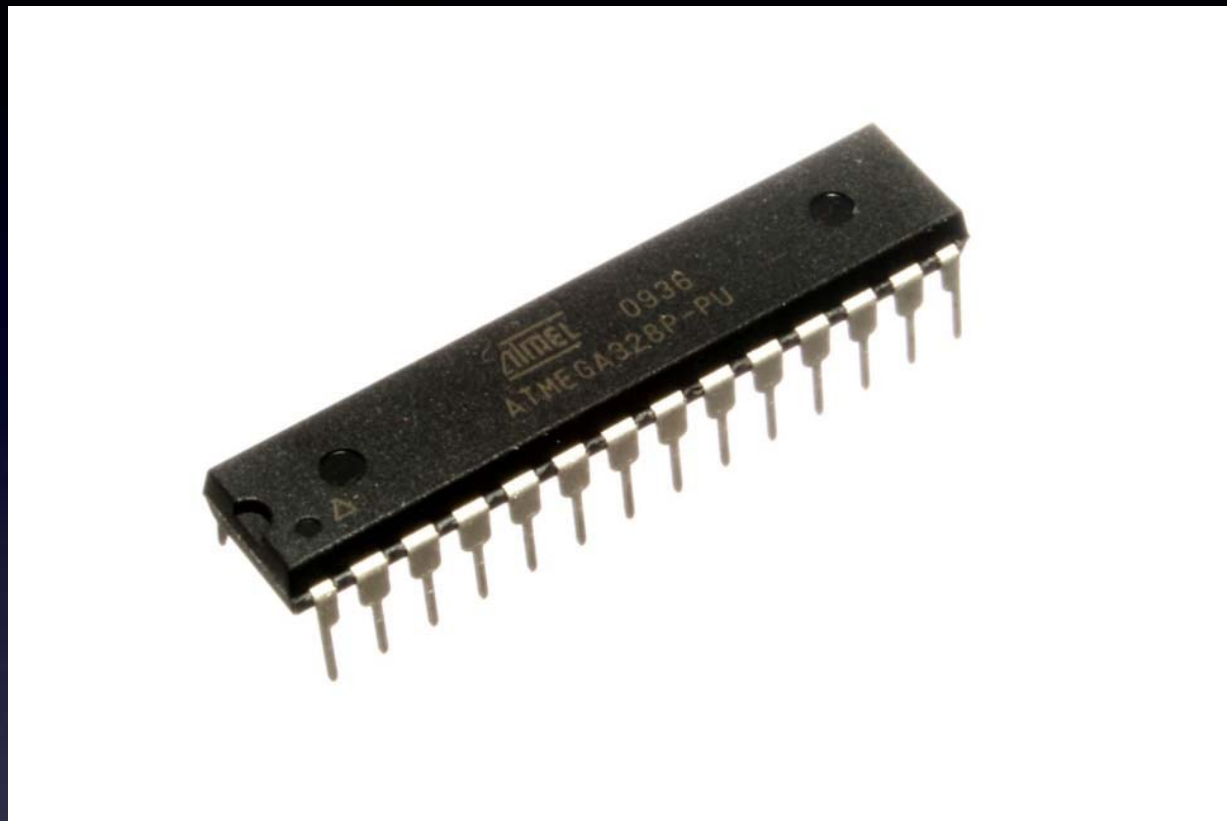
they control parts connected to their pins

***But,***

How do you  
connect parts to its pins?

How do you  
create and upload a program  
to control the parts?

# Intro to Arduino: microcontrollers



*A complete computer on a chip:*

they control parts connected to their pins

*But,*

How do you  
connect parts to its pins?

How do you  
create and upload a program  
to control the parts?

*Answer: Be a geek, and learn how!*



Or

# Intro to Arduino



*Use an Arduino board*



# Intro to Arduino



Super easy to  
connect parts  
to its microcontroller's pins

*Use an Arduino board*

# Intro to Arduino



Super easy to  
connect parts  
to its microcontroller's pins

*Use an Arduino board*

Super easy to  
create and upload a program  
to control the parts



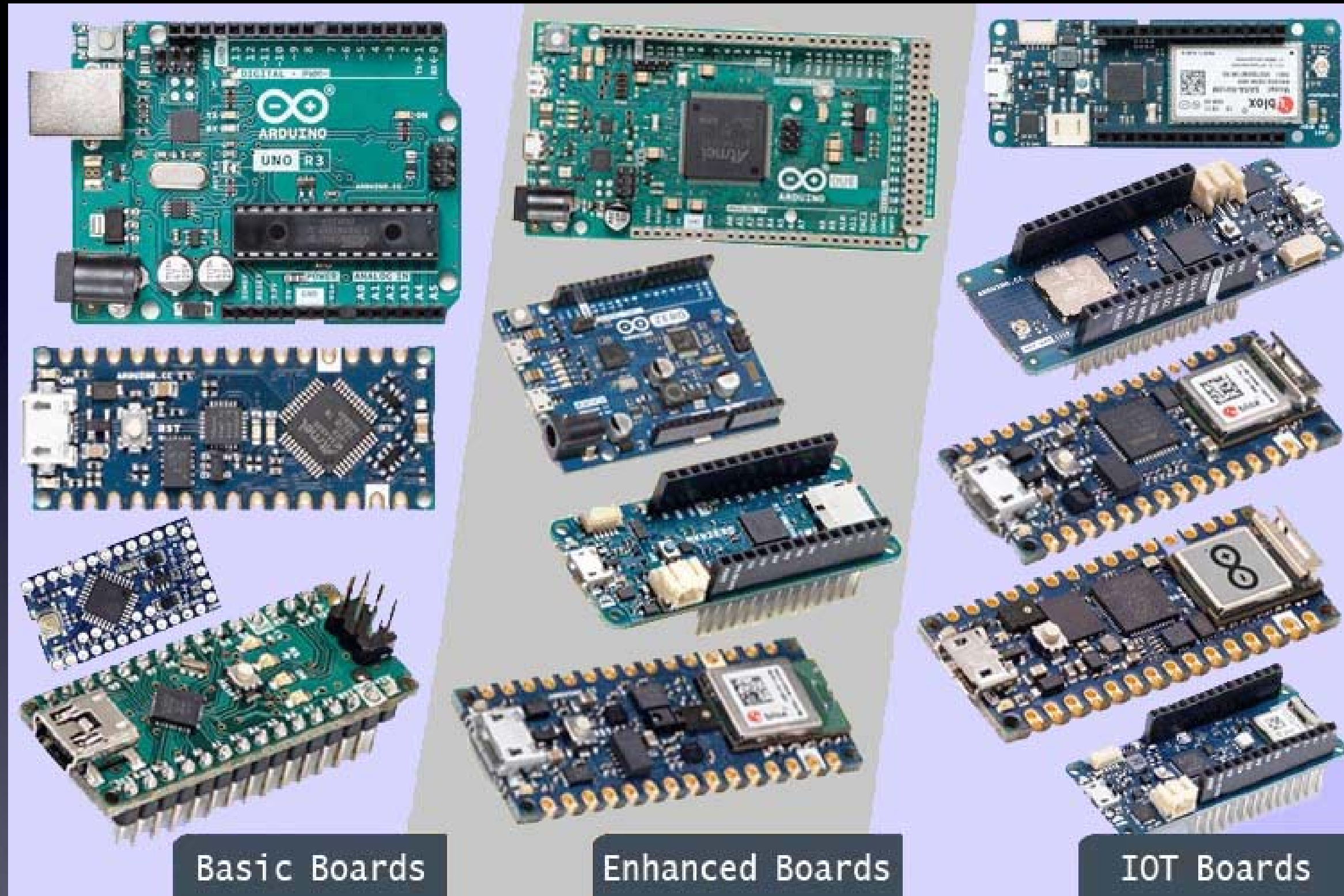
# Intro to Arduino



*Arduino board*

Designed for  
non-geeky artists

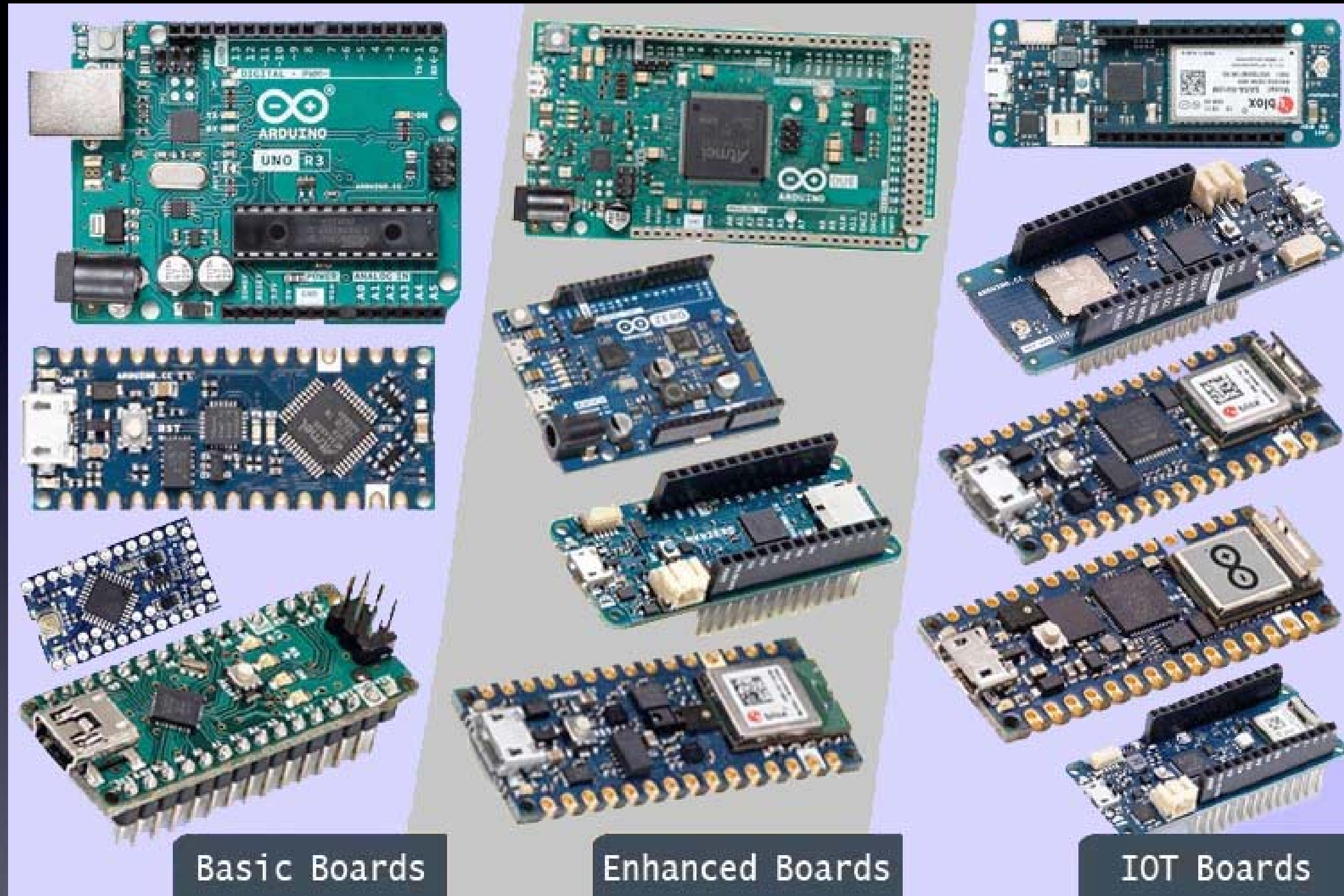
# Intro to Arduino



*Many Arduino boards to choose from*



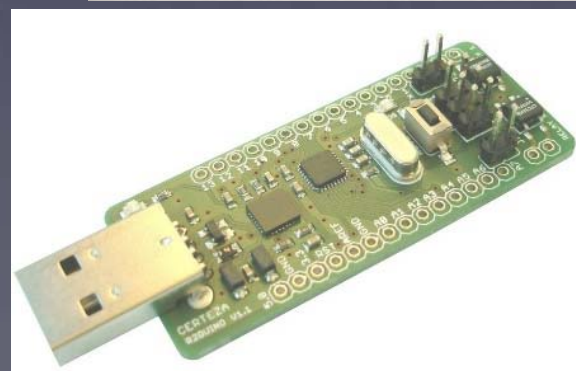
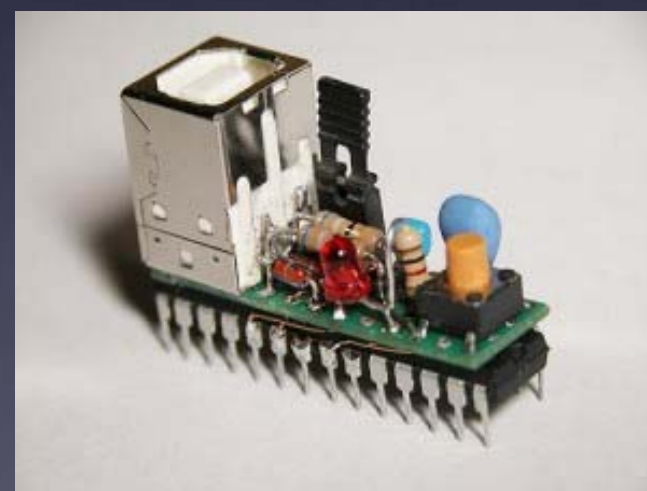
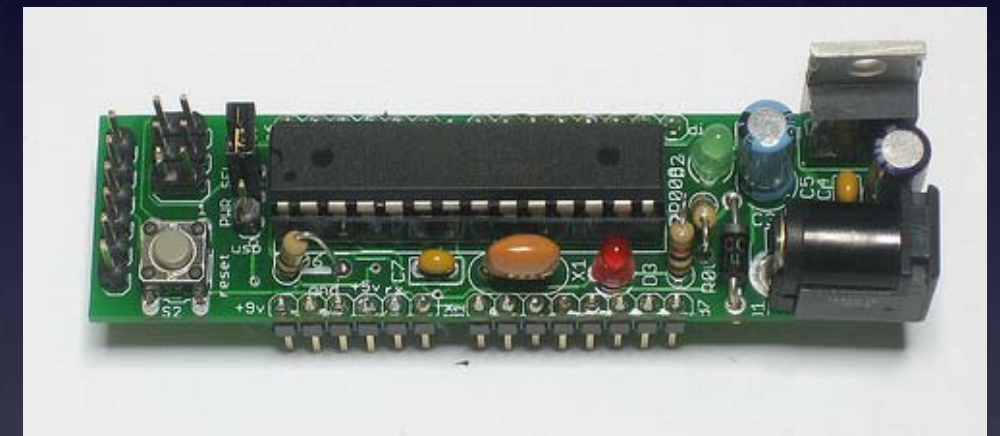
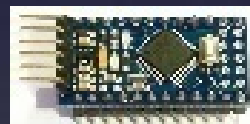
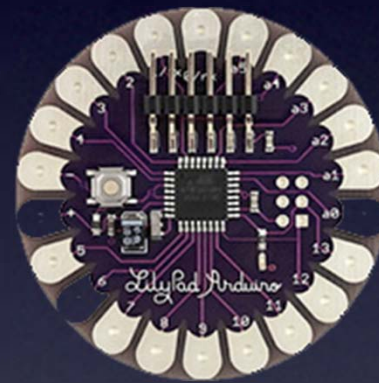
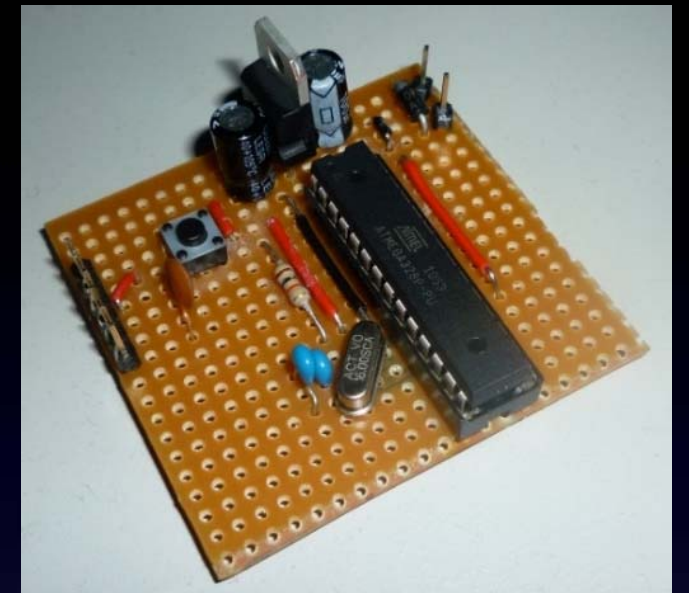
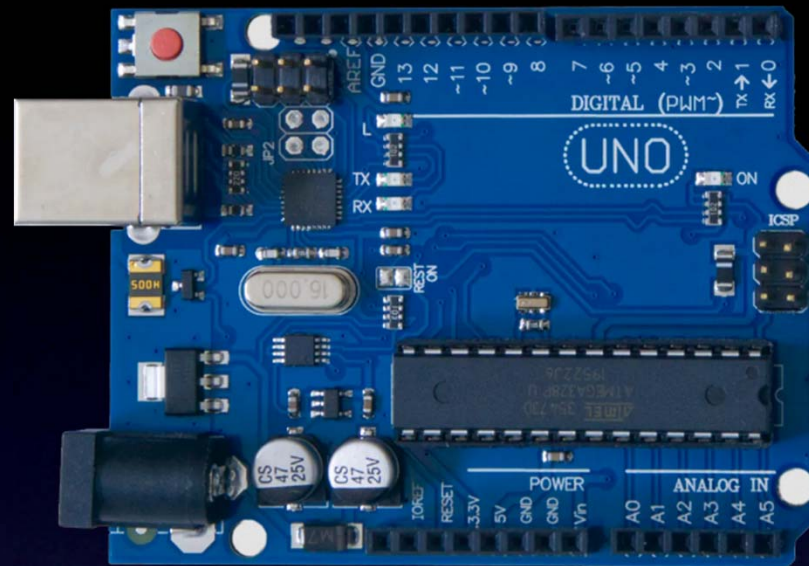
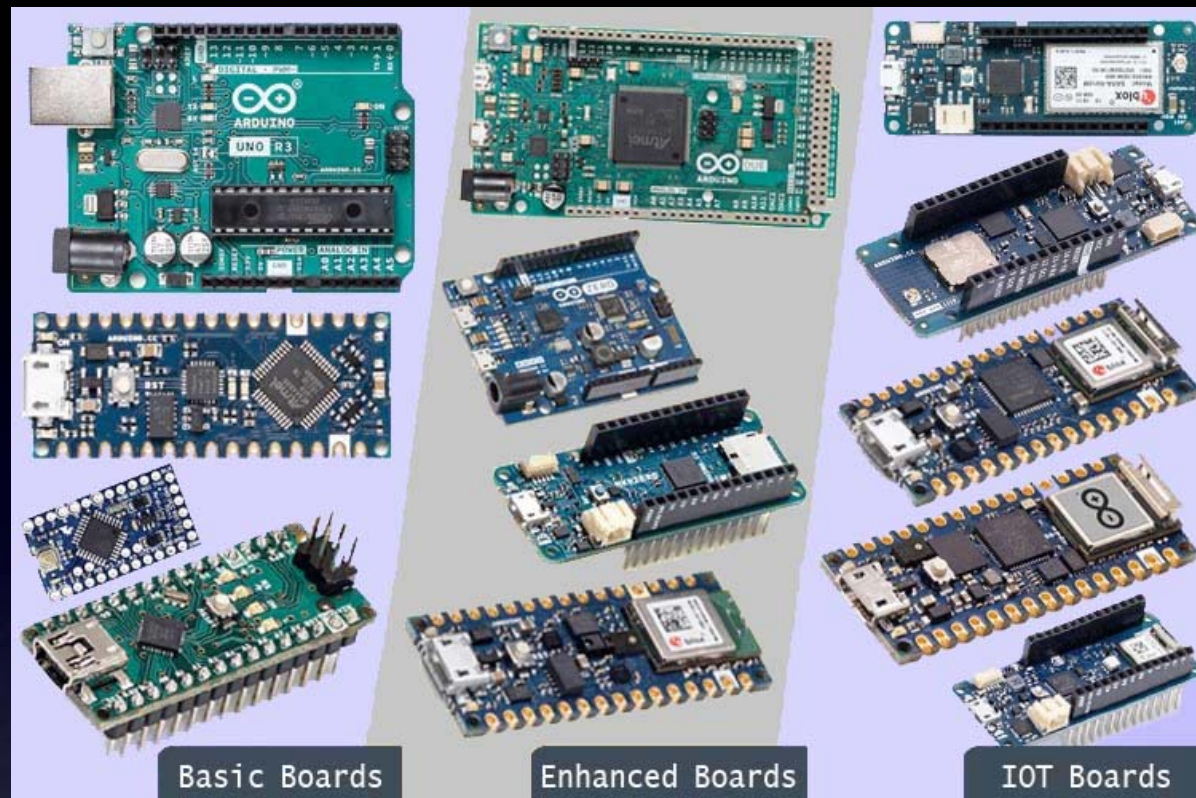
# Intro to Arduino



Open Source



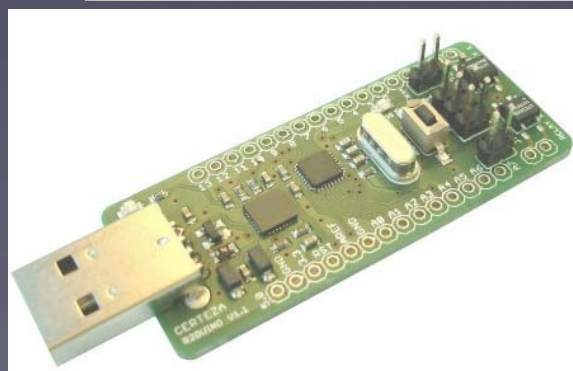
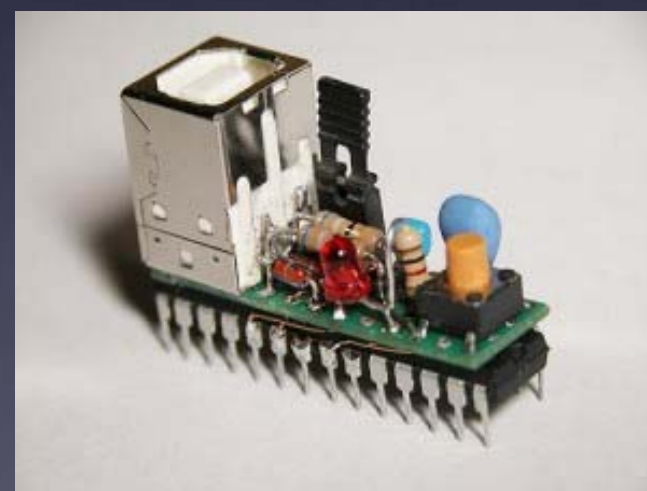
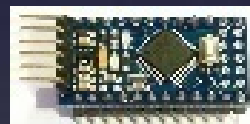
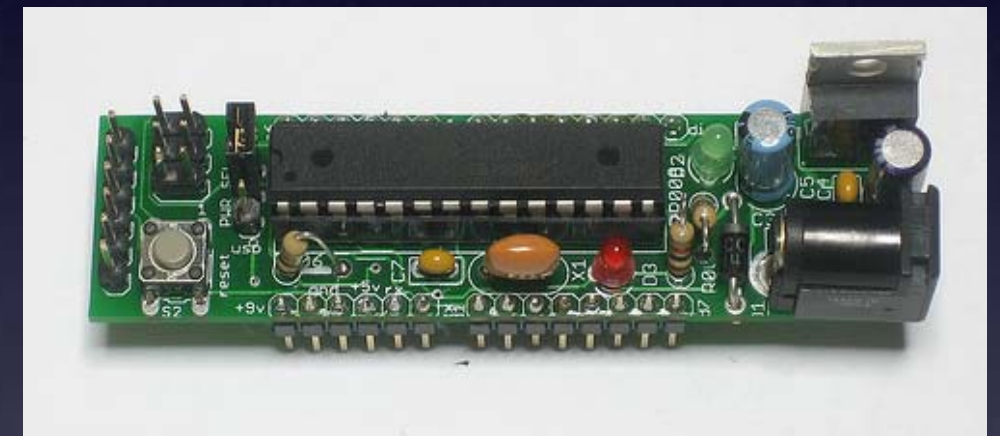
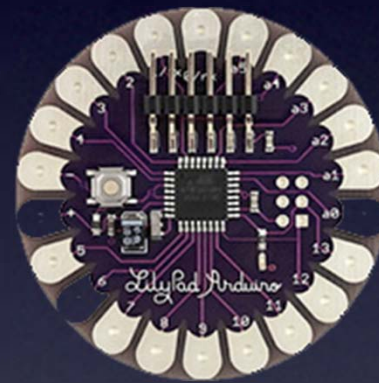
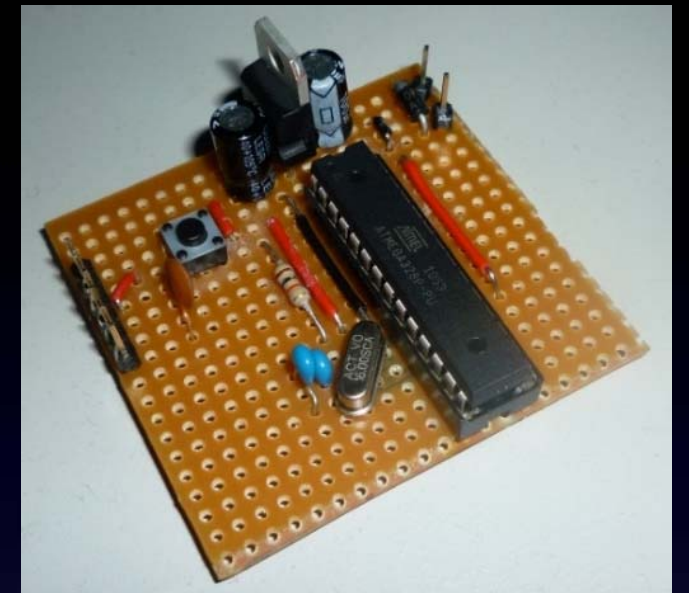
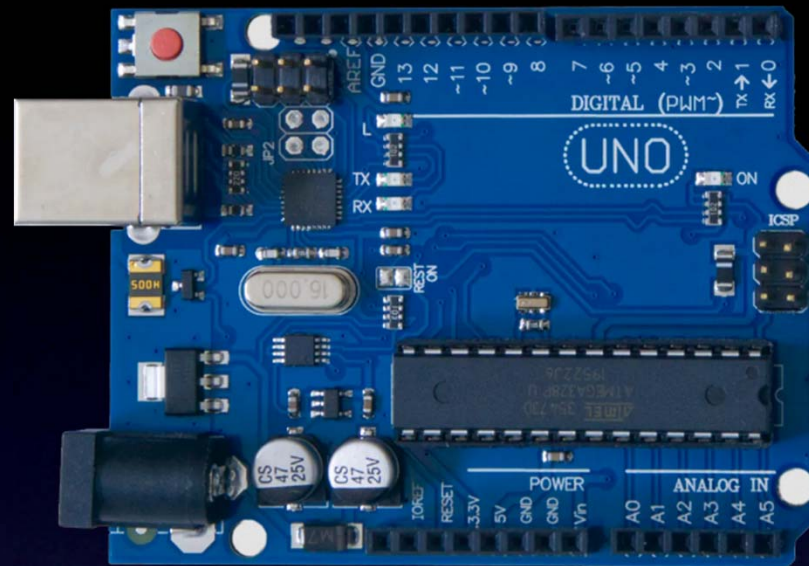
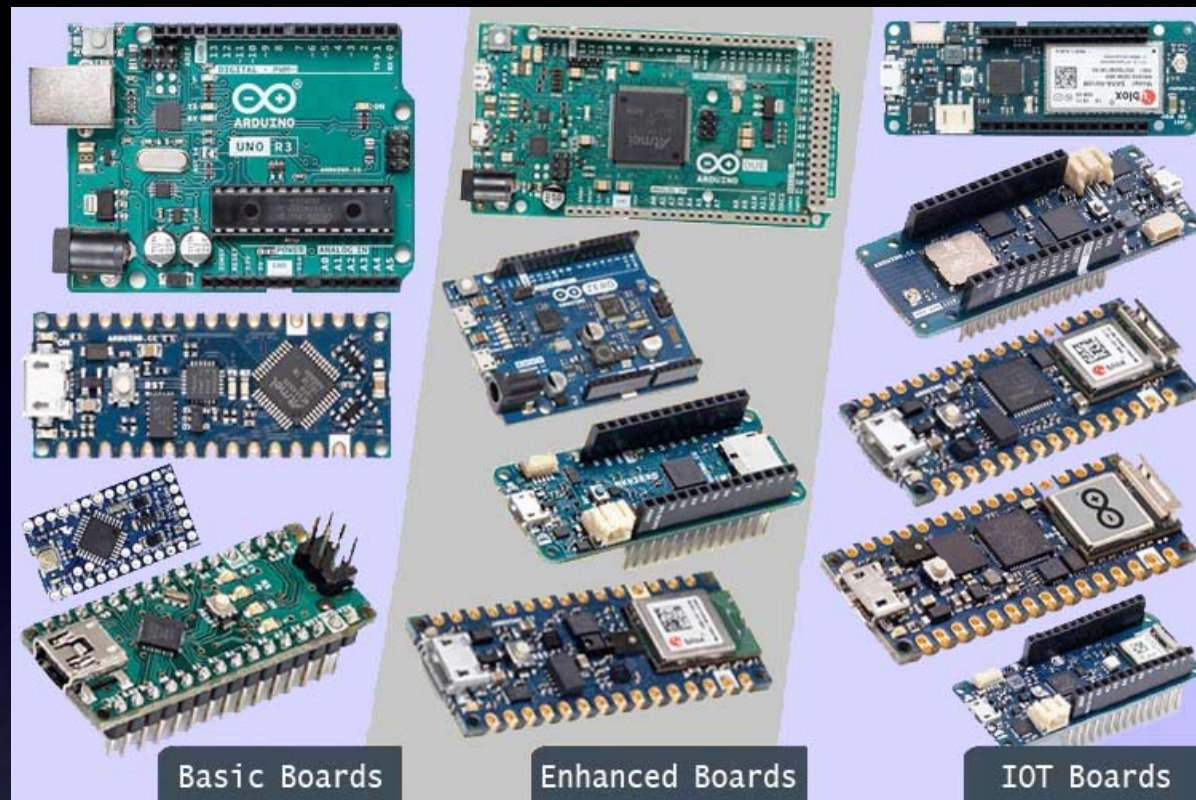
# Intro to Arduino



Open Source



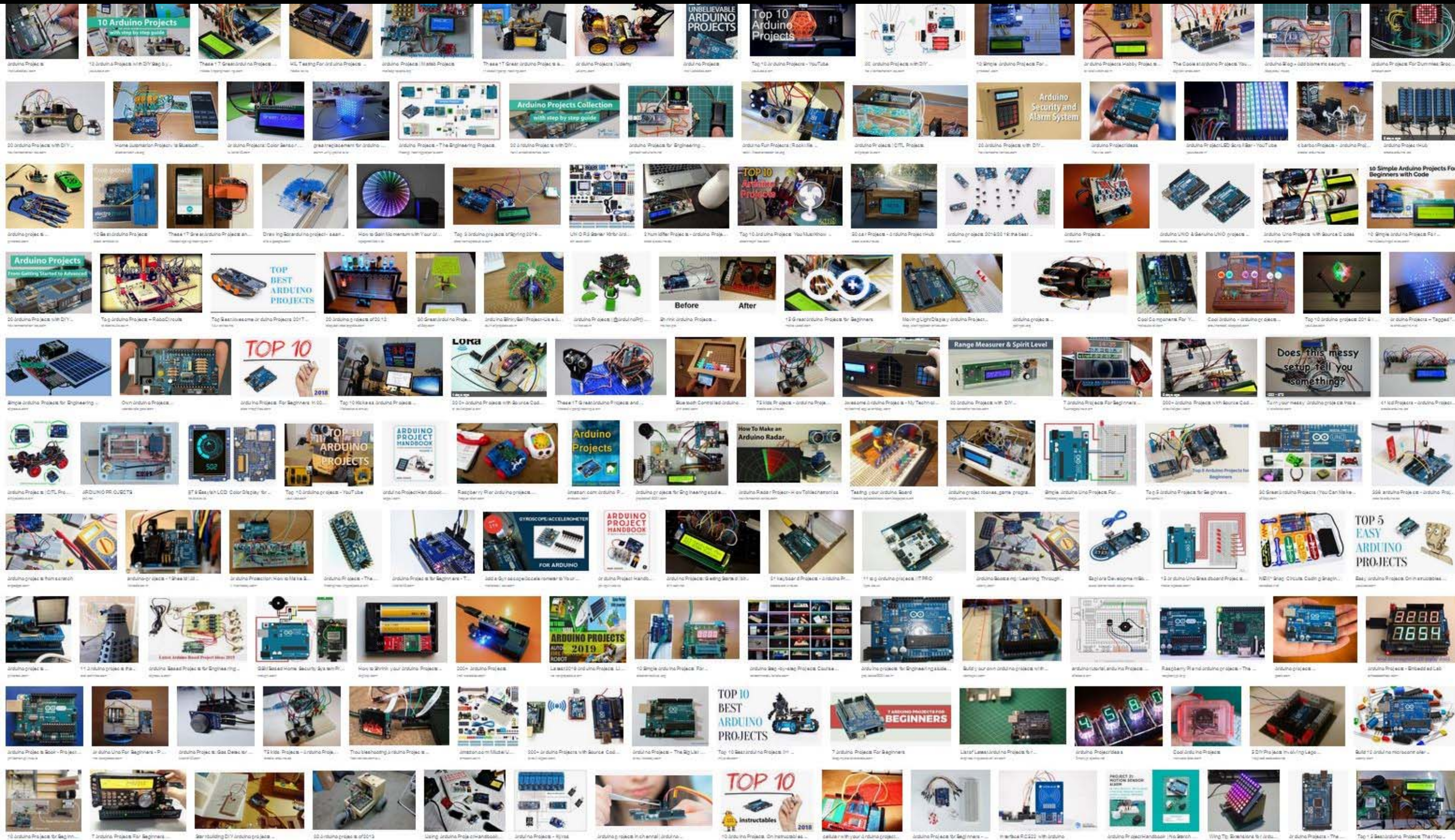
# Intro to Arduino



Arduino "Clones"



# Intro to Arduino



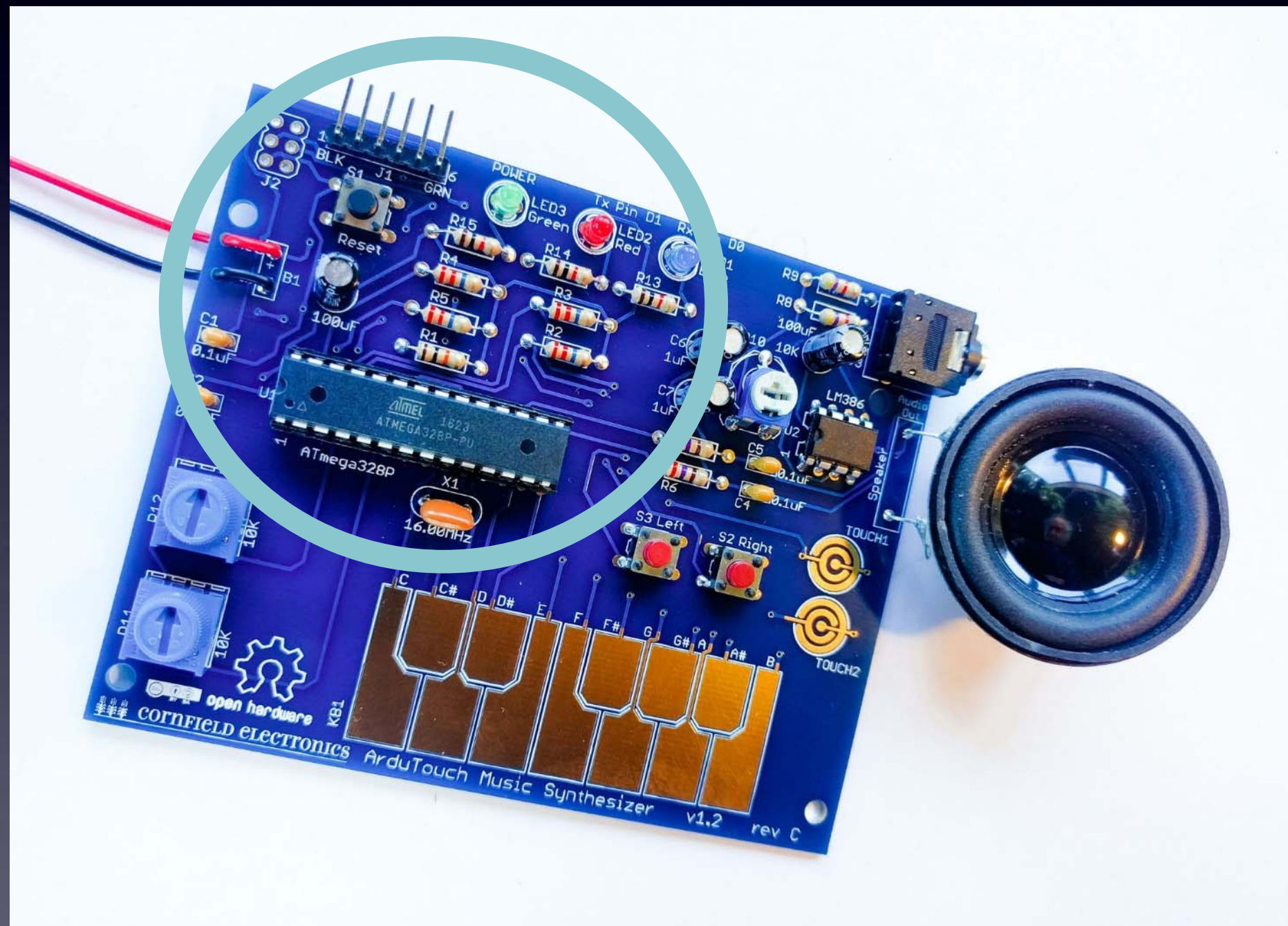
*hundreds of thousands of projects online!*



# Intro to Arduino

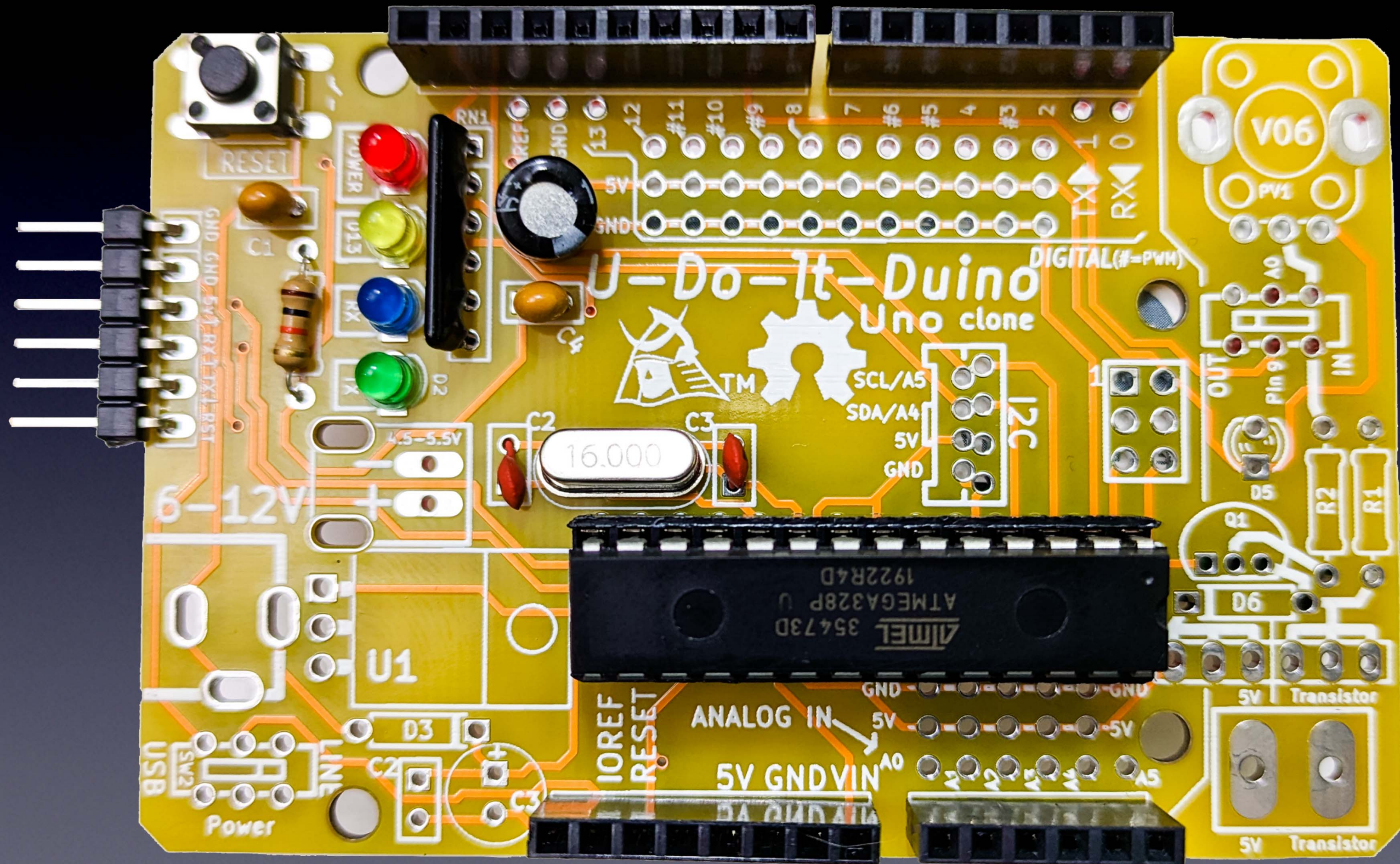
*“Arduino-Compatible”*

*ArduTouch  
music  
synthesizer  
kit*





# Intro

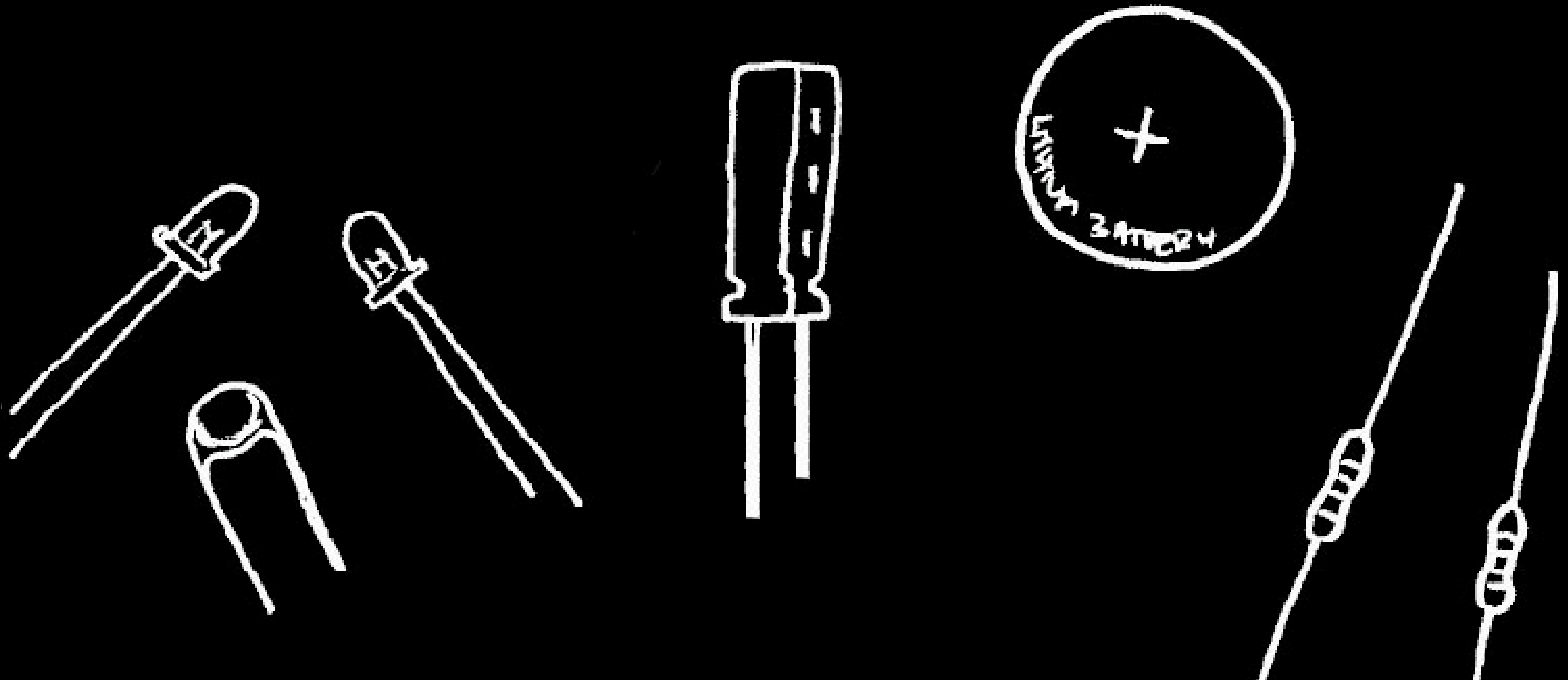




Intro

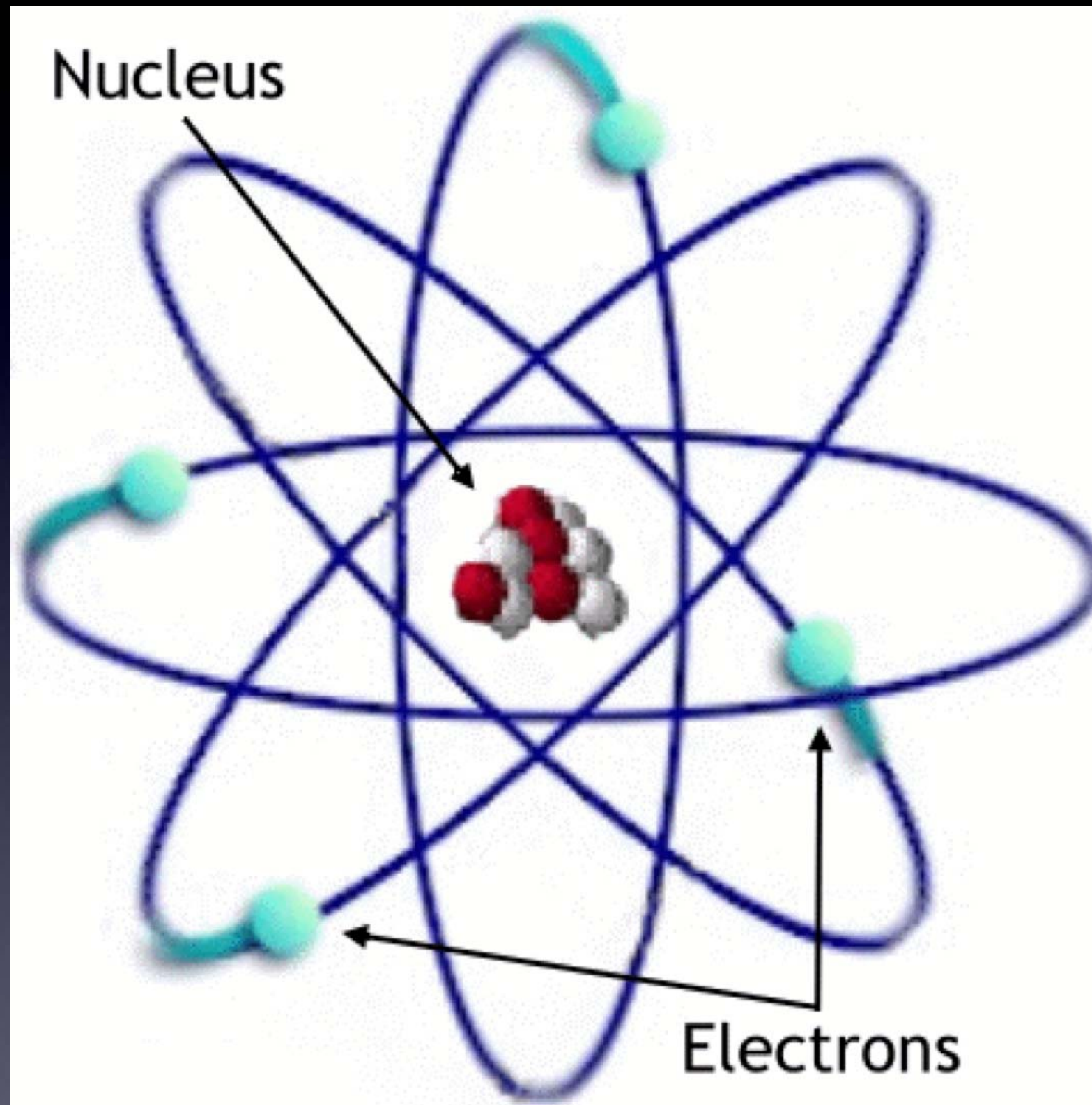
Questions?

# *Everything You Need to Know About Electronics*



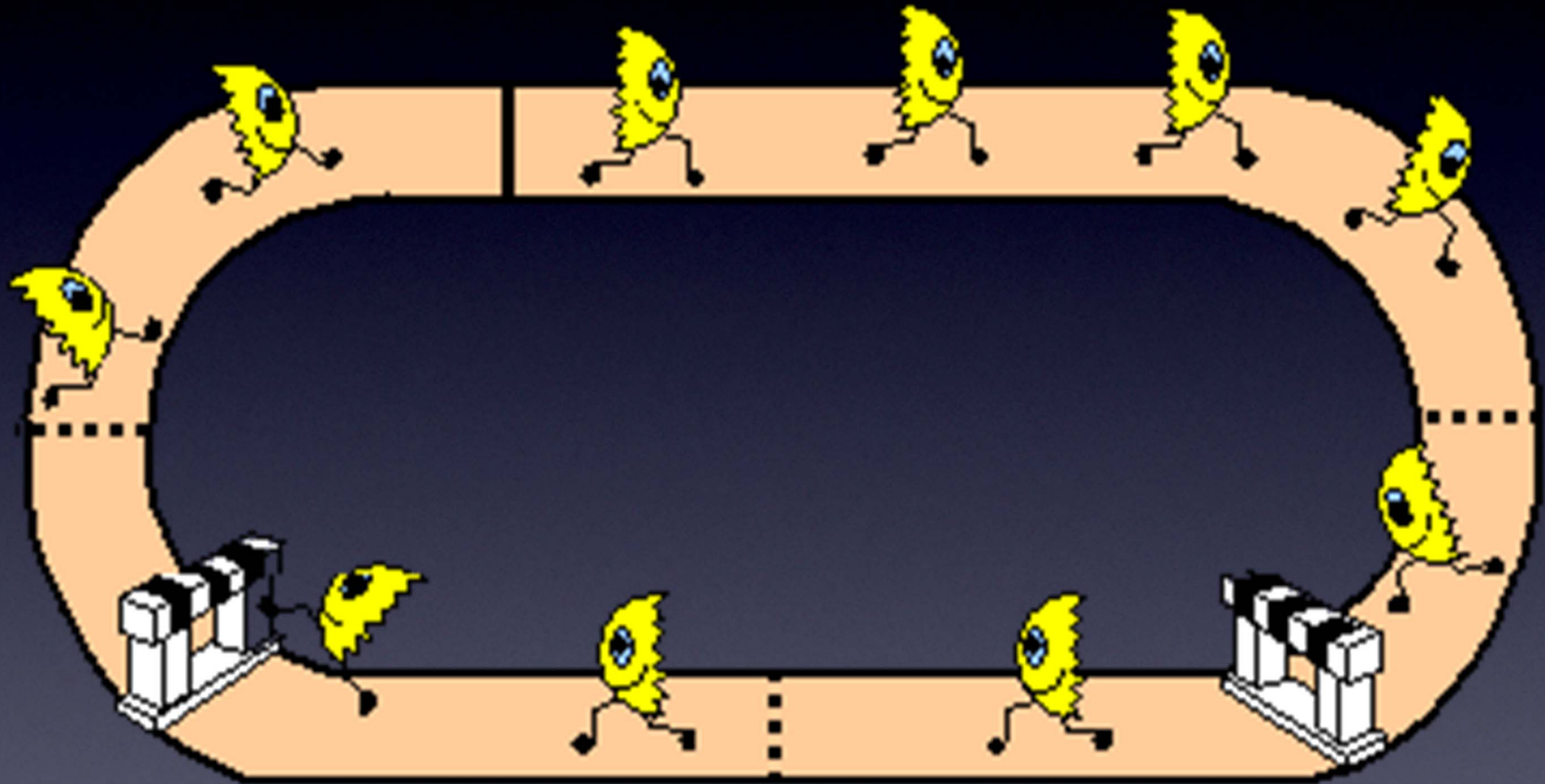


# Everything You Need to Know About Electronics



Electrons

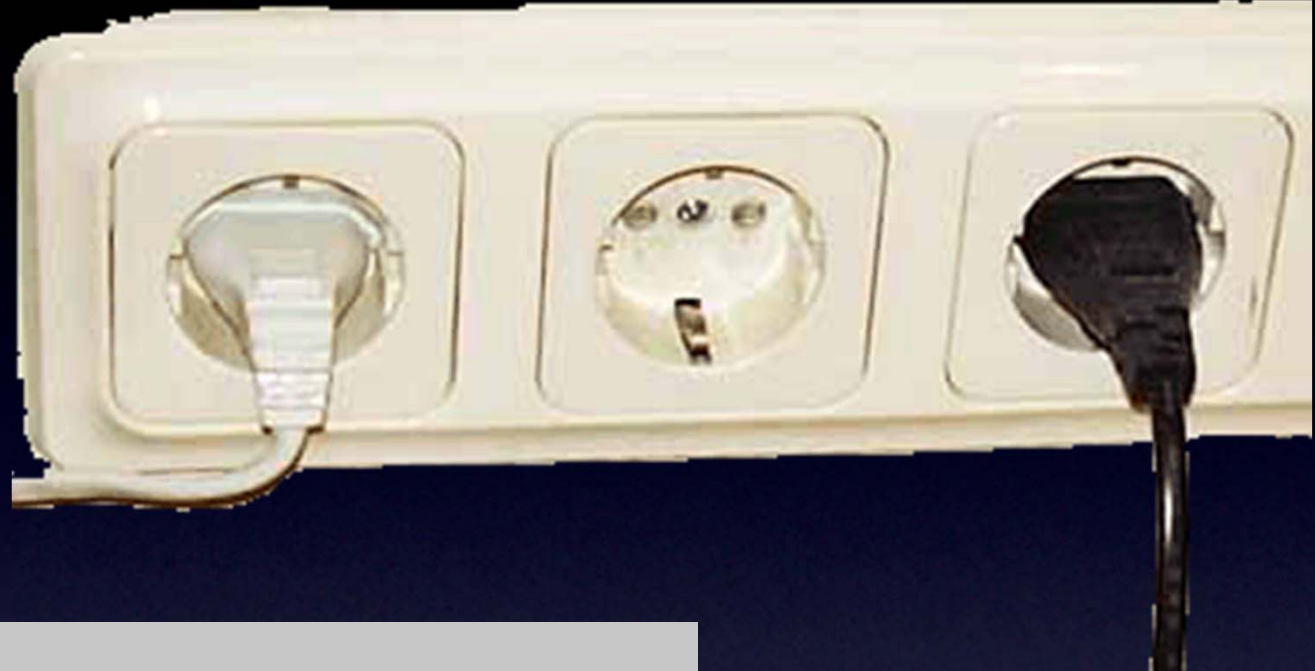
# Everything You Need to Know About Electronics



Circuit = Electrons going in complete circle = Magic!

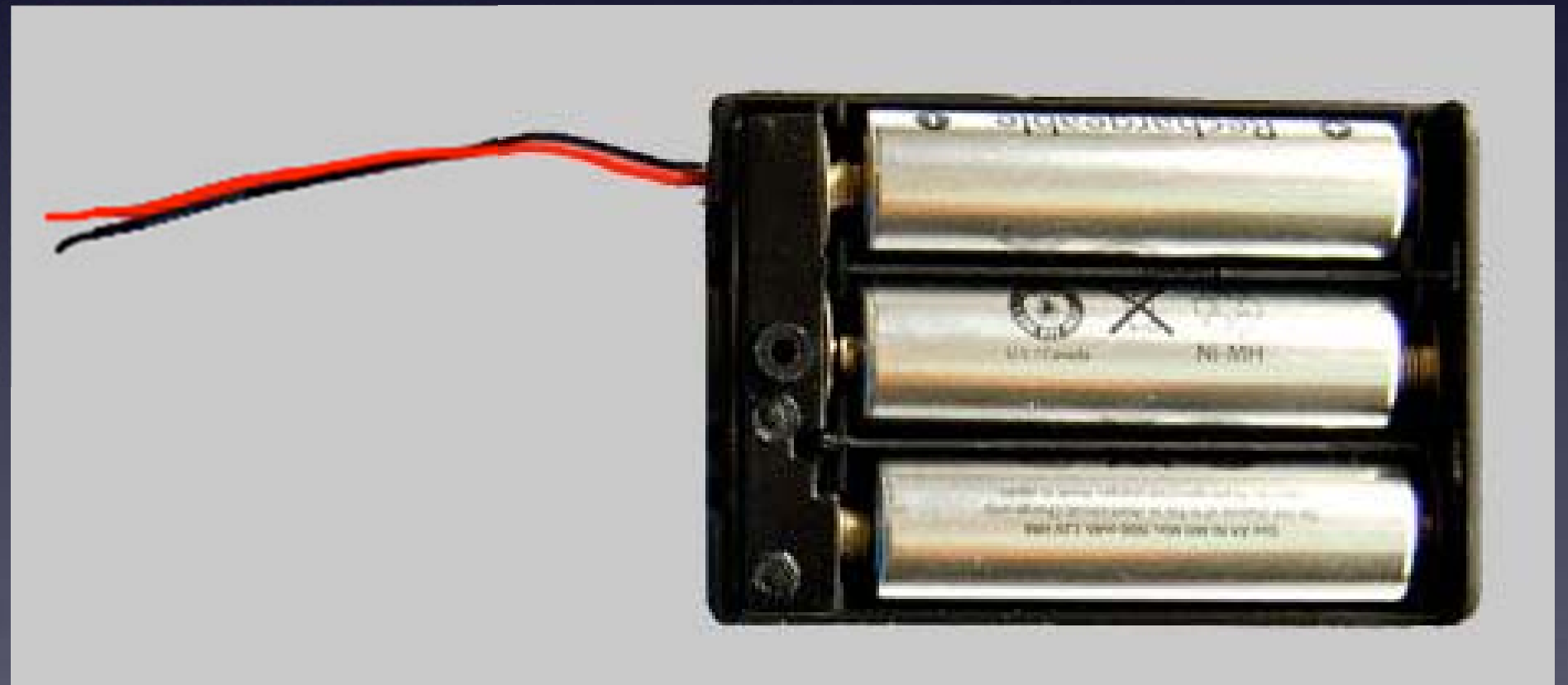


# Everything You Need to Know About Electronics



Power Supplies

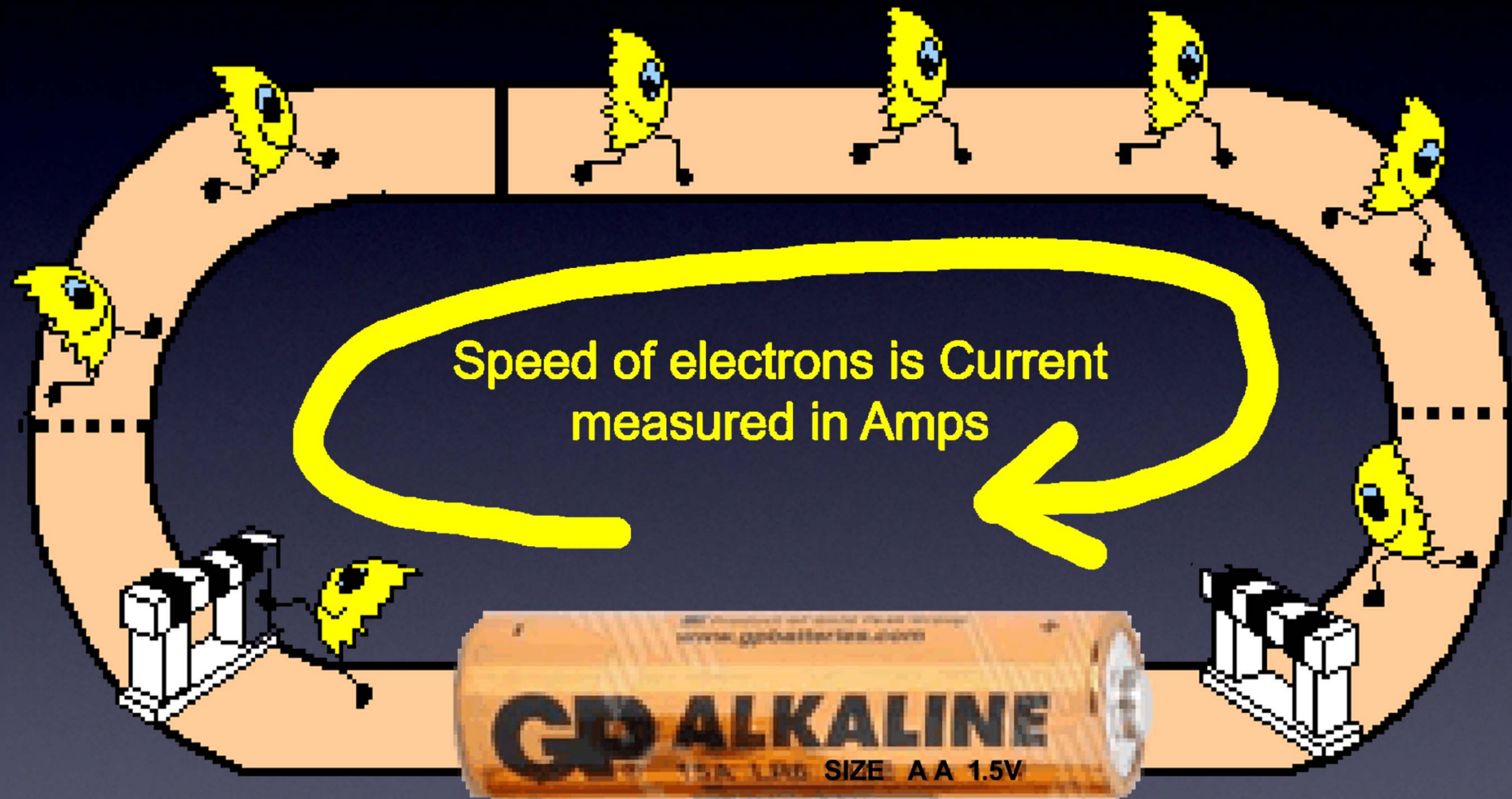
# Everything You Need to Know About Electronics



Voltage / **Volts**



# Everything You Need to Know About Electronics

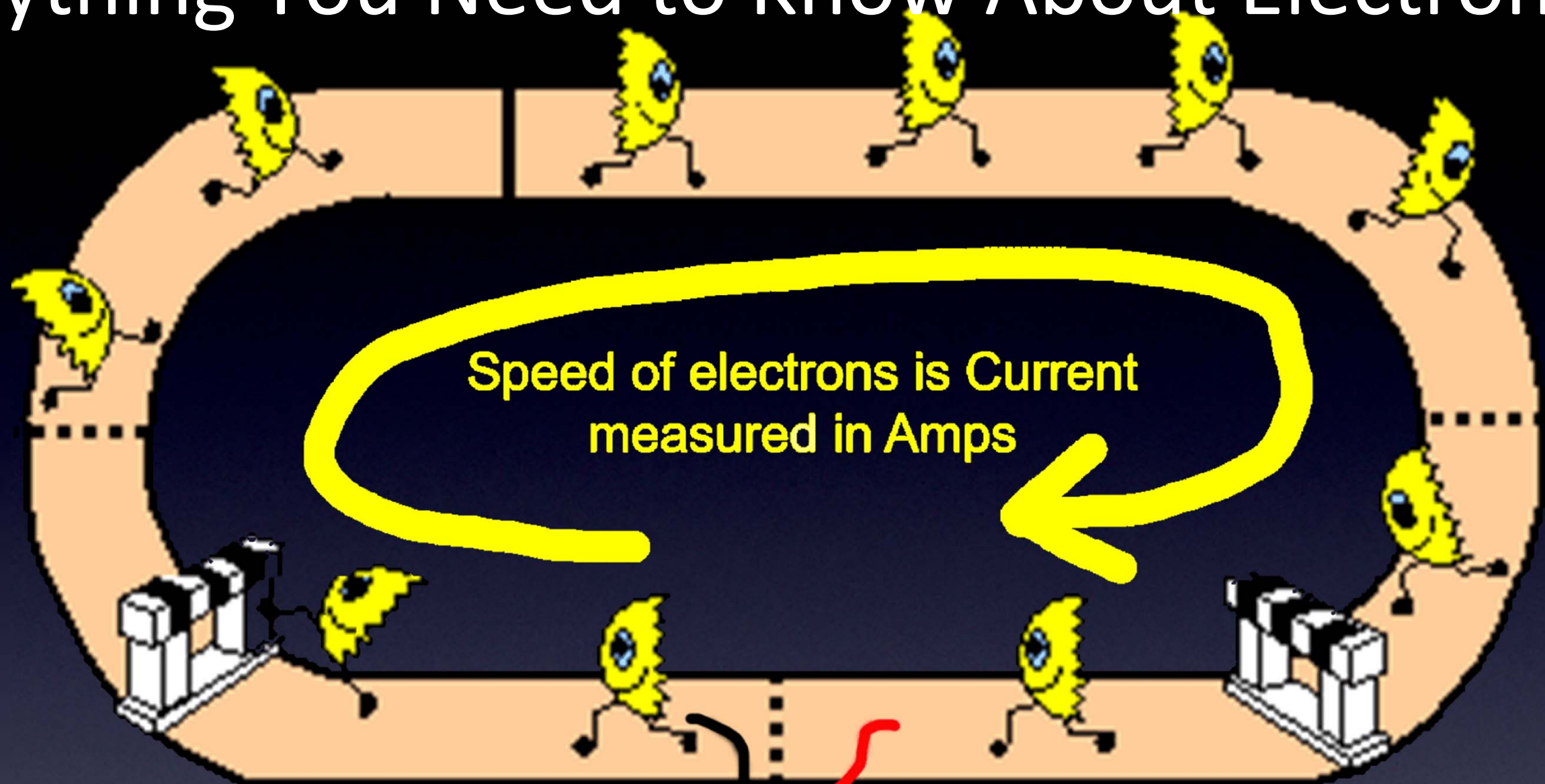


Electrons pushed with 1.5V.  
So, they move!

Current / **Amps**



# Everything You Need to Know About Electronics



3 times more Volts  
3 times more push  
3 times faster electrons  
3 times more current / Amps

Current / Amps



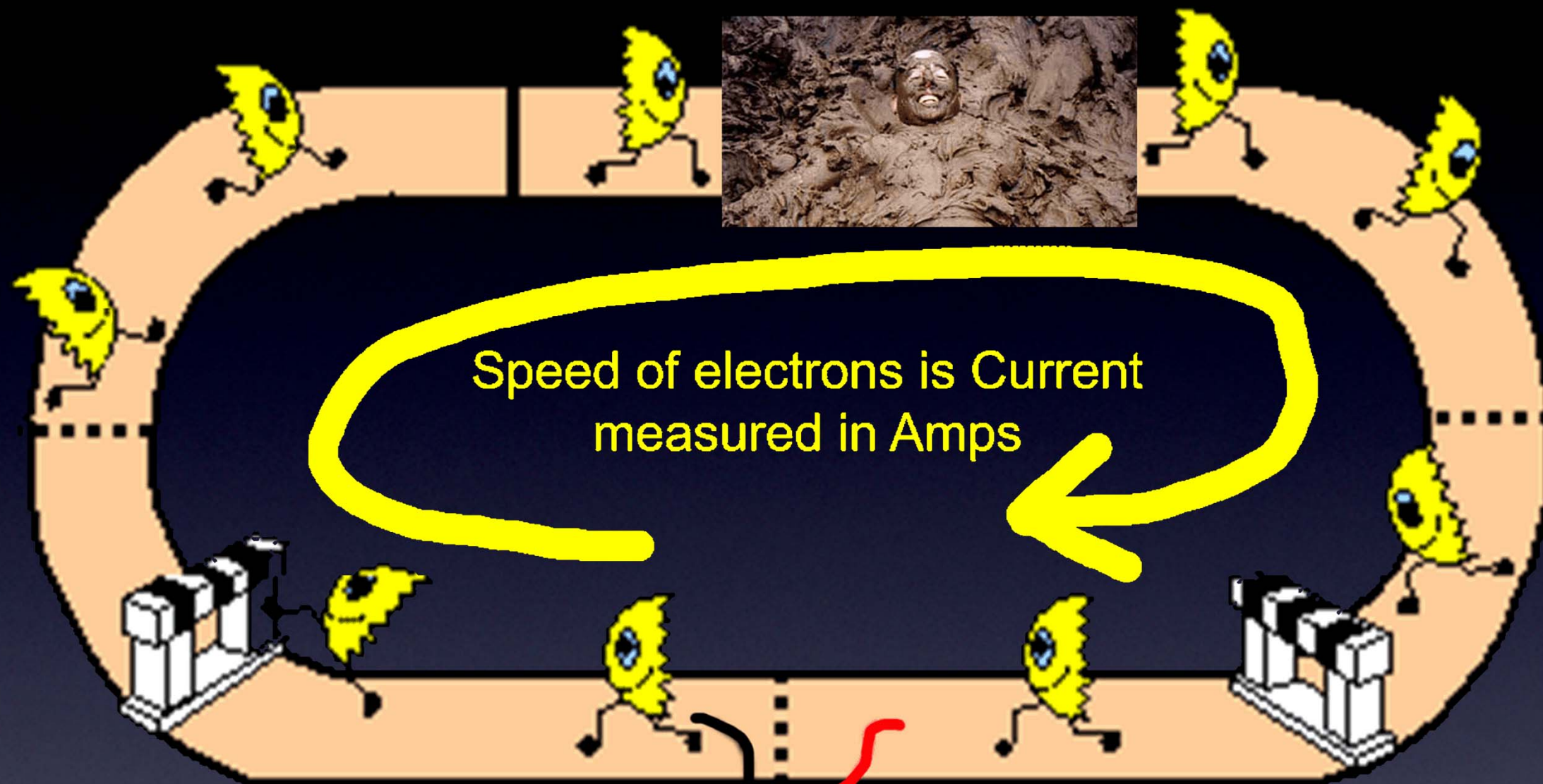
# Everything You Need to Know About Electronics

**Too much energy?**

**Lots of energy!**

Current / Amps

# Everything You Need to Know About Electronics

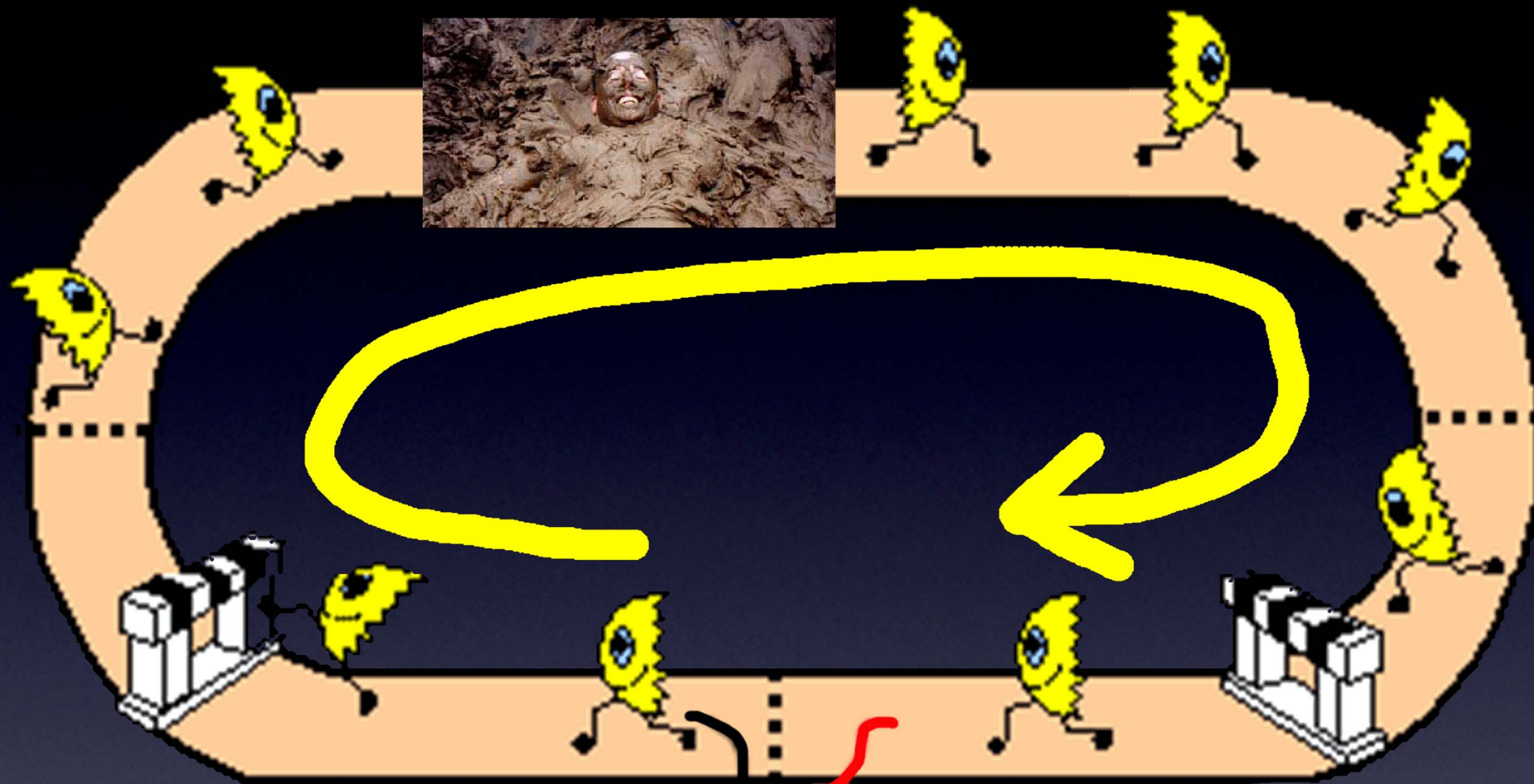


Resistance in the electrons' path slows them down, which means less current (less Amps).

Resistance / **Ohms**



# Everything You Need to Know About Electronics

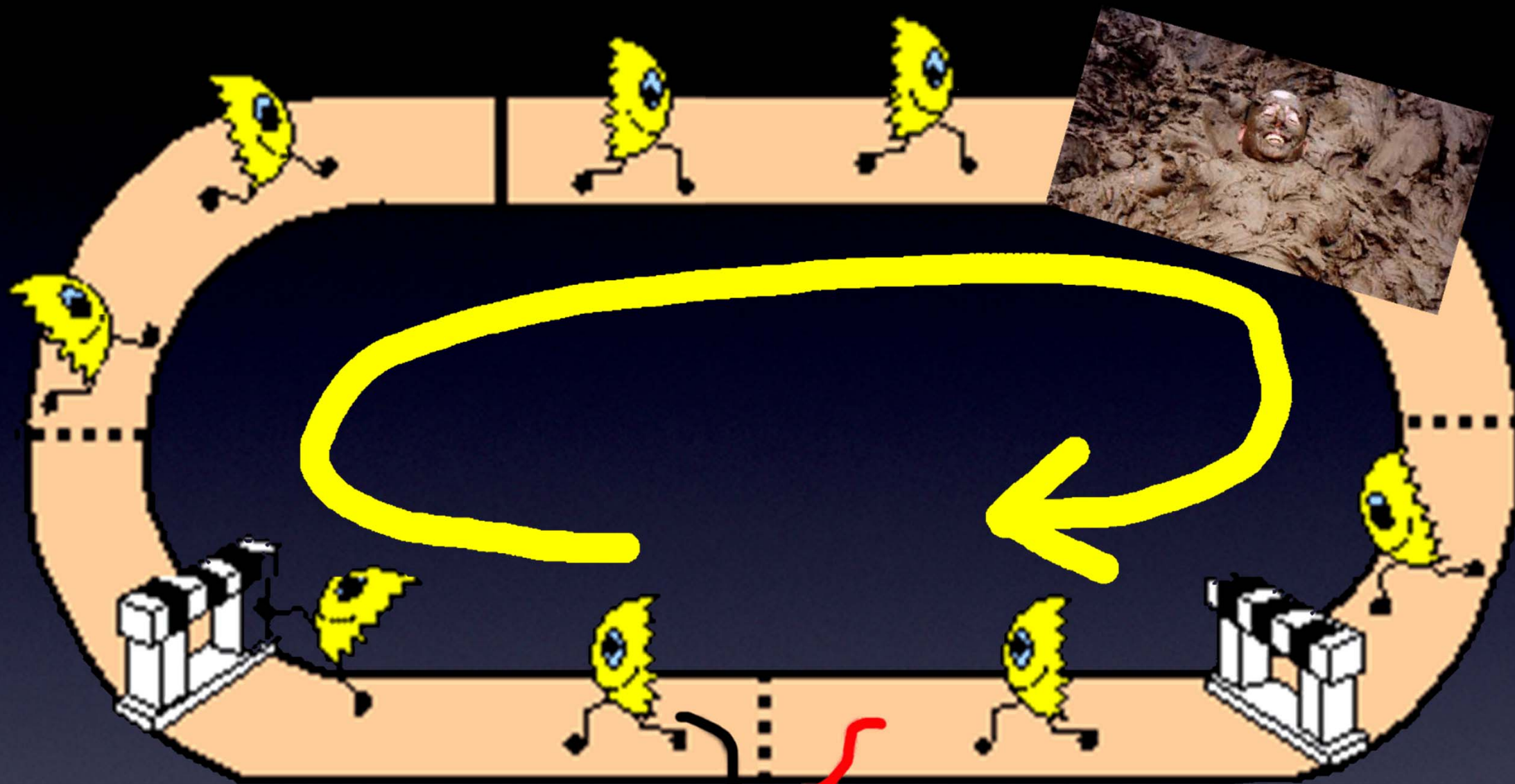


Resistance / Ohms

Same  
Circuit !



# Everything You Need to Know About Electronics

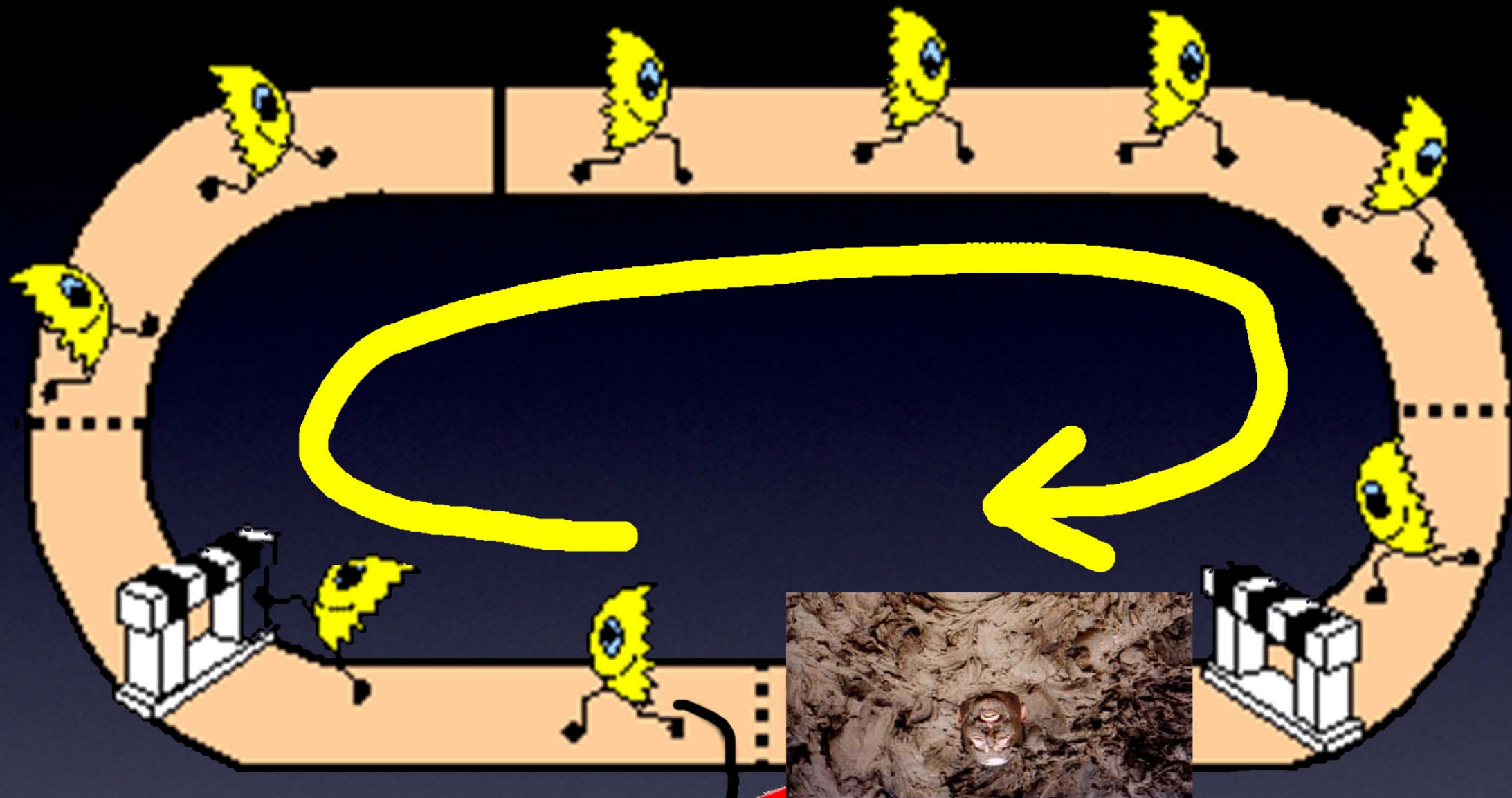


Resistance / Ohms

Same  
Circuit !



# Everything You Need to Know About Electronics

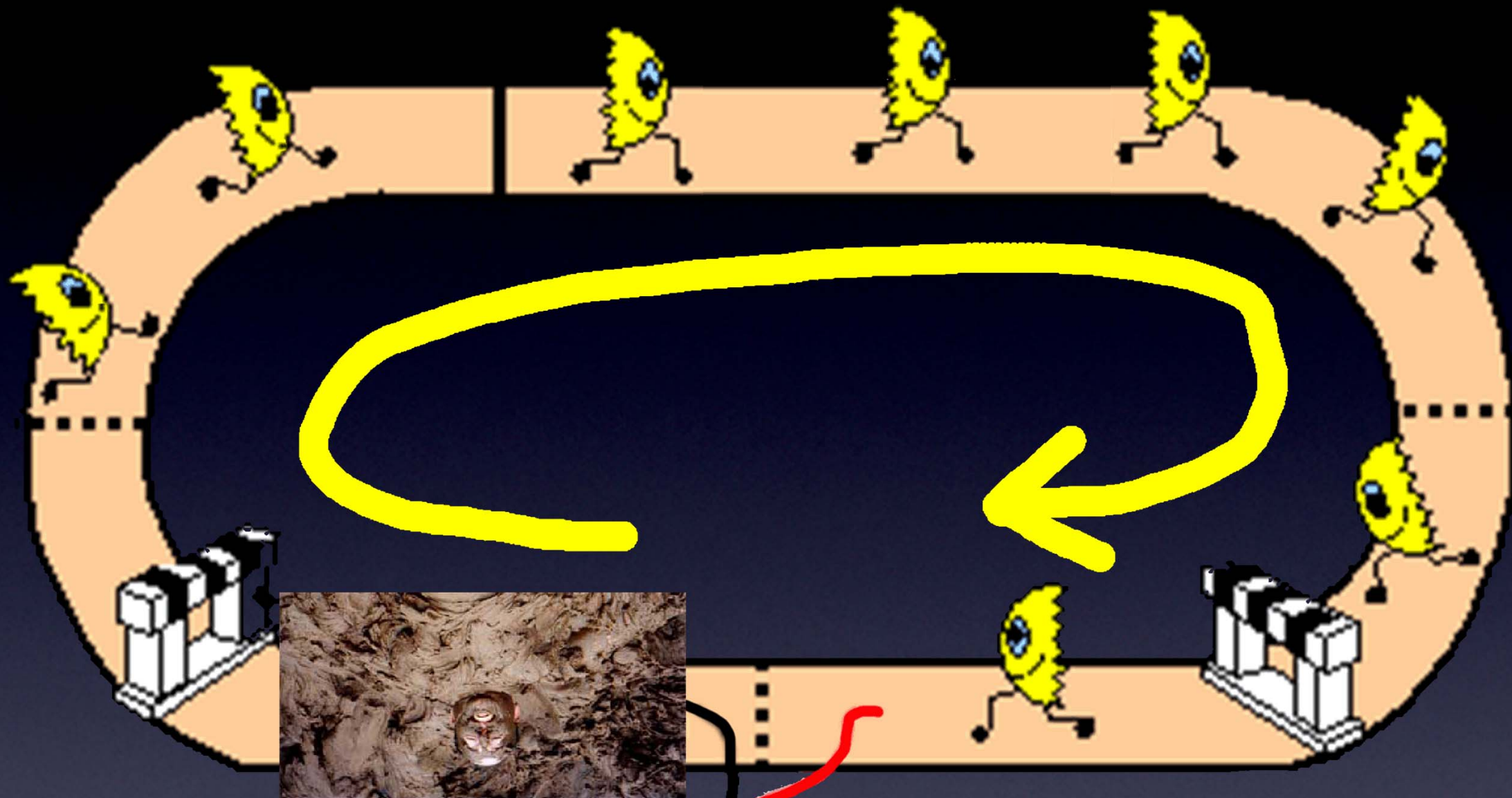


Resistance / Ohms

Same  
Circuit !



# Everything You Need to Know About Electronics



Resistance / Ohms

Same  
Circuit !



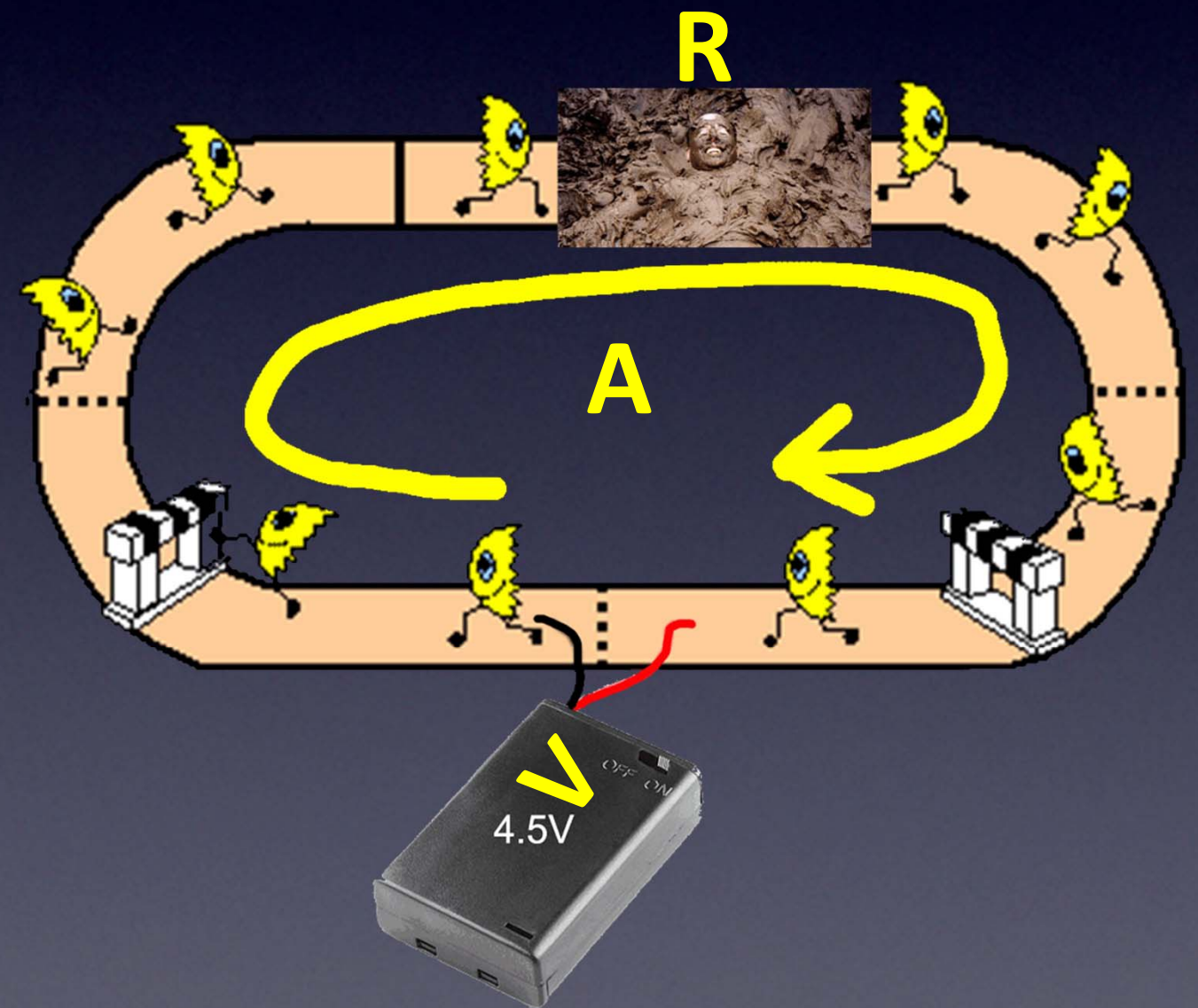
# Everything You Need to Know About Electronics

## Ohm's Law

**V**olts -- *force* pushing electrons

**A**mps -- *speed* of electrons

**O**hms -- *Resistance* to flow of electrons



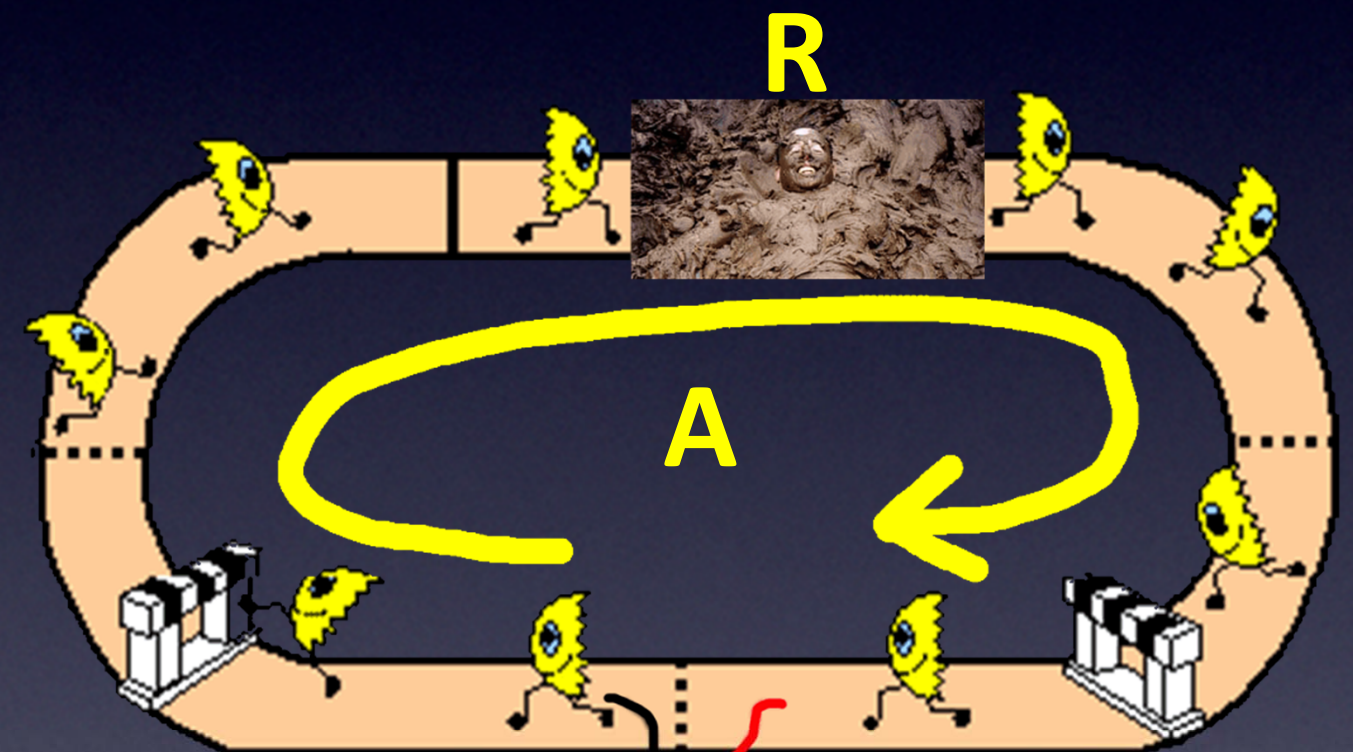
# Everything You Need to Know About Electronics

## Ohm's Law

**V**olts -- *force* pushing electrons

**A**mps -- *speed* of electrons

Ohms -- *Resistance* to flow of electrons



$$\mathbf{V_{olts} = A_{mps} \times R}$$



(Ohms)



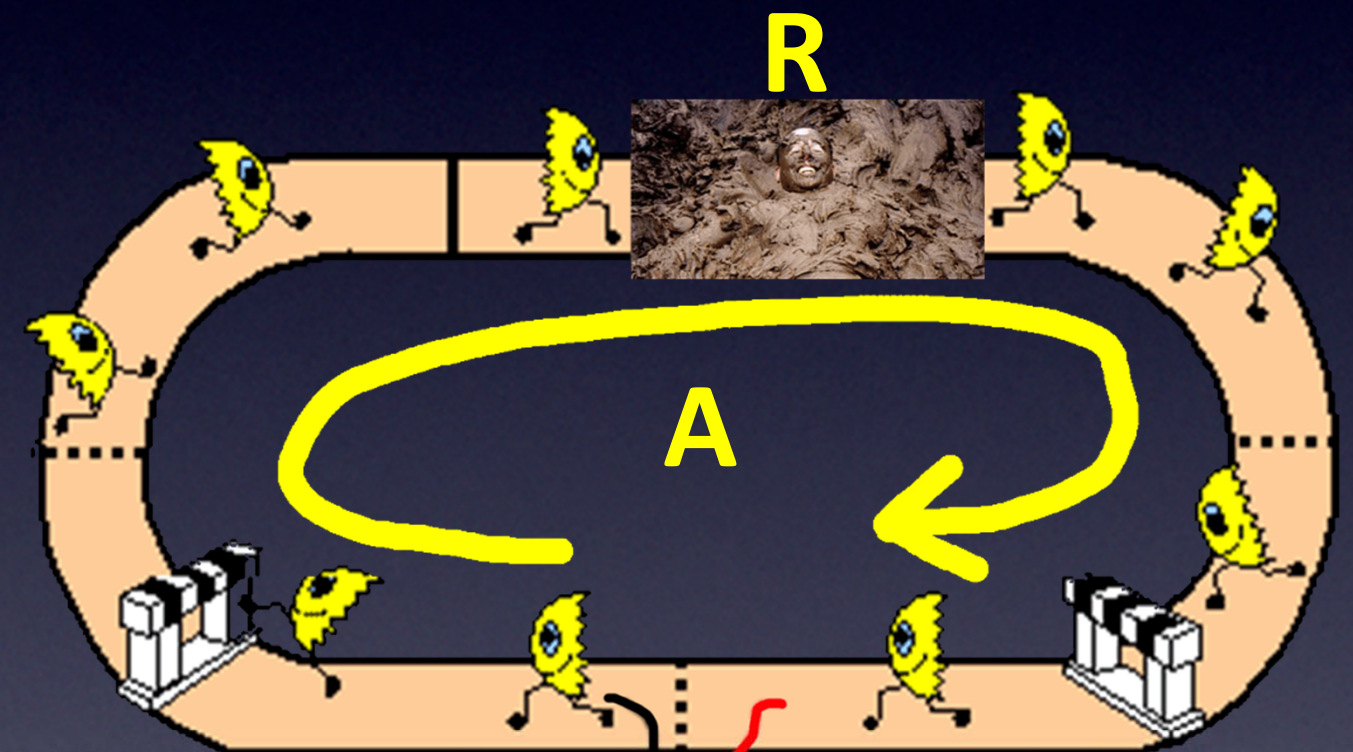
# Everything You Need to Know About Electronics

## Ohm's Law

**Volts** -- *force* pushing electrons

**Amps** -- *speed* of electrons

**Ohms** -- *Resistance* to flow of electrons



$$\text{Volts} = \text{Amps} \times R$$

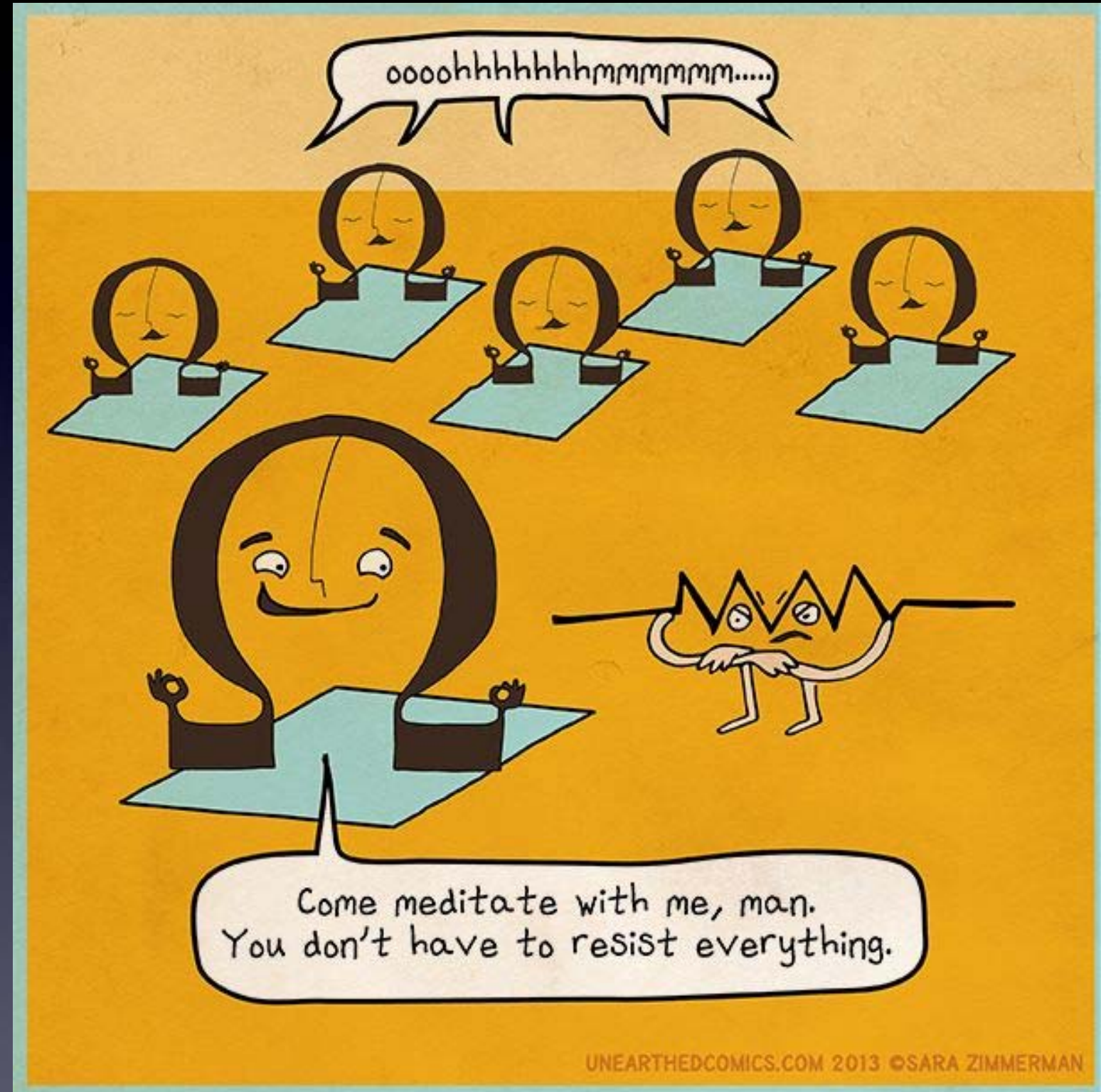
Also commonly written:  $E = I \times R$

(Ohms)

# Everything You Need to Know About Electronics

The symbol for  
**Resistance:**

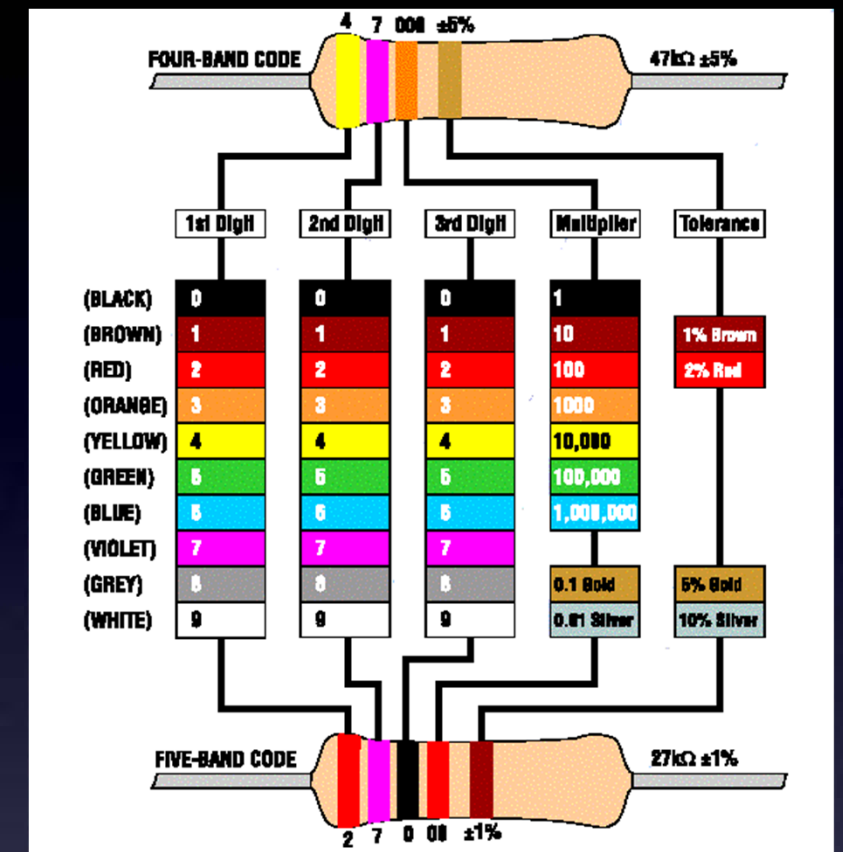
$\Omega$



Resistor / Ohms



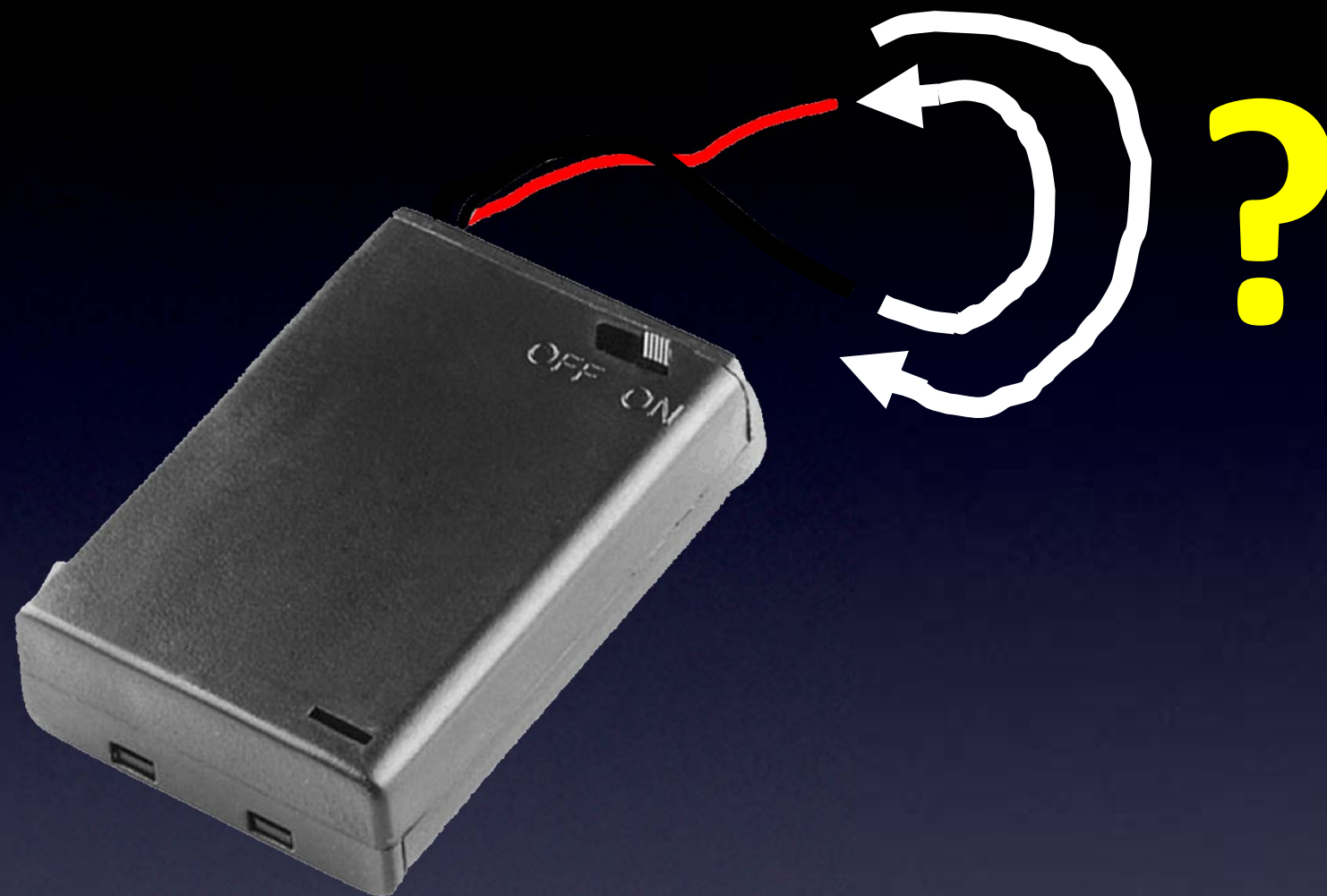
# What You Need to Know About Electronics



10KΩ: Brown, Black, Orange

Resistor / Ohms

# Everything You Need to Know About Electronics



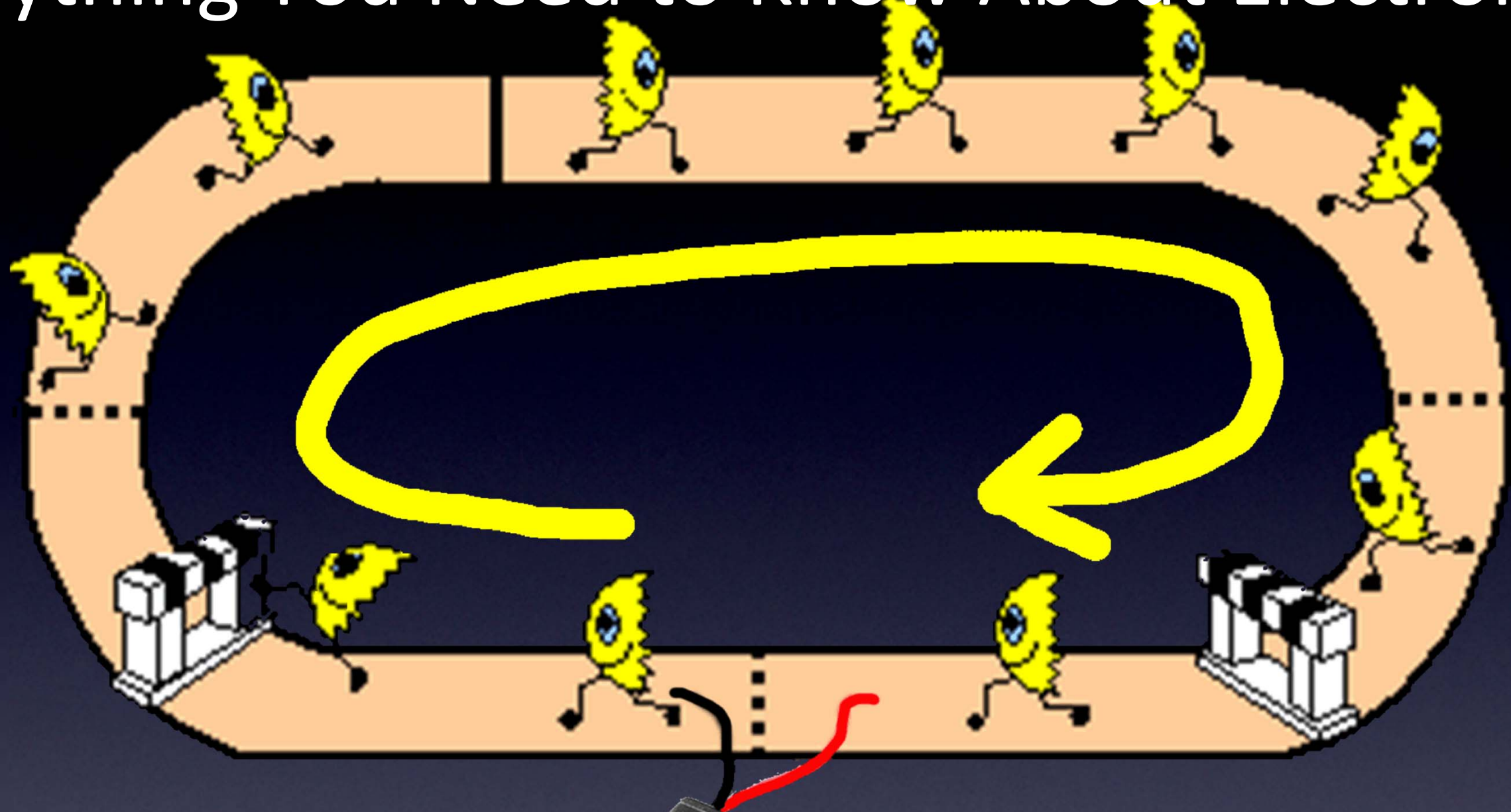
## What happens?

*polarity*

Power Supply – it matters how you connect it!



# Everything You Need to Know About Electronics



**Black Wire = “-”**

**Red Wire = “+”**



**Power Supply – it matters how you connect it!**

# Everything You Need to Know About Electronics



**Red wire:**  
**Power,**  
**Plus, Positive,**  
**4.5V,**  
**Vcc**

**Black wire:**  
**Minus, Negative,**  
**0V,**  
**Ground (GND)**

Power Supply – it matters how you connect it!



# Everything You Need to Know About Electronics



or



or

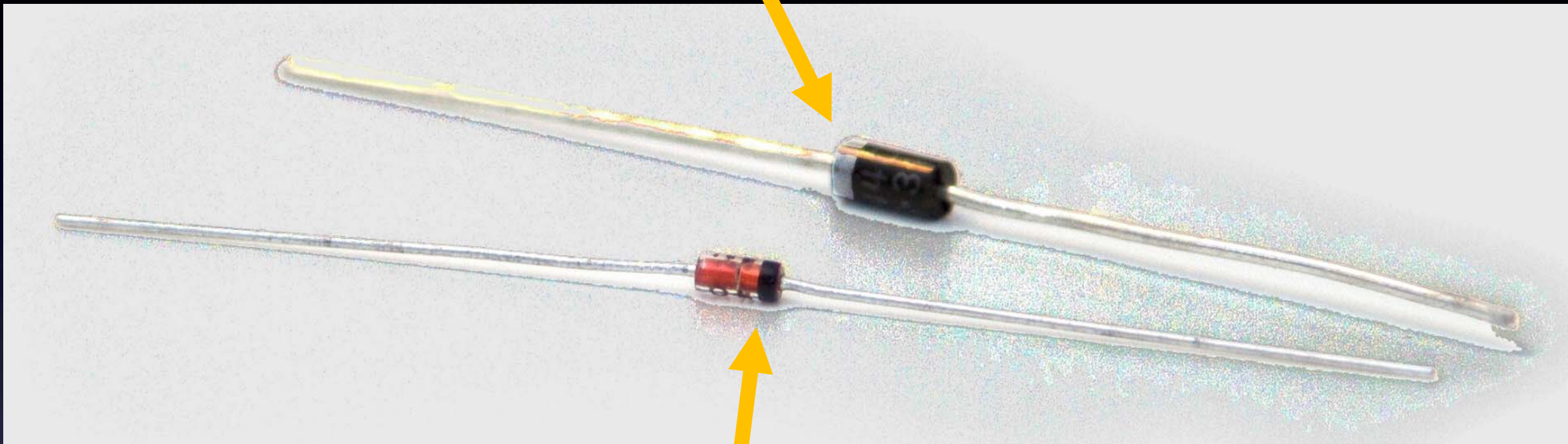


(electrons slowed down the same either way)

Resistors – it doesn't matter which way

# Everything You Need to Know About Electronics

**Minus / Negative side**



**Minus / Negative side**

**Diodes – One-Way valve for electrons**

Diodes – it matters which way!



# Everything You Need to Know About Electronics



**Short wire is Minus / Negative**

**Special kind of Diode – it Emits Light!**

LED – it matters which way!

# Everything You Need to Know About Electronics

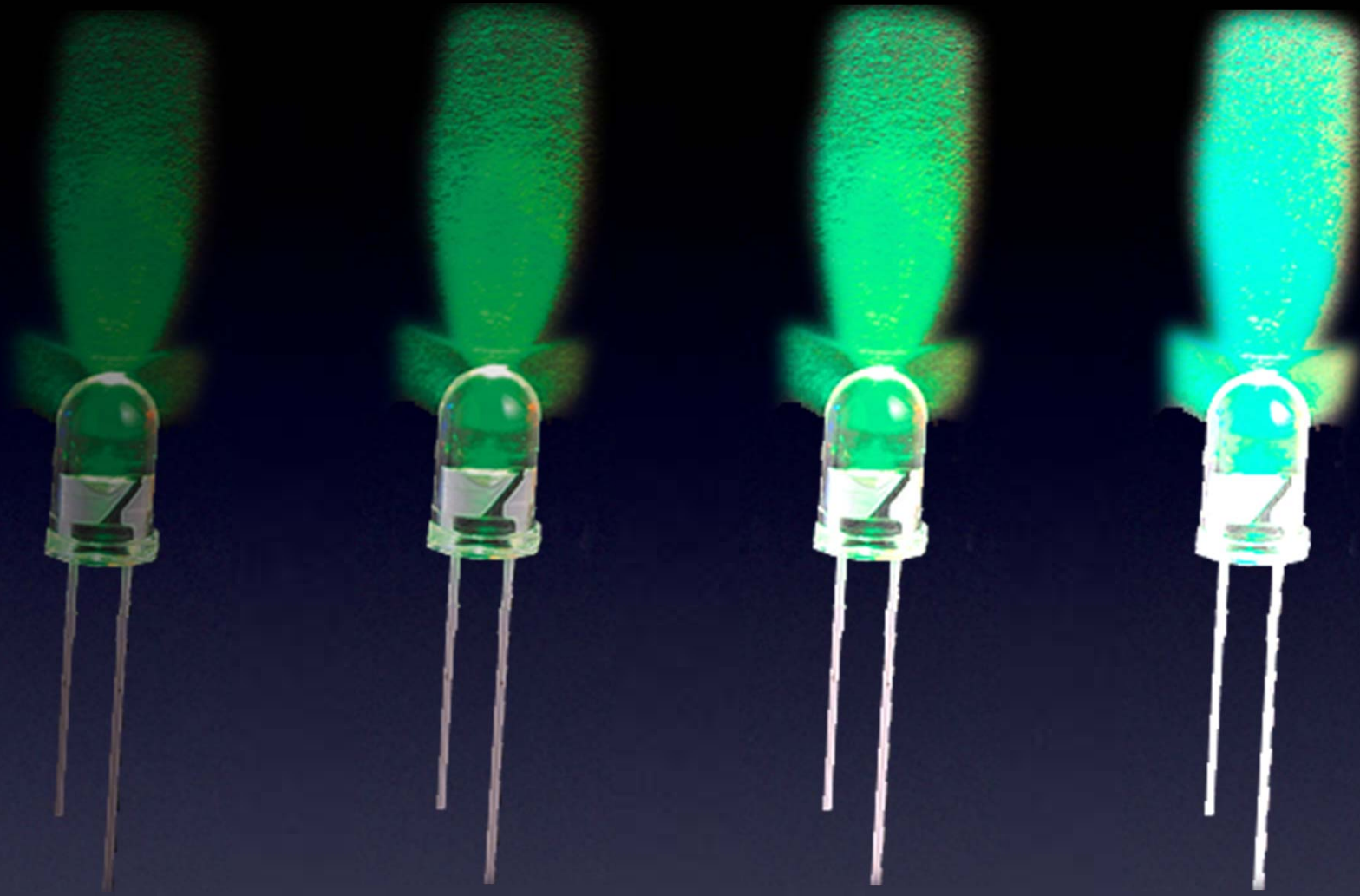


Lots of different colored LEDs! (including IR)

LED



# Everything You Need to Know About Electronics



More current → More brightness! (until...)

LED

# Everything You Need to Know About Electronics

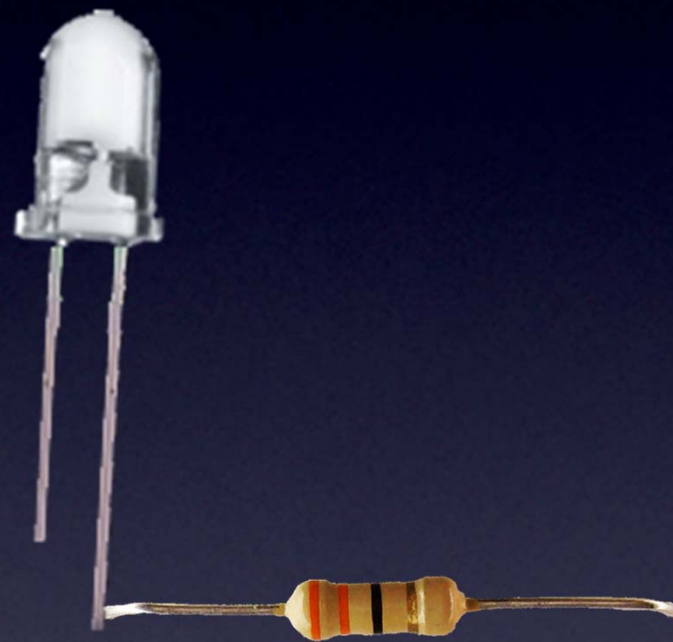


More current → More brightness! (until...)

LED



# Everything You Need to Know About Electronics

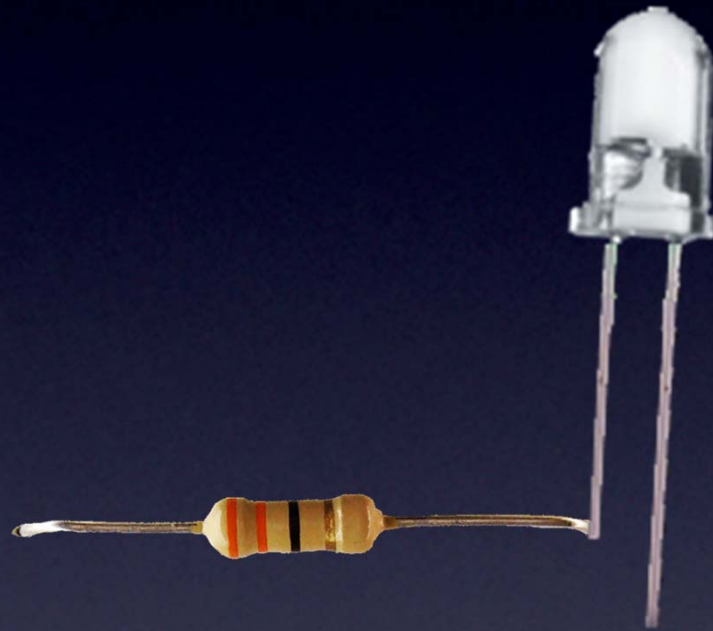


*(with a resistor  
so no magic smoke goes away)*

This is why we put a resistor in line with an LED

LED

# Everything You Need to Know About Electronics



*(the resistor can go on either side)*

This is why we put a resistor in line with an LED

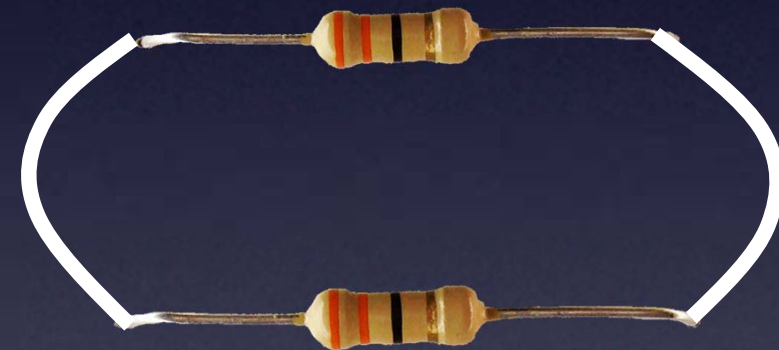
LED



# Everything You Need to Know About Electronics

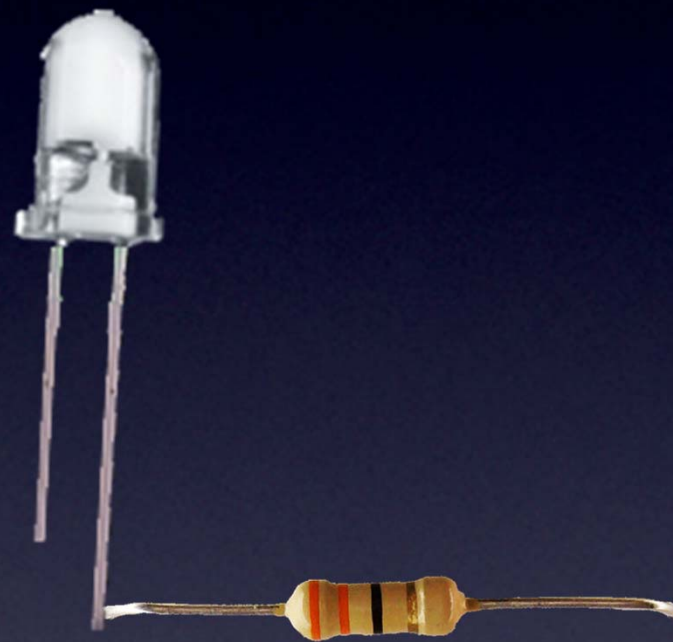


Series = in line



Parallel = across

# Everything You Need to Know About Electronics

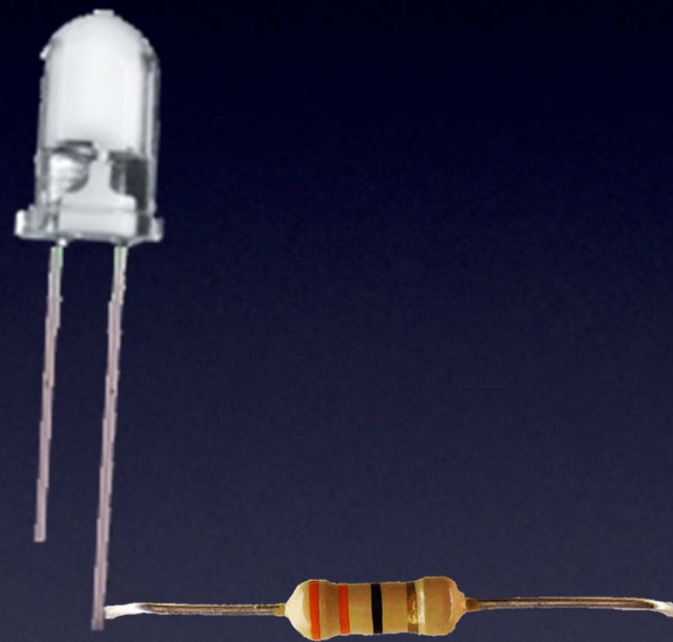
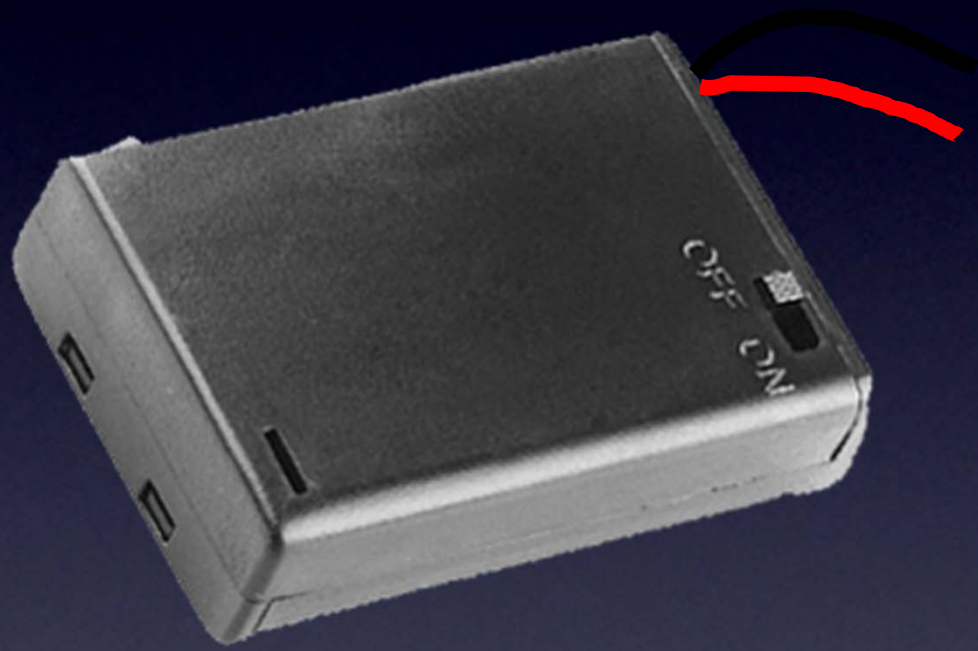


Let's make this light up!

LED



# Everything You Need to Know About Electronics

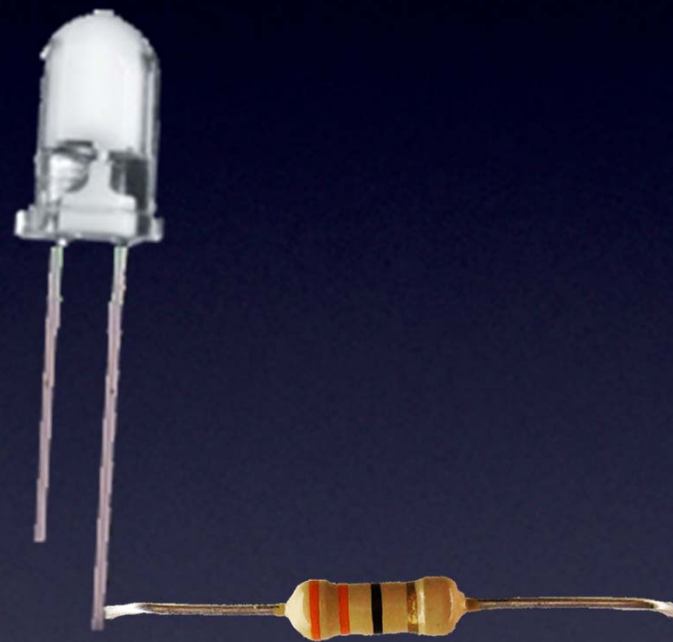
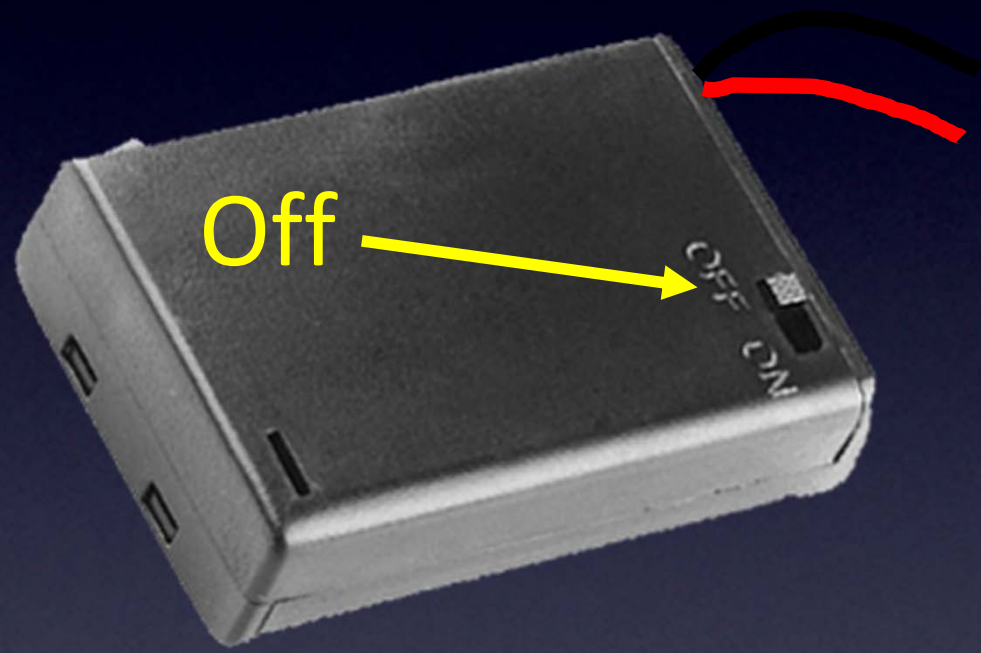


*(add a power supply)*

## Let's make this light up!

LED

# Everything You Need to Know About Electronics



Let's make this light up!

LED

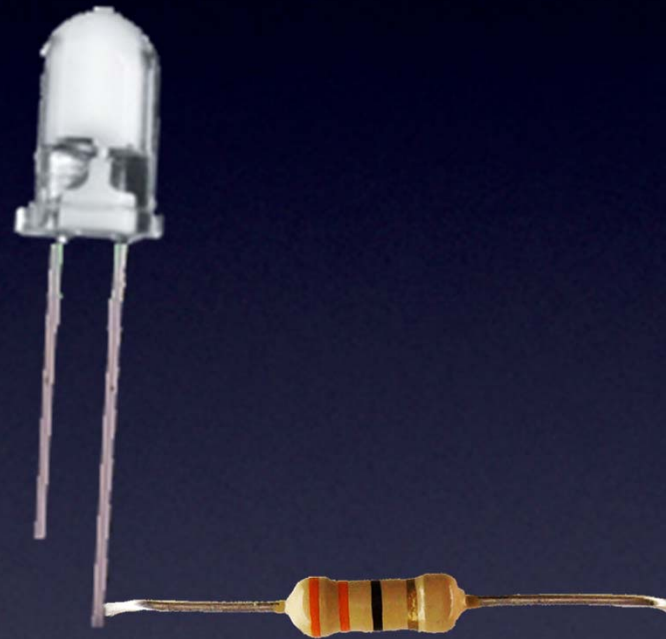


# Everything You Need to Know About Electronics

Black wire: “-” (ground)



Red wire: “+” power)



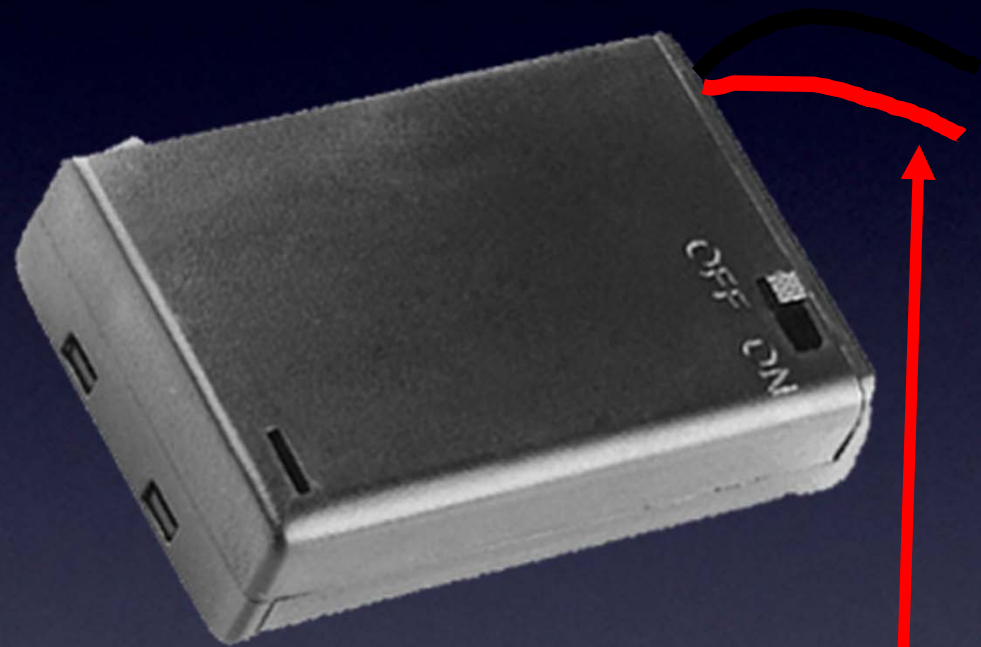
## Let's make this light up!

LED

# Everything You Need to Know About Electronics

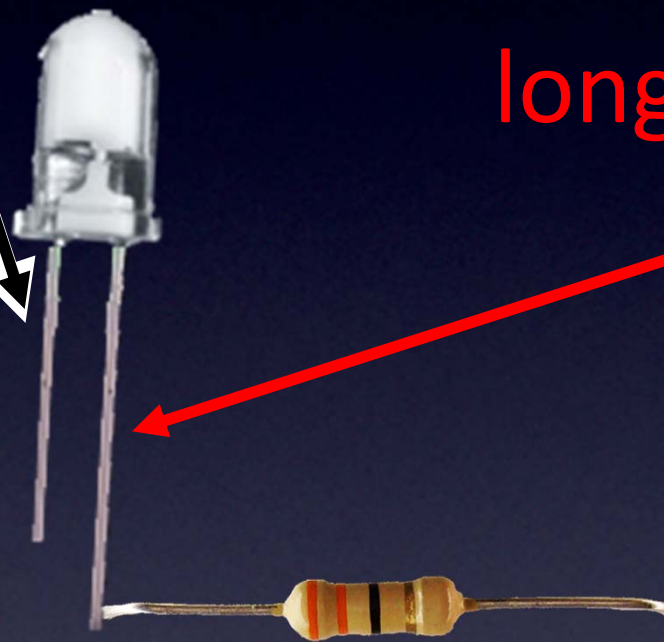
short lead: “-”

Black wire: “-” (ground)



Red wire: “+” power)

long lead: “+”



## Let's make this light up!

LED

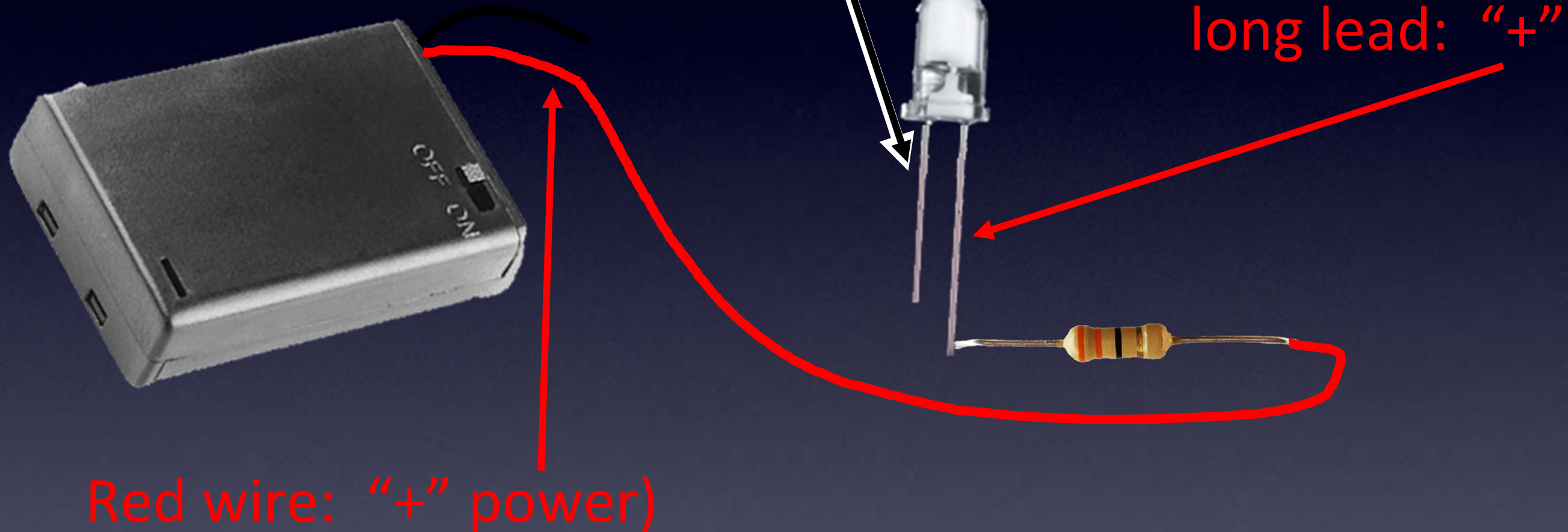


# Everything You Need to Know About Electronics

short lead: “-”

Black wire: “-” (ground)

long lead: “+”



## Let's make this light up!

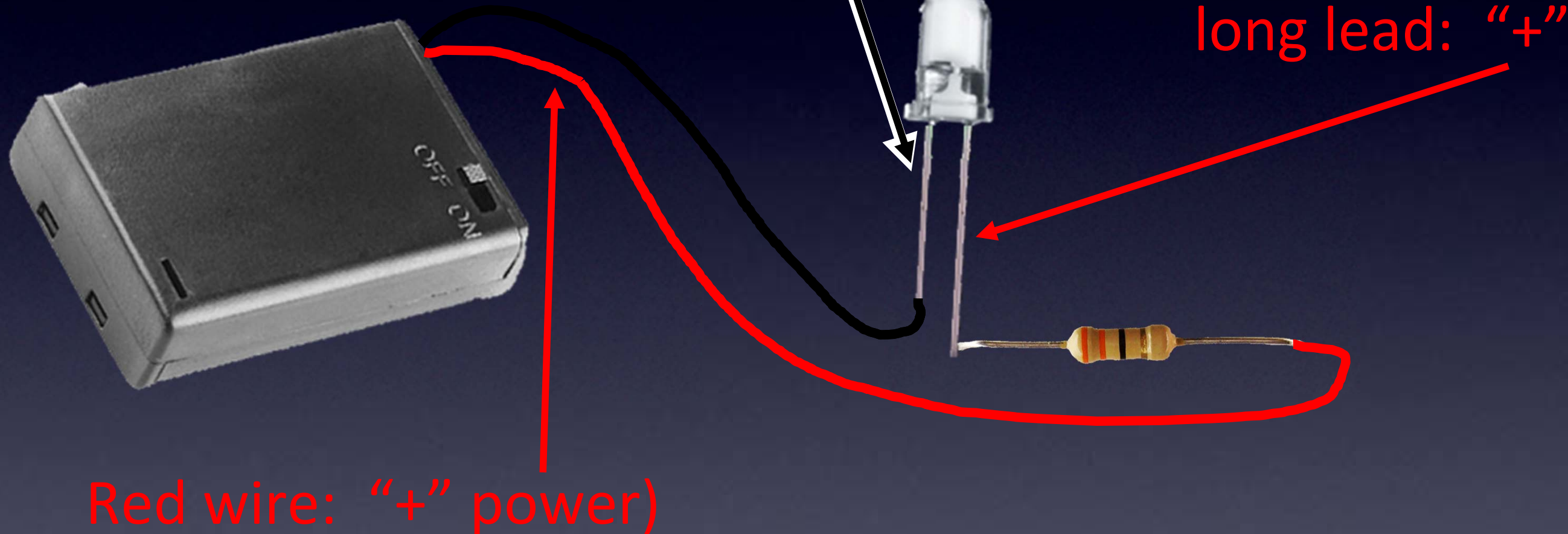
LED

# Everything You Need to Know About Electronics

short lead: “-”

Black wire: “-” (ground)

long lead: “+”

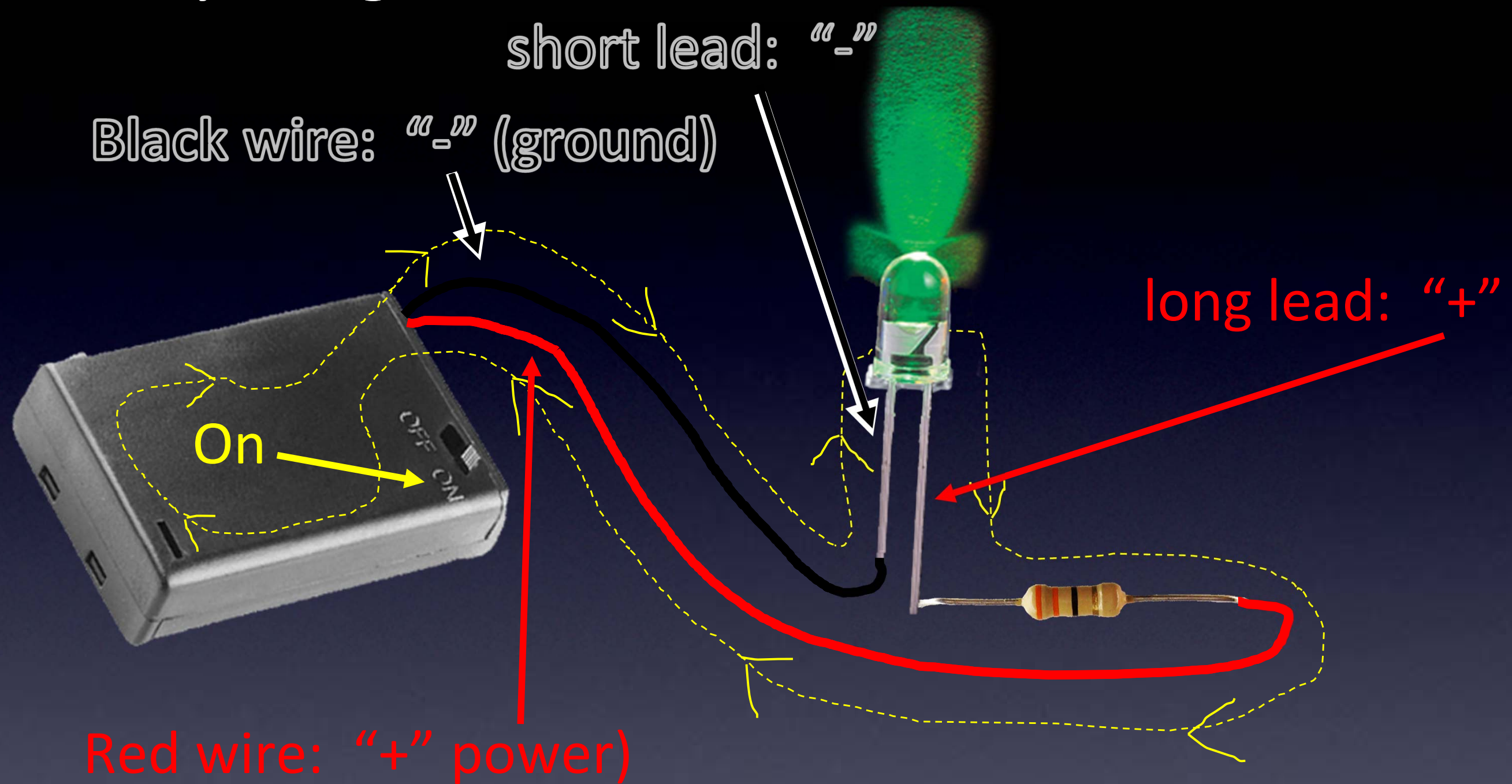


## Let's make this light up!

LED



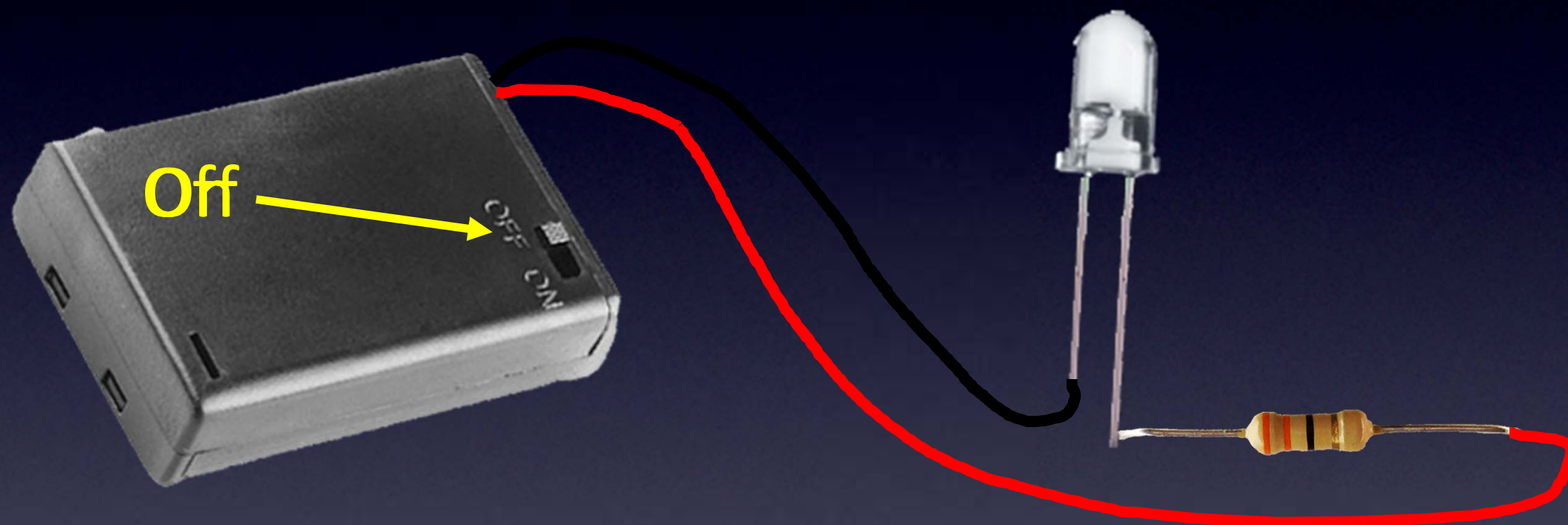
# Everything You Need to Know About Electronics



It lights!

LED

# Everything You Need to Know About Electronics



It's off

LED



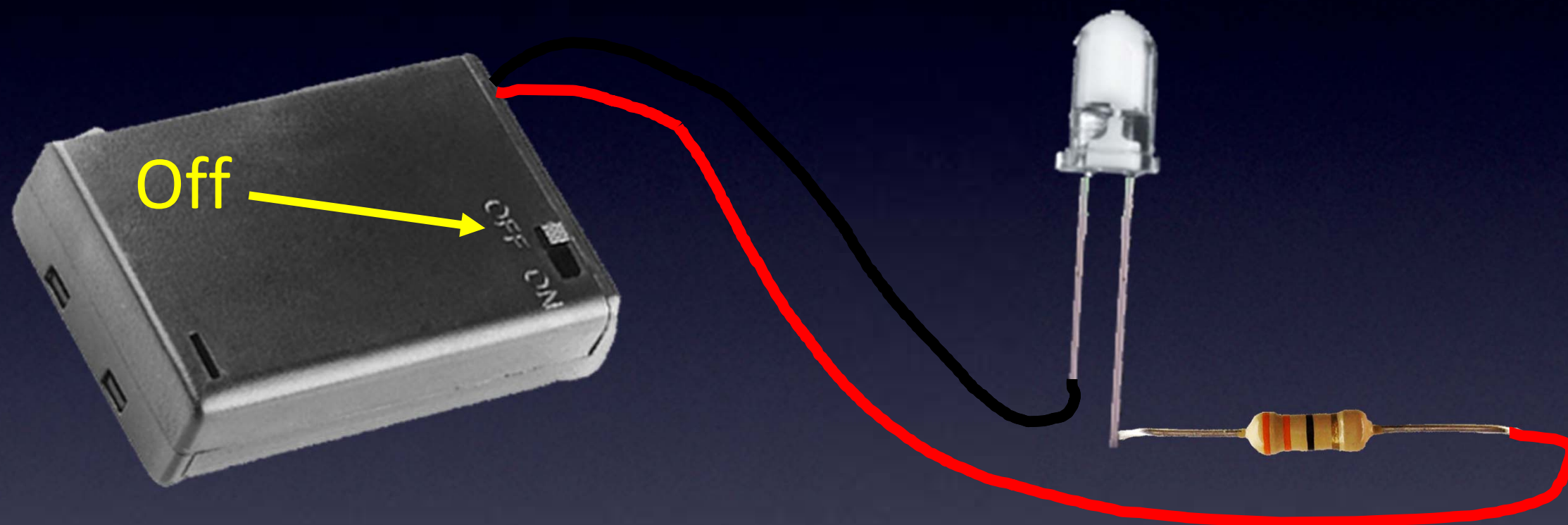
# Everything You Need to Know About Electronics



LED & battery

Our first circuit

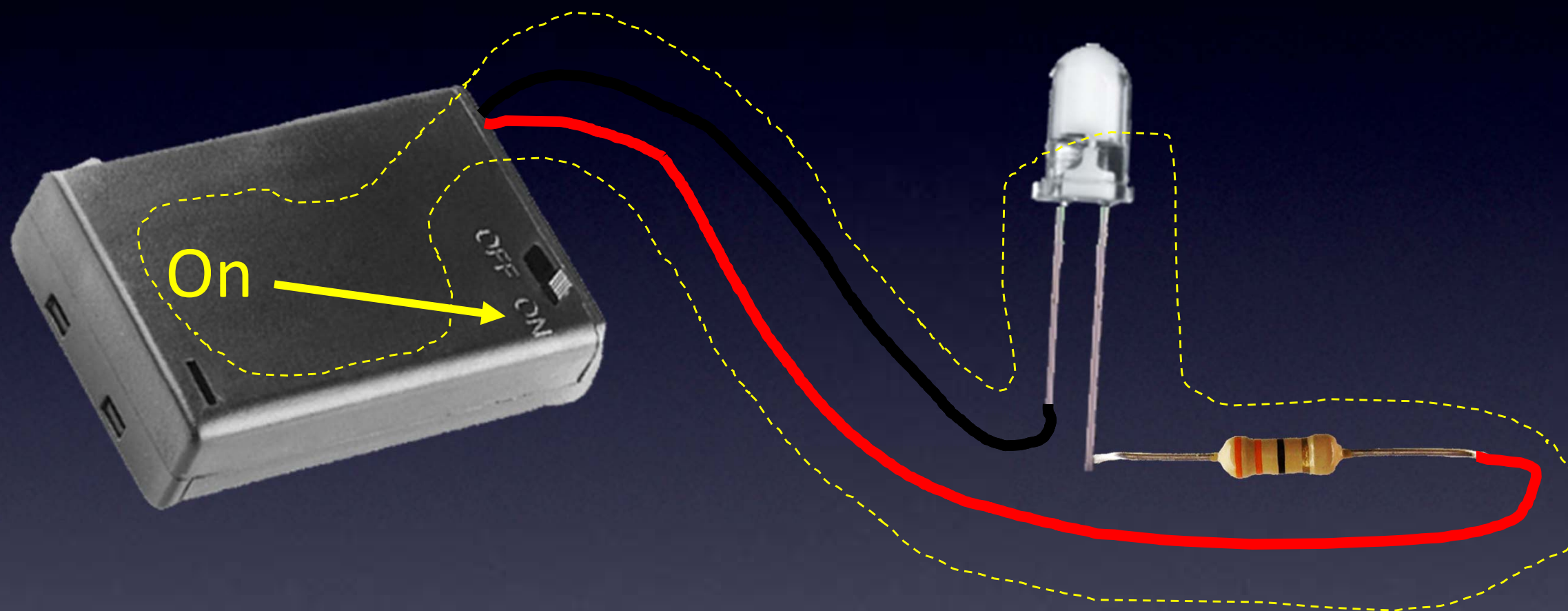
# Everything You Need to Know About Electronics



IR LED

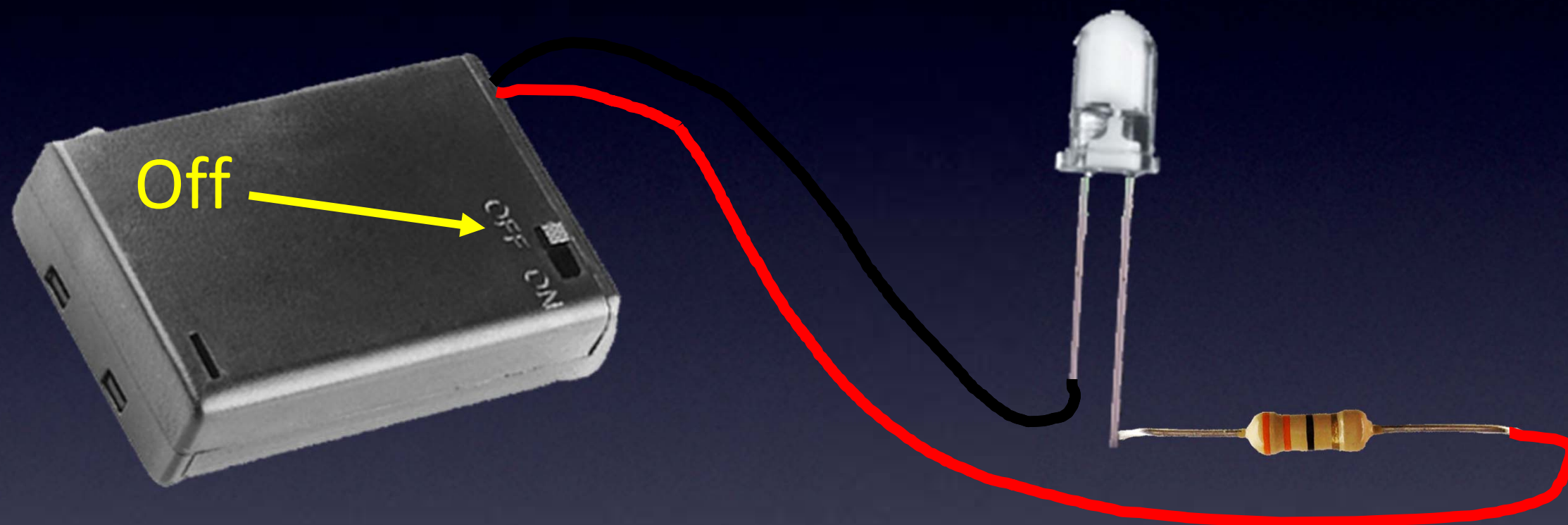


# Everything You Need to Know About Electronics



IR LED

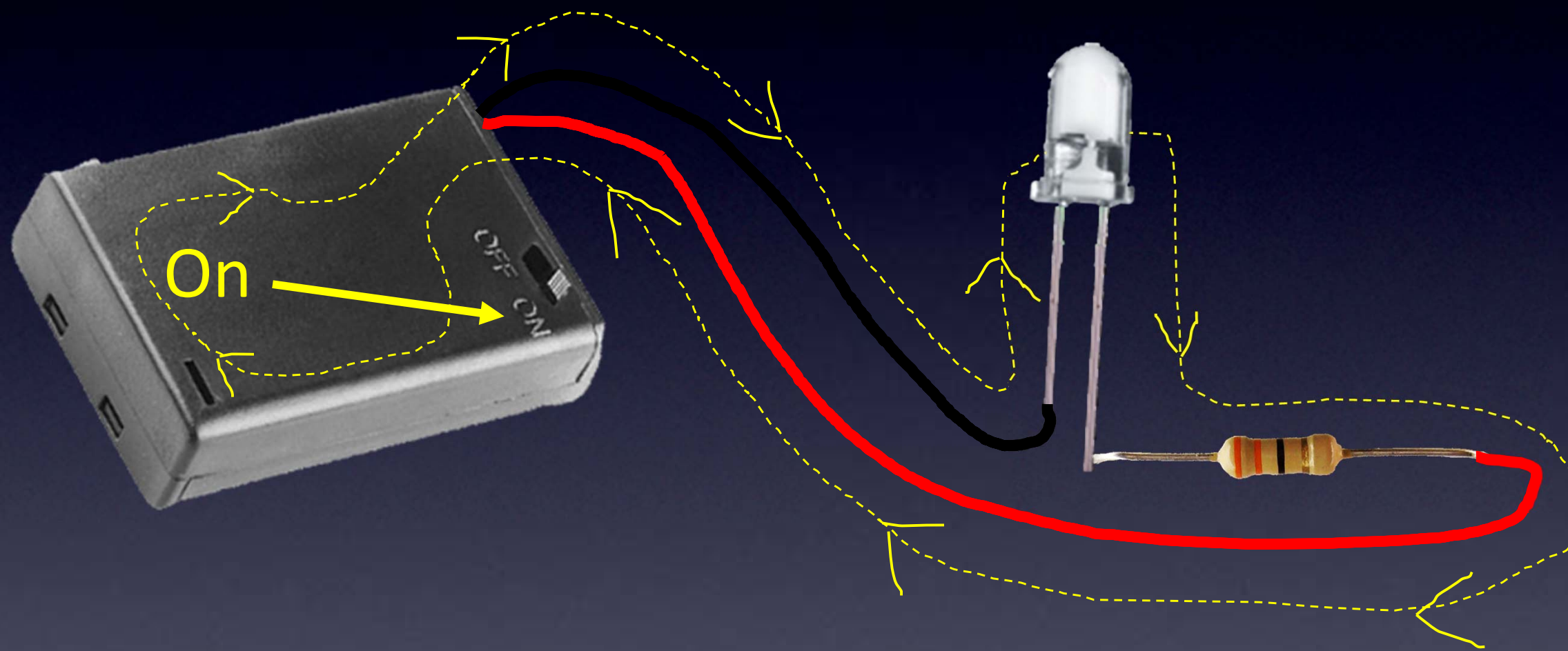
# Everything You Need to Know About Electronics



IR LED

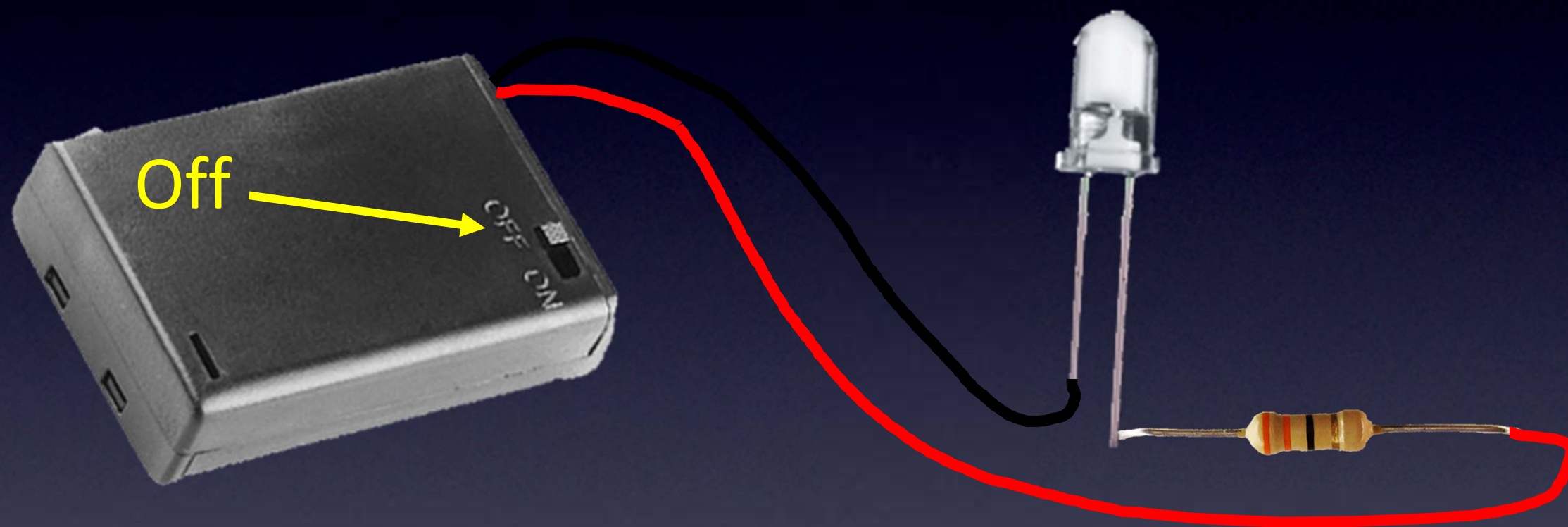


# Everything You Need to Know About Electronics



IR LED

# Everything You Need to Know About Electronics

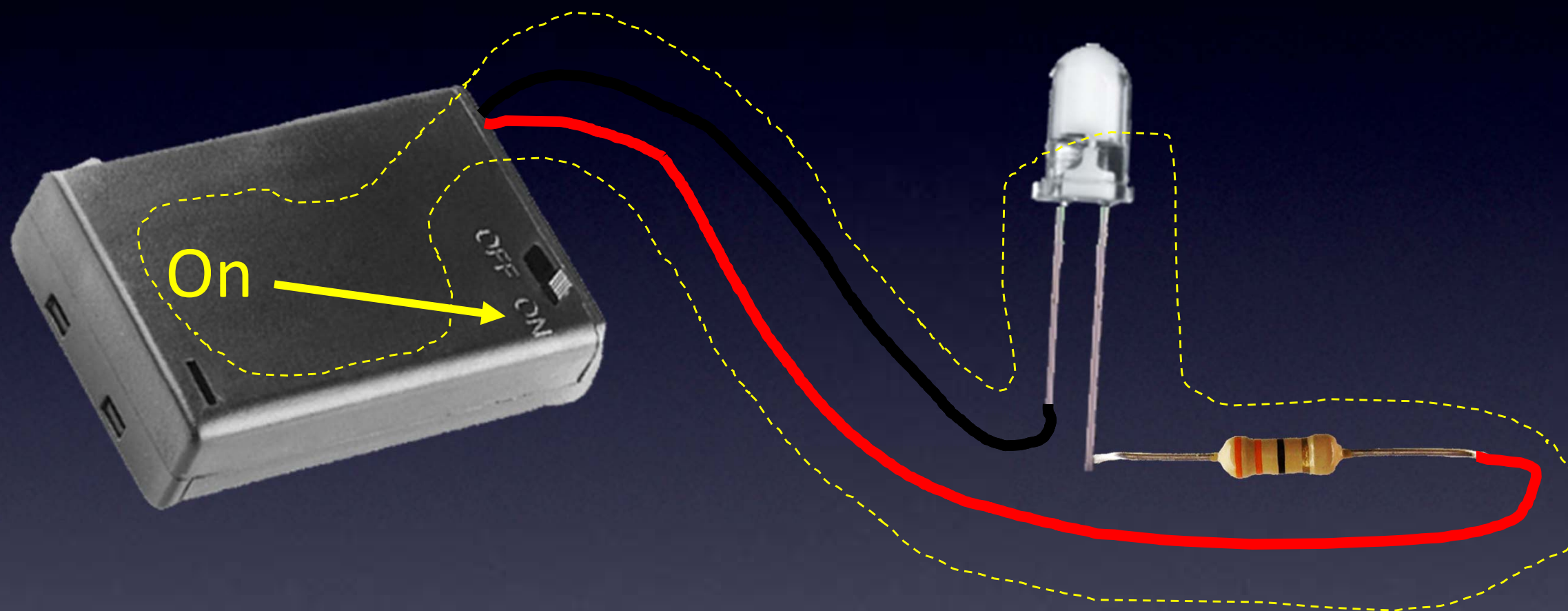


A “code” is IR light blinking on-off-on-off...

IR Remote Control



# Everything You Need to Know About Electronics



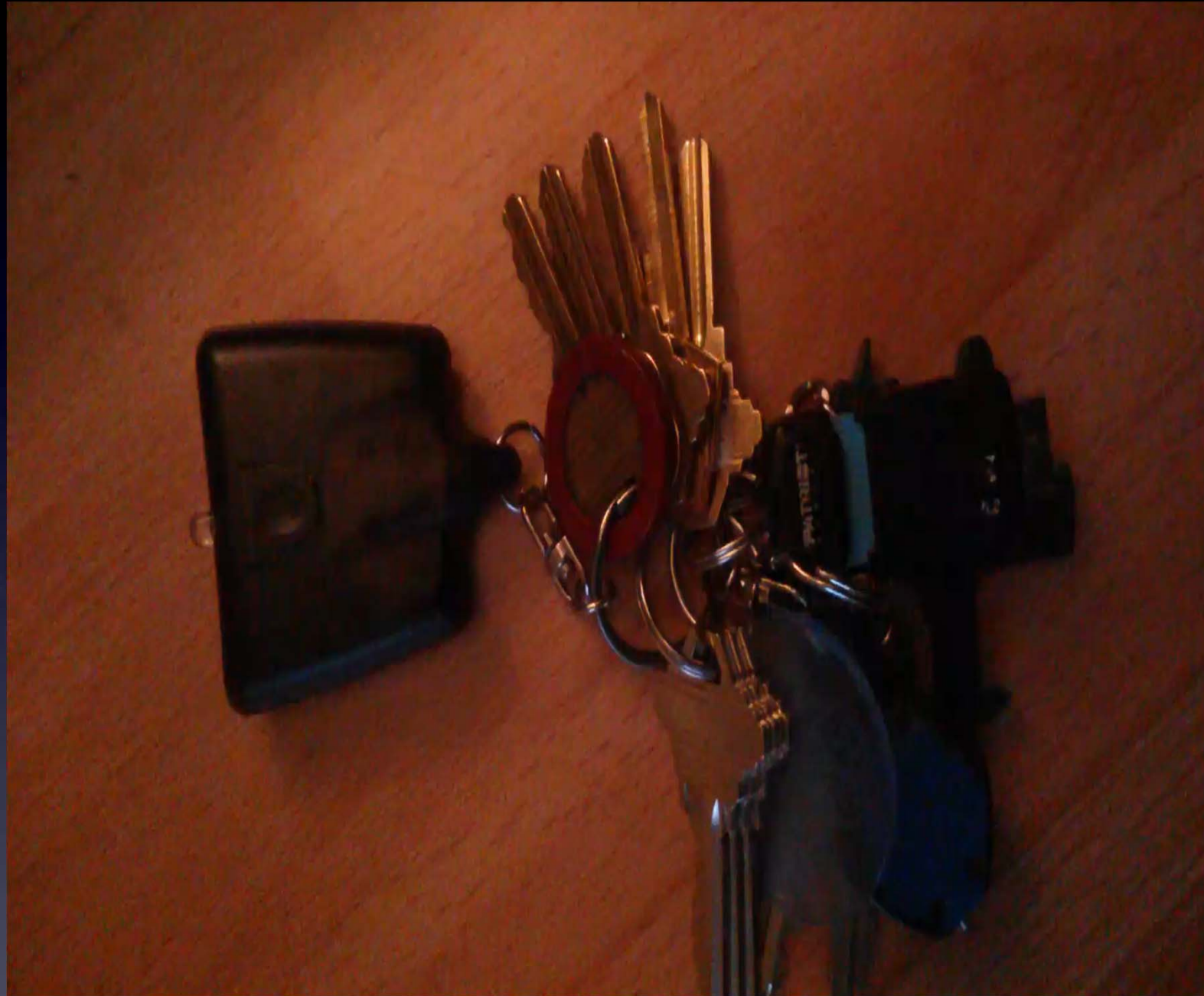
A “code” is IR light blinking on-off-on-off...

*(we can't do this, but microcontrollers can!)*

## IR Remote Control

# Everything You Need to Know About Electronics

Takes about 60 seconds

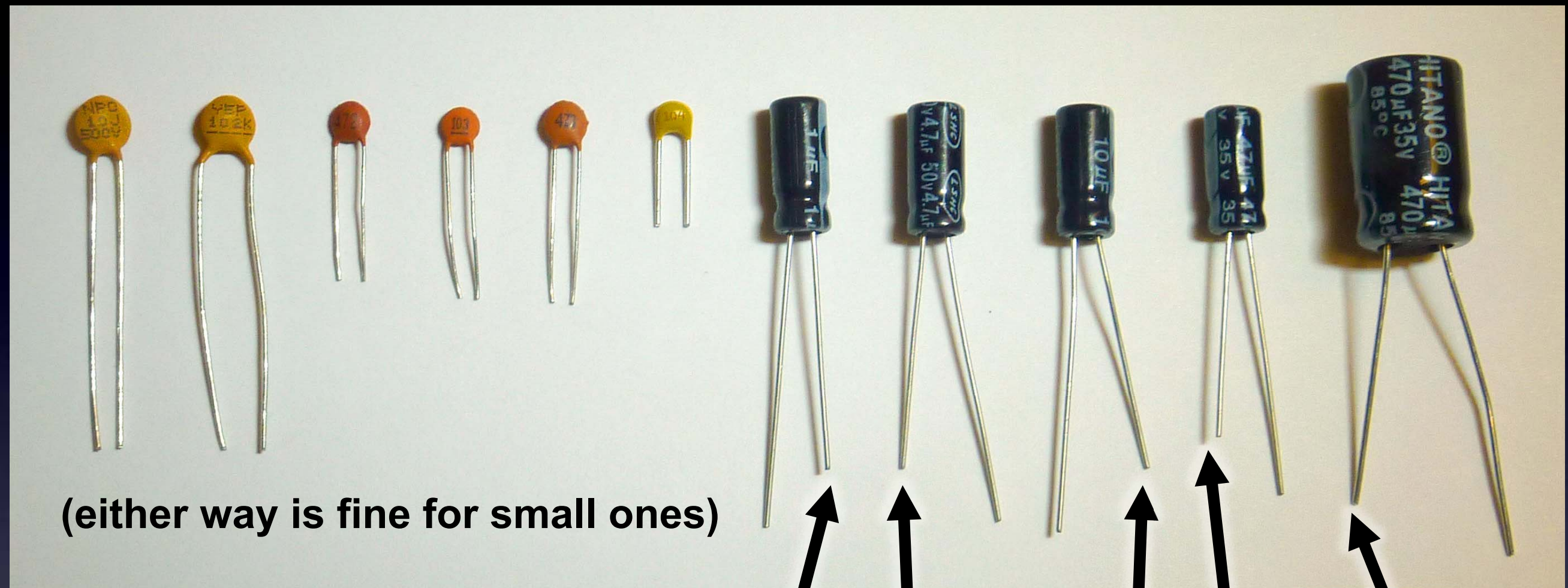


About 150 IR “OFF” codes (one per blink)

TV-B-Gone universal remote control



# Everything You Need to Know About Electronics



(either way is fine for small ones)

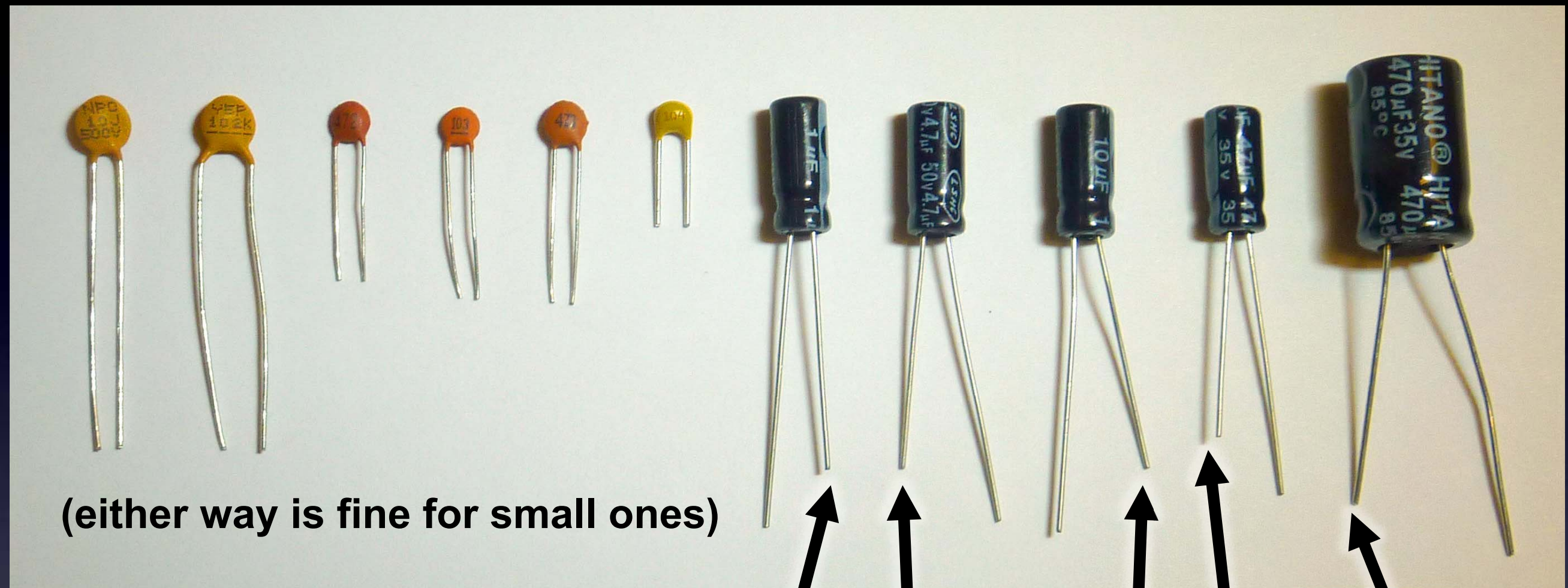
Short wire is Minus / Negative

Little buckets for electrons

Capacitor / Farads



# Everything You Need to Know About Electronics



(either way is fine for small ones)

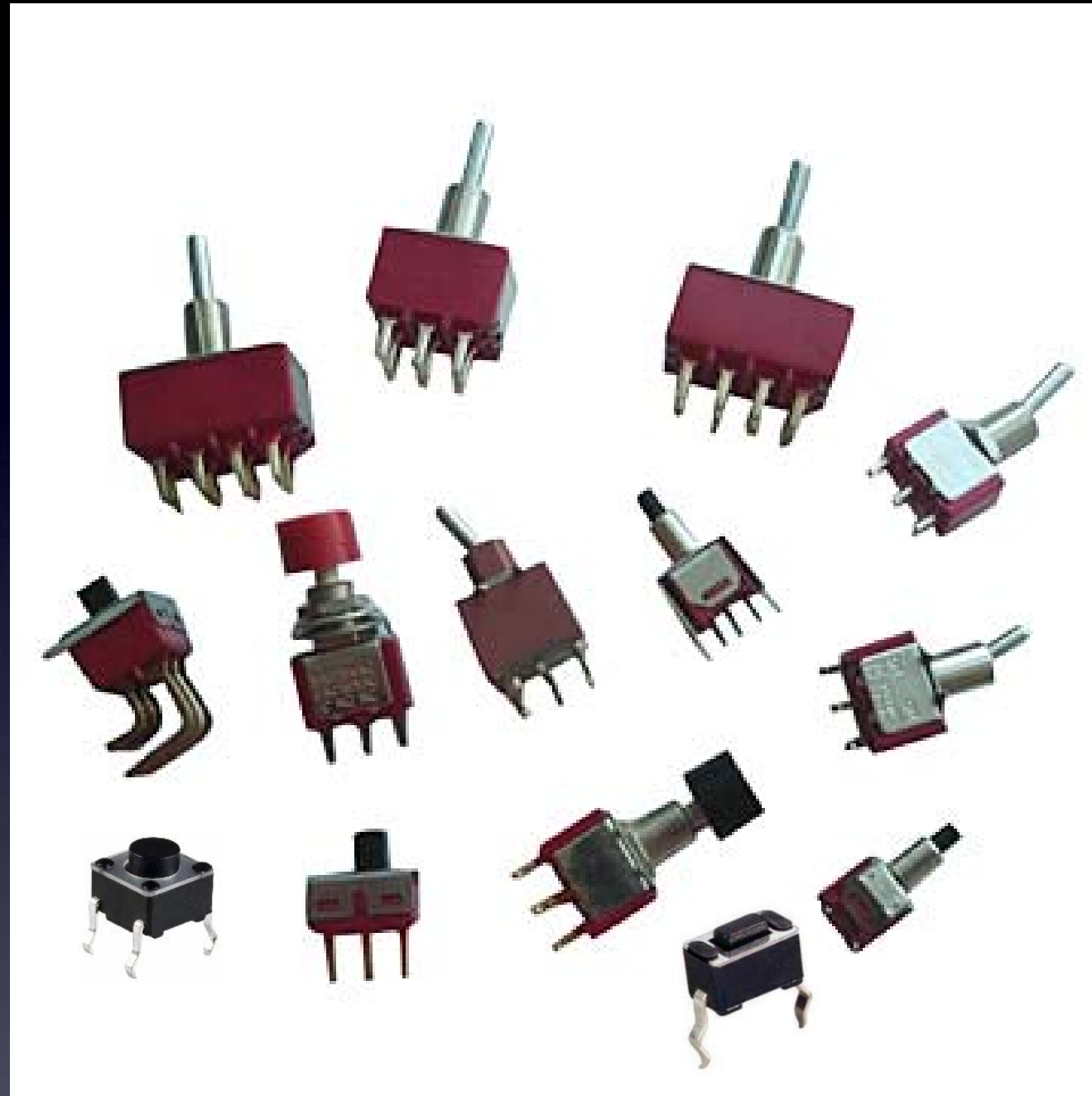
**Short wire is Minus / Negative**

**Little buckets for electrons**

Capacitor / **Farads**



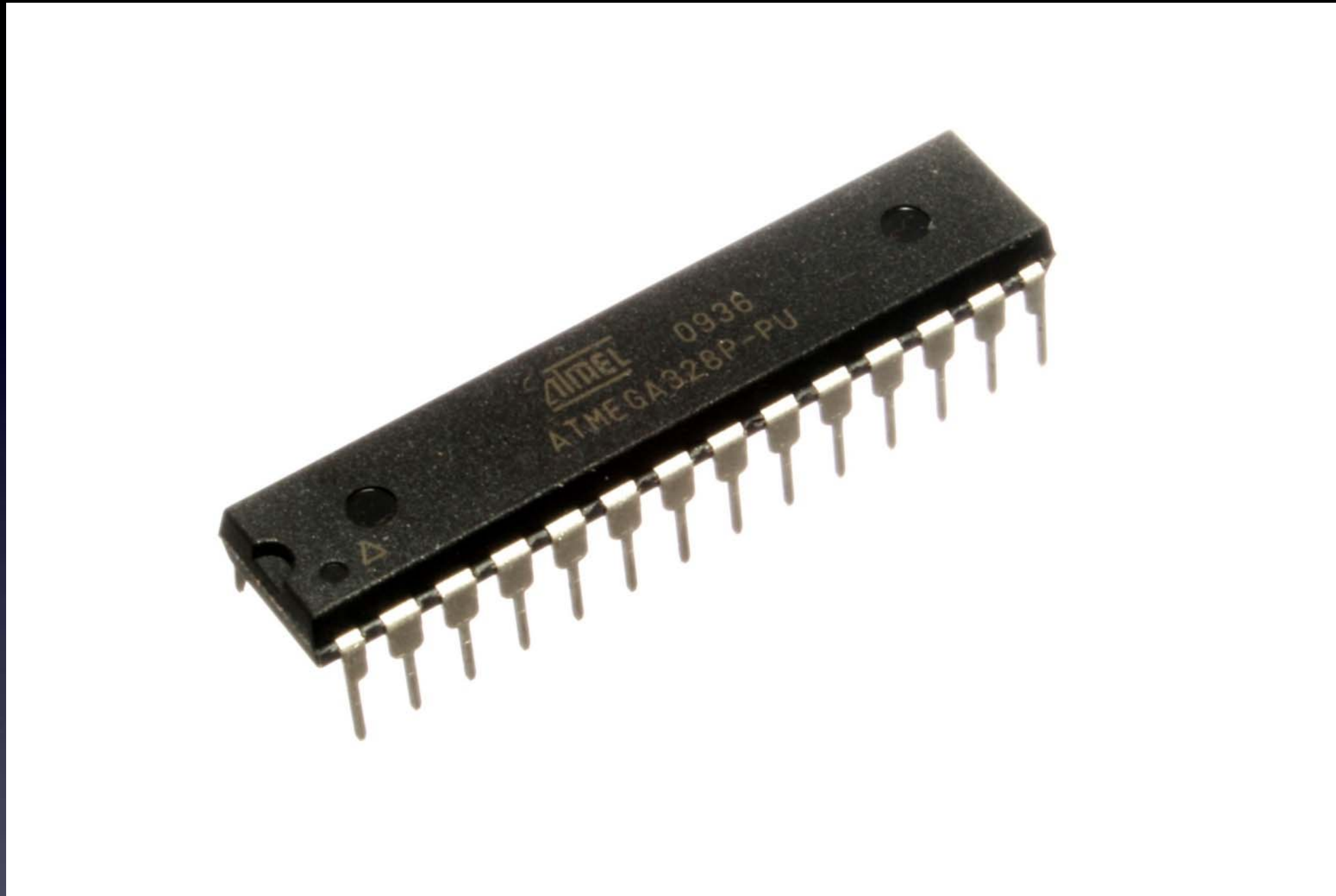
# Everything You Need to Know About Electronics



Strips of metal connected together – or not

Switch

# Everything You Need to Know About Electronics

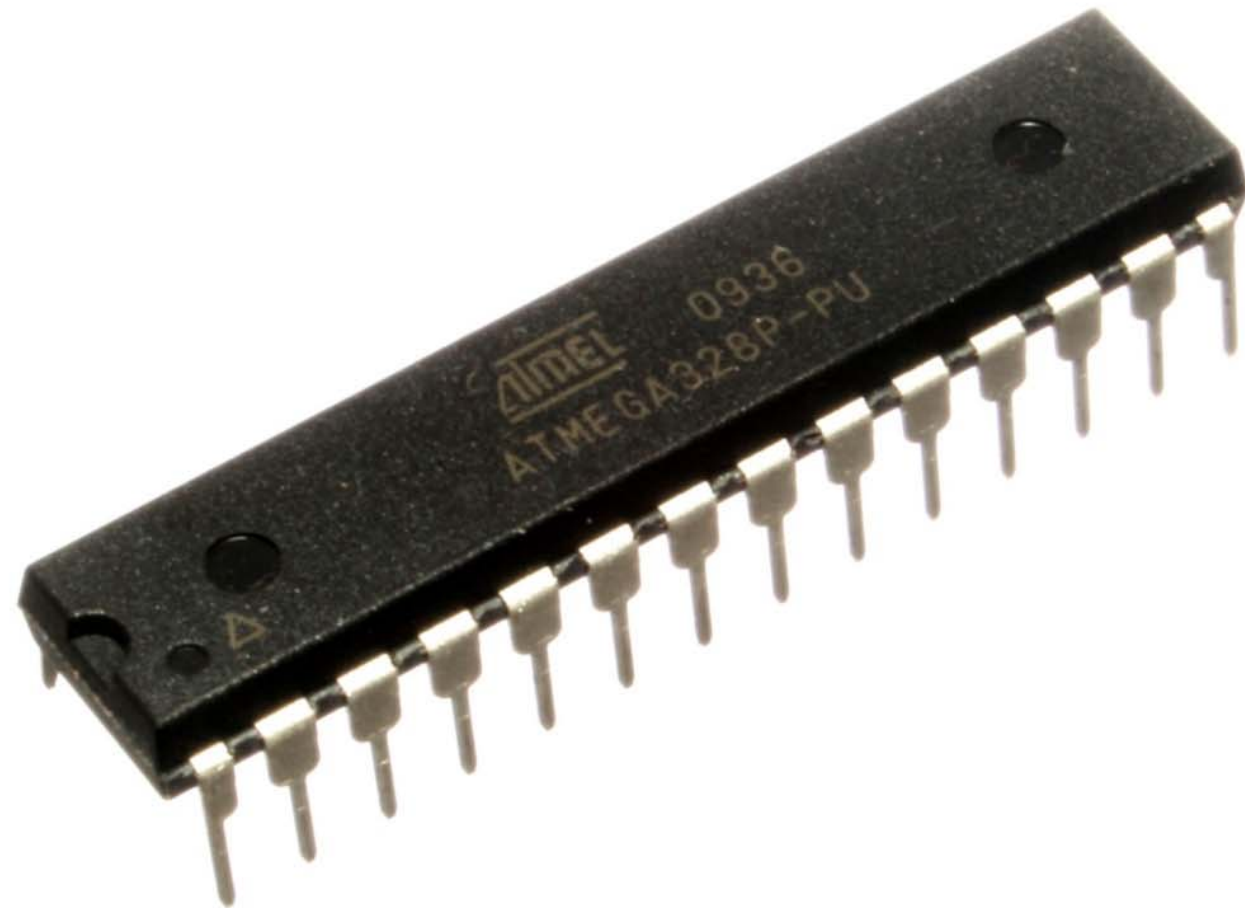


A complete computer on a chip

Microcontroller



# Everything You Need to Know About Electronics



**It runs programs**

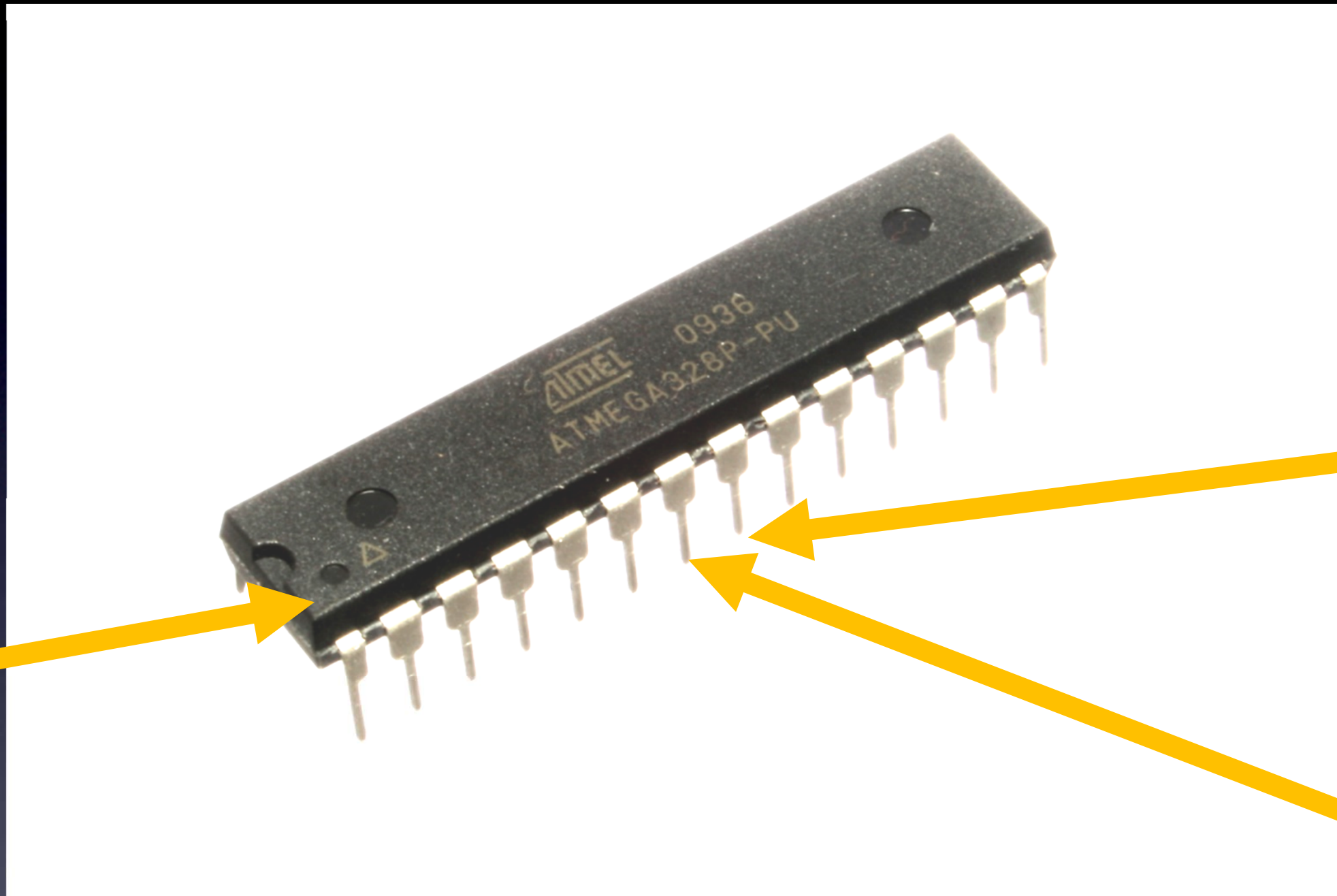
that control electronic parts connected to its pins.

Microcontroller

# Everything You Need to Know About Electronics

2 special pins:

Pin 1



Pin 8 =  
Ground

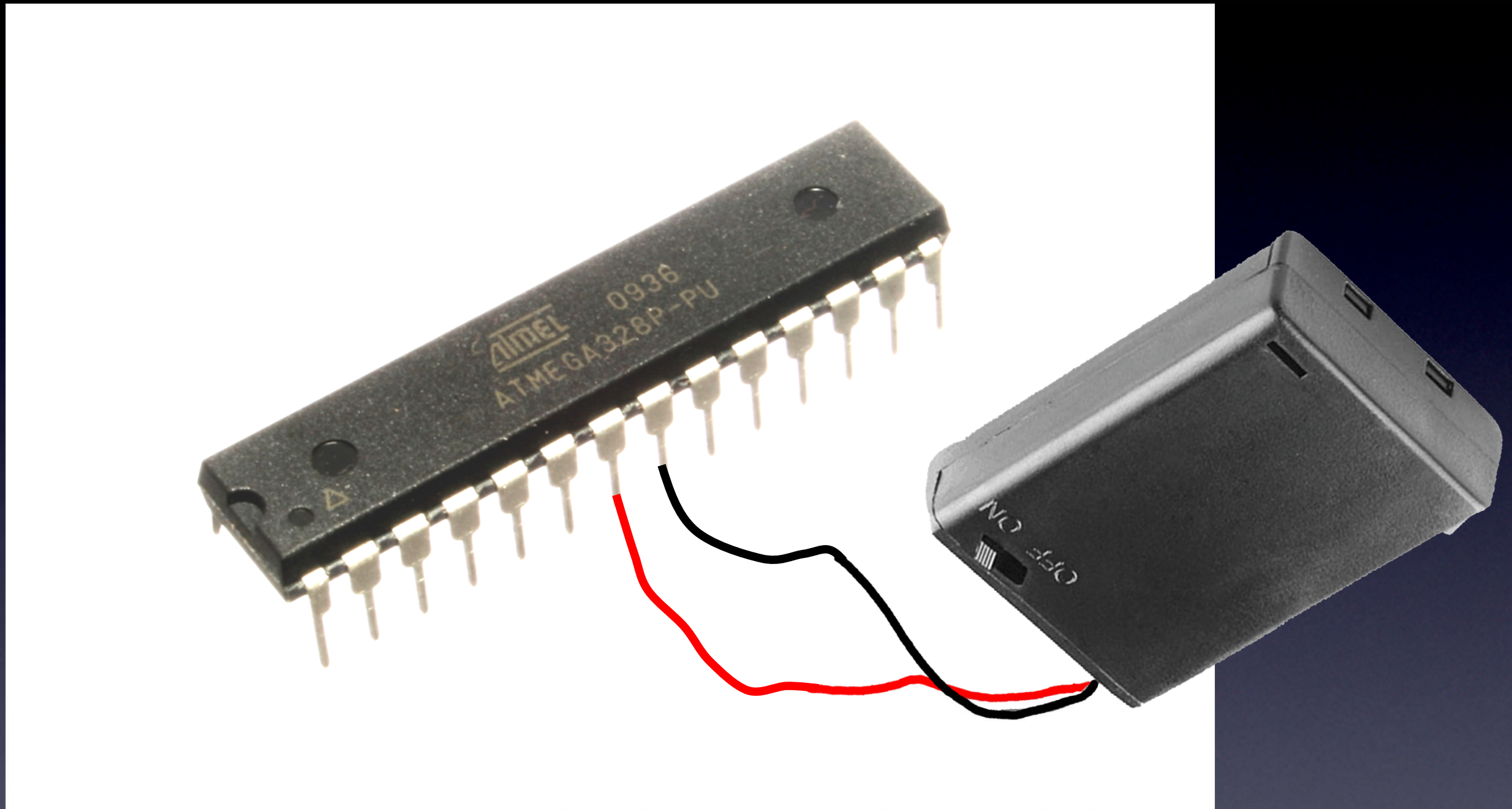
Pin 7 =  
Vcc

A complete computer on a chip

Microcontroller – it matters how you hook it up!



# Everything You Need to Know About Electronics

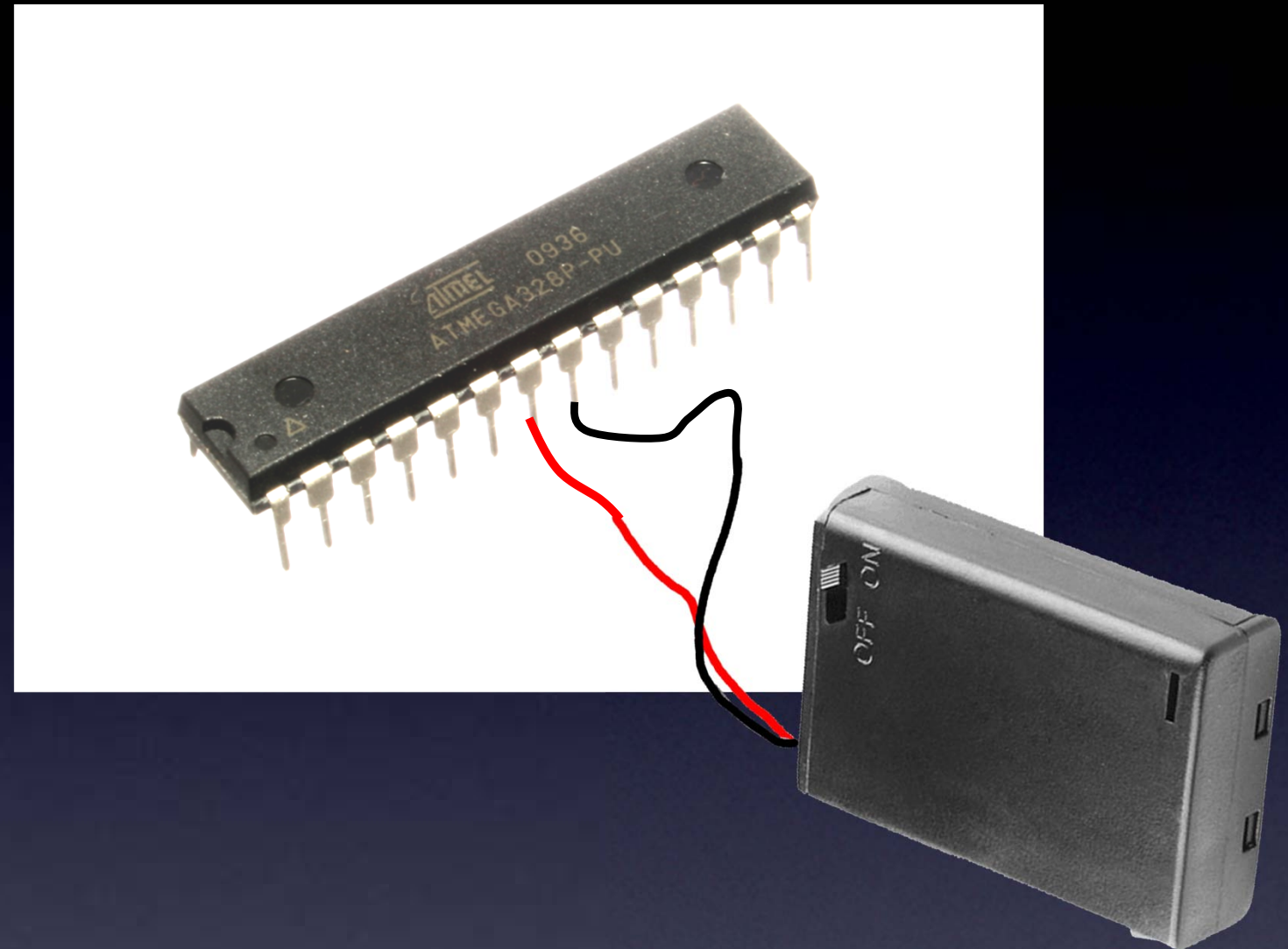


A complete computer – running a program!

Microcontroller – turned on!

# Everything You Need to Know About Electronics

all other pins are  
Input pins  
or  
Output pins



Your program controls electronics parts  
on these other pins

Microcontroller



# Everything You Need to Know About Electronics

## **Analog Electronics:**

Any voltage between Ground (0V) and  $V_{cc}$

## **Digital Electronics:**

Only 2 choices: Ground (0V) or  $V_{cc}$

2 types of electronics

# Everything You Need to Know About Electronics

**Ground (0V)**

**Low**

**Off**

**0**

**Power / Vcc**

**High**

**On**

**1**

(without Voltage / with Voltage)

(without current / with current)

**Digital Electronics:**

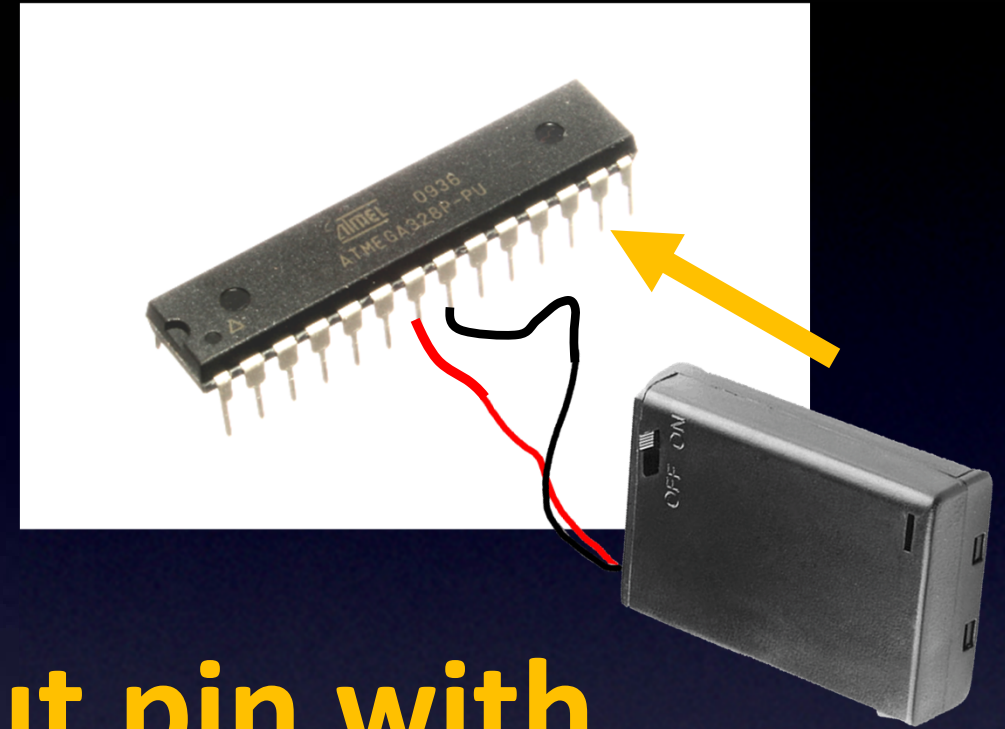
Only 2 choices: Ground (0V) or Vcc

Digital Electronics



# Everything You Need to Know About Electronics

To make a pin an  
**Output pin**



you tell it to become an Output pin with  
a statement in your program

Let's tell pin 13 to be an Output pin

Microcontroller – Output pins

# Everything You Need to Know About Electronics

Low

Off

(0V)

High

On

(Power supply voltage)

-- like the Red wire of our power supply  
-- *but controlled by our program!*

Only 2 choices: High or Low

Microcontroller – Output pins



Everything You Need to Know About Electronics

**A real world example**

**How to make an LED blink?**

Hello World

Microcontroller

# Everything You Need to Know About Electronics

## Software

Type:

Hello World  
on your screen

## Microcontrollers

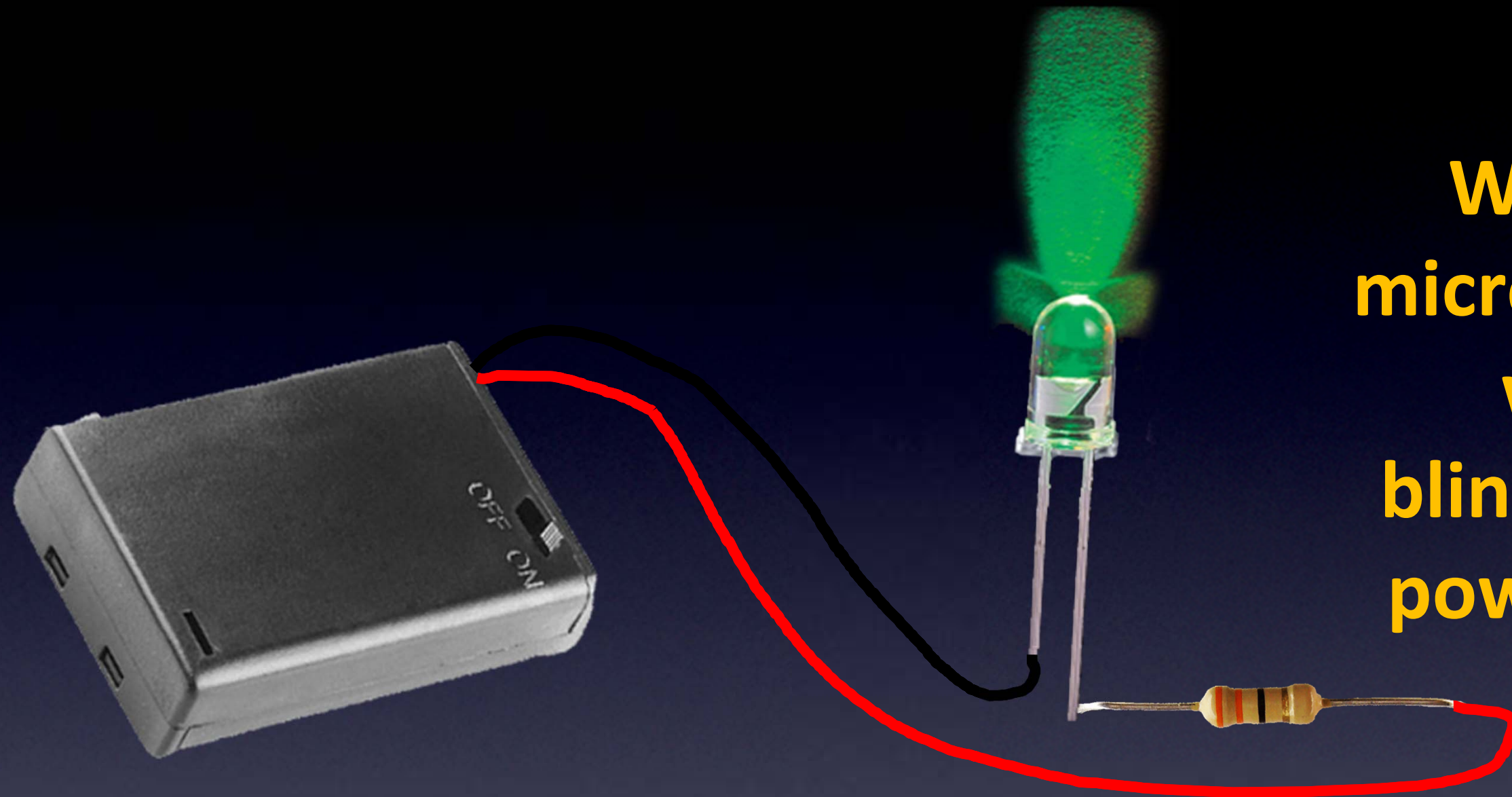
make an LED blink

Hello World

Microcontroller



# Everything You Need to Know About Electronics



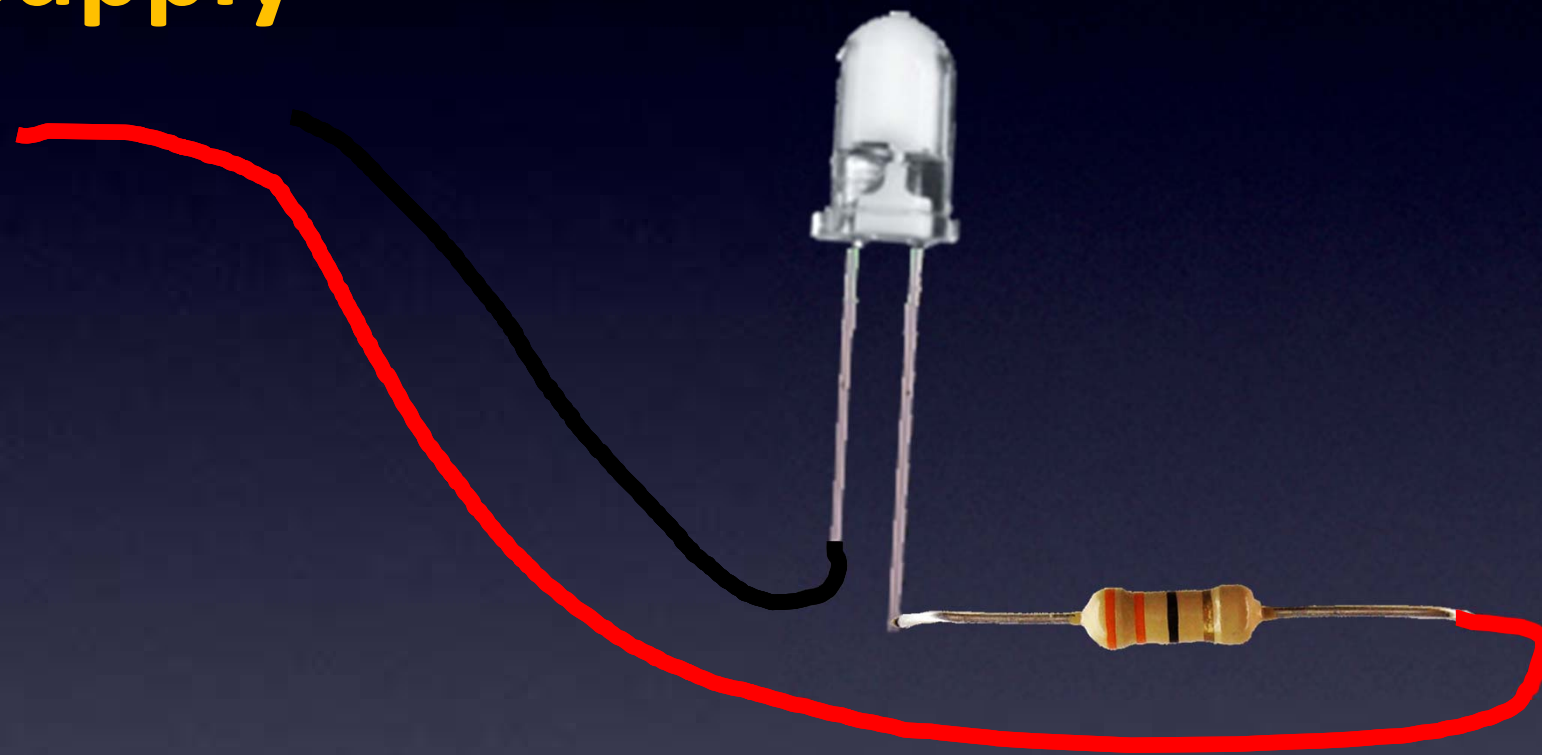
**Without a  
microcontroller  
we can  
blink with our  
power supply**

**Turning an LED on and off**

(Leading up to Hello World)

# Everything You Need to Know About Electronics

**Let's replace  
the power supply**

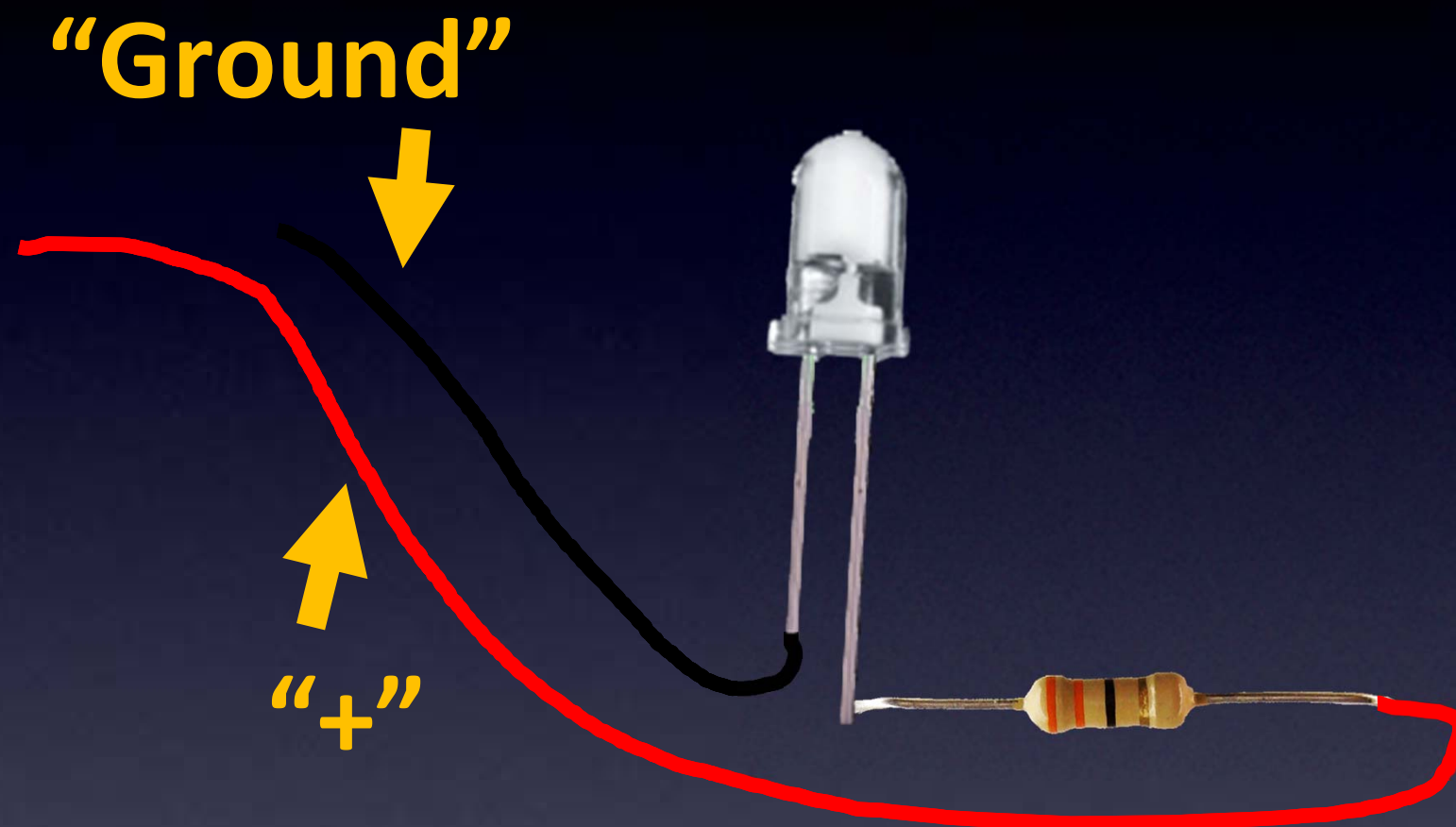


**Turning an LED on and off**

(Leading up to Hello World)



# Everything You Need to Know About Electronics

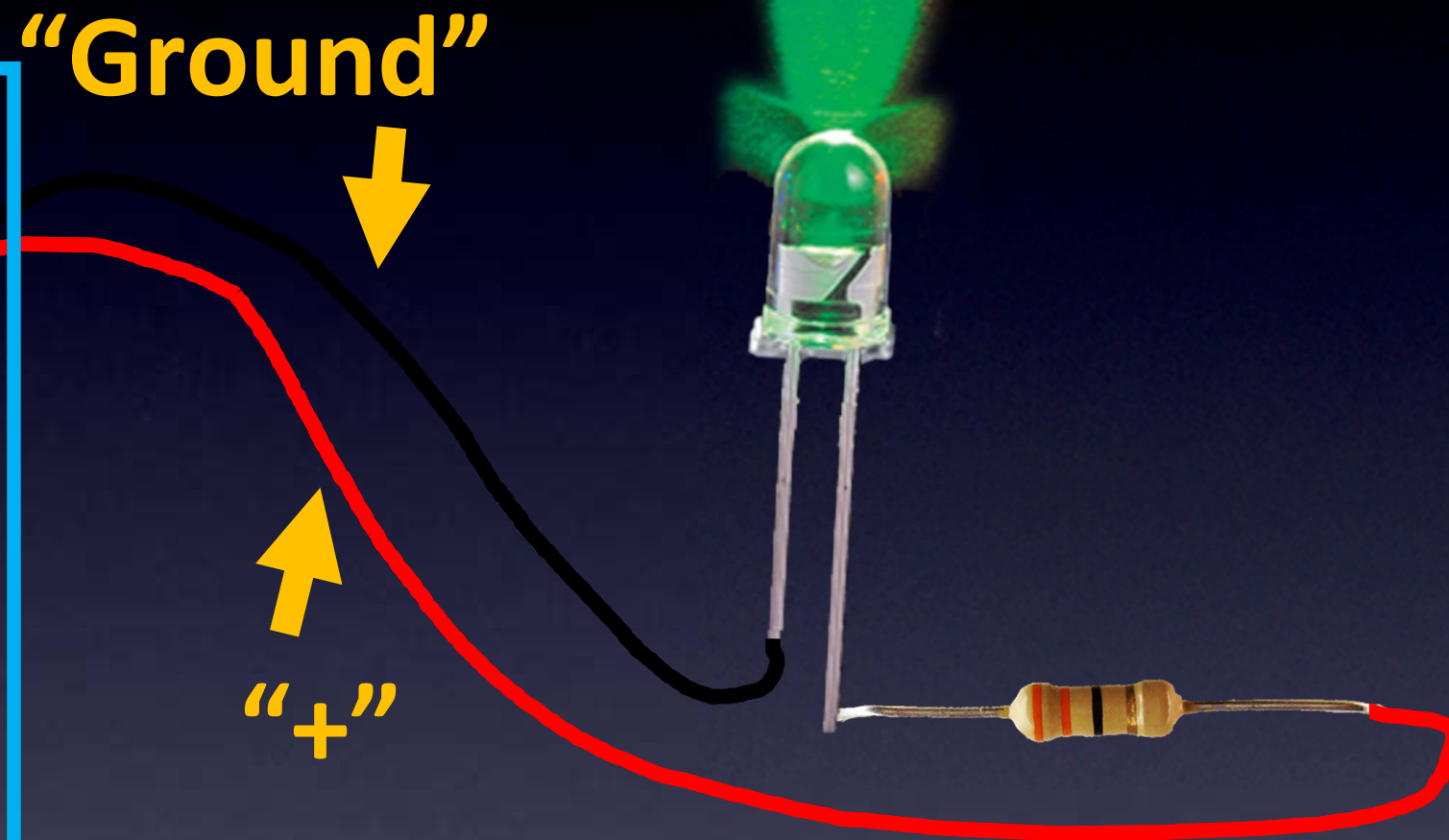


Turning an LED on and off

(Leading up to Hello World)

# Everything You Need to Know About Electronics

We can use an  
Output pin  
(and Ground)  
as our power  
supply

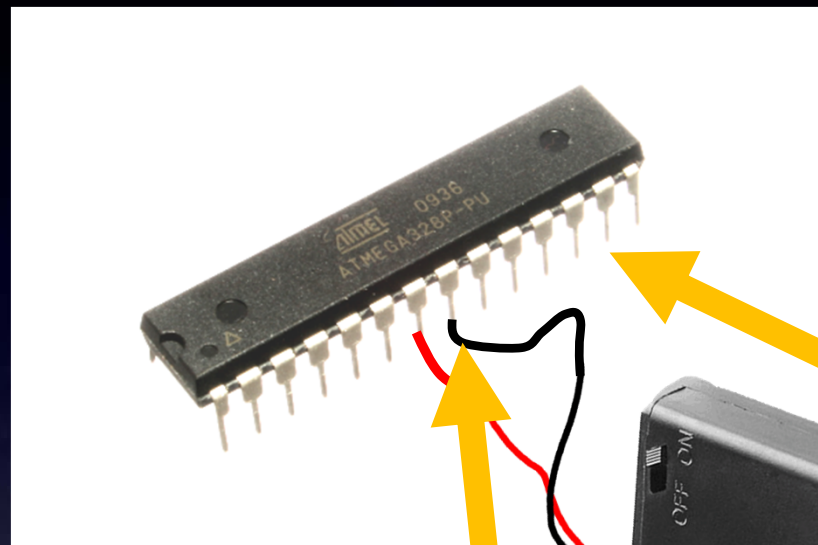


Turning an LED on and off

(Leading up to Hello World)



# Everything You Need to Know About Electronics



**Ground**

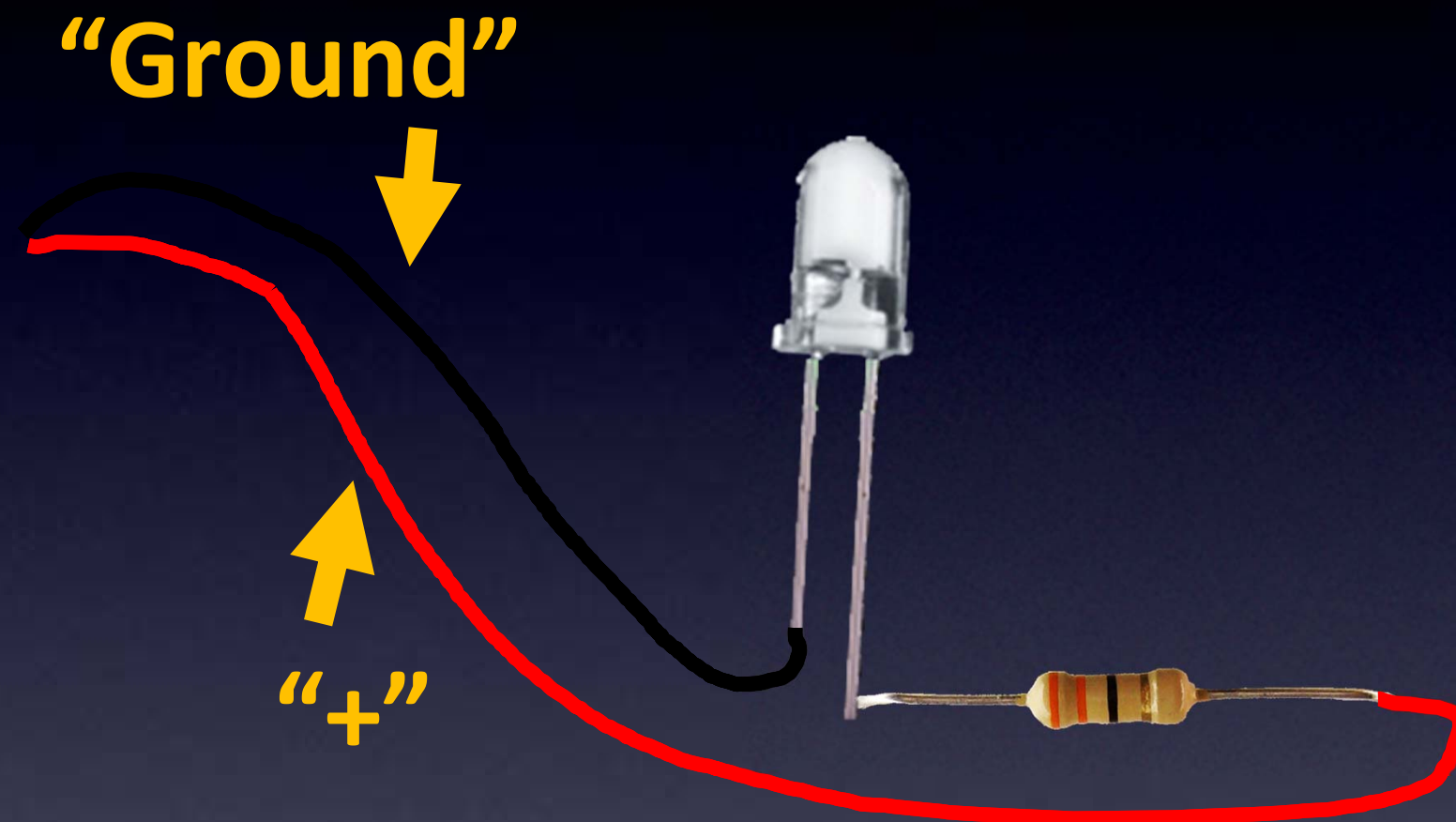
**Pin 13**

is our **Output pin**  
*(which can be On or Off)*

Turning an LED on and off

(Leading up to Hello World)

# Everything You Need to Know About Electronics

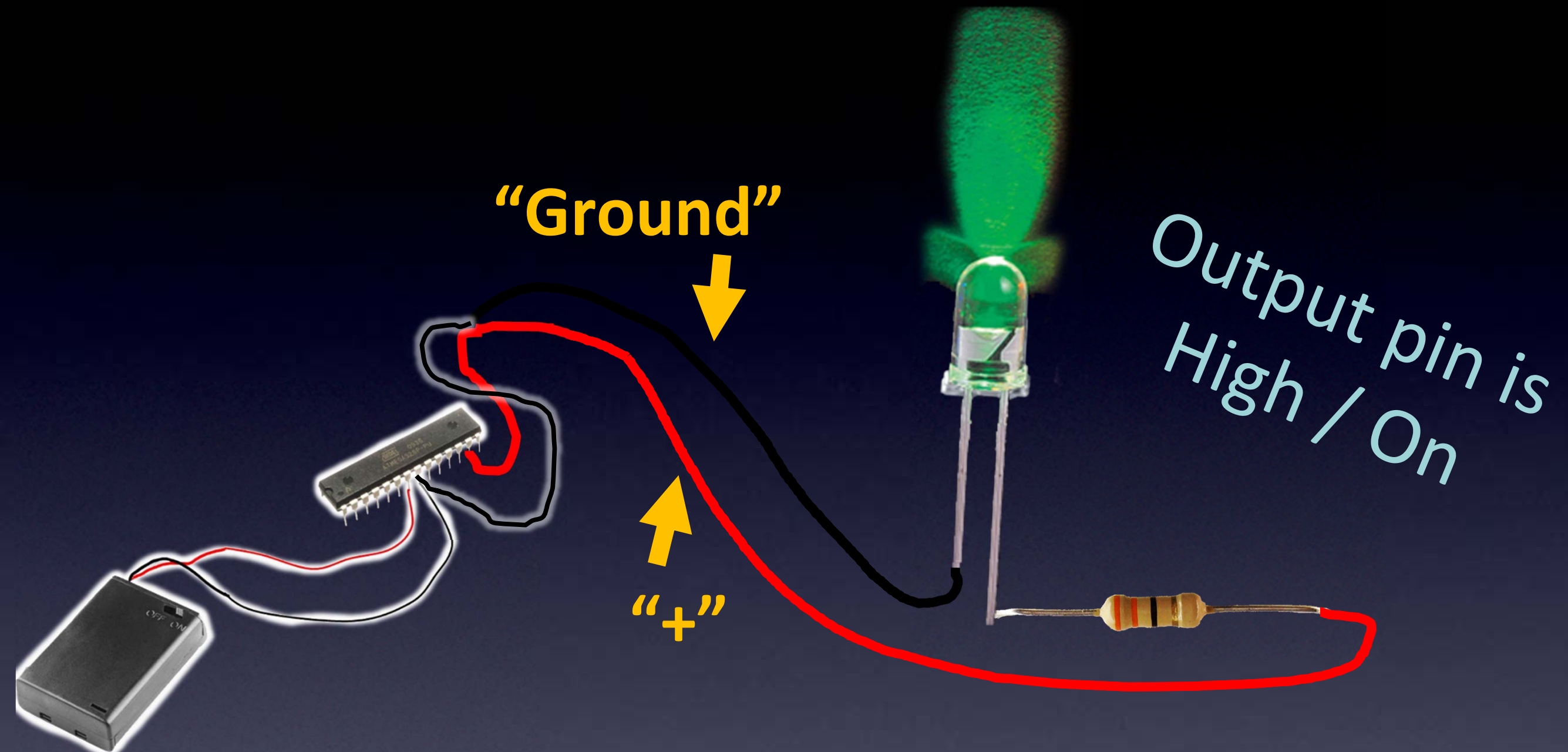


Turning an LED on and off

(Leading up to Hello World)



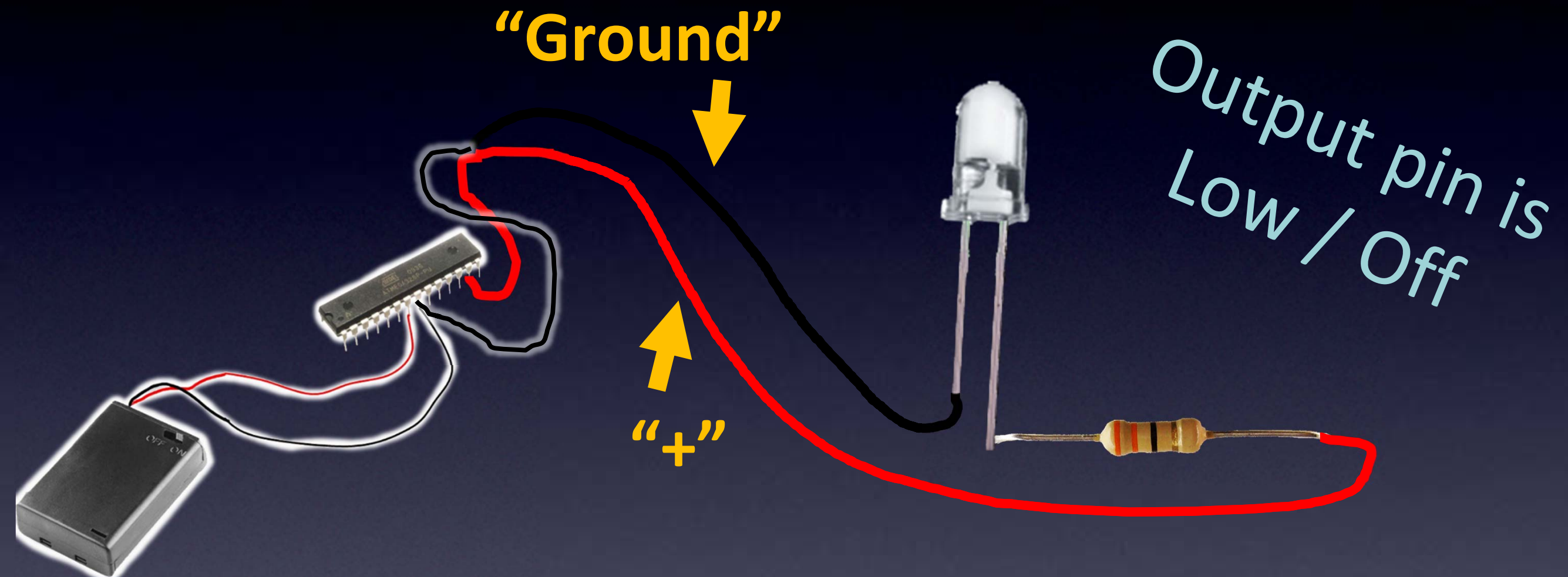
# What You Need to Know About Electronics



Turning an LED on and off

(Leading up to Hello World)

# What You Need to Know About Electronics



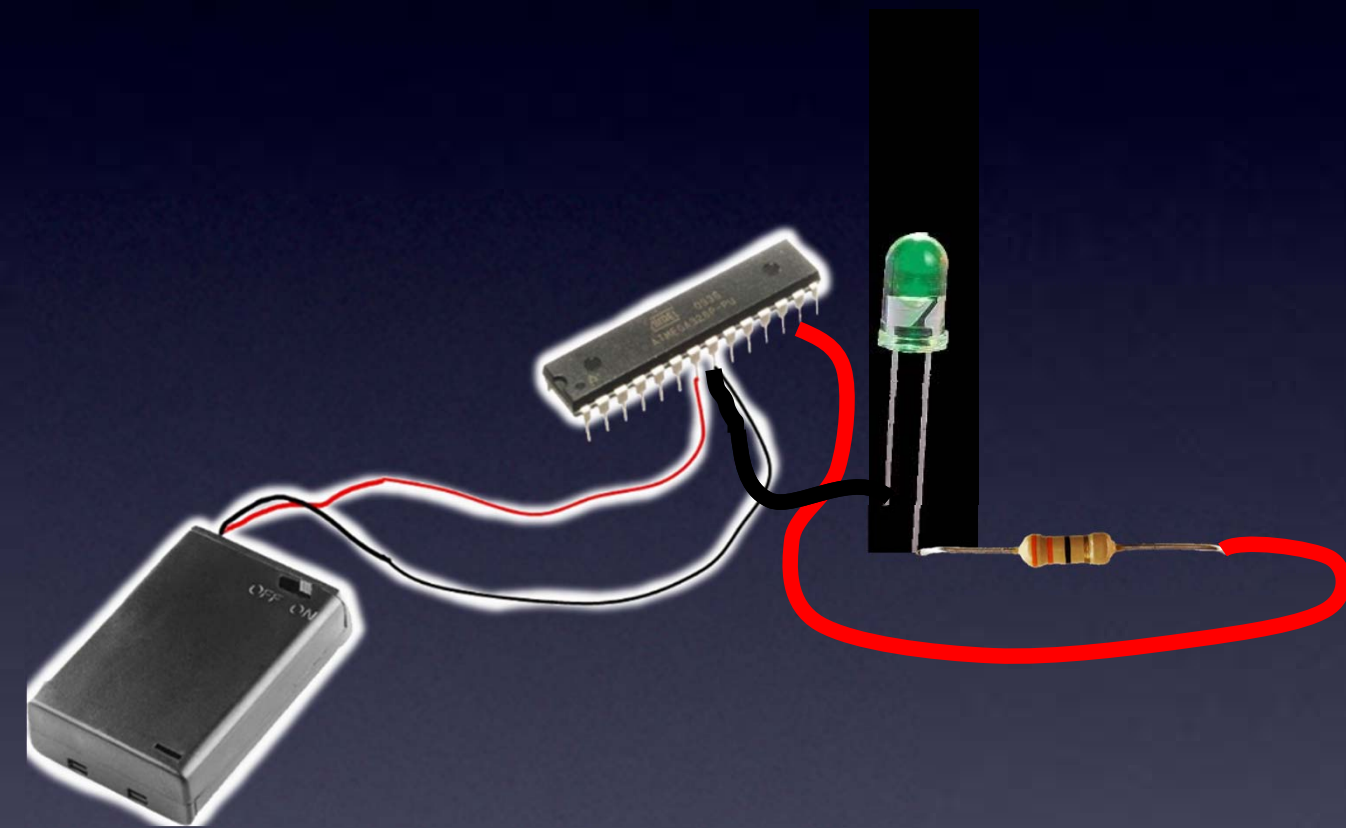
Turning an LED on and off

(Leading up to Hello World)



# Everything You Need to Know About Electronics

This is our Hardware for Hello World!

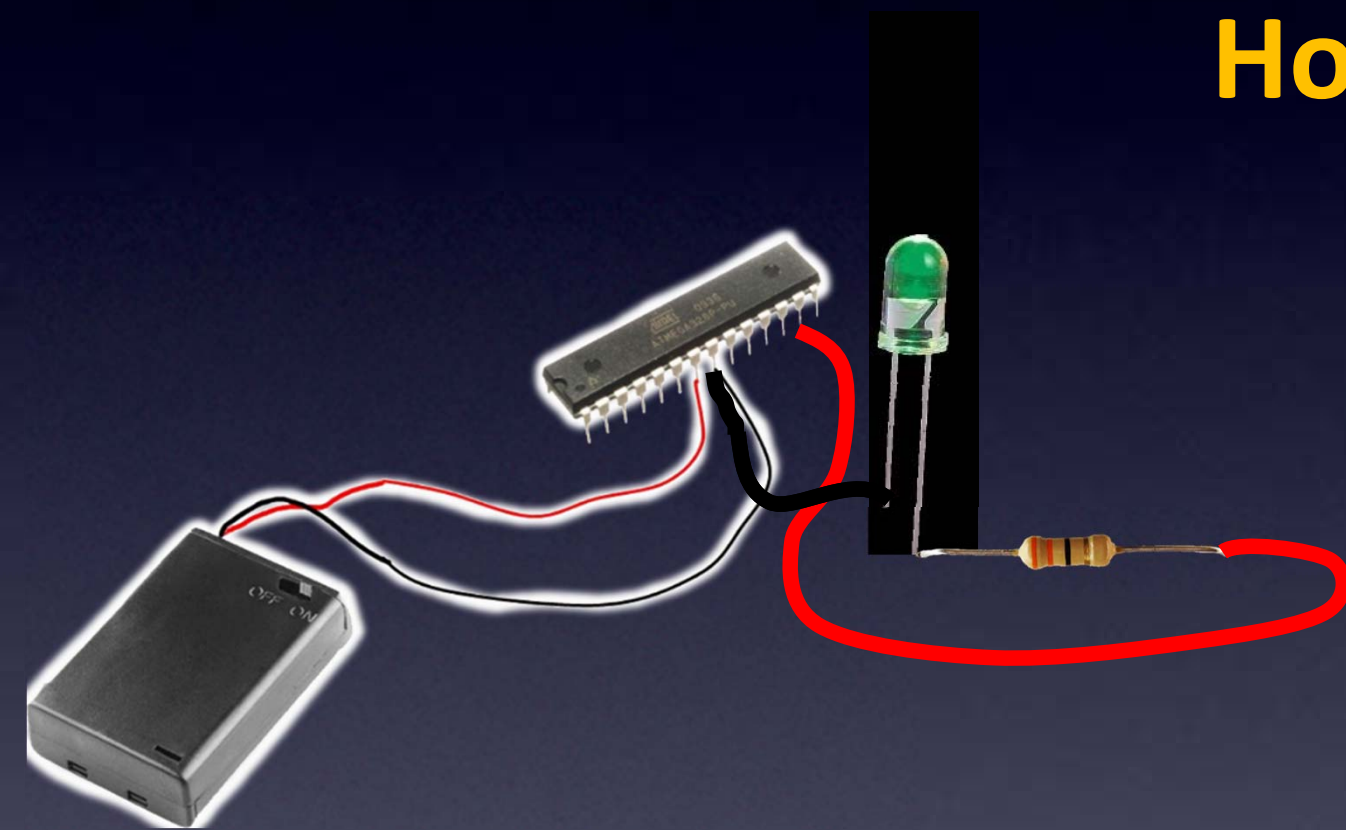


Turning an LED on and off

Hello World

# Everything You Need to Know About Electronics

How about our program?



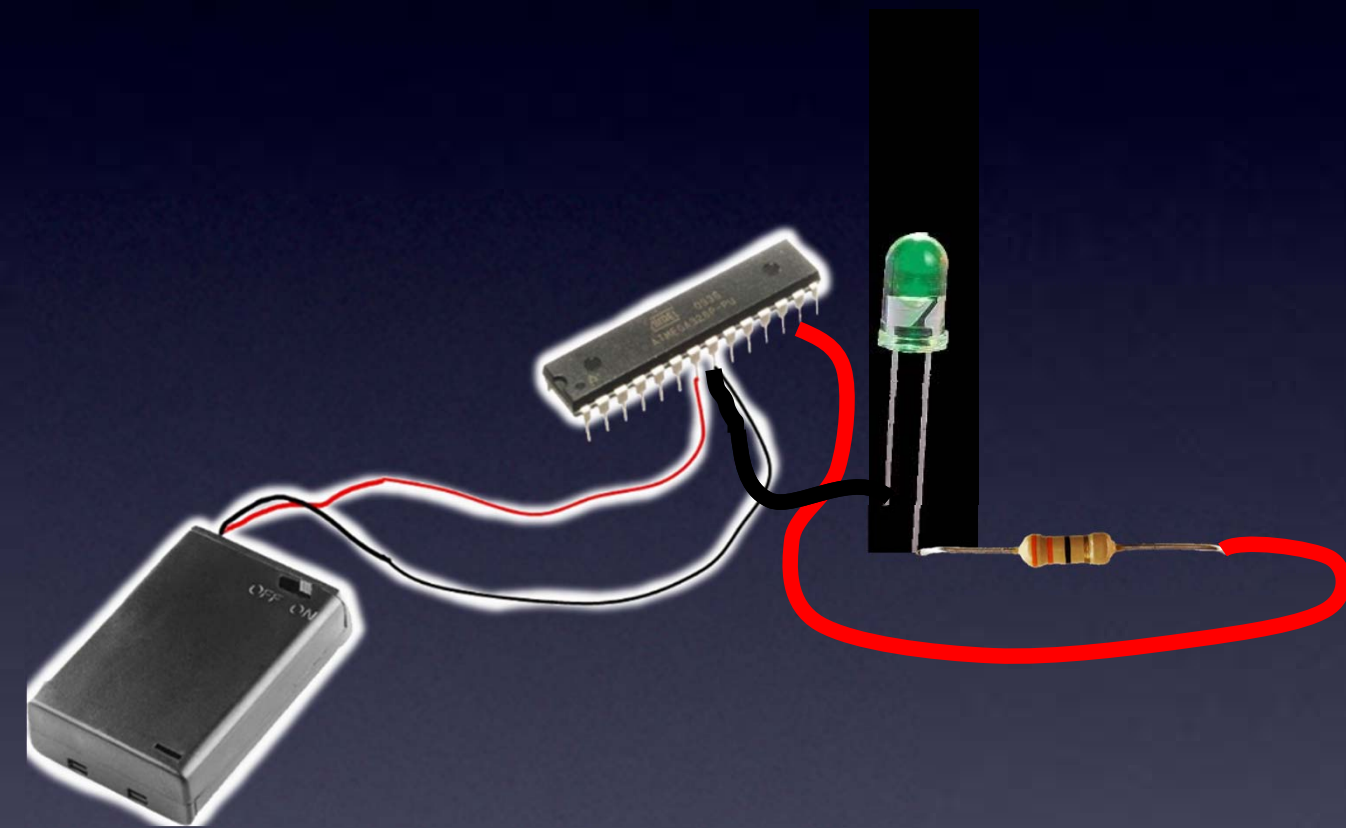
Turning an LED on and off

Hello World



# Everything You Need to Know About Electronics

Programs on microcontrollers are called **“Firmware”**



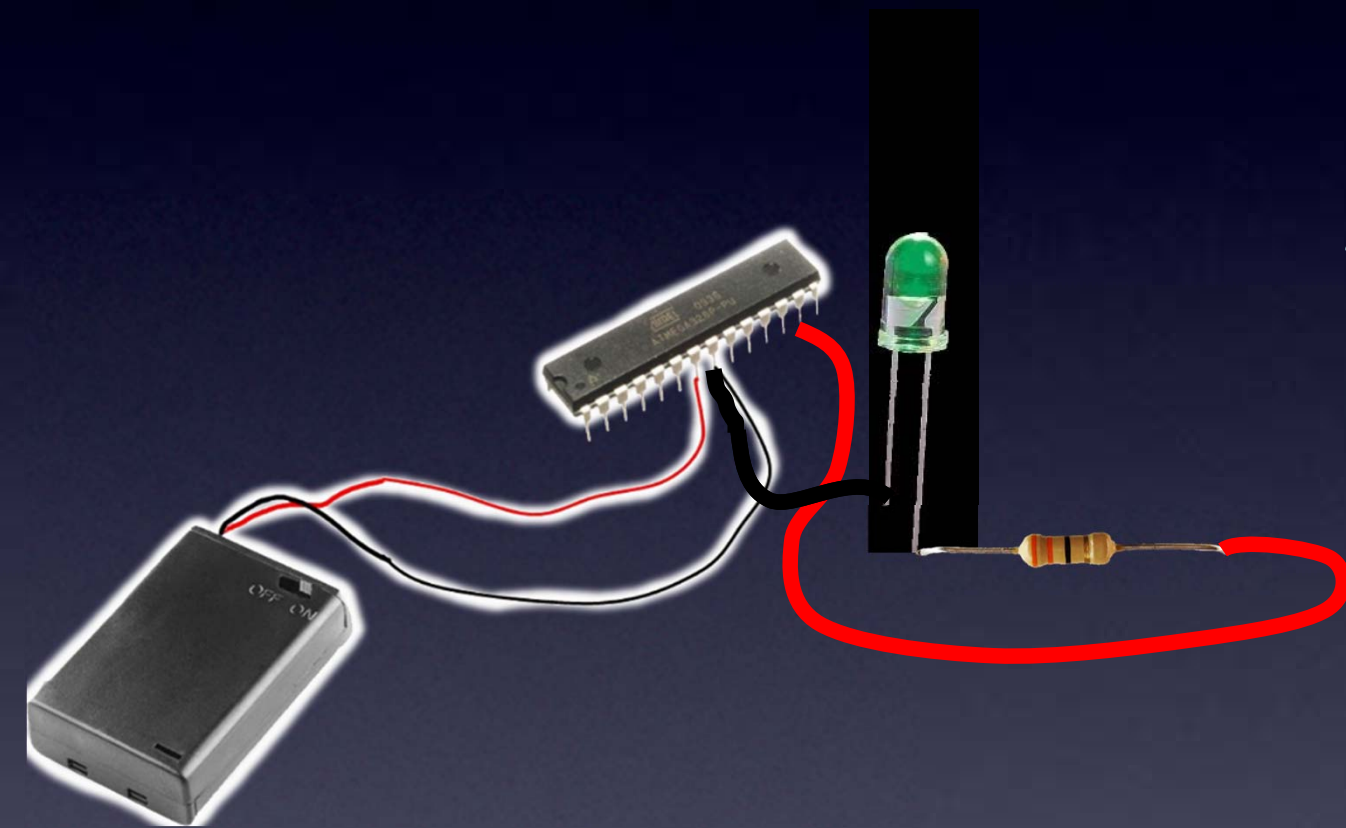
Turning an LED on and off

Hello World

# What You Need to Know About Electronics

Programs on microcontrollers are called **“Firmware”**

A programs for Arduino is called a  
**“Sketch”**

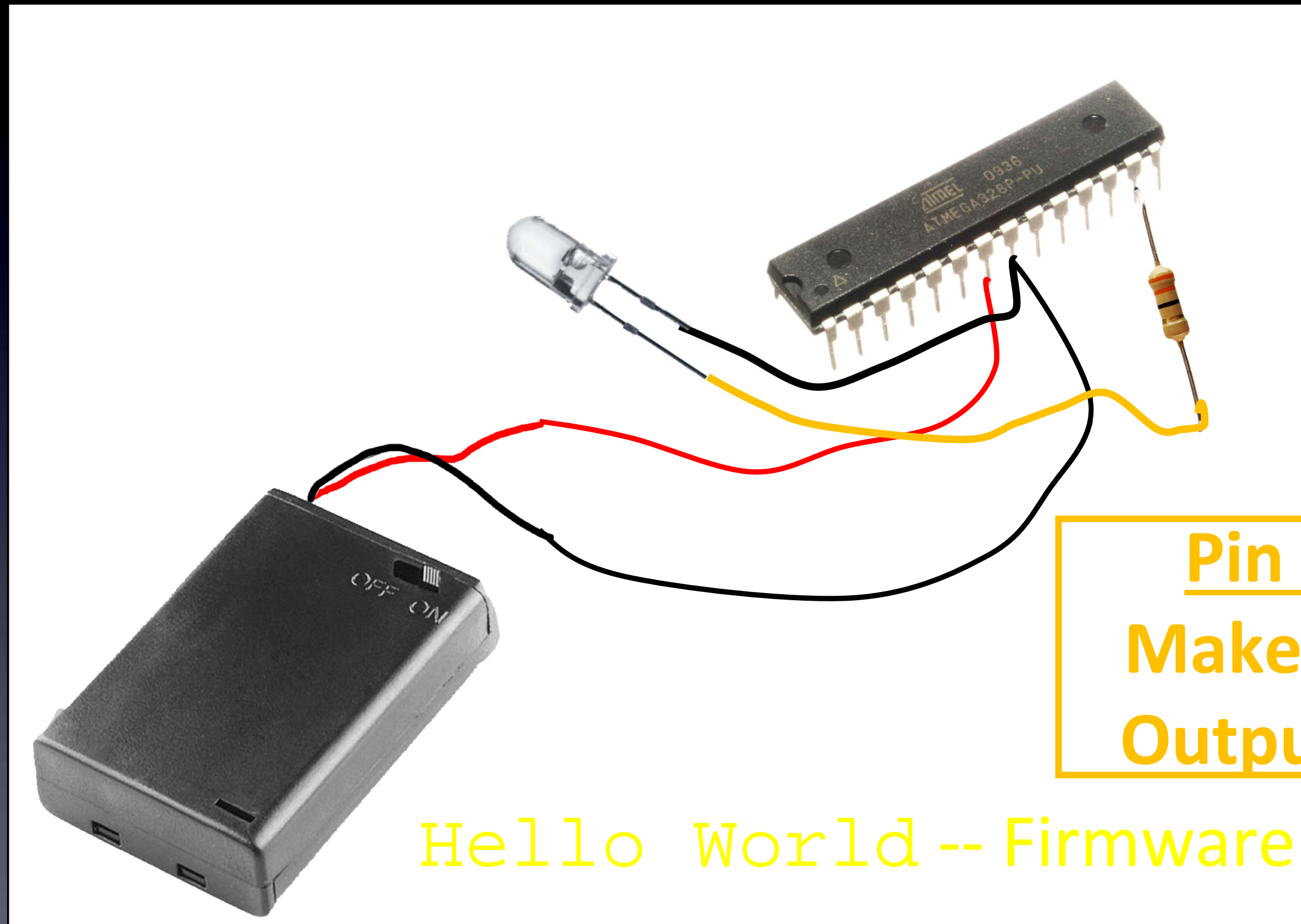


Turning an LED on and off

Hello World



# Everything You Need to Know About Electronics

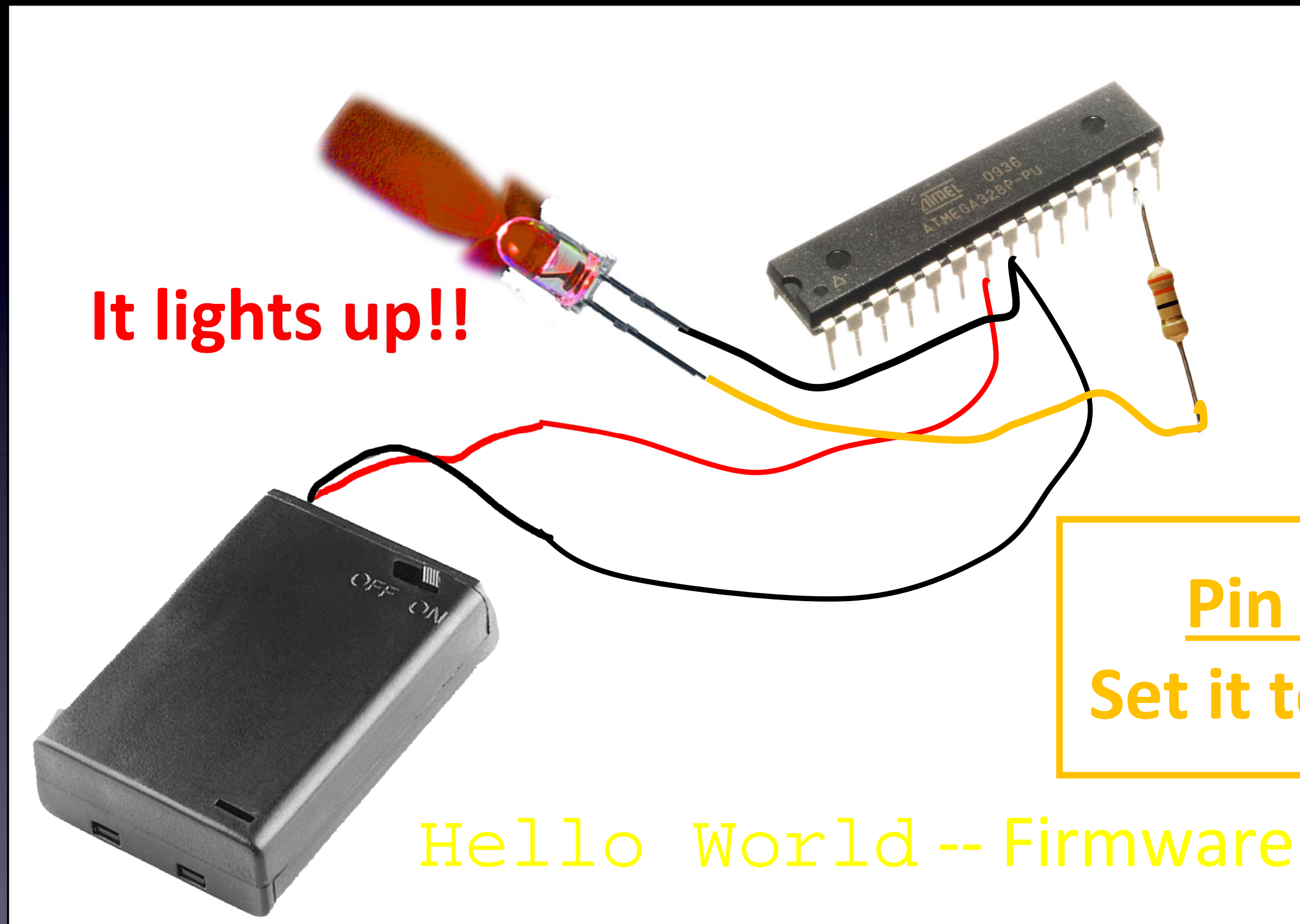


Pin 13:  
Make it an  
Output pin

Hello World -- Firmware

Microcontroller

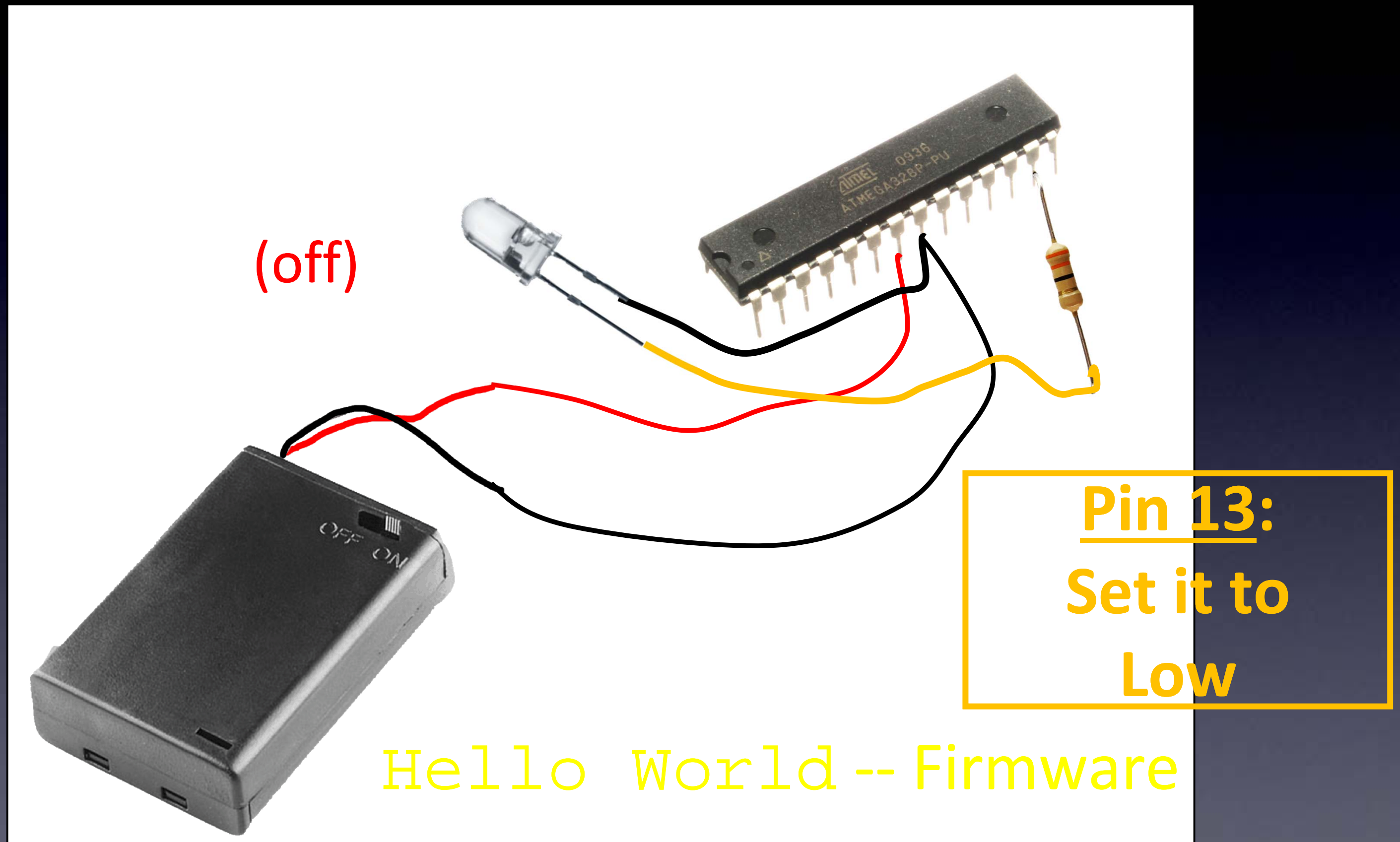
# Everything You Need to Know About Electronics



Microcontroller



# Everything You Need to Know About Electronics



Microcontroller

# Everything You Need to Know About Electronics

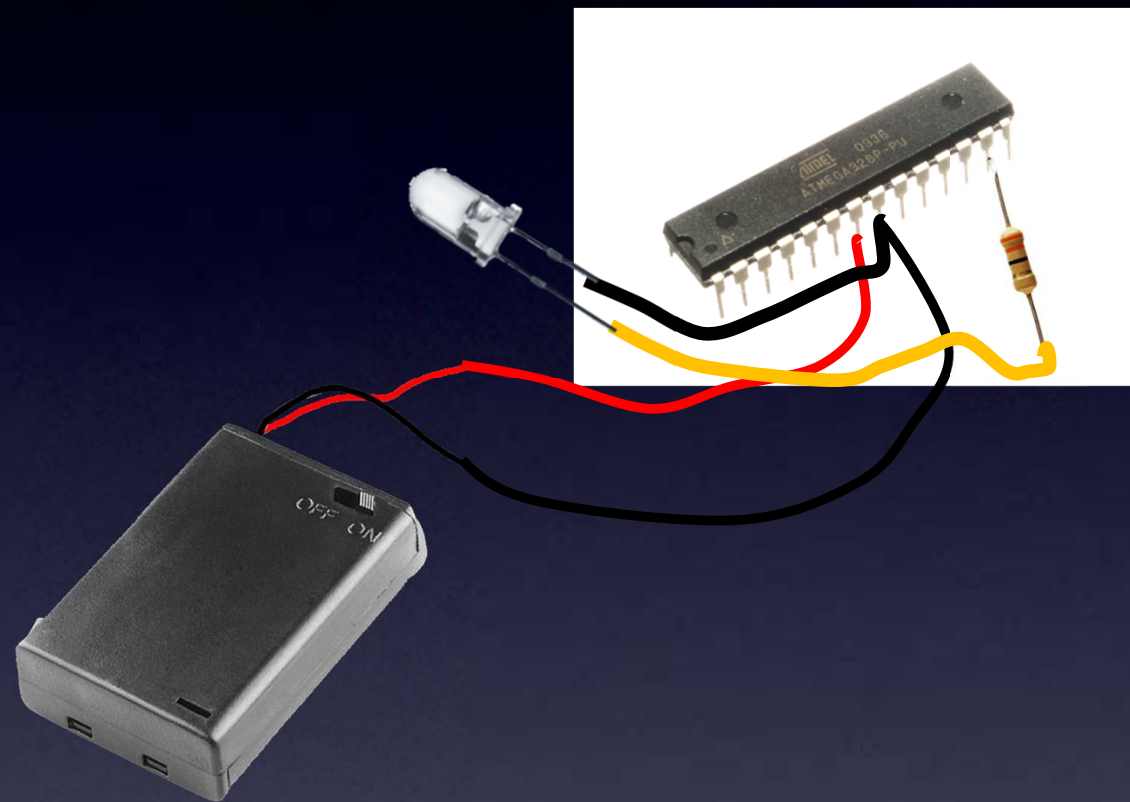


We now have  
Hello World !

Microcontroller



# Everything You Need to Know About Electronics



We now have  
Hello World !

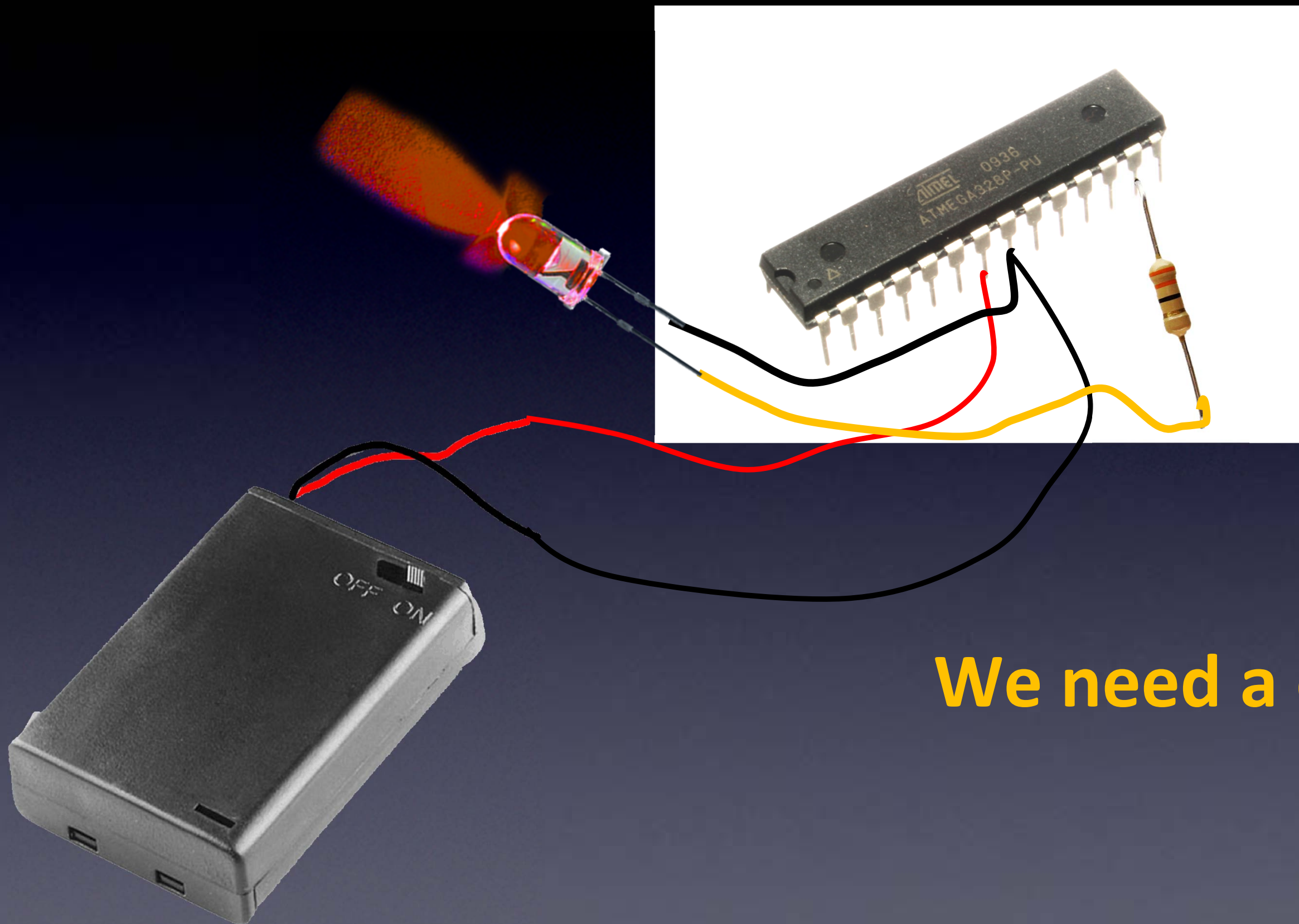
Microcontroller

Except

We won't see it



# Everything You Need to Know About Electronics



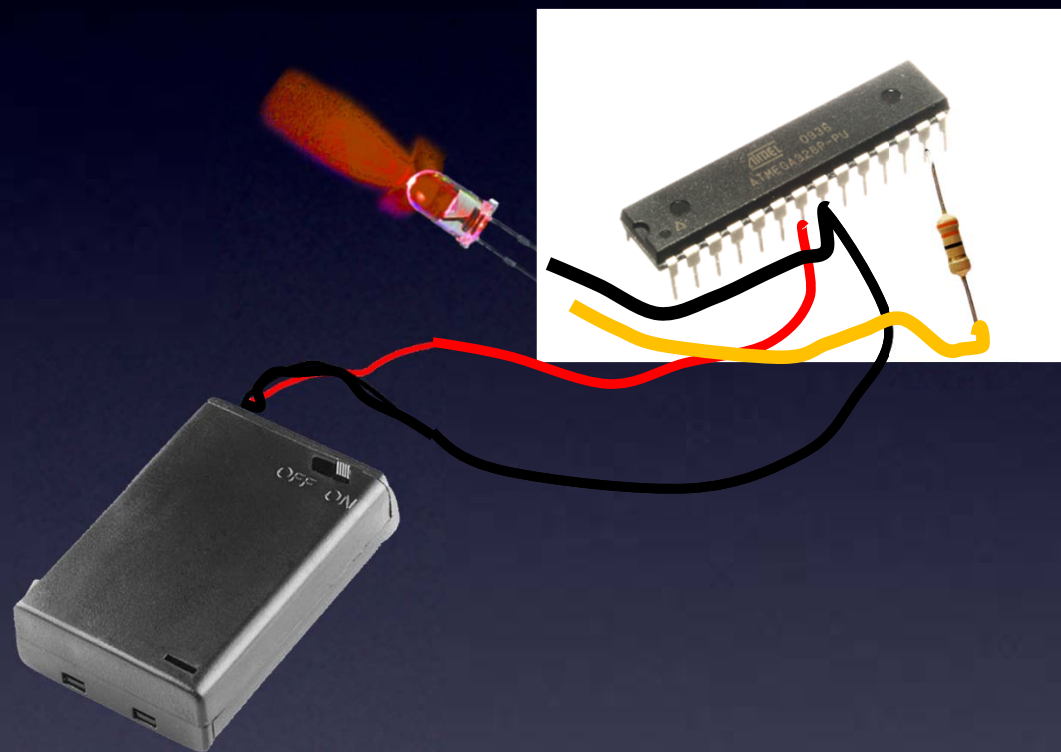
**We need a delay**

Microcontrollers – they go really fast!



# Everything You Need to Know About Electronics

## Hardware



## Firmware

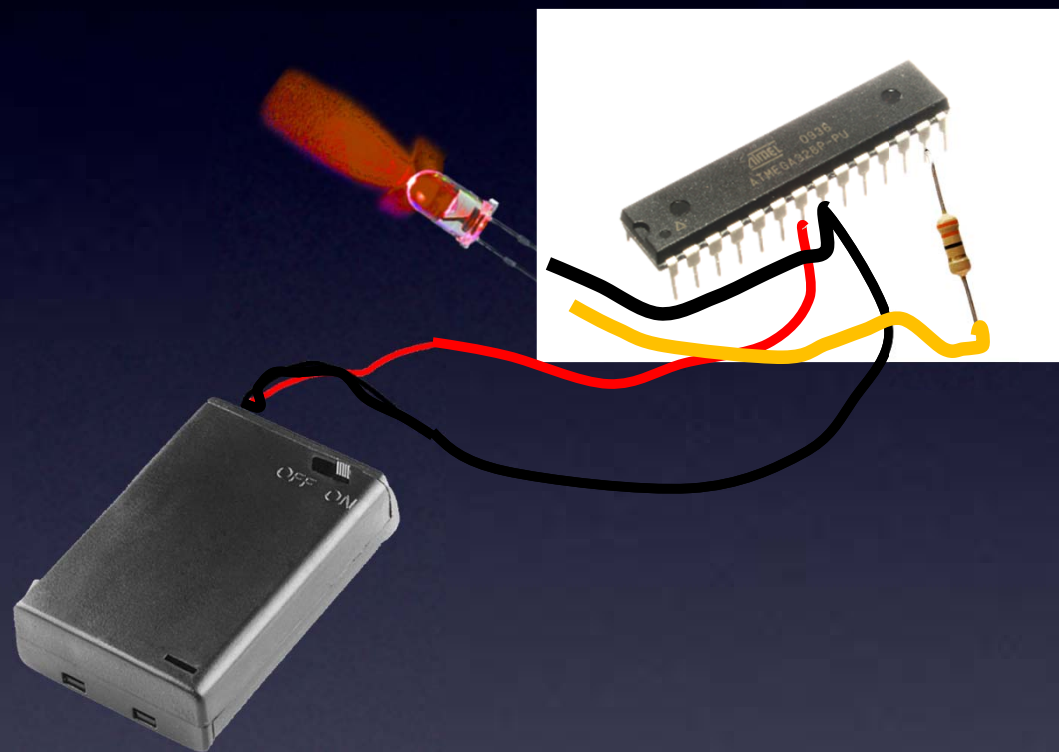
- pin 13 is Output pin
- set pin 13 High
- delay
- set pin 13 Low

Hello World – for real now!

Microcontroller – Firmware

# What You Need to Know About Electronics

## Hardware



## Firmware

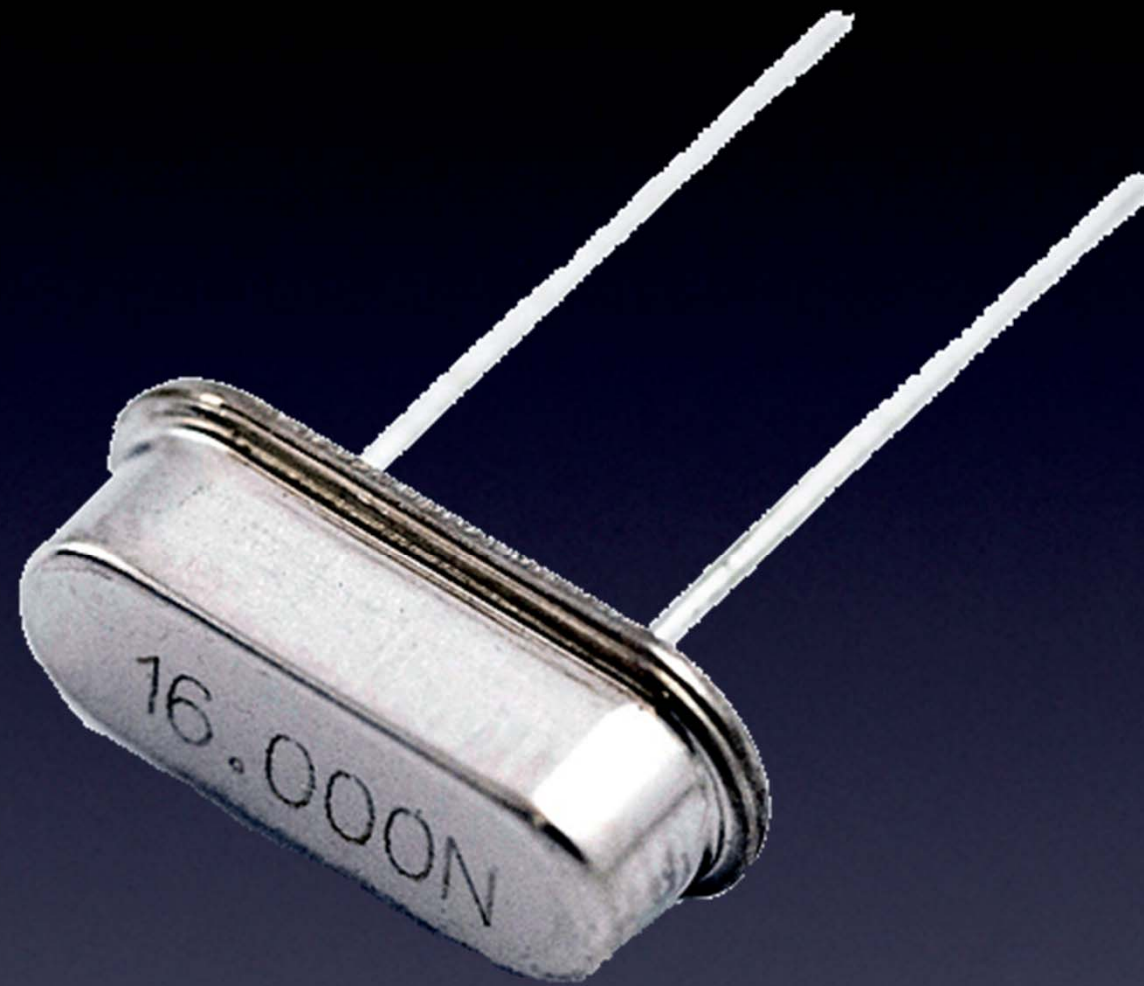
- pin 13 is Output pin
- set pin 13 High
- delay
- set pin 13 Low
- delay

Hello World – for real now!

Microcontroller – Firmware



# Everything You Need to Know About Electronics



**A precision cut piece of quartz crystal**

**For precise timing**

Crystal / Hertz

# Everything You Need to Know About Electronics



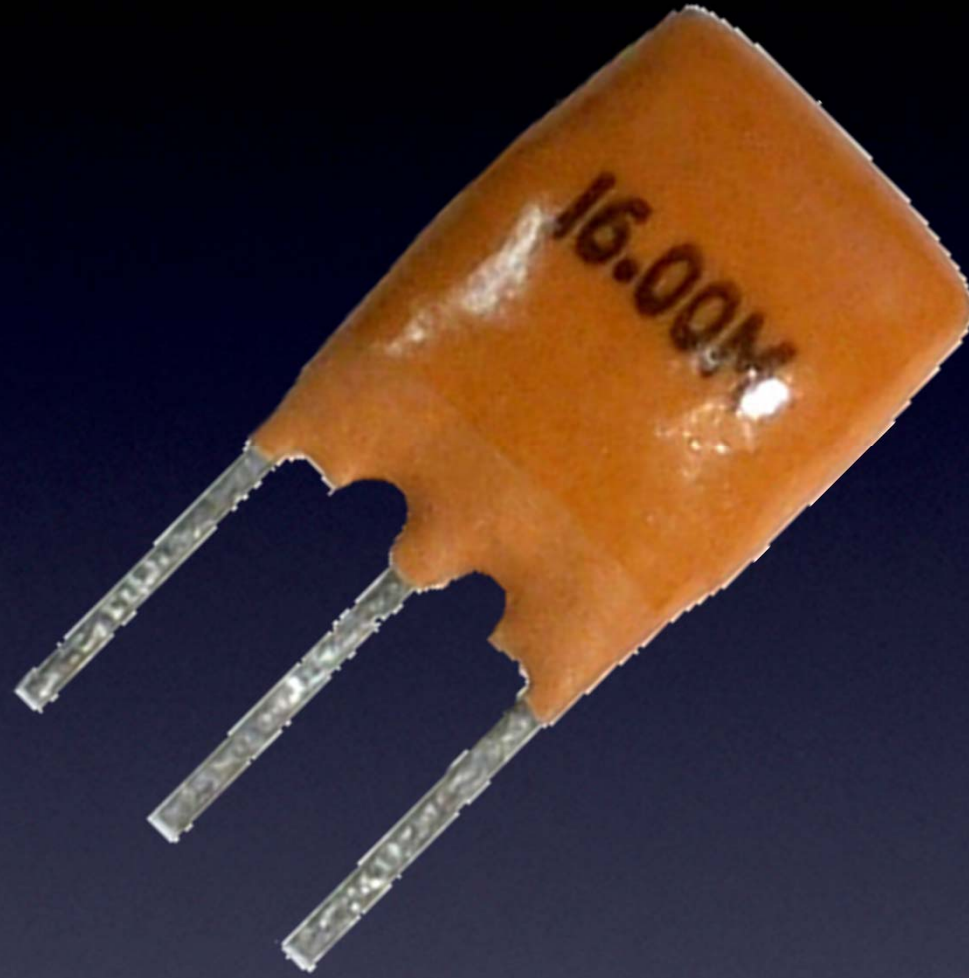
Frequency, measured in **Hertz**

For precise timing (but less than a crystal)

Crystal / **Hertz**



# Everything You Need to Know About Electronics



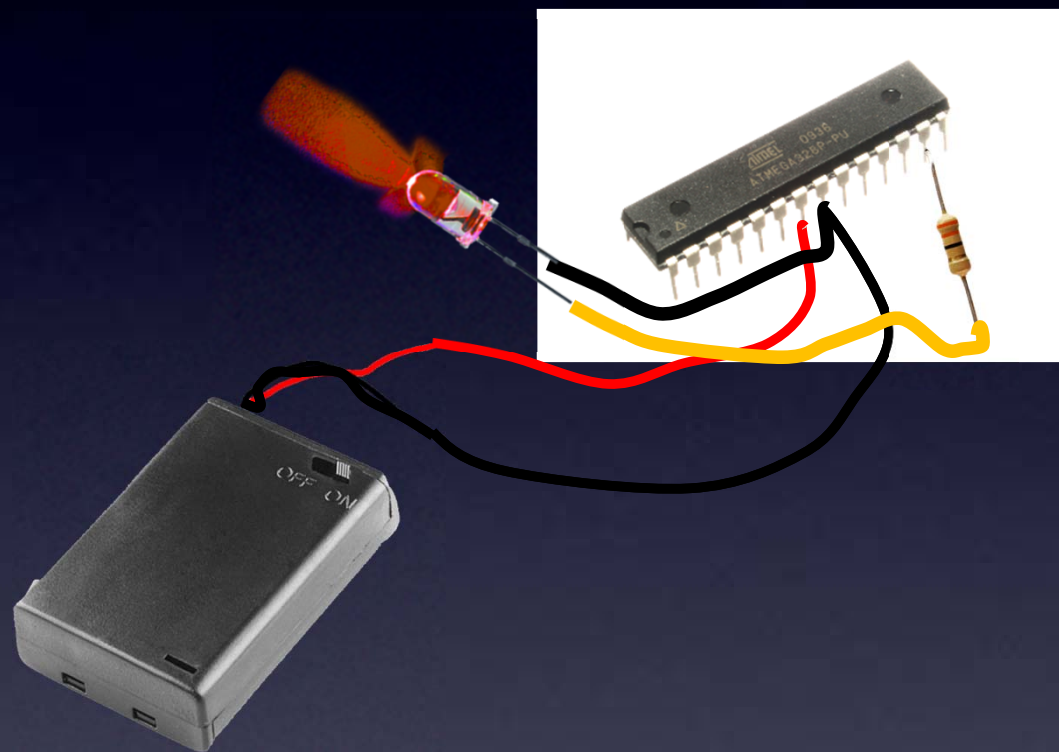
**A bunch of resistors and capacitors**

**For precise timing (but less than a crystal)**

Ceramic Resonator / Hertz

# Everything You Need to Know About Electronics

## Hardware



## Firmware

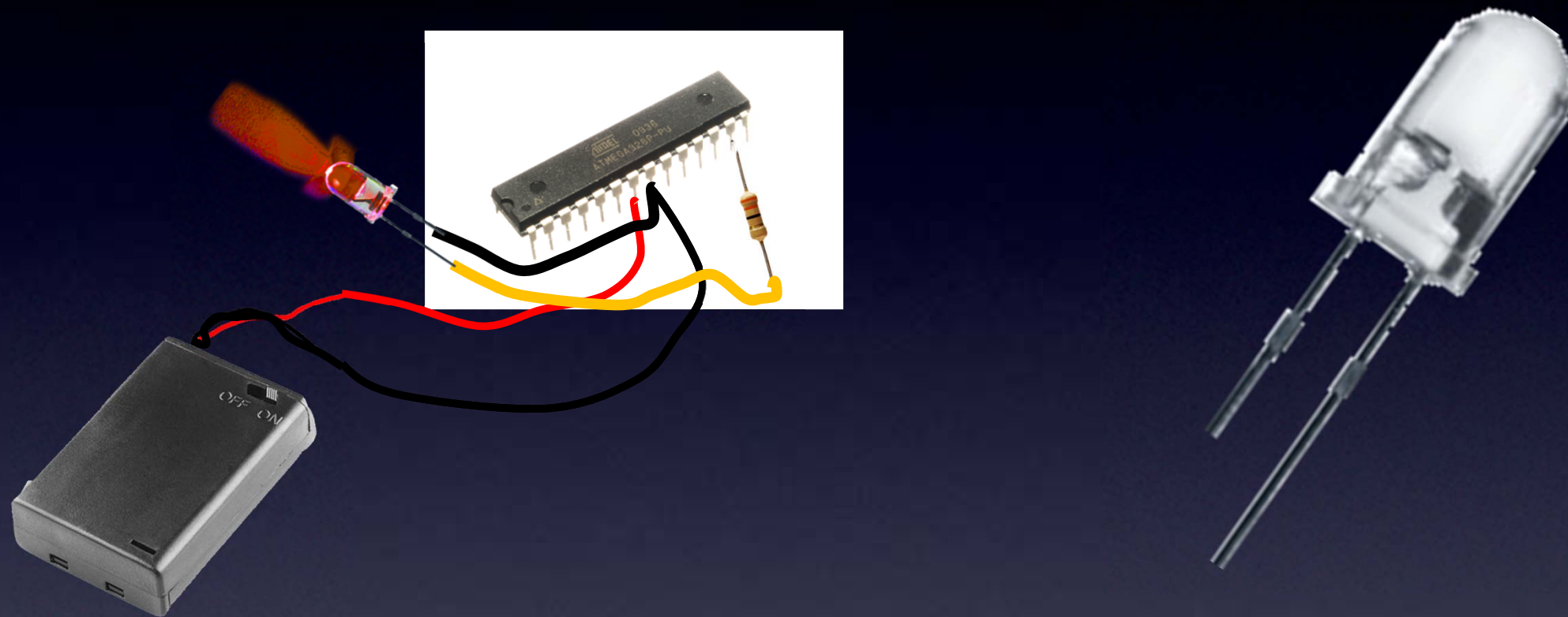
- pin 13 is Output pin
- set pin 13 High
- delay
- set pin 13 Low

# Let's hack Hello World!

Microcontroller



# Everything You Need to Know About Electronics

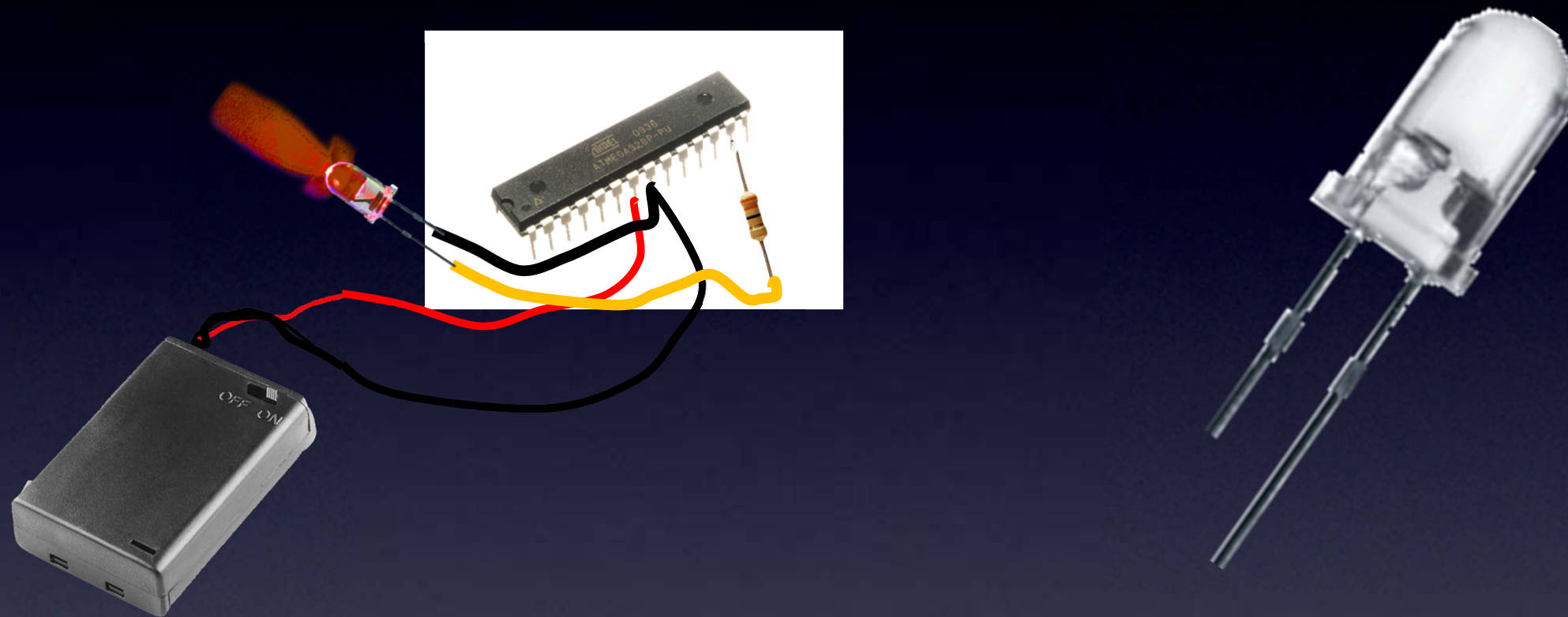


**Add an IR LED to another pin**

**IR “OFF” codes**

Microcontroller

# Everything You Need to Know About Electronics



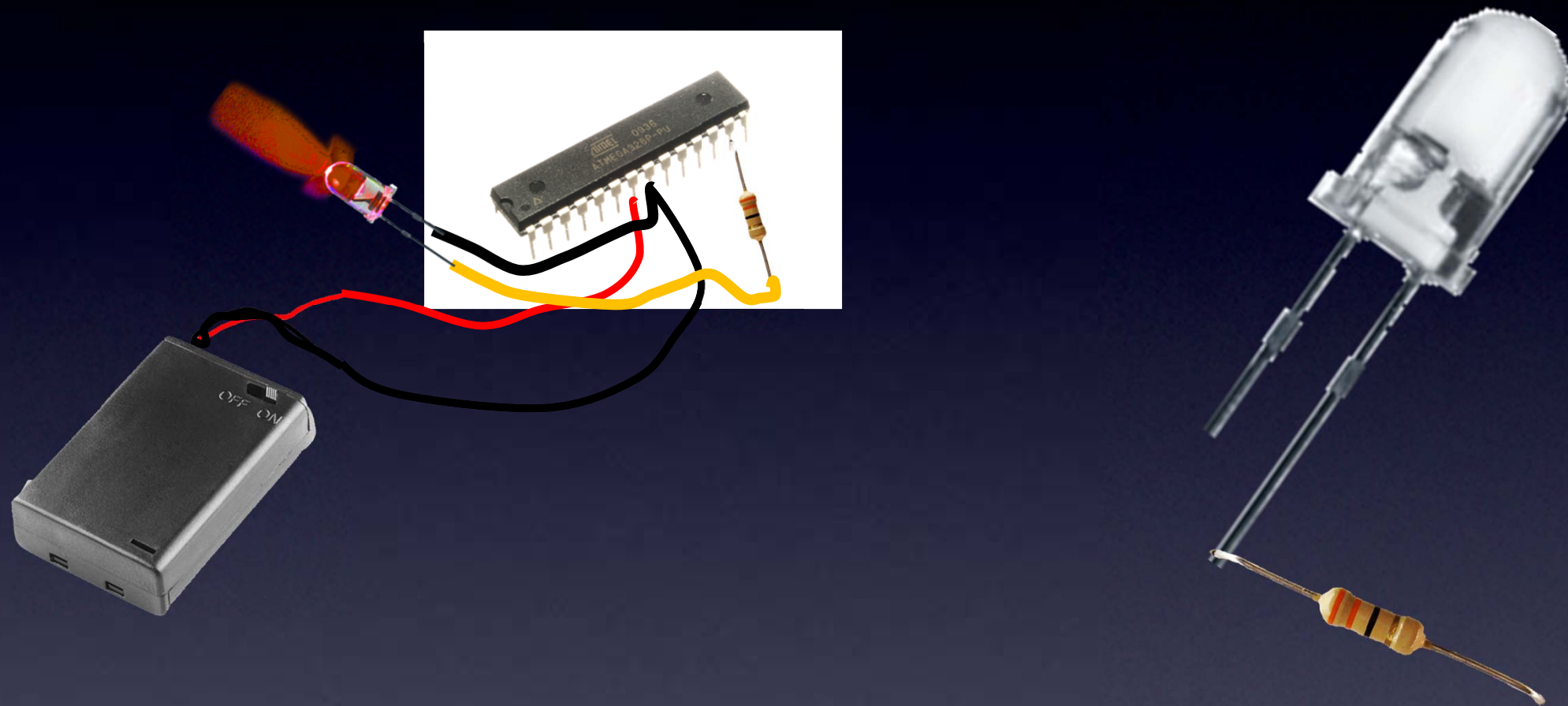
**Add an IR LED to another pin (say, pin3)**

IR “OFF” codes

Microcontroller



# Everything You Need to Know About Electronics

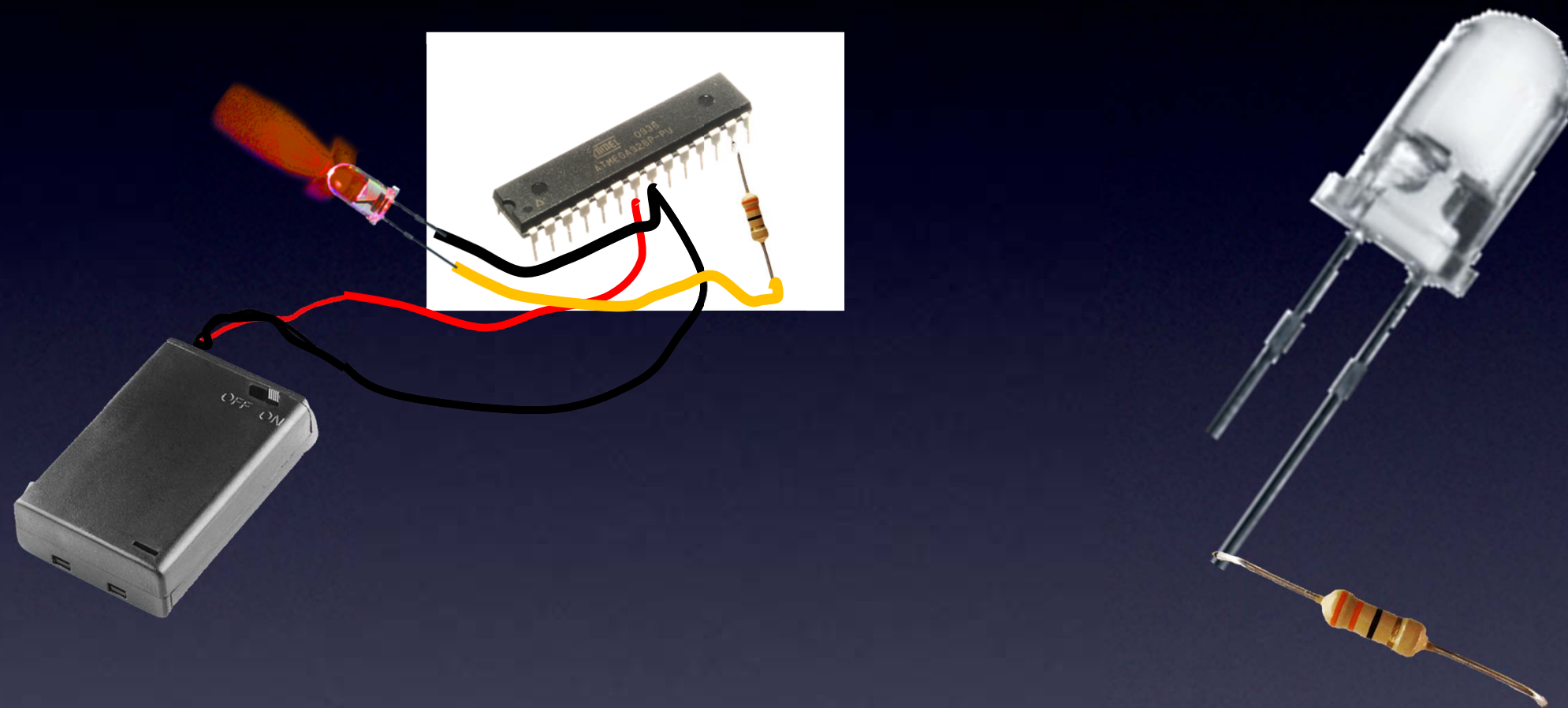


**Add an IR LED to another pin (say, pin3)**  
*and a resistor so no magic smoke goes away*

**IR “OFF” codes**

Microcontroller

# Everything You Need to Know About Electronics



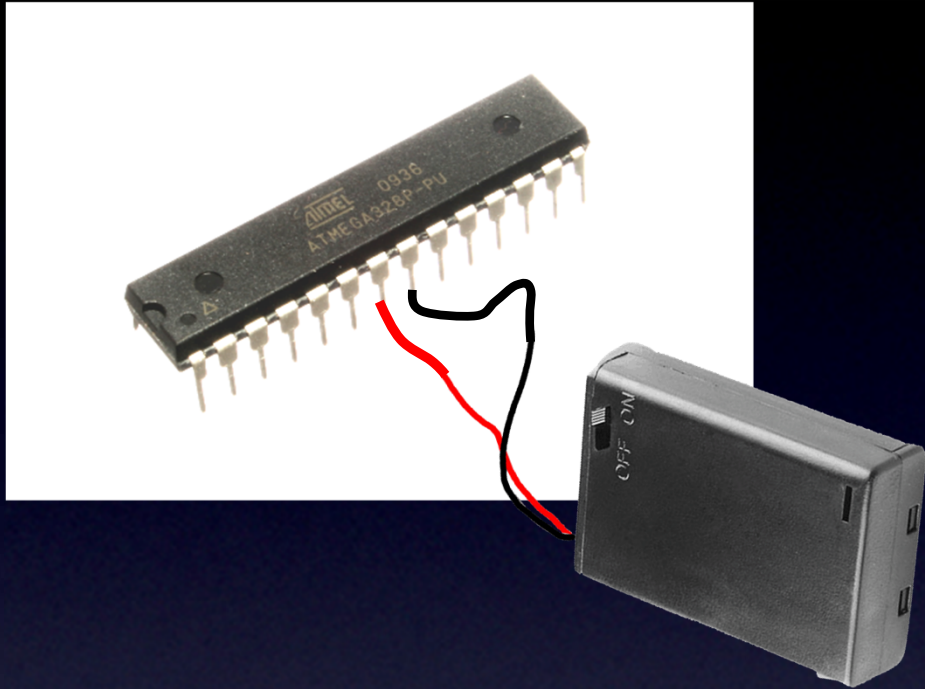
But, *When?*

IR “OFF” codes

Microcontroller



# Everything You Need to Know About Electronics



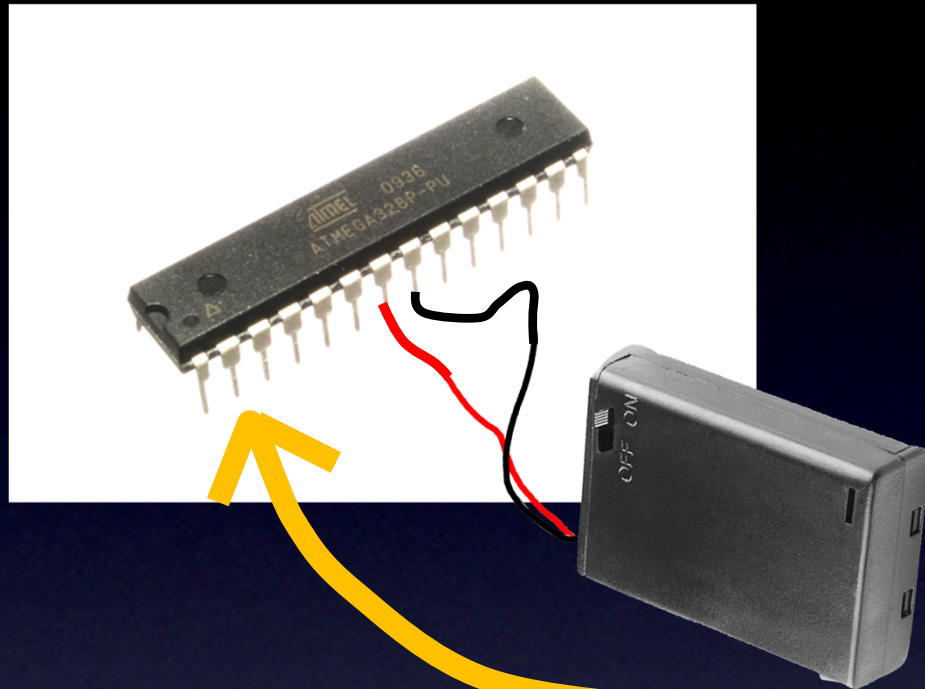
Let's add an Input pin!

and

We can add a Start button

Microcontroller

# Everything You Need to Know About Electronics



**How do we make a pin an Input pin?**

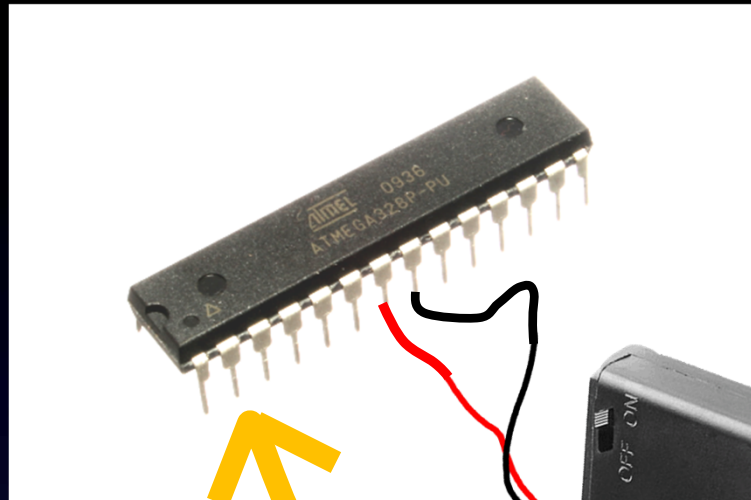
**We tell it to be one – with our program.**

***Any pin can be an Input pin (like, pin 2).***

Microcontroller – Input pins



# Everything You Need to Know About Electronics

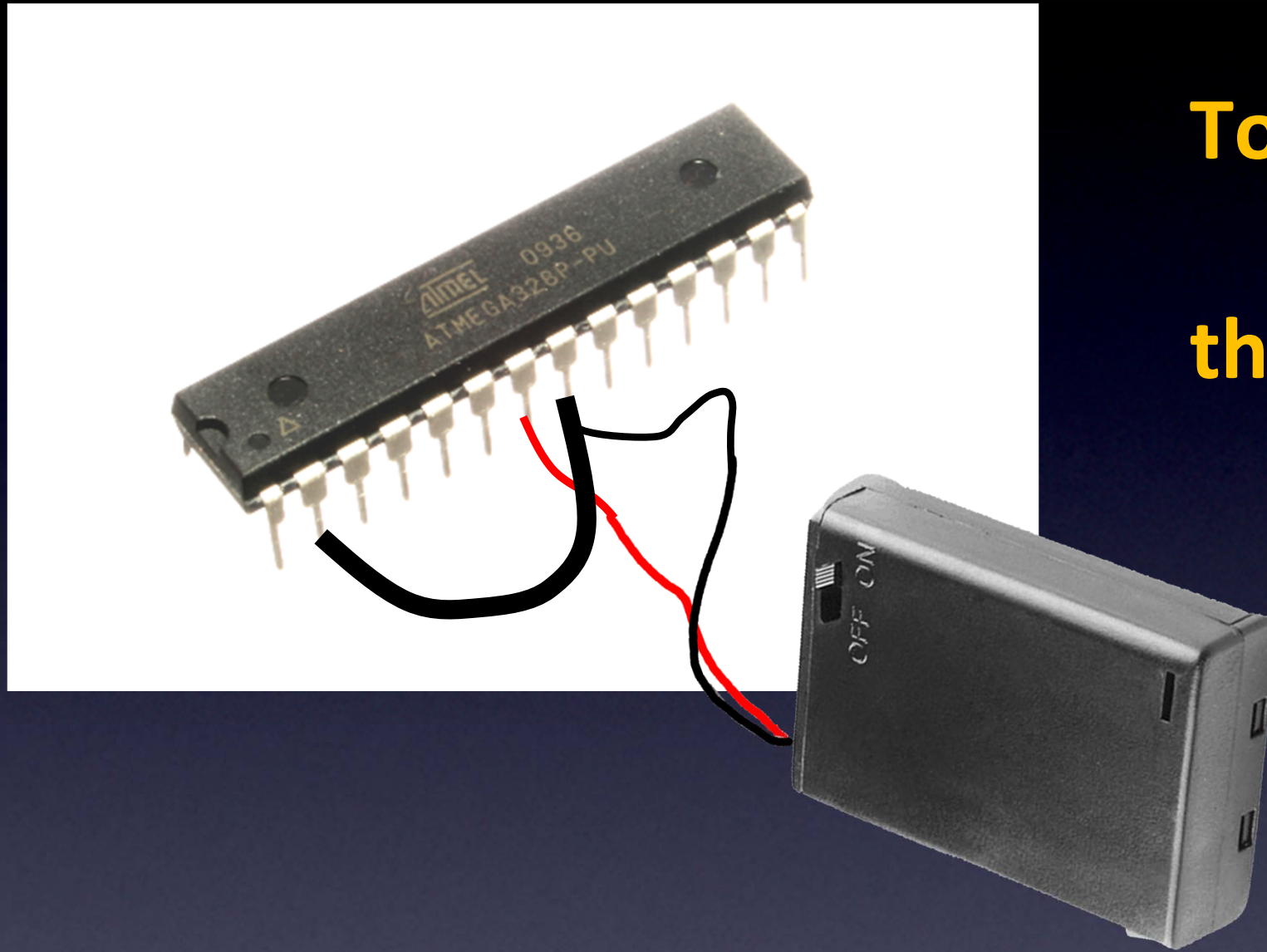


Once we have an **Input pin** (like, pin 2):

only 2 choices – is the Input pin: **High** or **Low** ?

Microcontroller – Input pins

# Everything You Need to Know About Electronics



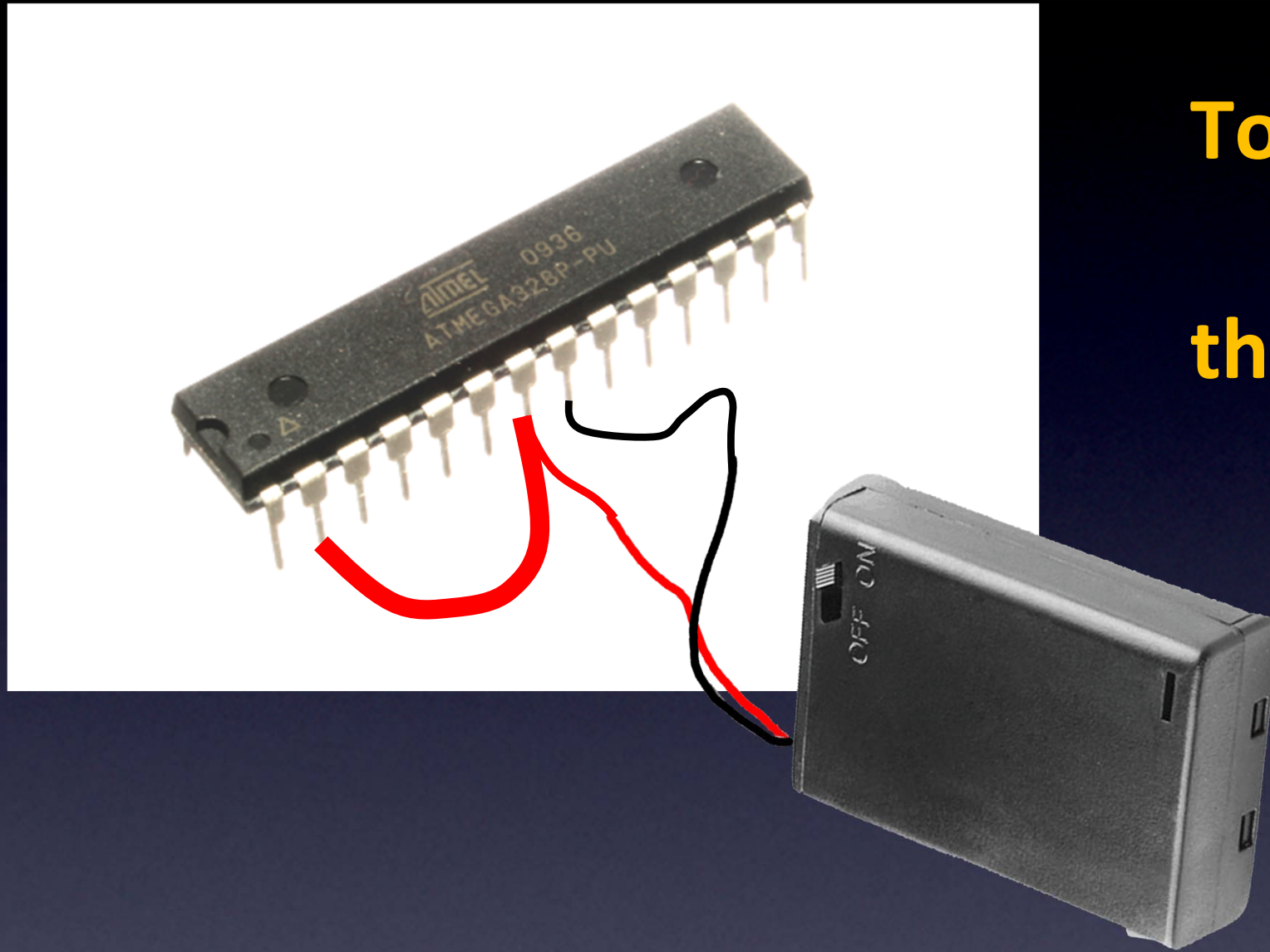
**To make the Input pin Low,  
connect it to  
the Black wire of our power  
supply (Ground).**

**Low**

Microcontroller – Input pins



# Everything You Need to Know About Electronics

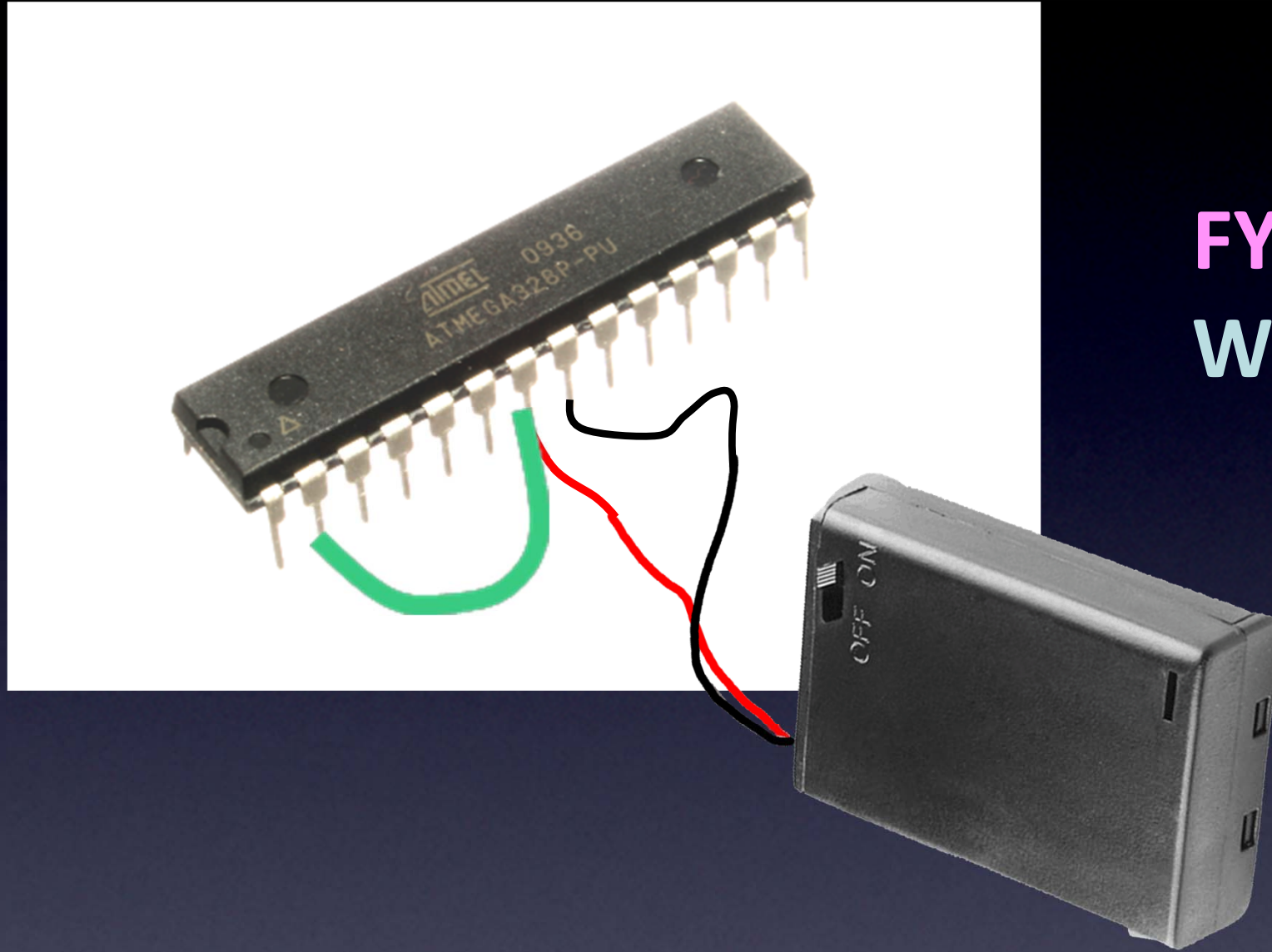


**To make the Input pin High,  
connect it to  
the Red wire of our power  
supply (Vcc).**

**High**

Microcontroller – Input pins

# Everything You Need to Know About Electronics



FYI:

Wire color does **not** matter !

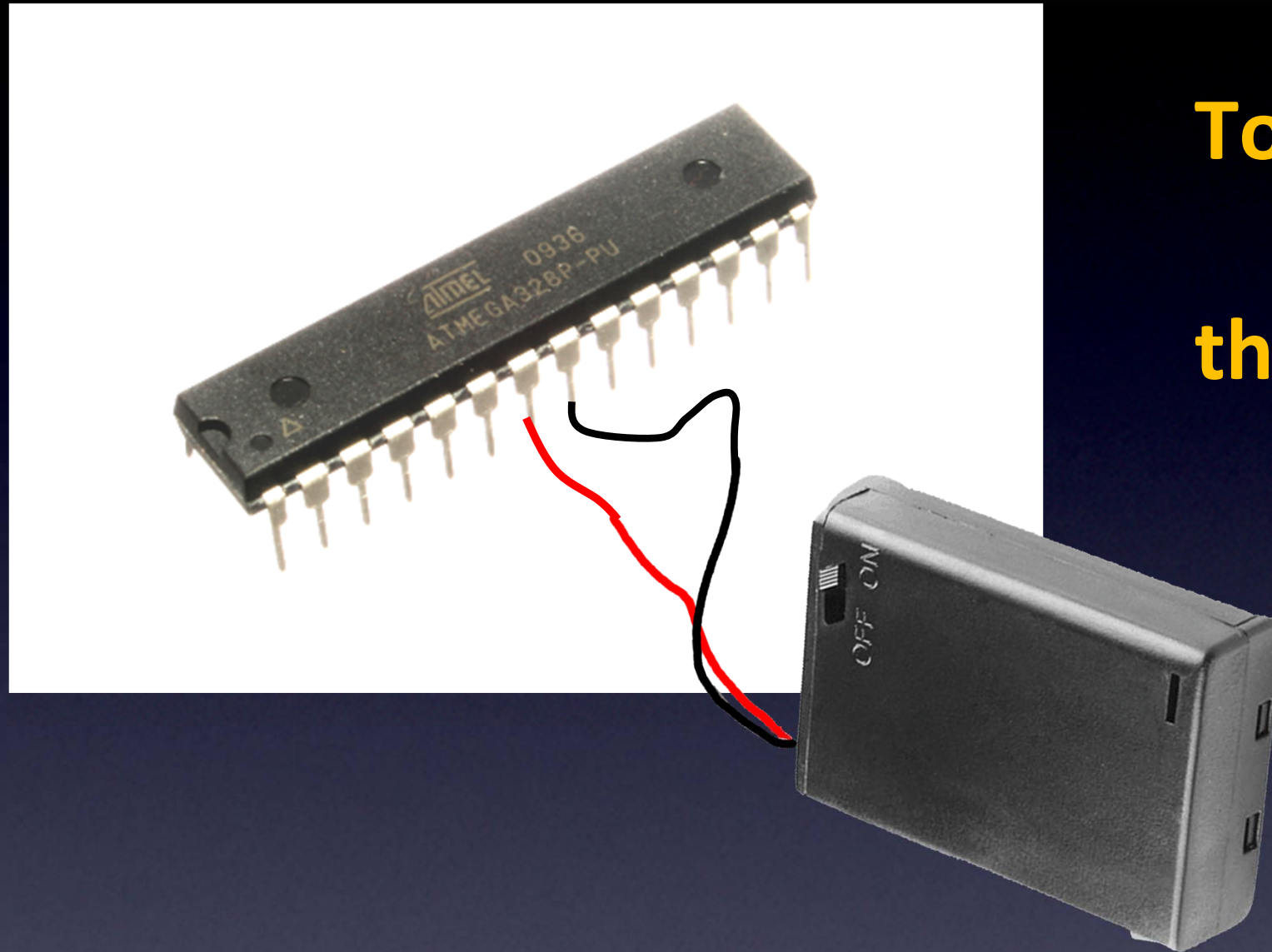
(electrons don't care)

**High**

Microcontroller – Input pins



# Everything You Need to Know About Electronics



**To make the Input pin High,  
connect it to  
the Red wire of our power  
supply (Vcc).**

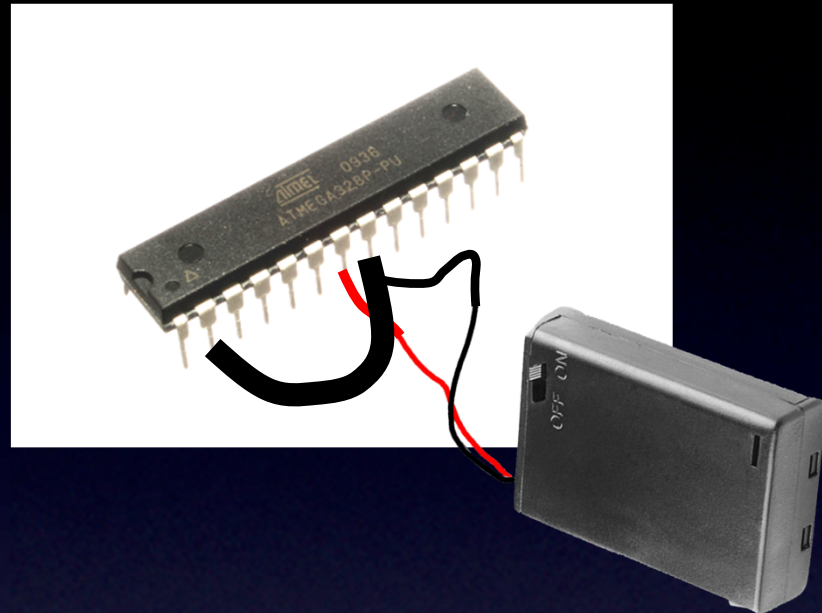
**OR:**

**just leave it blank**  
(built-in resistors on each pin)

**High**

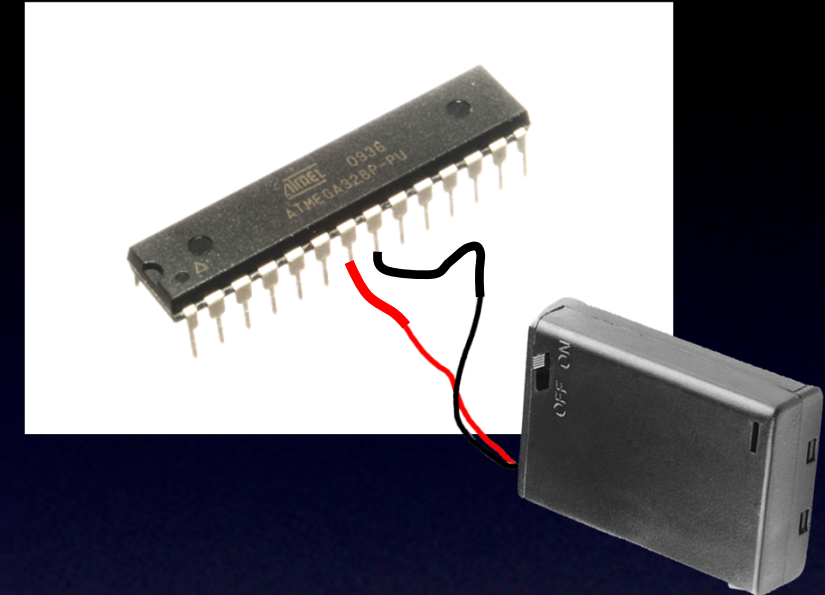
Microcontroller – Input pins

# Everything You Need to Know About Electronics



If firmware looks at  
Pin 2 when it's like this,  
it reports back:

**Low**



If firmware looks at  
Pin 2 when it's like this,  
it reports back:

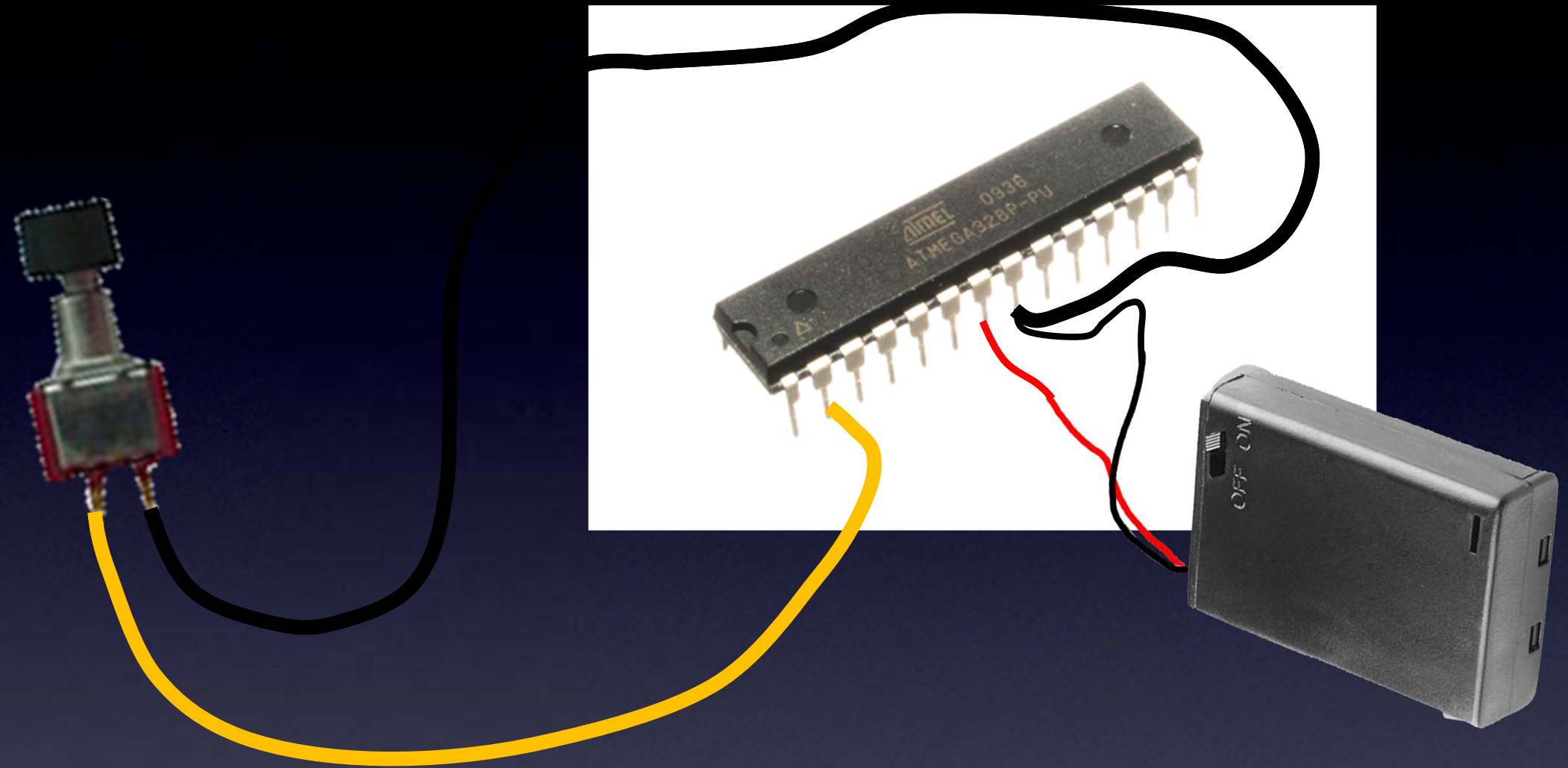
**High**

Reading the Input pin

Microcontroller – Input pins



# Everything You Need to Know About Electronics

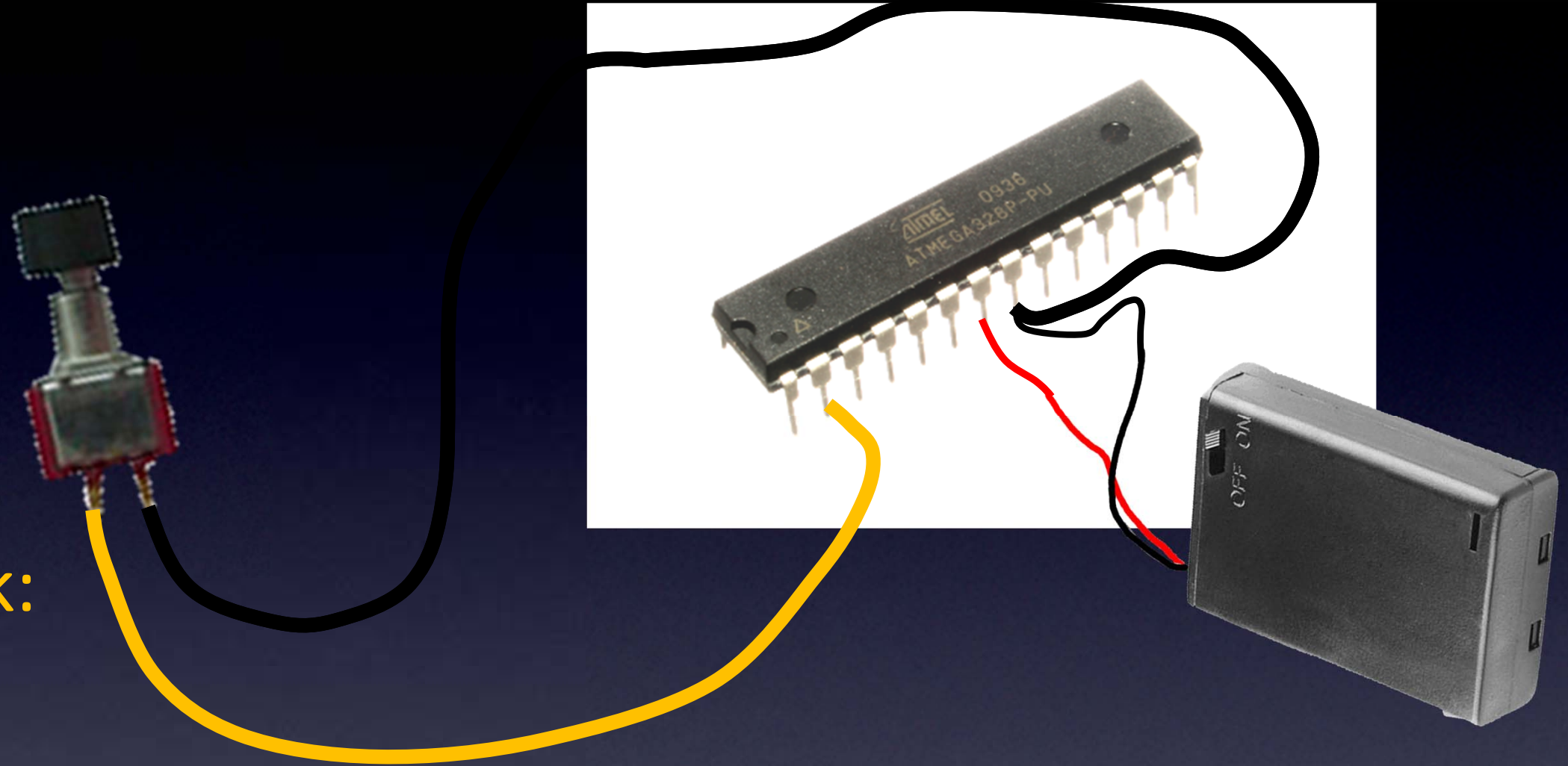


Reading the Input pin, with Switch

Microcontroller – Input pins

# Everything You Need to Know About Electronics

If firmware  
looks at Pin 2  
when switch  
NOT pushed,  
it reports back:  
**High**

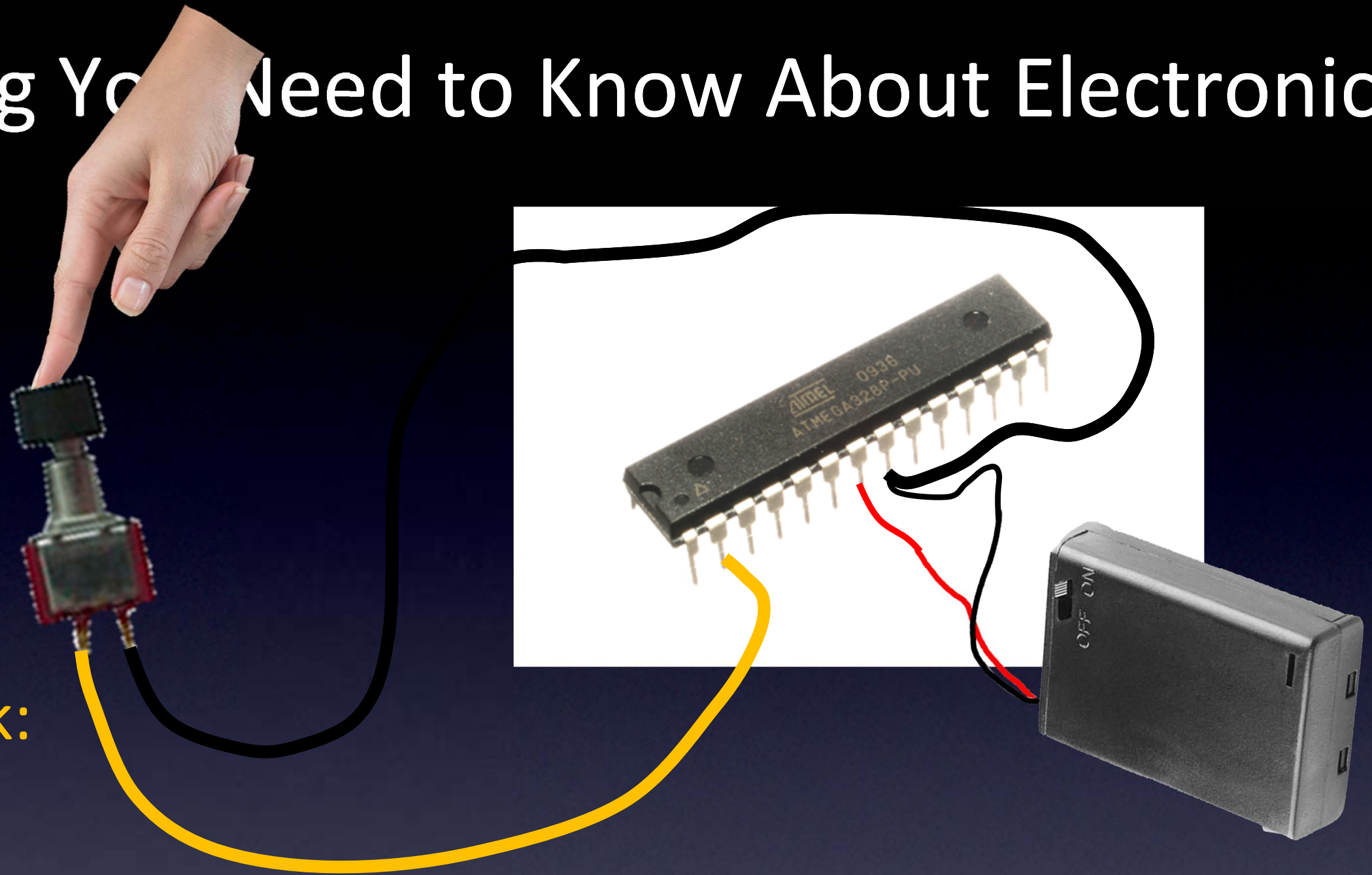


Reading the Input pin, with Switch

Microcontroller – Input pins

# Everything You Need to Know About Electronics

If firmware  
looks at Pin 2  
when switch  
pushed,  
it reports back:  
**Low**



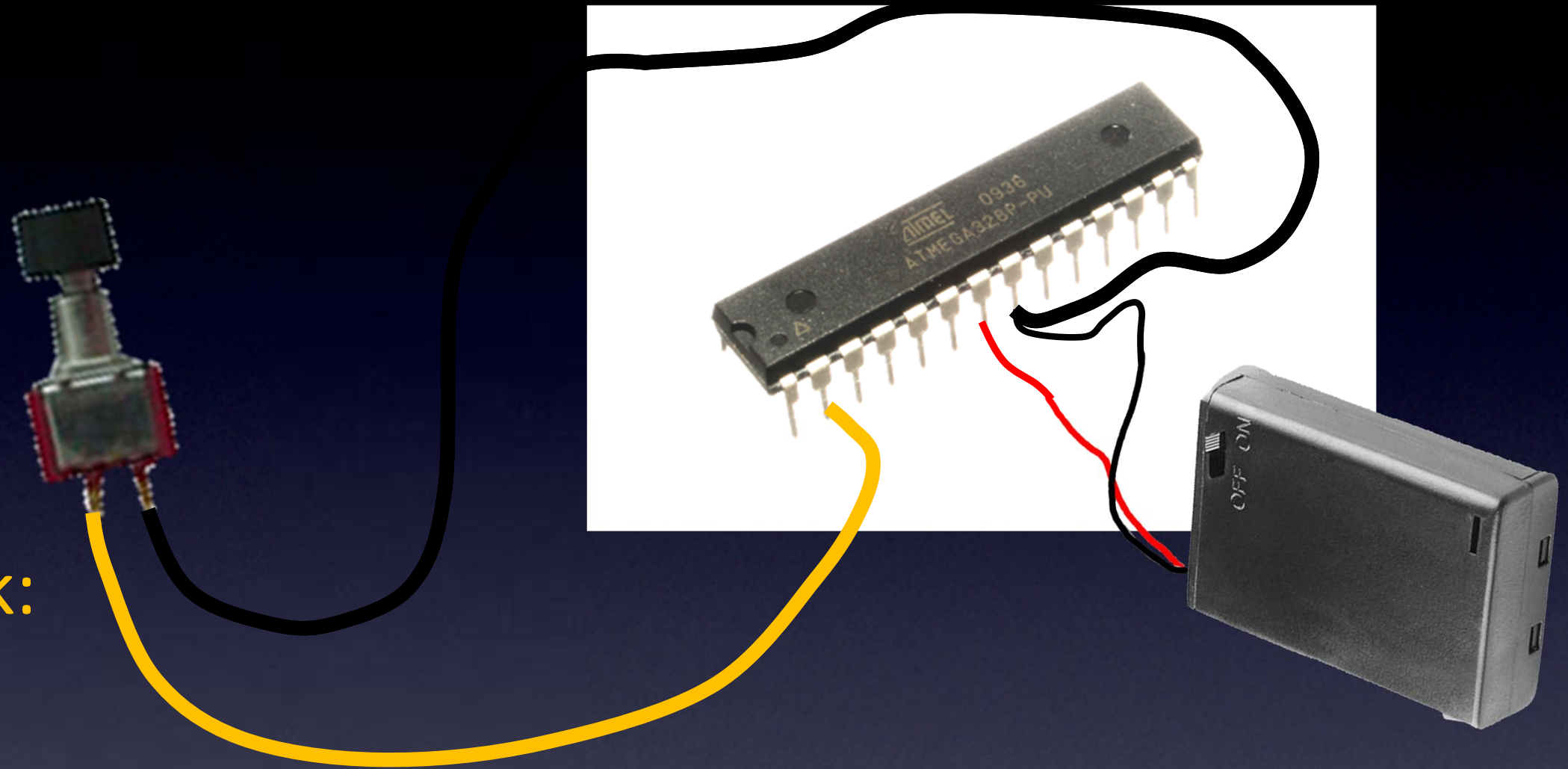
Reading the Input pin, with Switch

Microcontroller – Input pins



# Everything You Need to Know About Electronics

If firmware  
looks at Pin 2  
when switch  
NOT pushed,  
it reports back:  
**High**

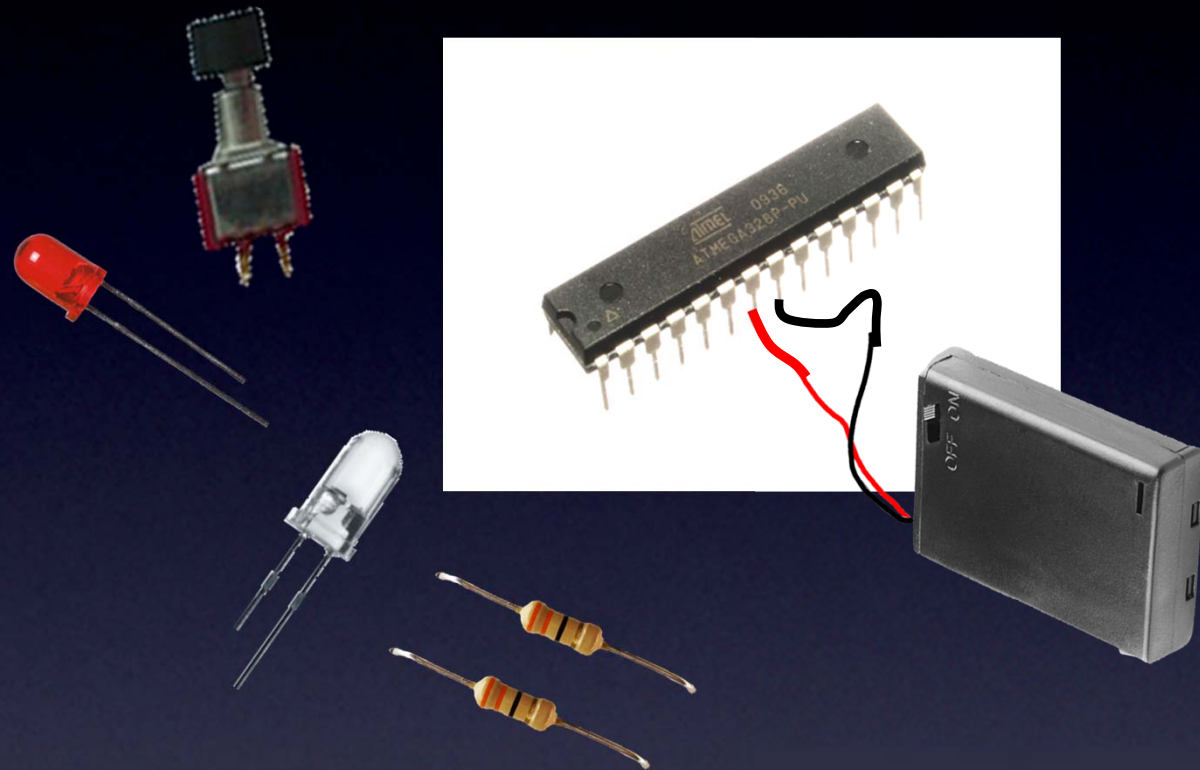


Reading the Input pin, with Switch

Microcontroller – Input pins

# Everything You Need to Know About Electronics

## Hardware



## Firmware

Pin 13 Output – visible LED pin

Pin 3 Output – IR LED pin

Pin 2 Input – Push Button

Wait for Switch to be Low

Blink visible LED:

High, Delay, Low

Pulse IR LED for Sony “OFF” code:

High, Delay, Low, Delay...

Blink visible LED:

High, Delay, Low

Pulse IR LED for Panasonic “OFF” code:

High, Delay, Low, Delay...

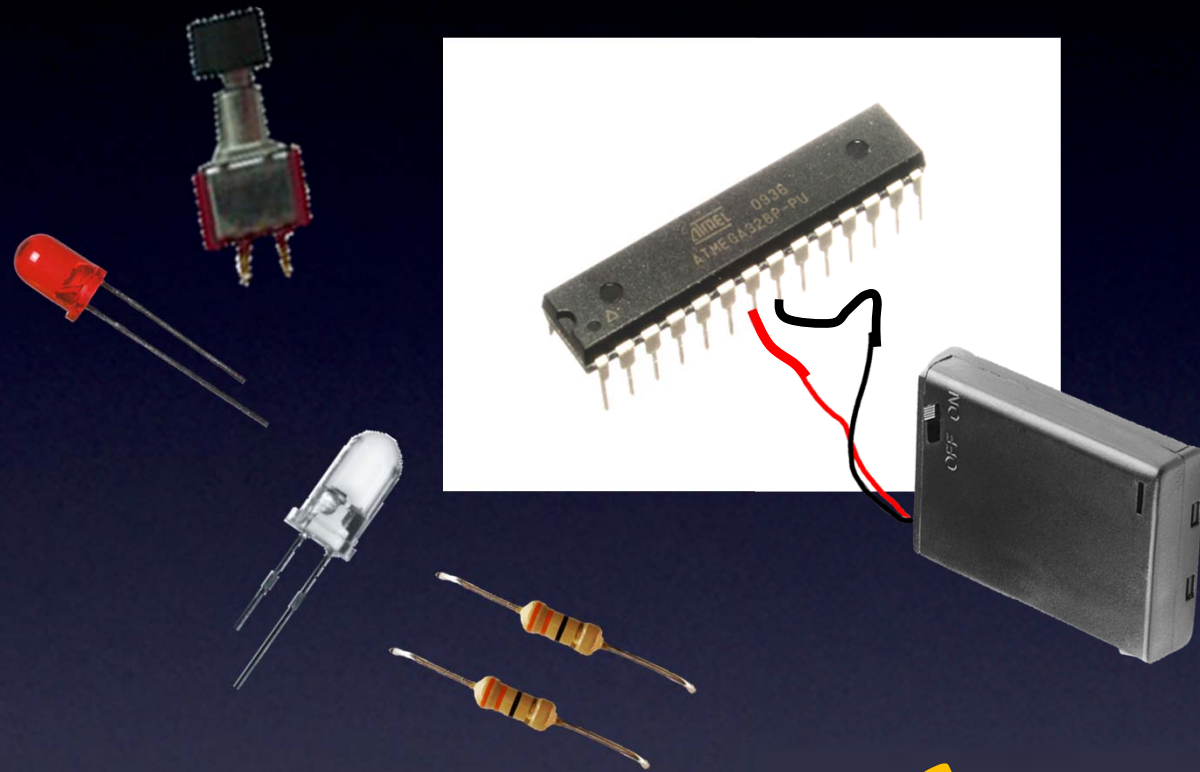
Etc for all “OFF” codes

## TV-B-Gone remote control

## Microcontroller

# Everything You Need to Know About Electronics

## Hardware



**Except**  
doesn't go very far

## Firmware

Pin 13 Output – visible LED pin

Pin 3 Output – IR LED pin

Pin 2 Input – Push Button

Wait for Switch to be Low

Blink visible LED:

High, Delay, Low

Pulse IR LED for Sony “OFF” code:

High, Delay, Low, Delay...

Blink visible LED:

High, Delay, Low

Pulse IR LED for Panasonic “OFF” code:

High, Delay, Low, Delay...

Etc for all “OFF” codes

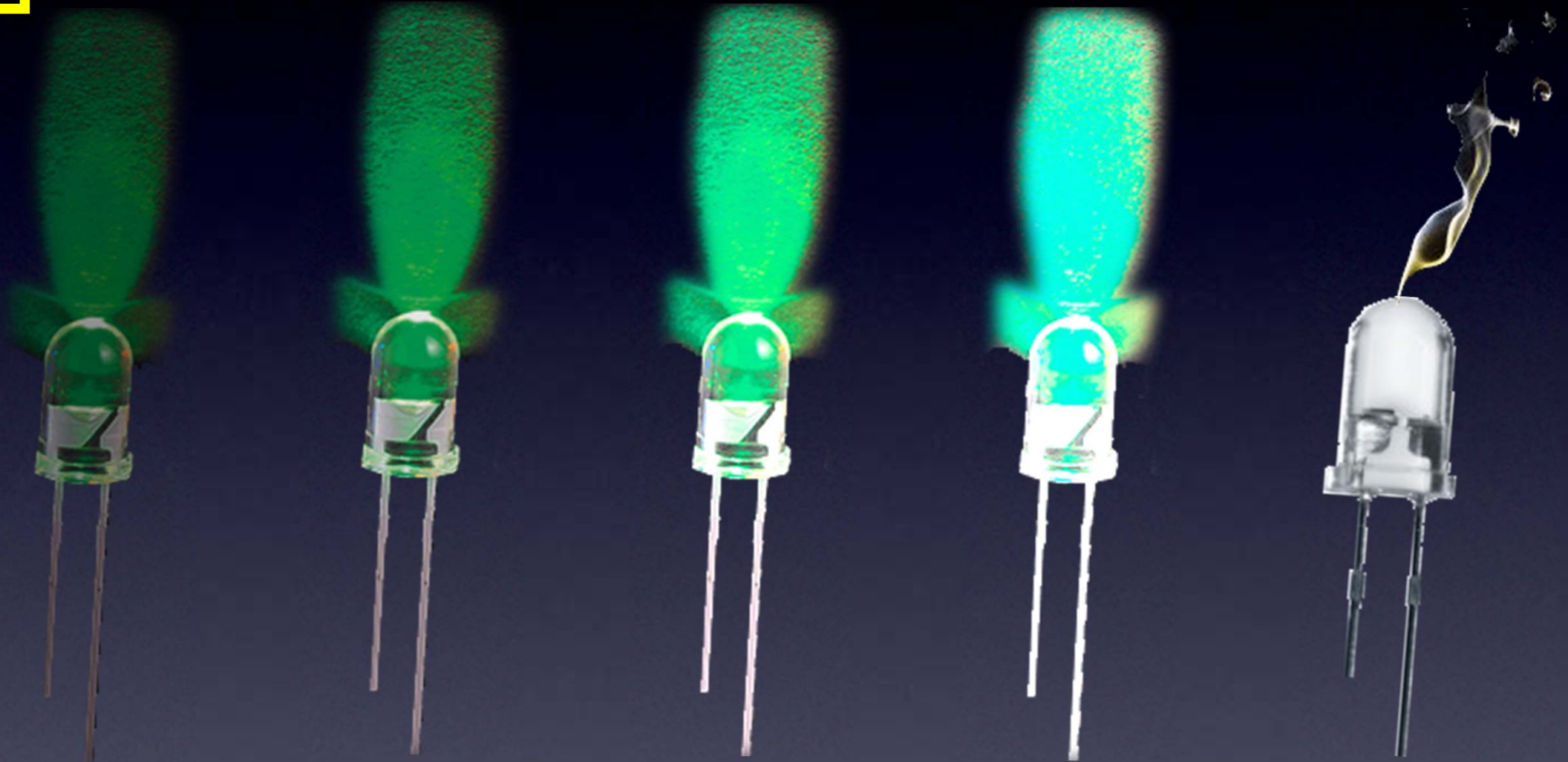
## TV-B-Gone remote control

## Microcontroller



# Everything You Need to Know About Electronics

## Review:



More current → More brightness! (until...)

LED

# Everything You Need to Know About Electronics

Output pin – only 2 choices:

Low

Off

(0V)

High

On

(Power supply voltage

-- *controlled by our Firmware!*)

Output pins

only allow

limited current

(built-in resistors on each pin)



Microcontroller – Output pins

# Everything You Need to Know About Electronics



dimly lit LED

Output pin – only 2 choices:

Low

High

Off

On

(0V)

(Power supply voltage  
-- controlled by our Firmware!)

So,  
IR LED  
can only light up  
dimly  
from the Output pin

Output pin – only limited current

Microcontroller – Output pins



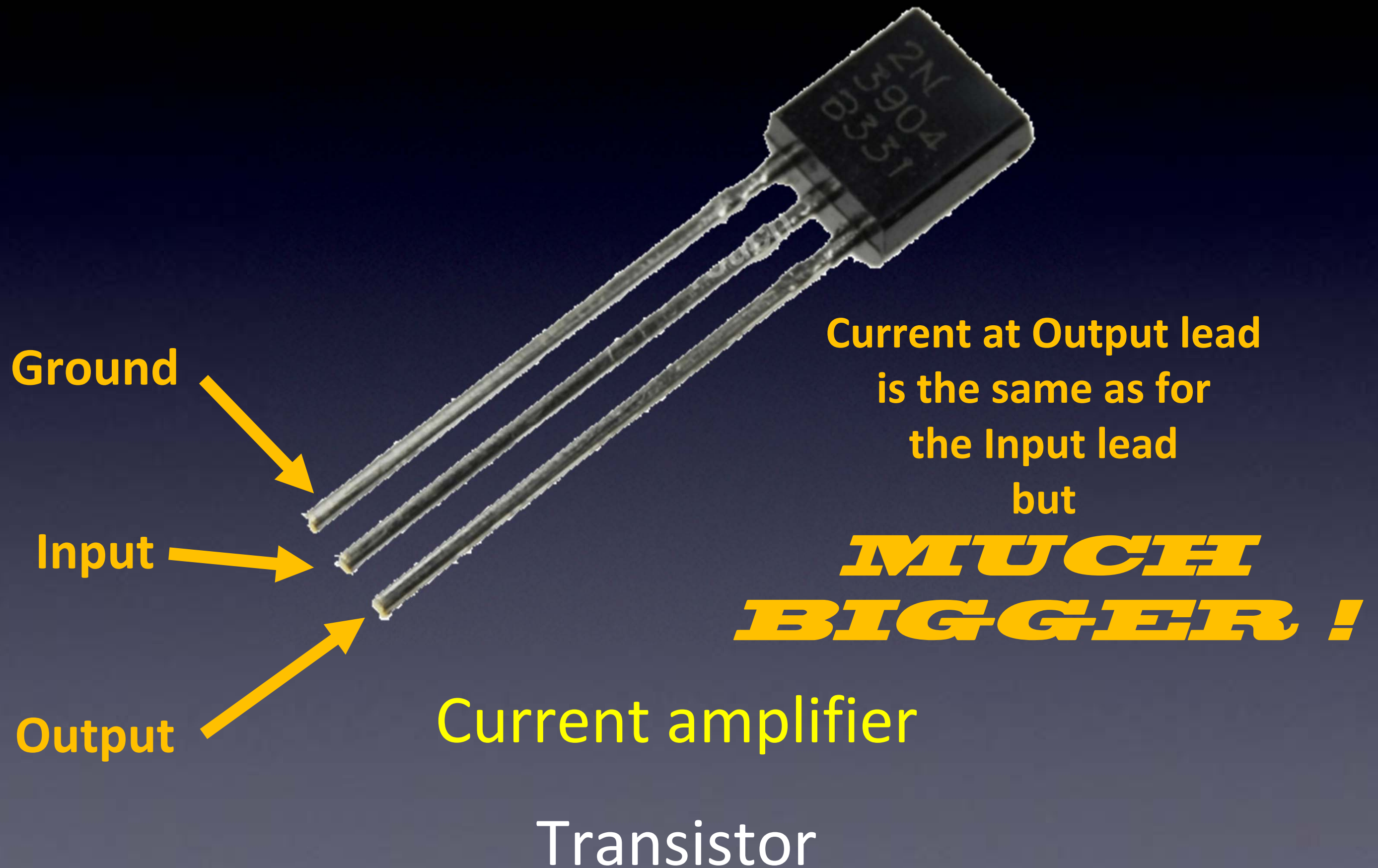
# Everything You Need to Know About Electronics

So,  
*let's amplify*  
*the current*  
from the Output pin

with  
a

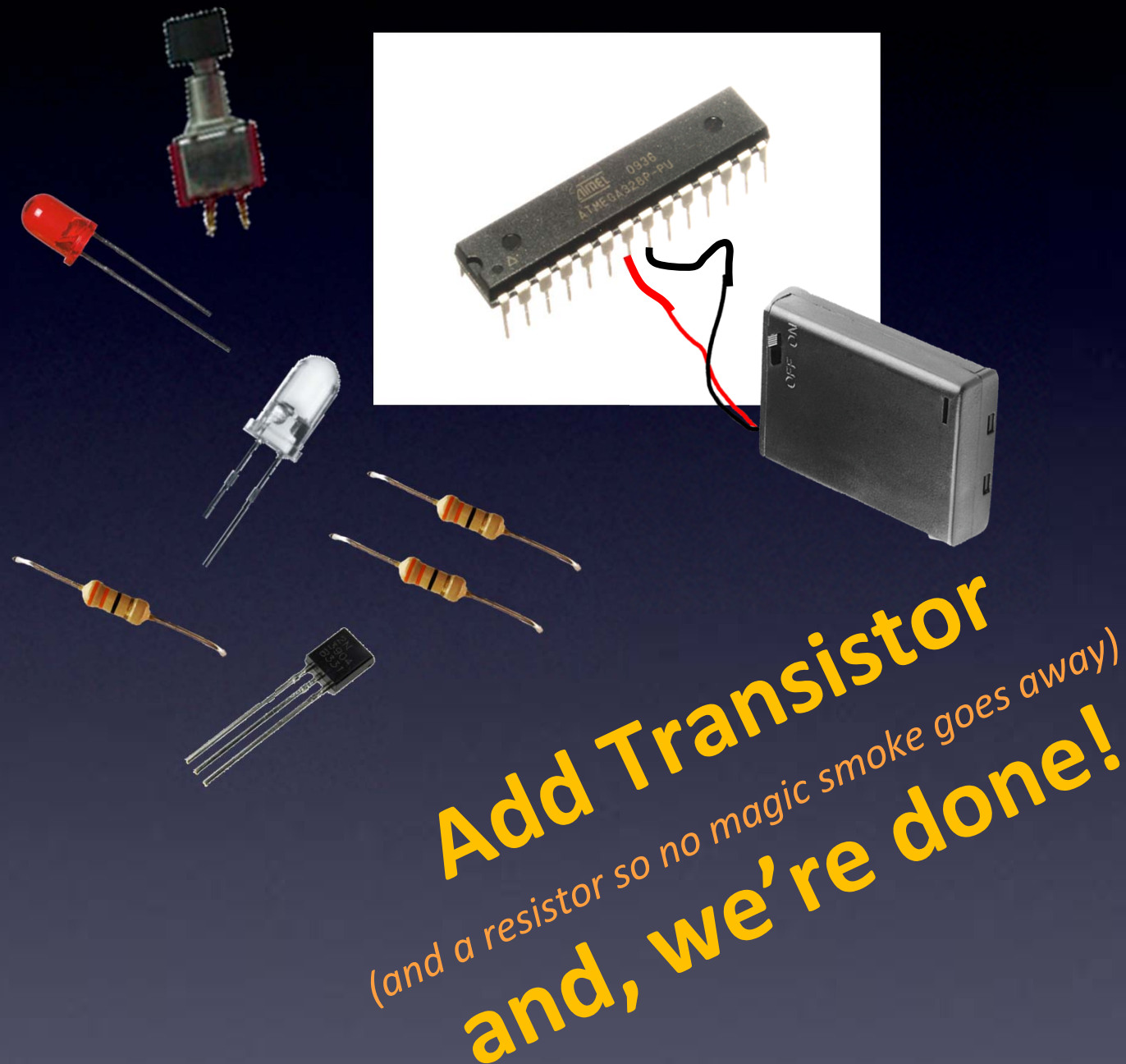
Current amplifier !

# Everything You Need to Know About Electronics



# Everything You Need to Know About Electronics

## Hardware



## Firmware

Pin 13 Output – visible LED pin

Pin 3 Output – IR LED pin

Pin 2 Input – Push Button

Wait for Switch to be Low

Blink visible LED:

High, Delay, Low

Pulse IR LED for Sony “OFF” code:

High, Delay, Low, Delay...

Blink visible LED:

High, Delay, Low

Pulse IR LED for Panasonic “OFF” code:

High, Delay, Low, Delay...

Etc for all “OFF” codes

# TV-B-Gone remote control – we're done!

## Microcontroller



And, that is

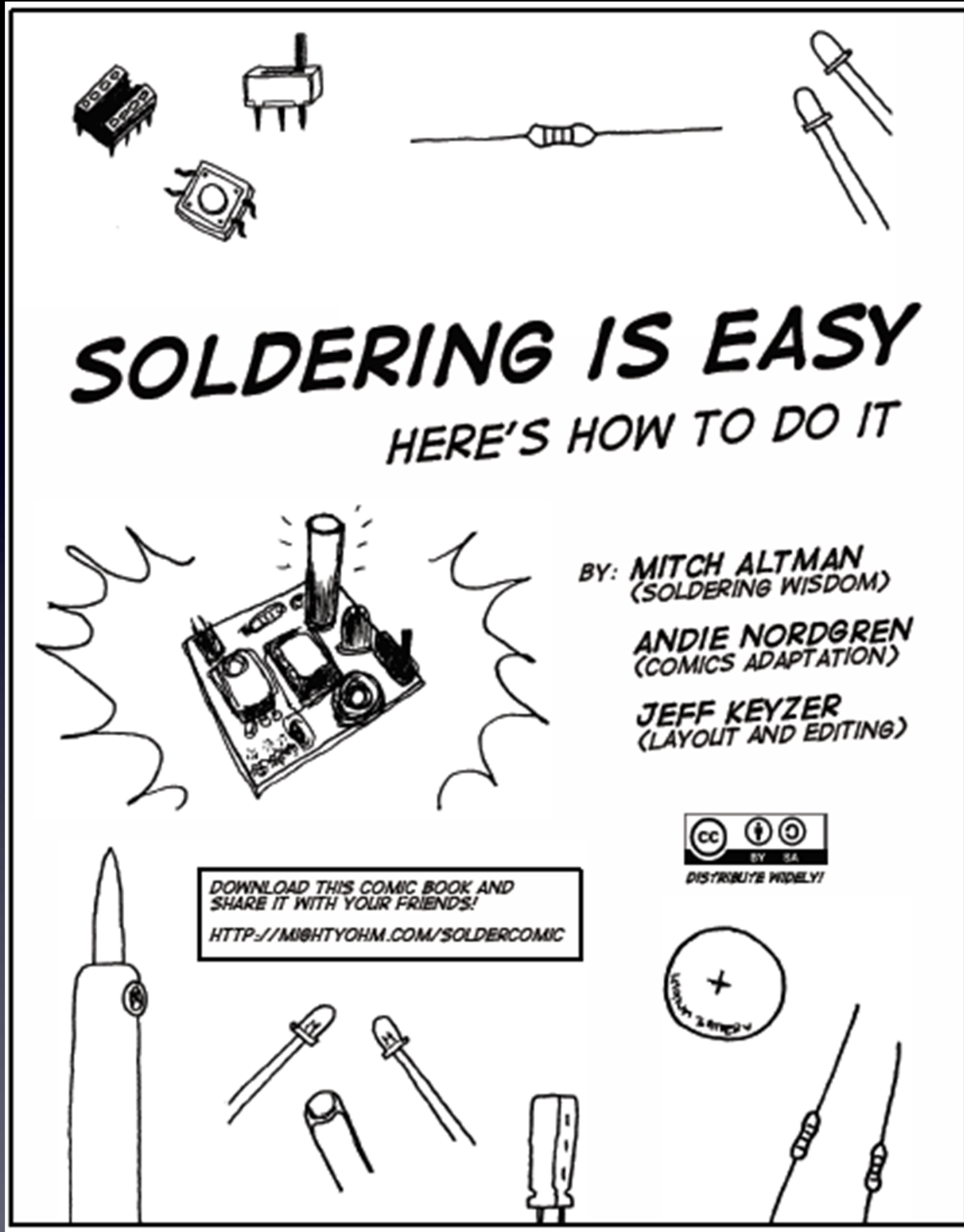
And, that is

*Everything You Need to Know About*  
***Electronics***

Questions?

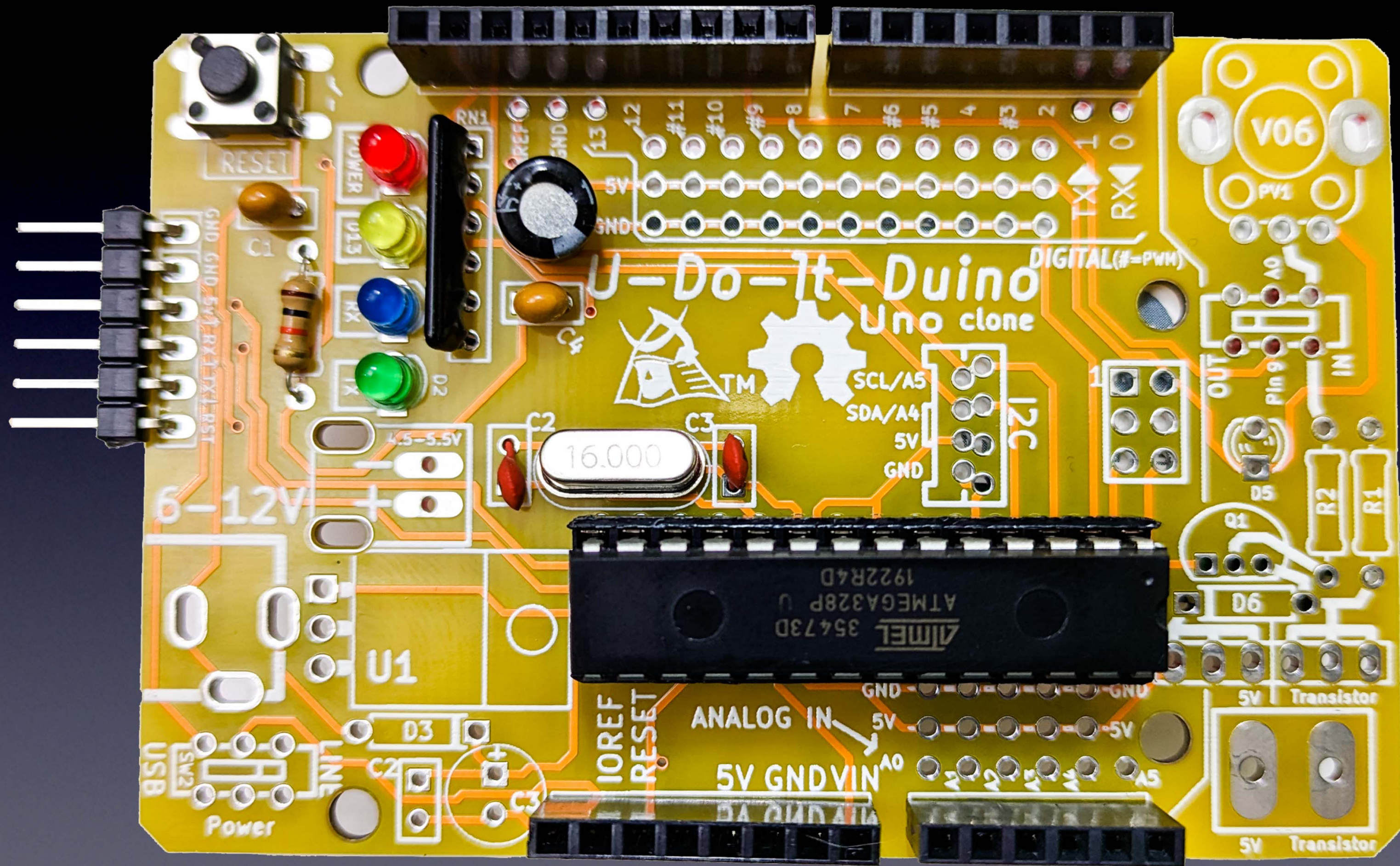


# Learn To Solder



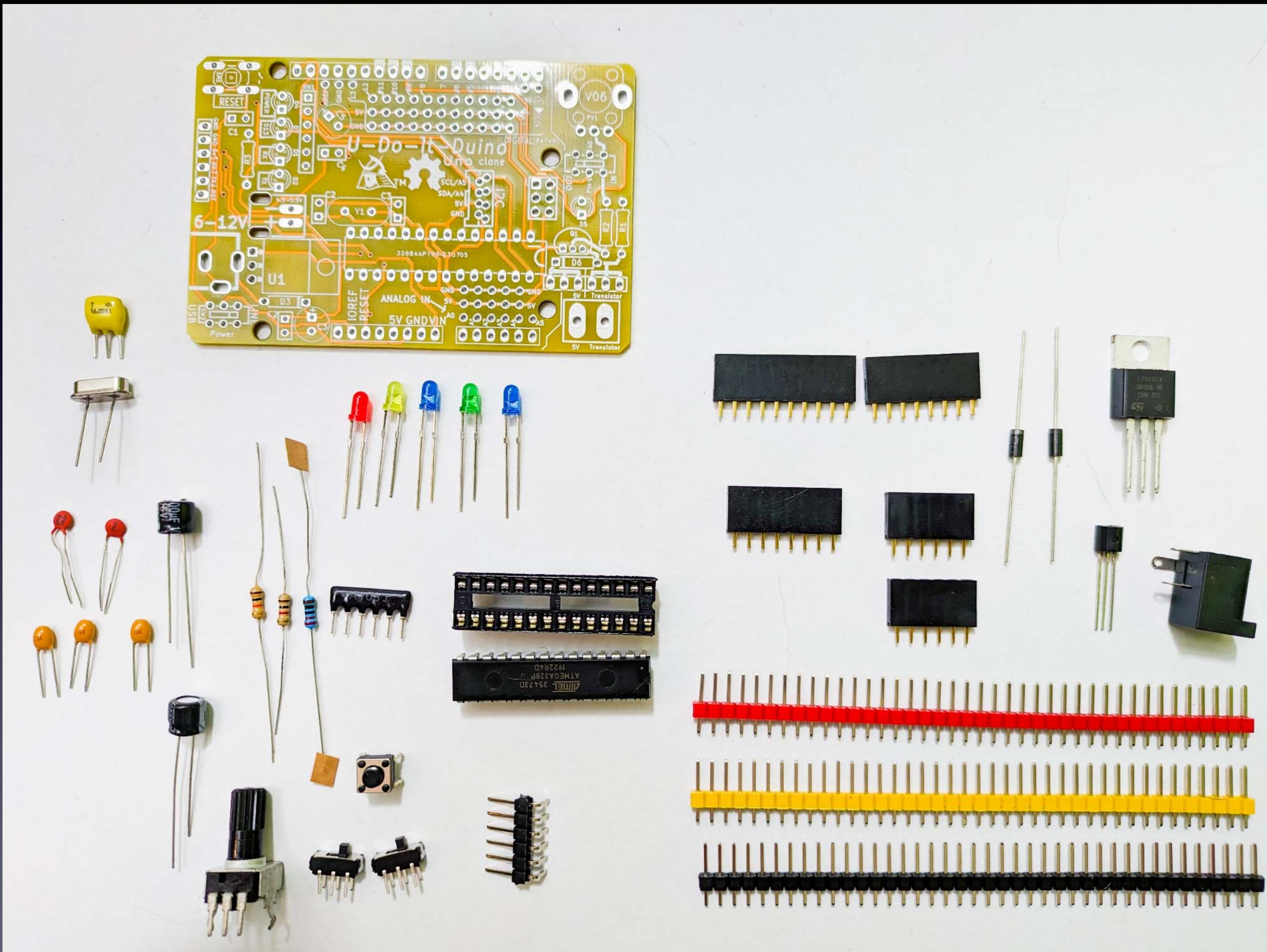
download for free at:  
<http://mightyohm.com/soldercomic>





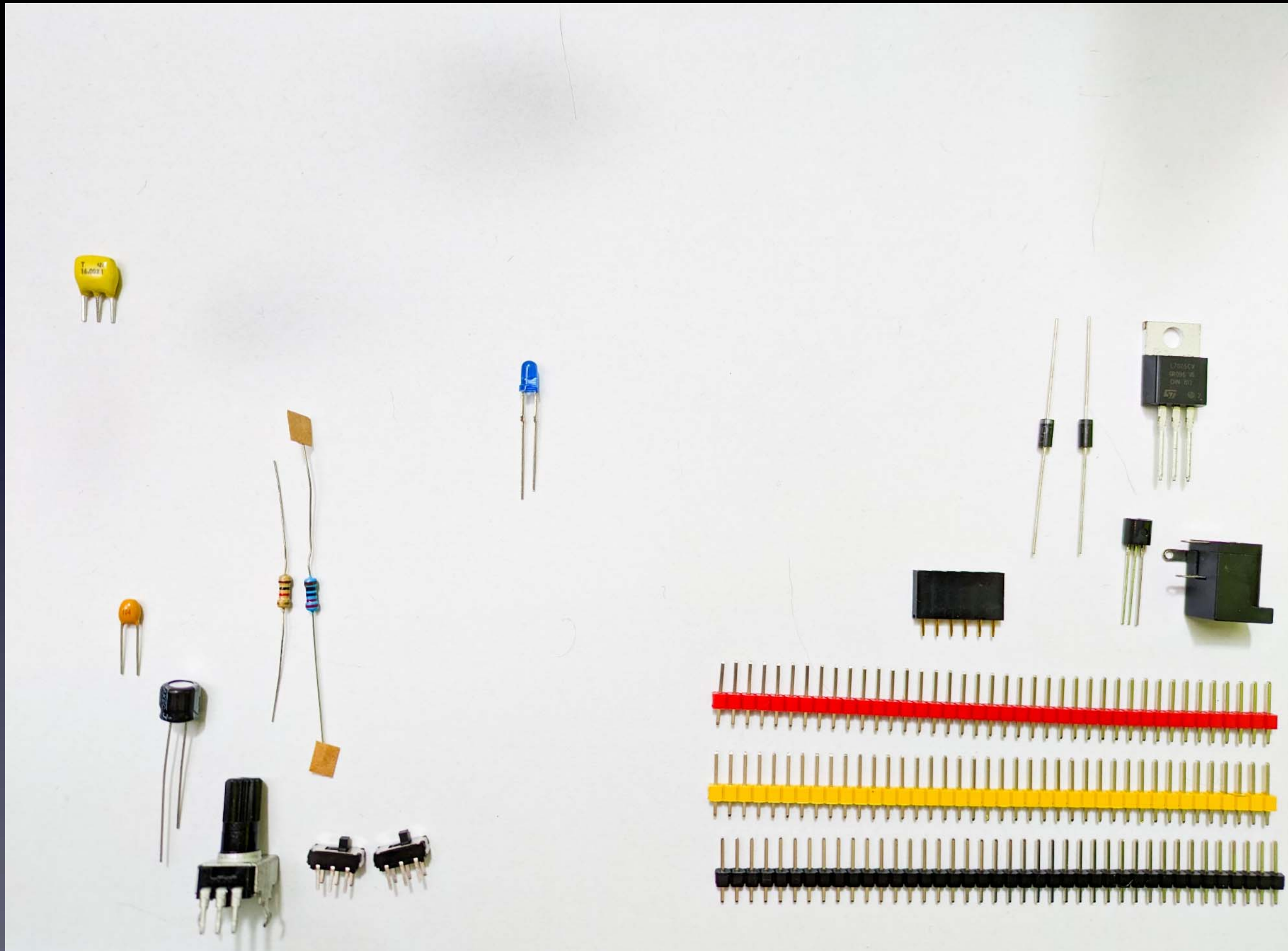


# U-Do-It-Duino – we won't use all of the parts



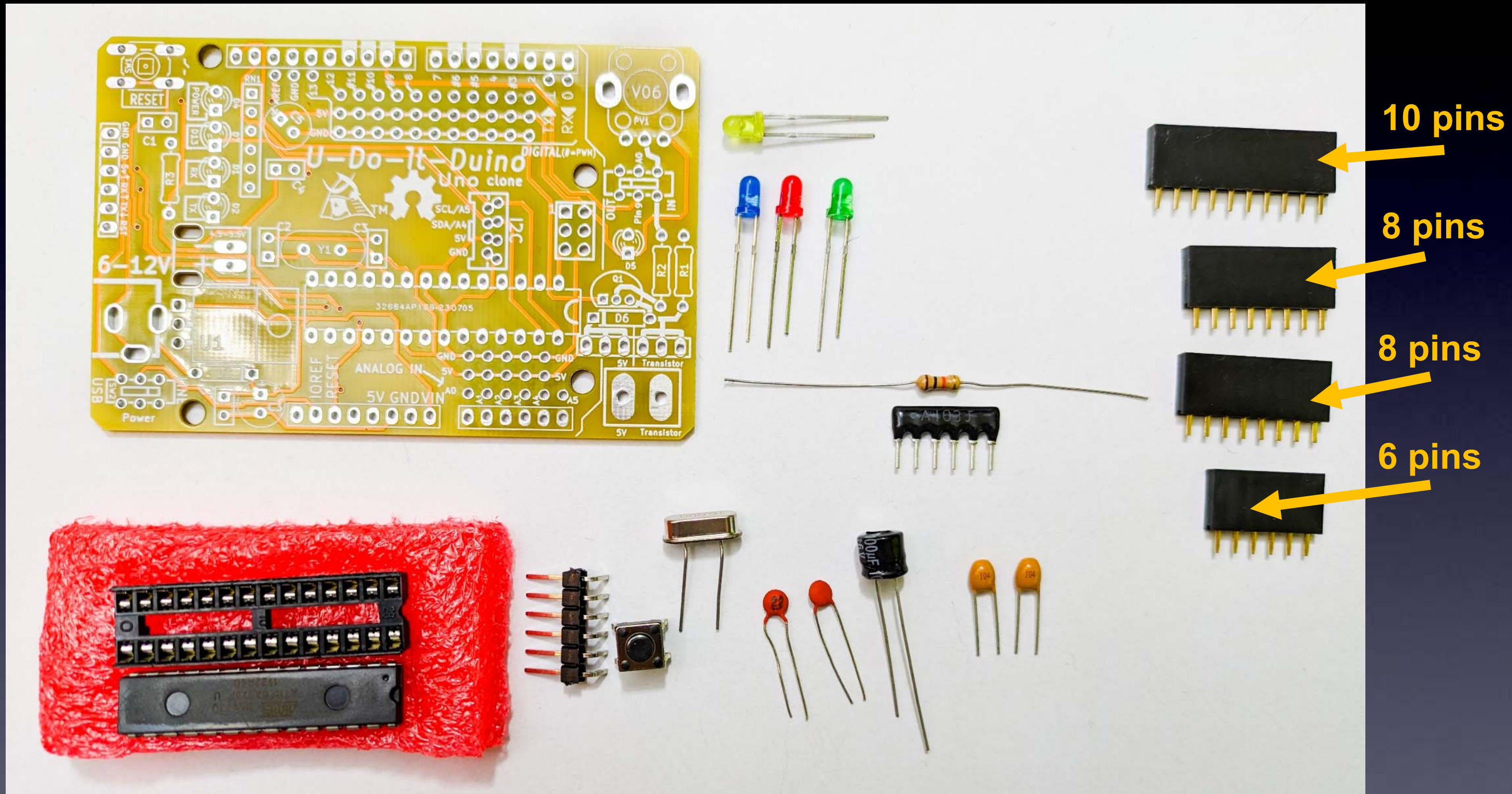


# Unused parts



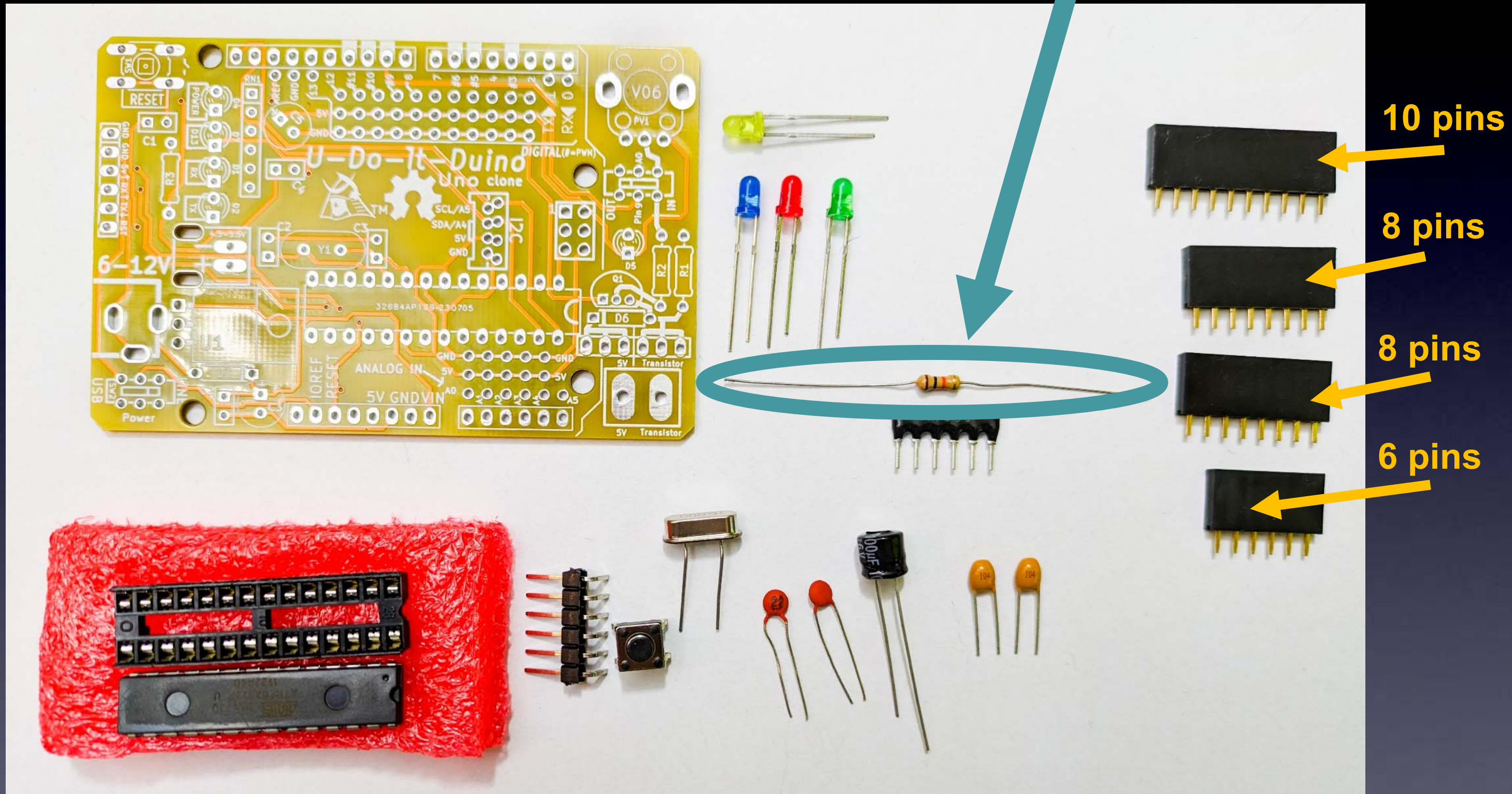


# Parts to use





# Our first part to solder: R3





# Our first part to solder: R3



**R3: Brown, Black, Orange, Gold**

*(NOT Brown, Black, Red, Gold)*

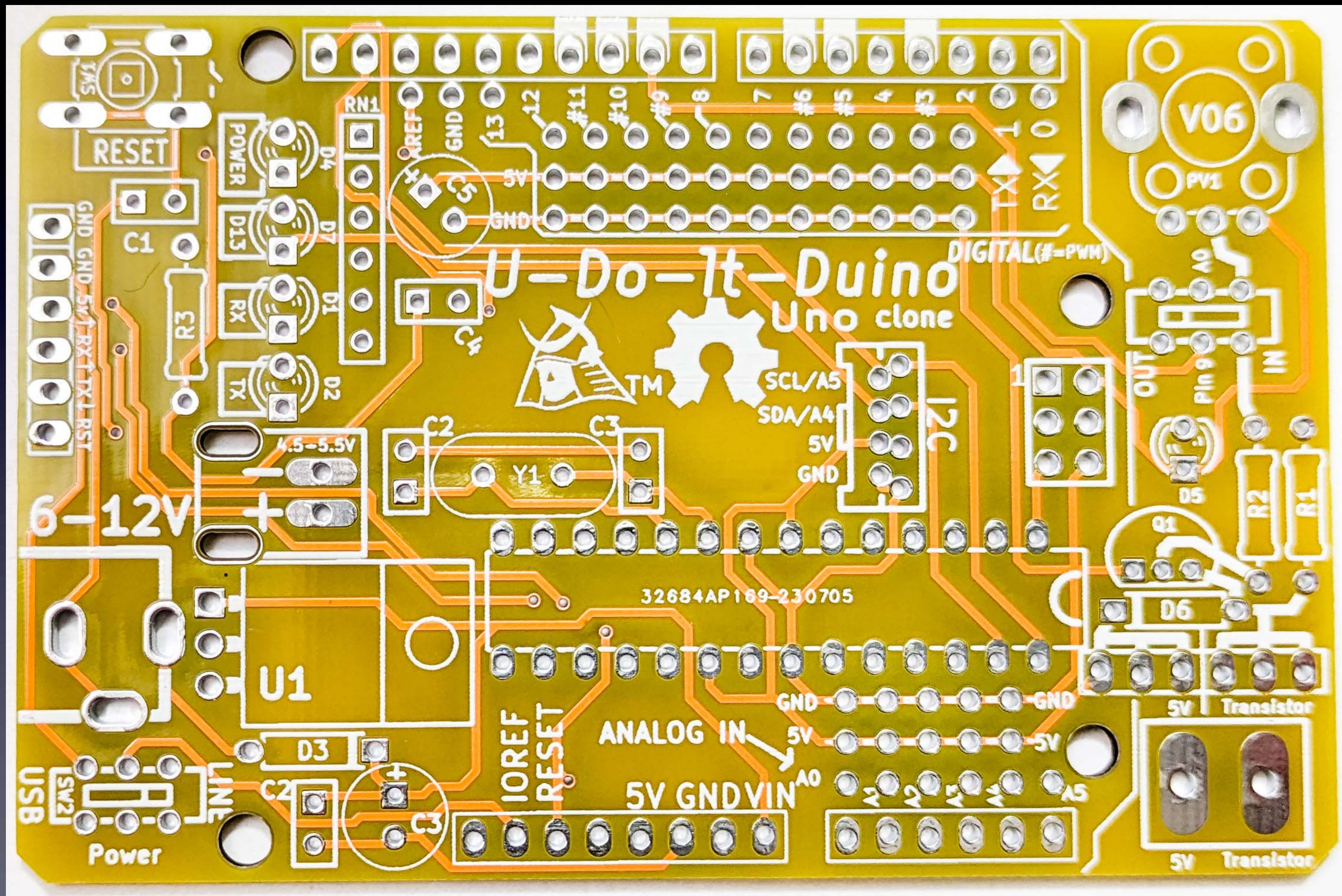
**R3: Look down at the shape of this part**



**R3: Brown, Black, Orange, Gold**

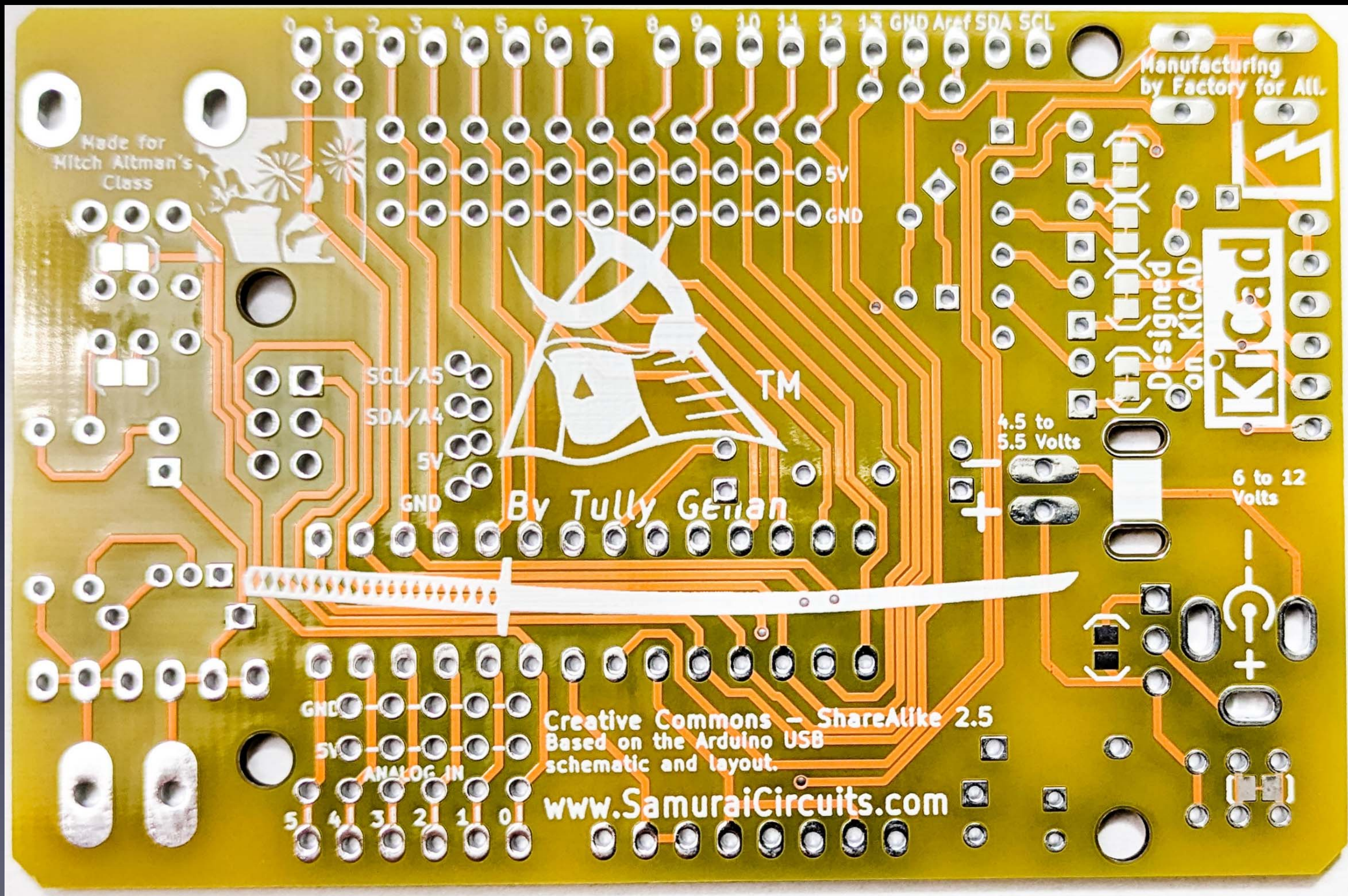


# Top of Board



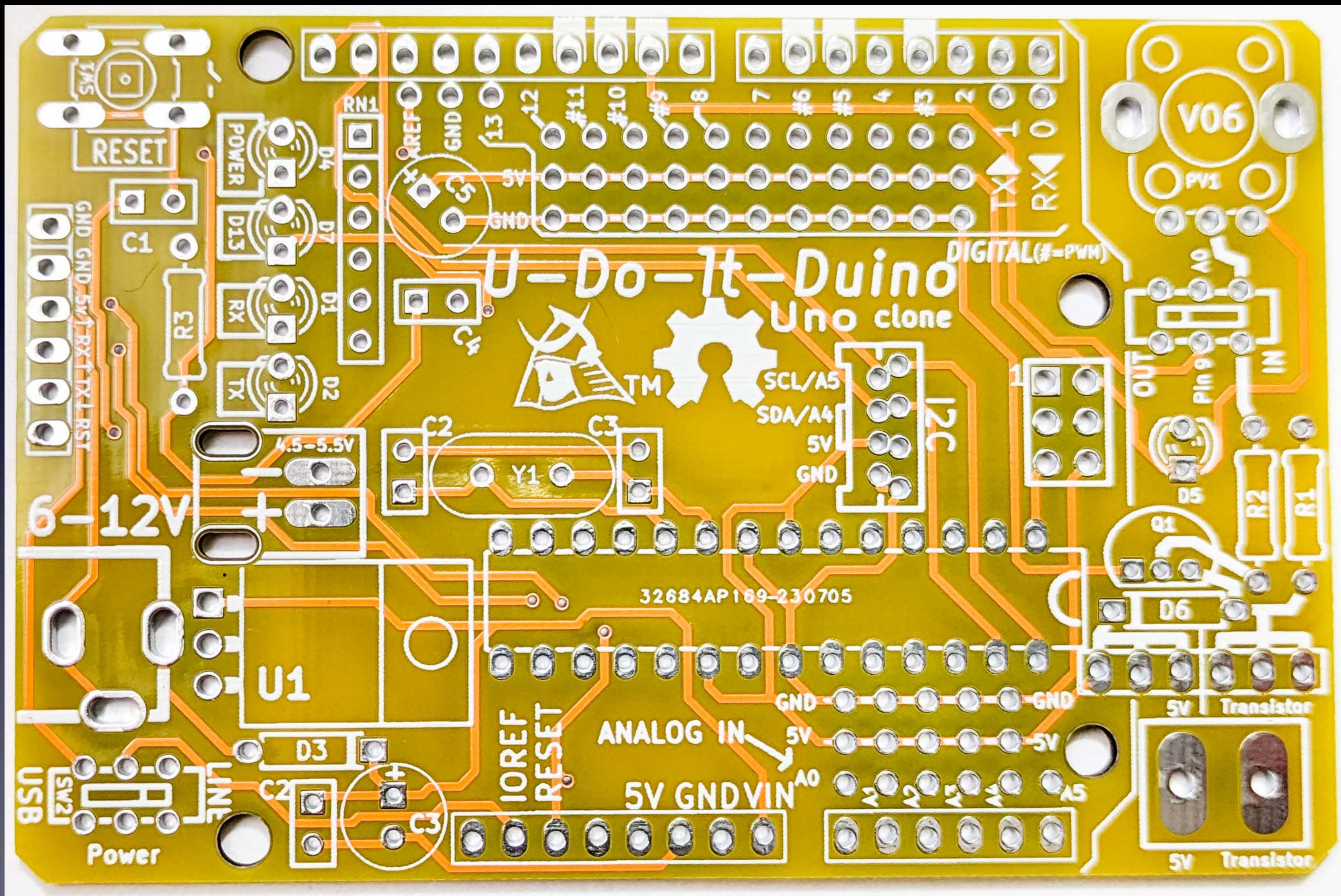


# Bottom of Board

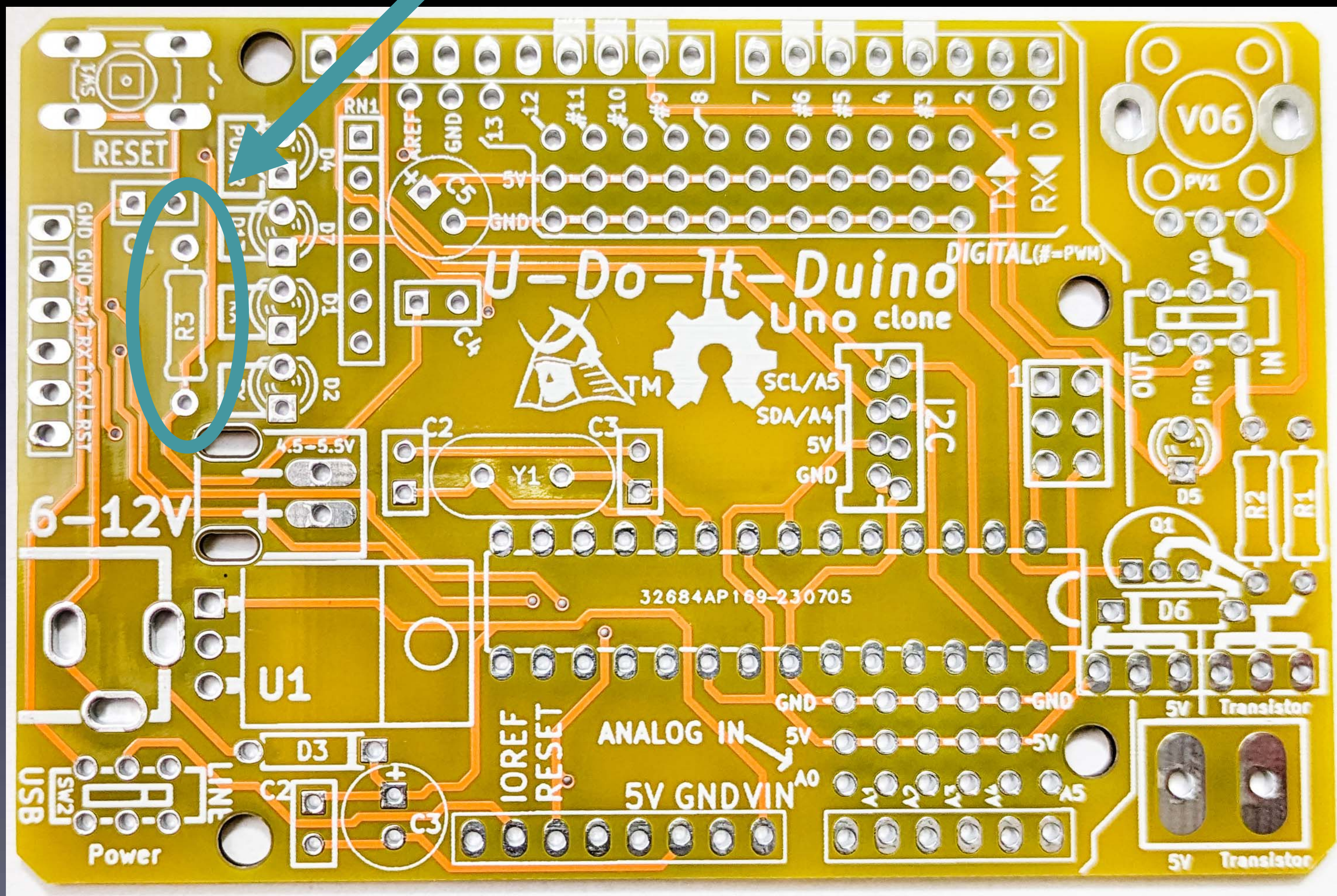




**All Parts are soldered on the Top of Board !**





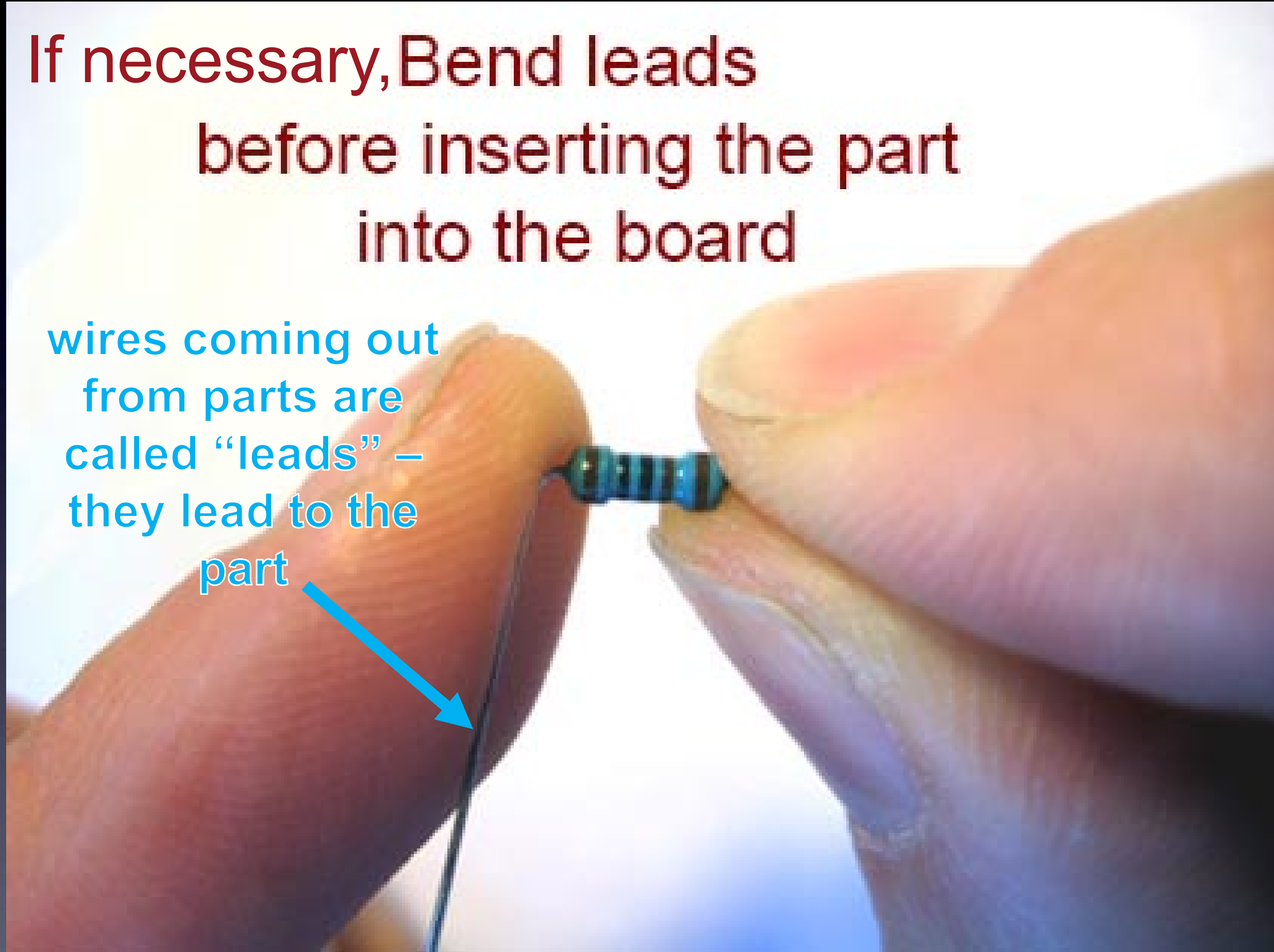




# Some parts, such as resistors, need their leads bent first

If necessary, Bend leads  
before inserting the part  
into the board

wires coming out  
from parts are  
called “leads” –  
they lead to the  
part





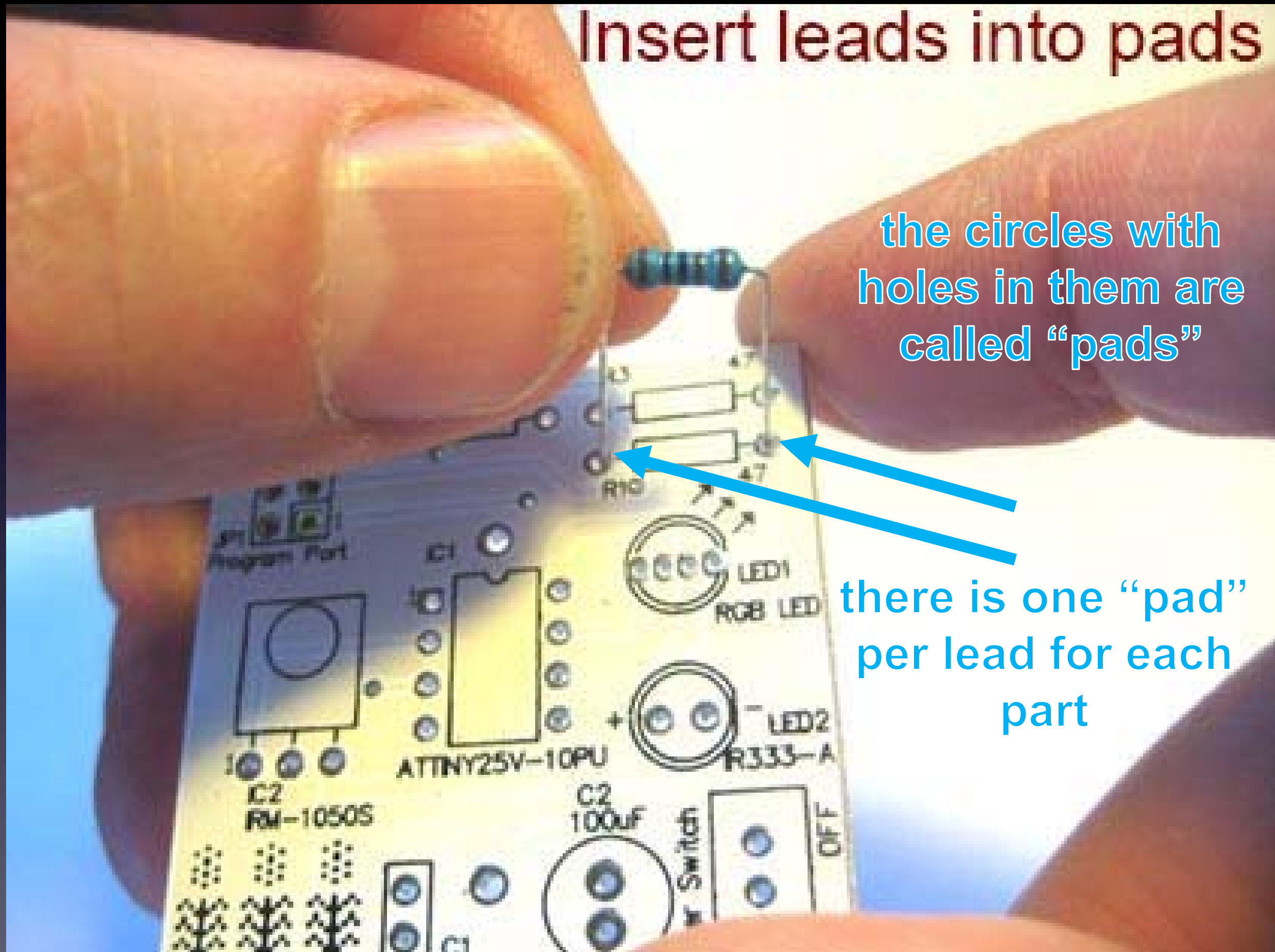
**This is how a resistor looks *before* inserting it into the board**



Insert leads into pads

the circles with  
holes in them are  
called "pads"

there is one "pad"  
per lead for each  
part



**R3: leads inserted  
into their pads**





R3: board upside down



Bend leads  
half way out

(only half way) like a “V”

so that the part won't fall out while soldering it

R3: board upside down



Bend leads  
half way out

(only half way) like a “V”

**Ready to Solder !**





# How to hold a soldering iron

(Like a pencil – held from underneath)

**Important**

The perfect kind of solder for  
electronics:

60/40 rosin core,  
0.031" (0.7mm) diameter (or smaller)

*(63/37 is also good)*

Important:

Use solder WITH lead (Pb) !!  
lead-free solder  
has very poisonous fumes!



# 3 Safety Tips...

# Safety Tip #1:

## Hot !!

(When you touch the tip,  
*you will* let go quickly -- every time!)



# Safety Tip #2:

Lead (Pb) is toxic

But it easily washes off your hands  
with soap and water

Safety Tip #3:

*(coming soon)*



2 secrets  
to good soldering...

Secret #1:

Clean the tip!

(before every solder connection)

Bang (lightly) 3 times,

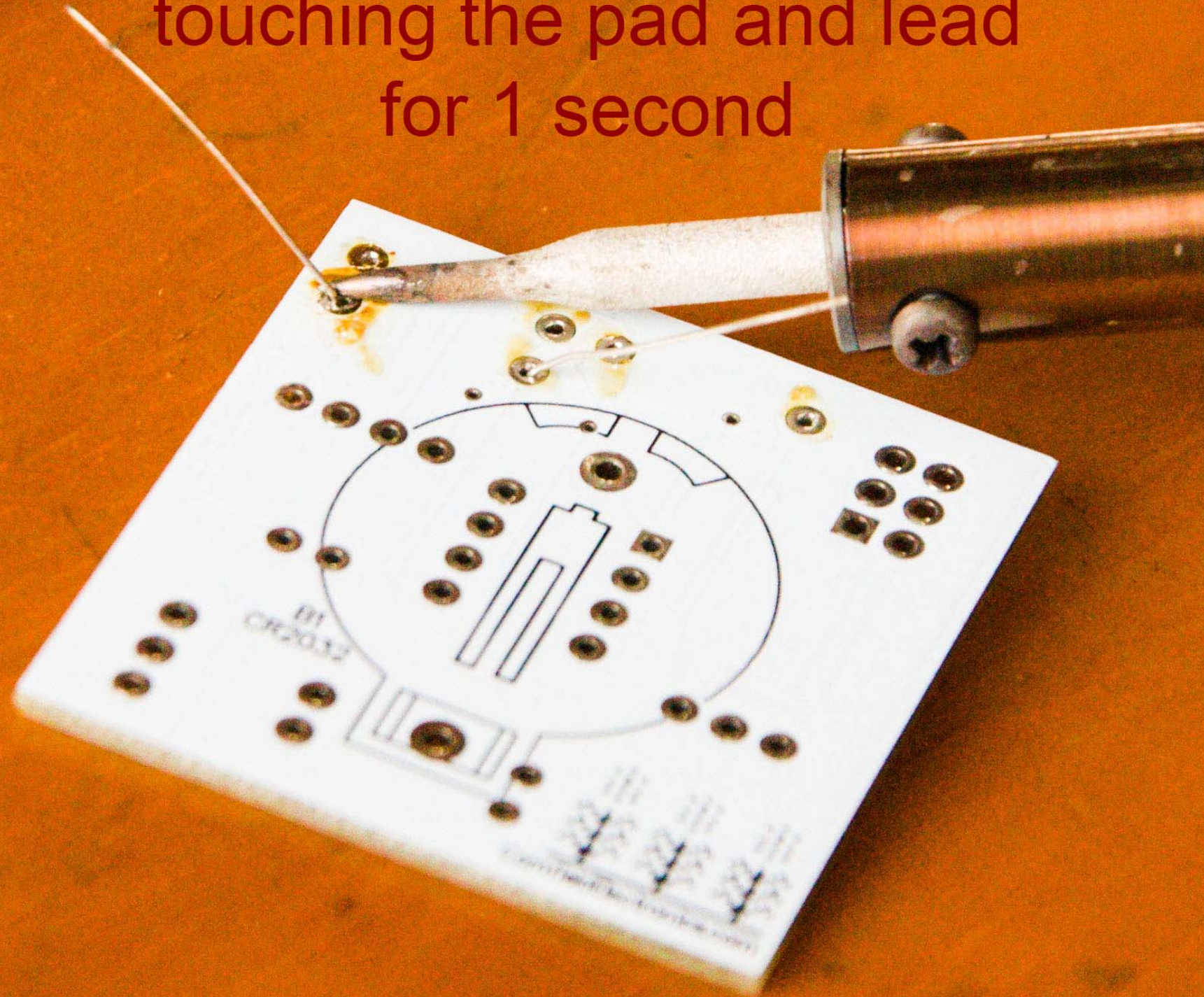
Swipe, Rotate, Swipe (on the sponge):

Keep the tip shiny silver!

knock solder off the tip

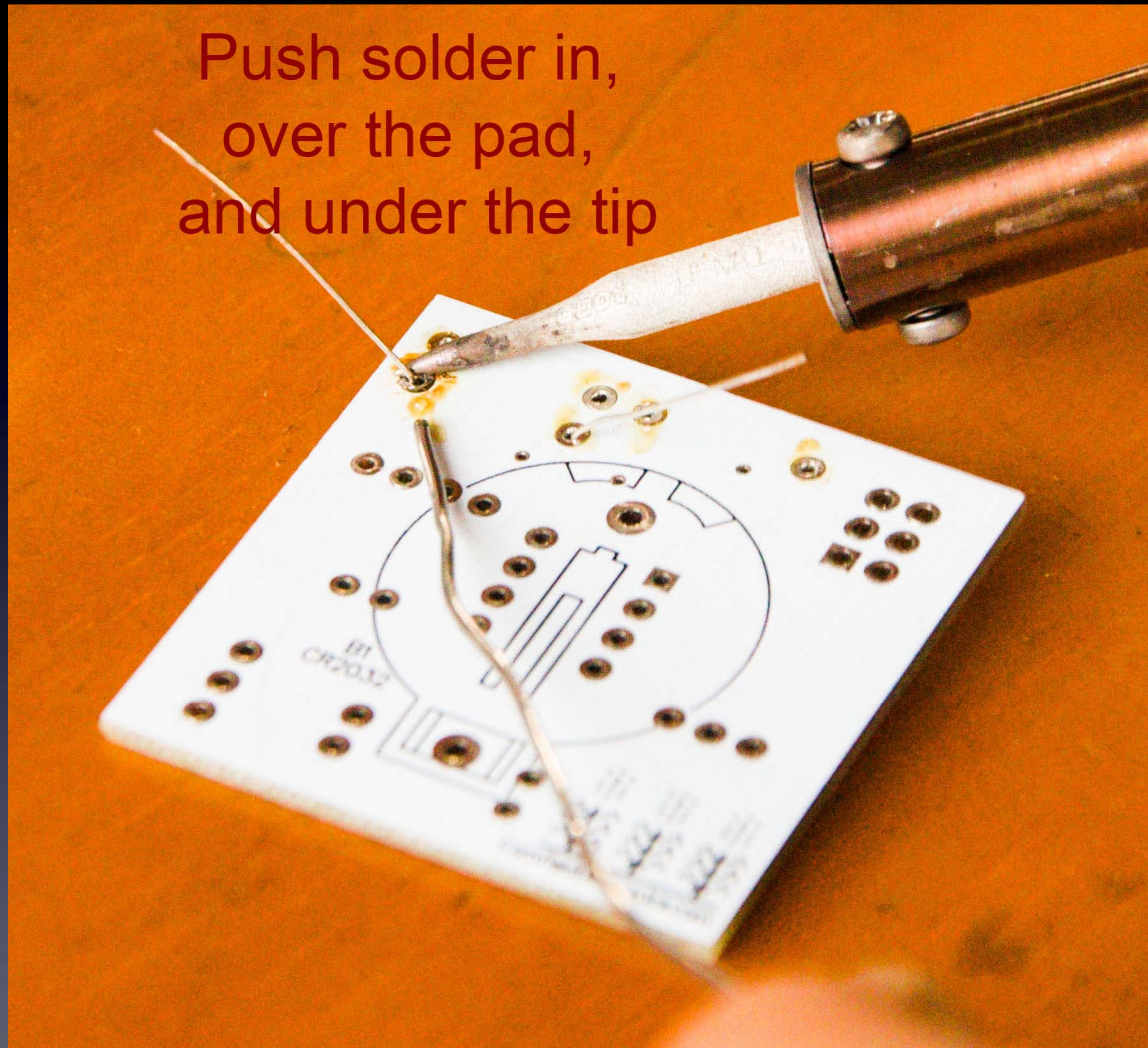


Lay clean tip across half of the pad,  
touching the pad and lead  
for 1 second





Do this quickly (slowly doesn't work well) – solder in & out in about 1 second

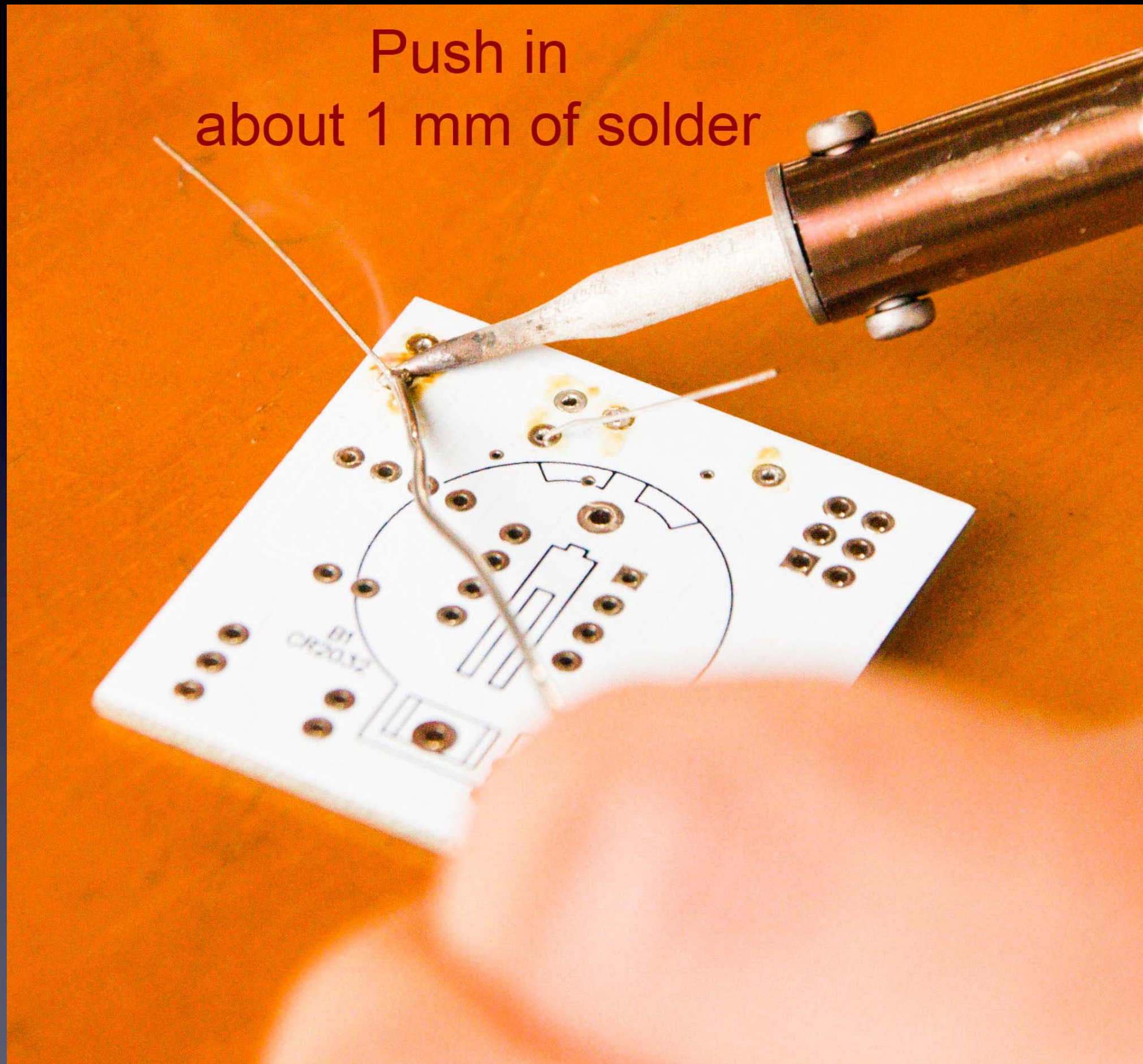


Push solder in,  
over the pad,  
and under the tip

Make sure solder melts on the underside of the soldering iron tip  
(not the side or top of the soldering iron tip)!

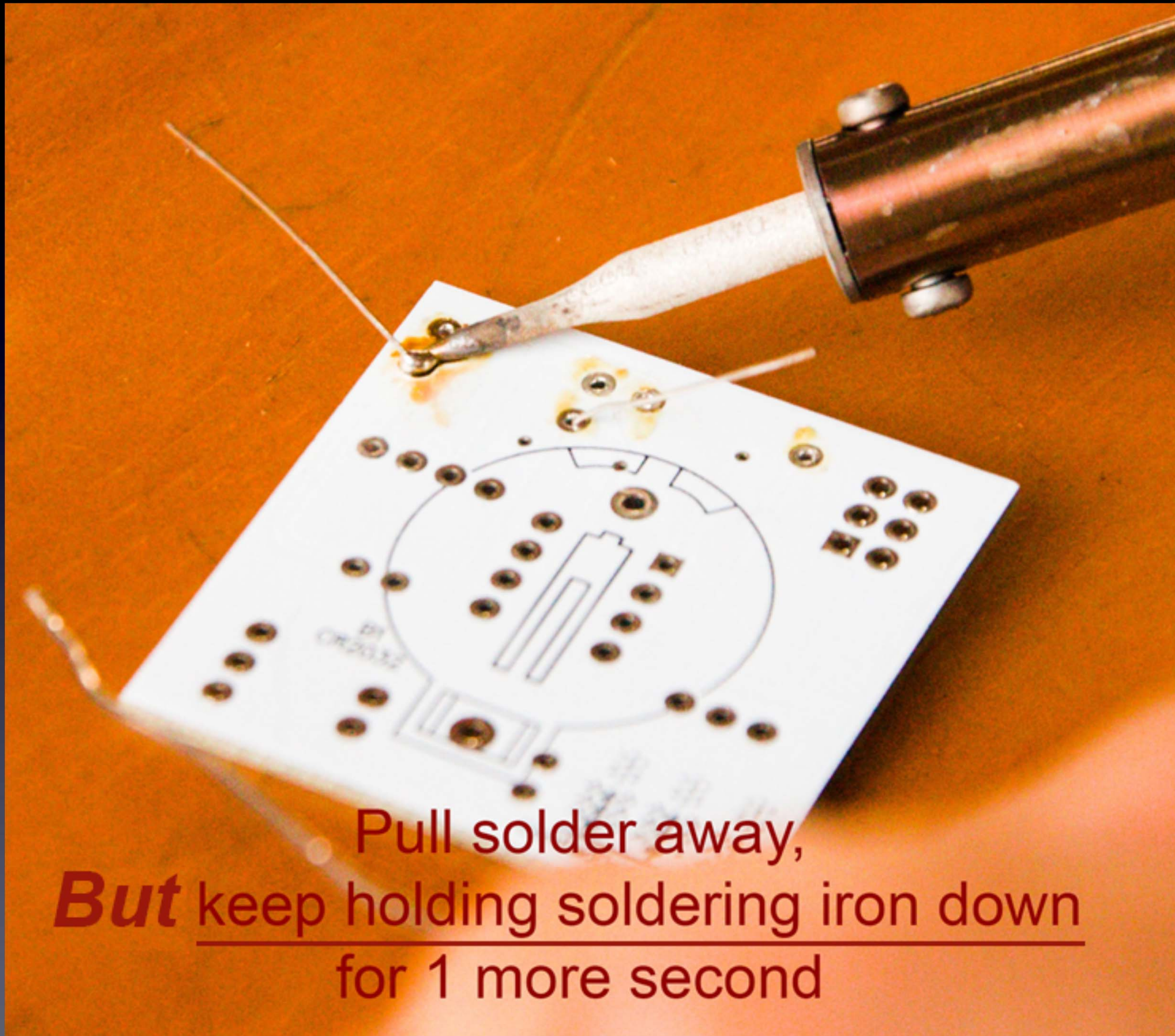


Do this quickly (slowly doesn't work well) – solder in & out in about 1 second



Make sure solder melts on the underside of the soldering iron tip  
(not the side or top of the soldering iron tip)!





Pull solder away,  
***But*** keep holding soldering iron down  
for 1 more second



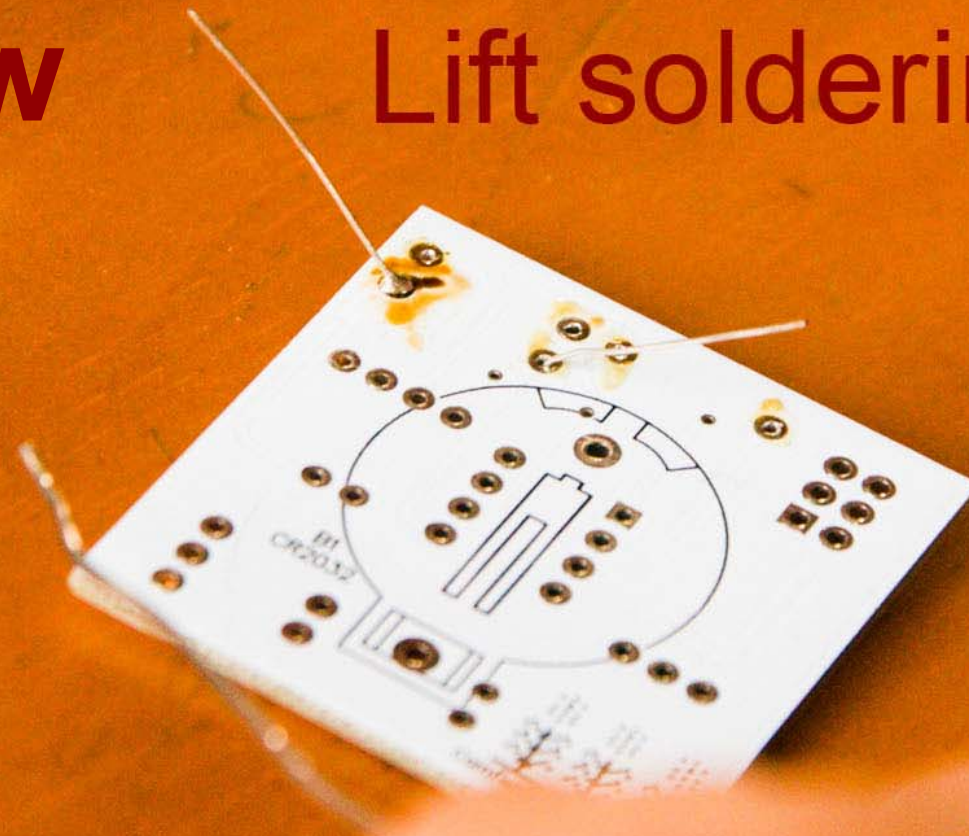
Secret #2:

Keep hot tip down  
1 second  
for solder to flow !!

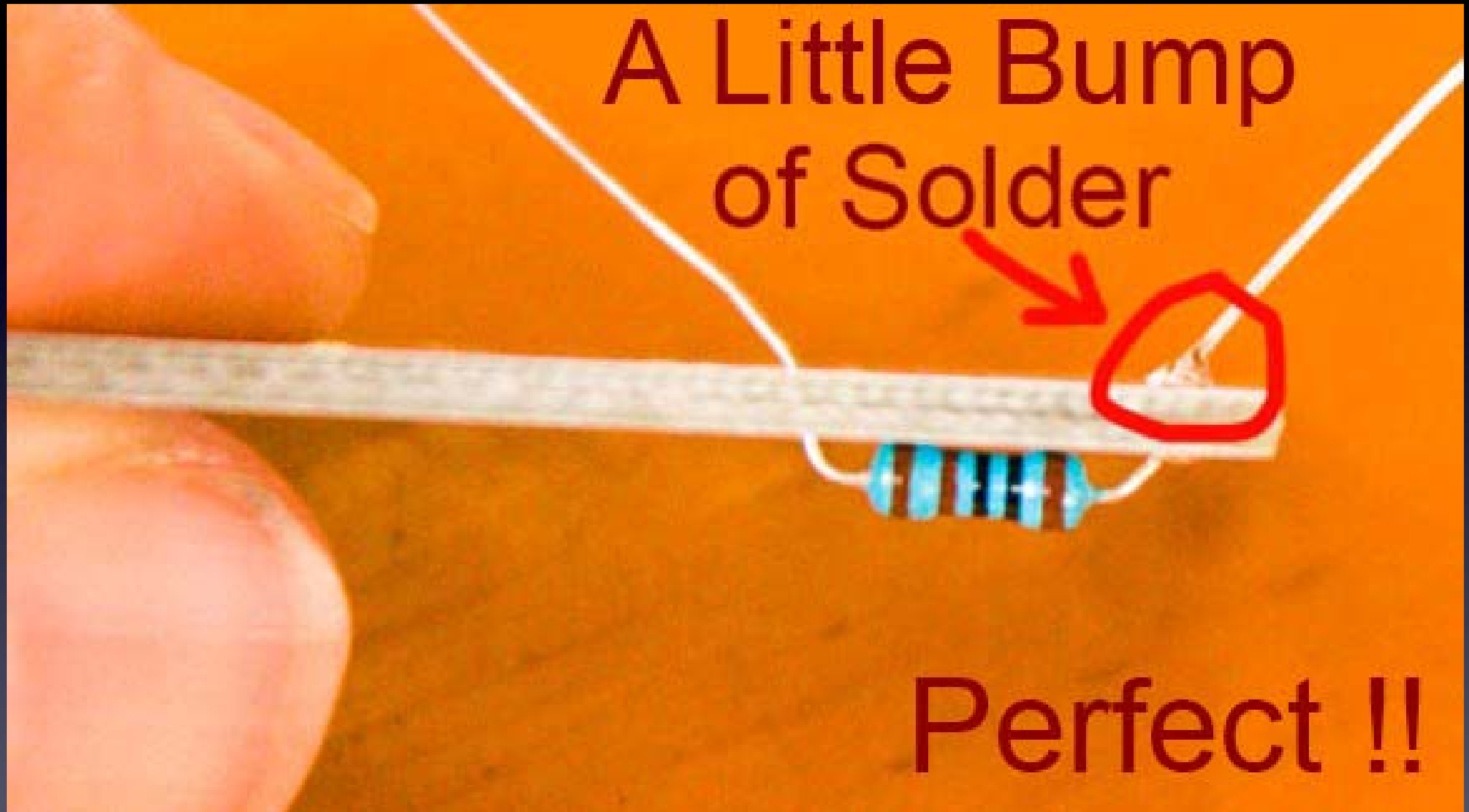


**Now**

**Lift soldering iron**







If you can see any of the pad, or the hole, you need more solder  
– so, just do all the steps again to make it perfect.

# The Rhythm !

and speed (about 1 second per step)





The Rhythm !

and speed (about 1 second per step)

Clean the tip



**The Rhythm !**  
and speed (about 1 second per step)



Tip **Down**



**The Rhythm !**  
and speed (about 1 second per step)



Solder **In**

**The Rhythm !**  
and speed (about 1 second per step)



Solder **Out**



**The Rhythm !**  
and speed (about 1 second per step)



***WAIT !***

**The Rhythm !**  
and speed (about 1 second per step)



**Lift** Tip



# The Rhythm !

and speed (about 1 second per step)



The Rhythm !

and speed (about 1 second per step)

Clean the tip





**The Rhythm !**  
and speed (about 1 second per step)



Tip **Down**

**The Rhythm !**  
and speed (about 1 second per step)



Solder **In**



**The Rhythm !**  
and speed (about 1 second per step)



Solder **Out**

**The Rhythm !**  
and speed (about 1 second per step)



***WAIT !***

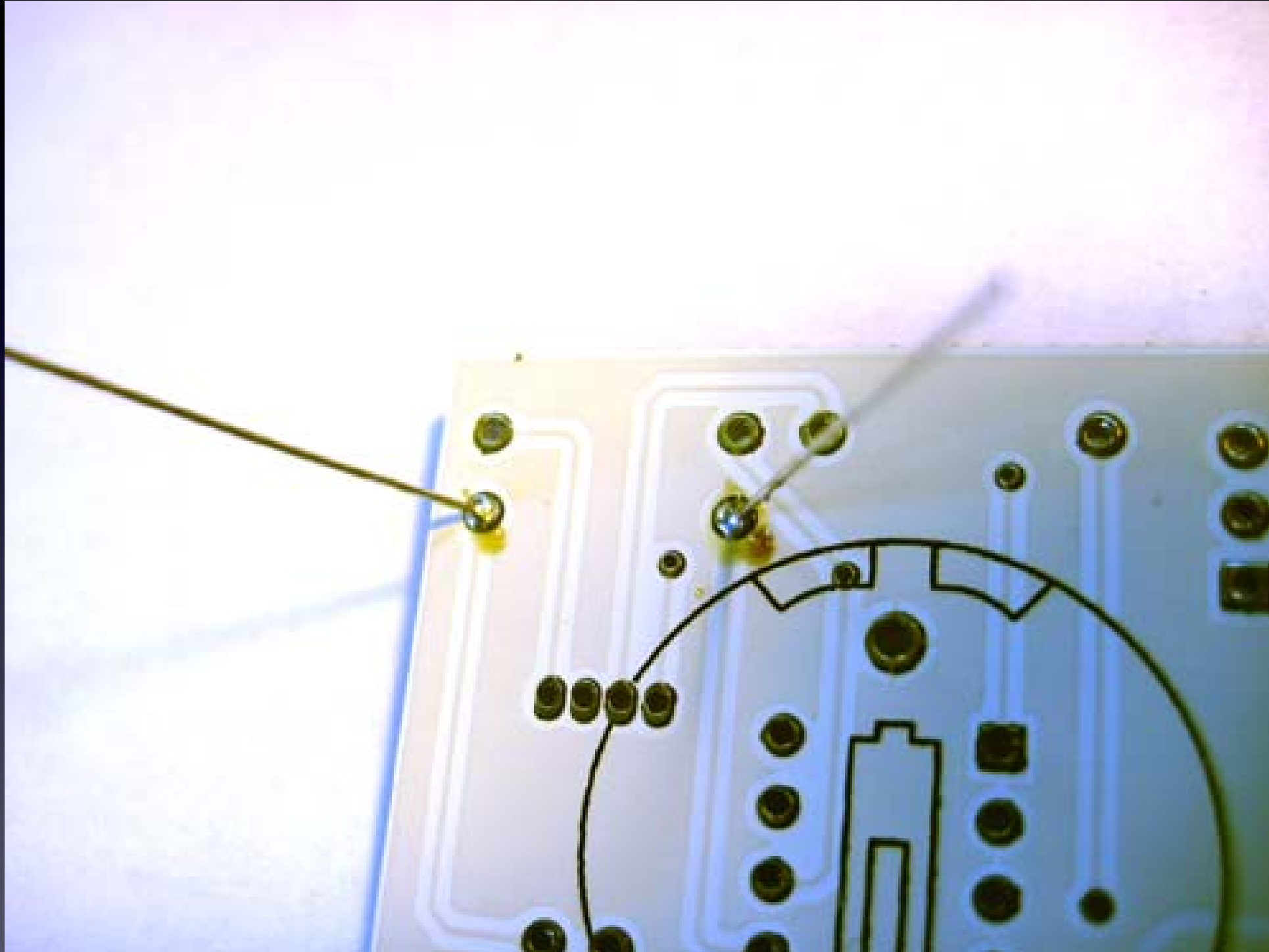


**The Rhythm !**  
and speed (about 1 second per step)



**Lift** Tip

# Solder all of the leads of the part to the board



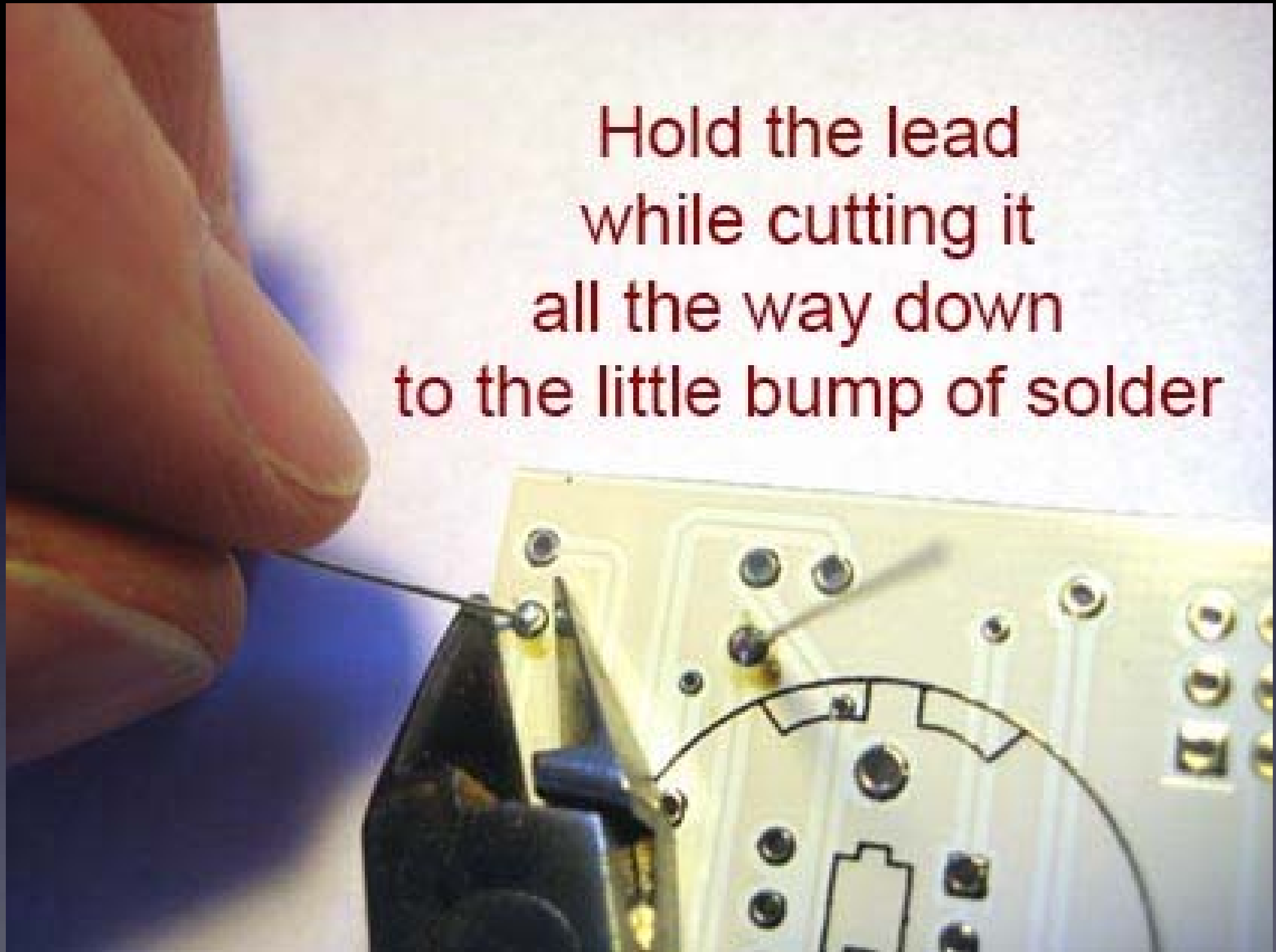
For this part, there are two leads

Here you can see two good solder connections



# Now cut the leads short

Hold the lead  
while cutting it  
all the way down  
to the little bump of solder



Cutting with the tip of the wire cutter gives you more control

Safety Tip #3:

Hold or cover the lead !

(or it will fly into your eye!)

*(They like doing that – so please hold or cover the lead when you cut.)*





All done !

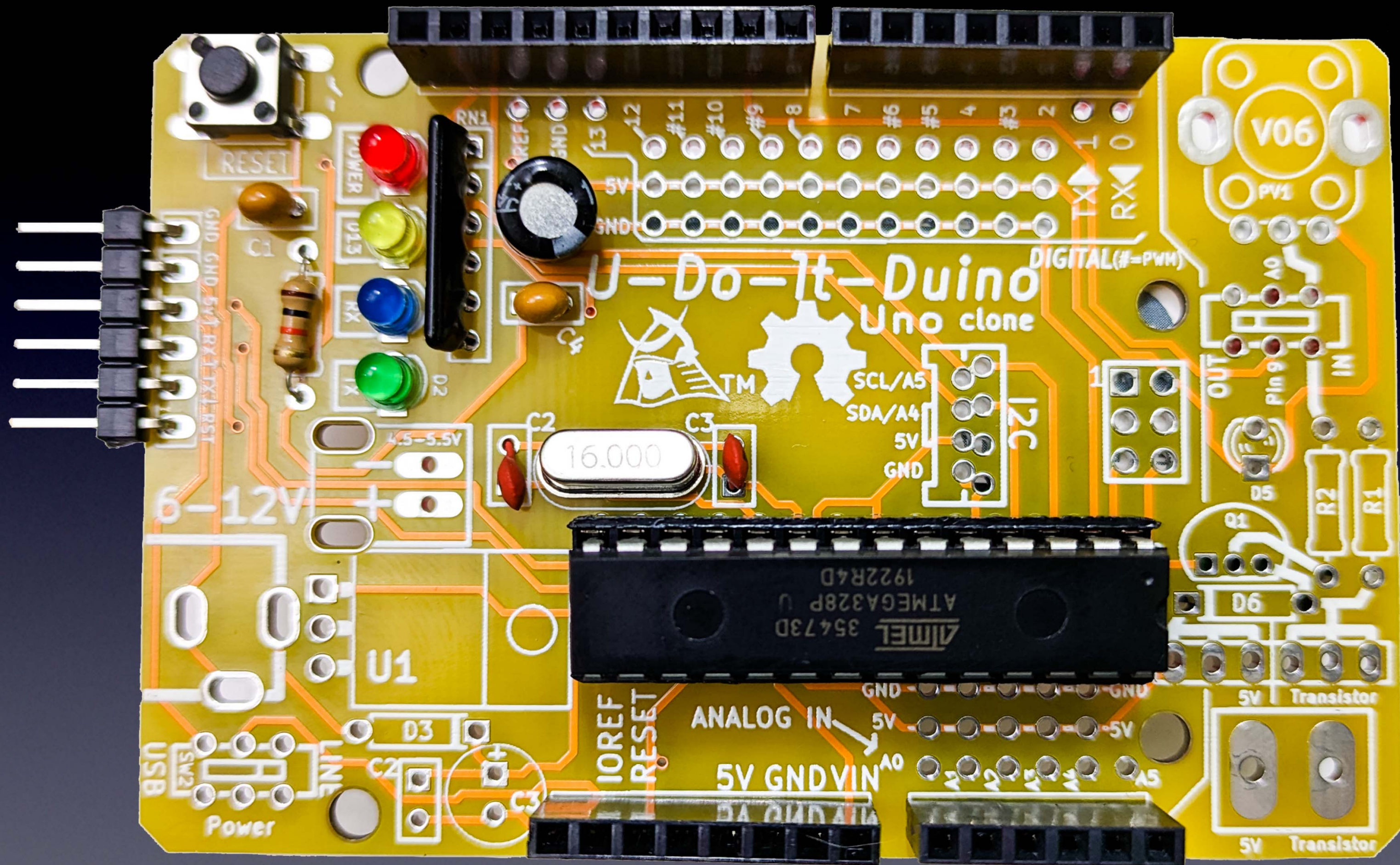
No wires sticking out

One part at a time



Till all the parts are soldered





And it will look like this when you're done.



Then test with battery pack,

Turn it on,

And it works!

(Or you start debugging.)

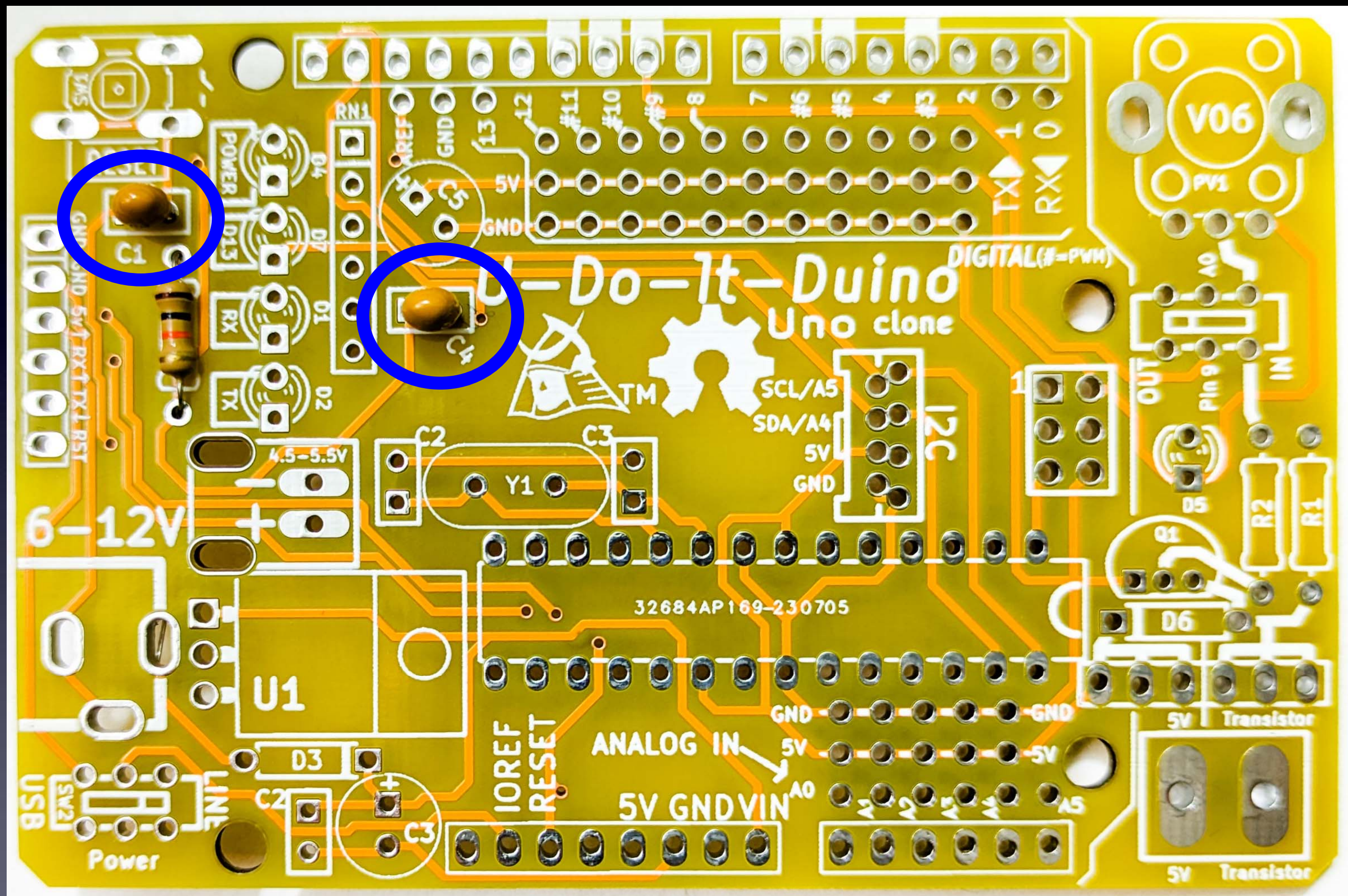
Let's start!





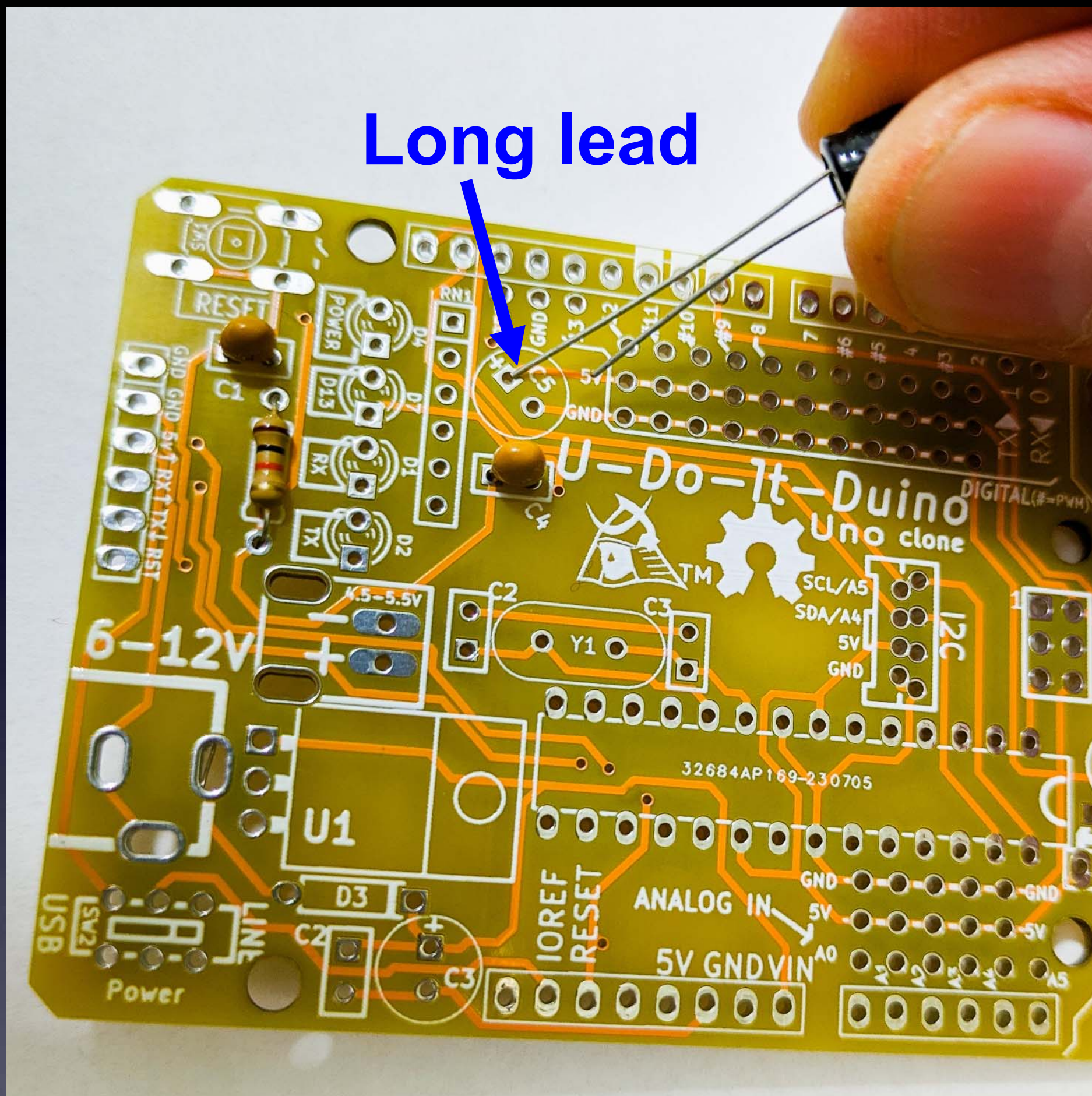
# R3





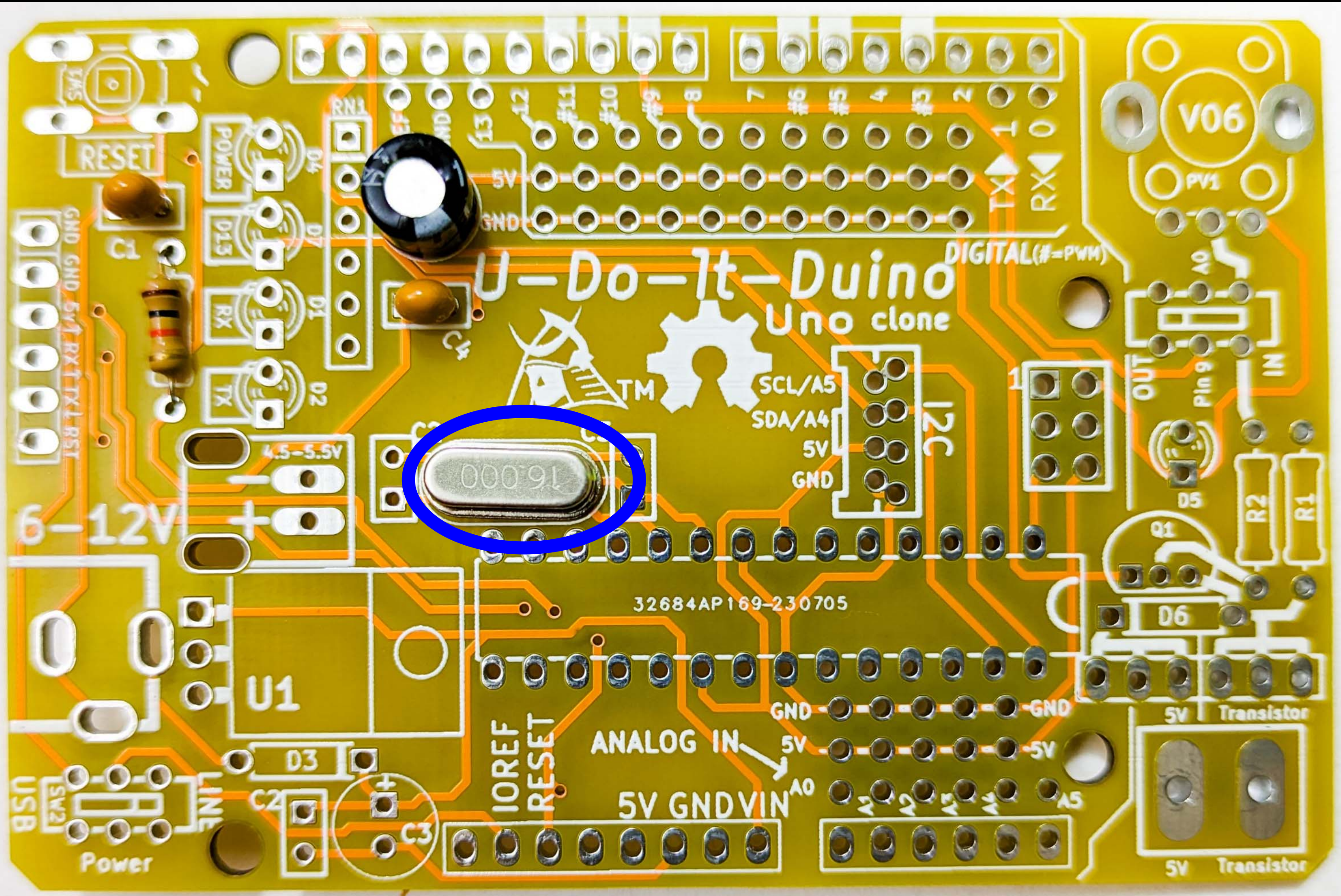
C1 and C4





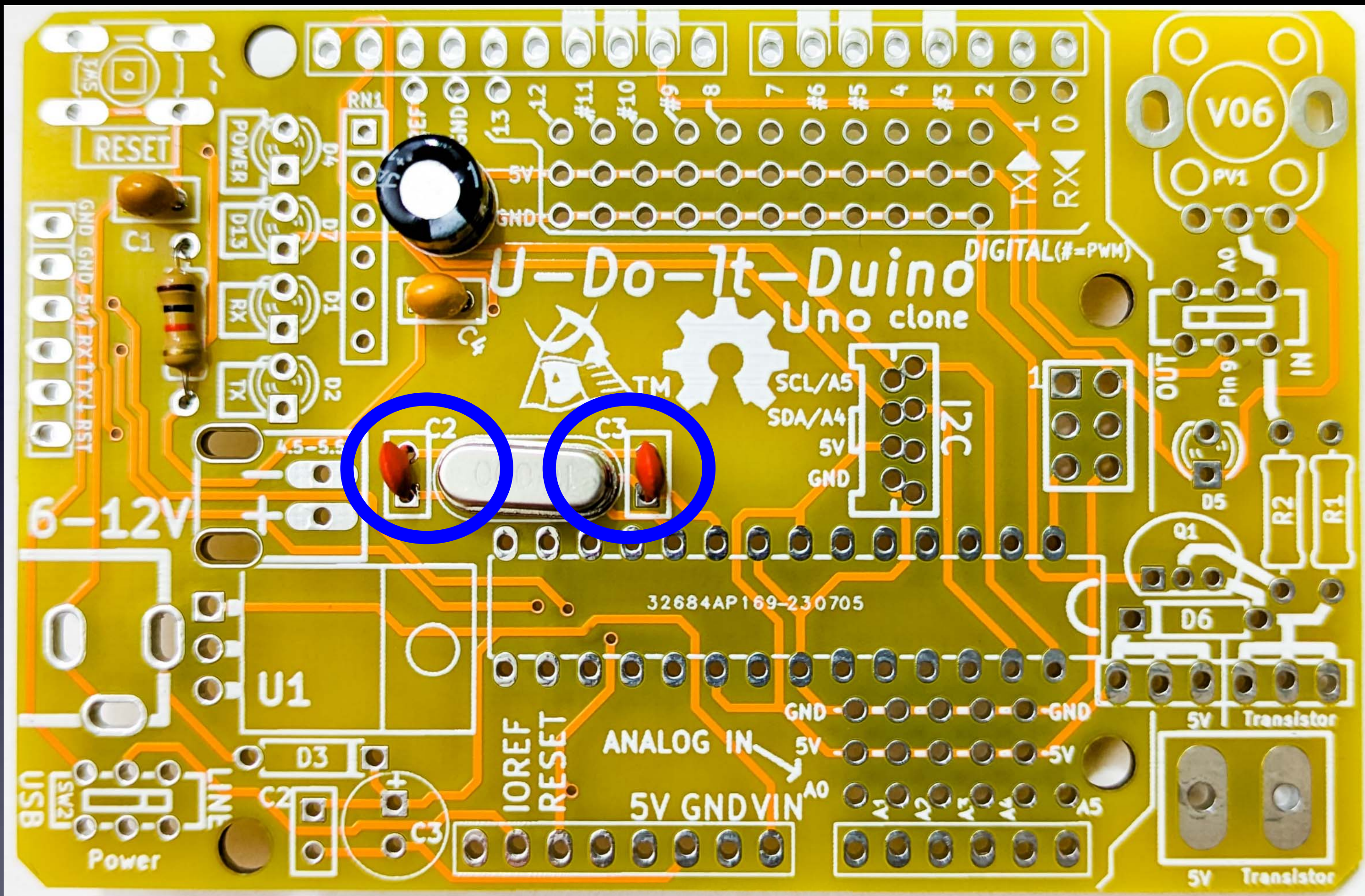
**C5: Long lead “+” (square pad)**





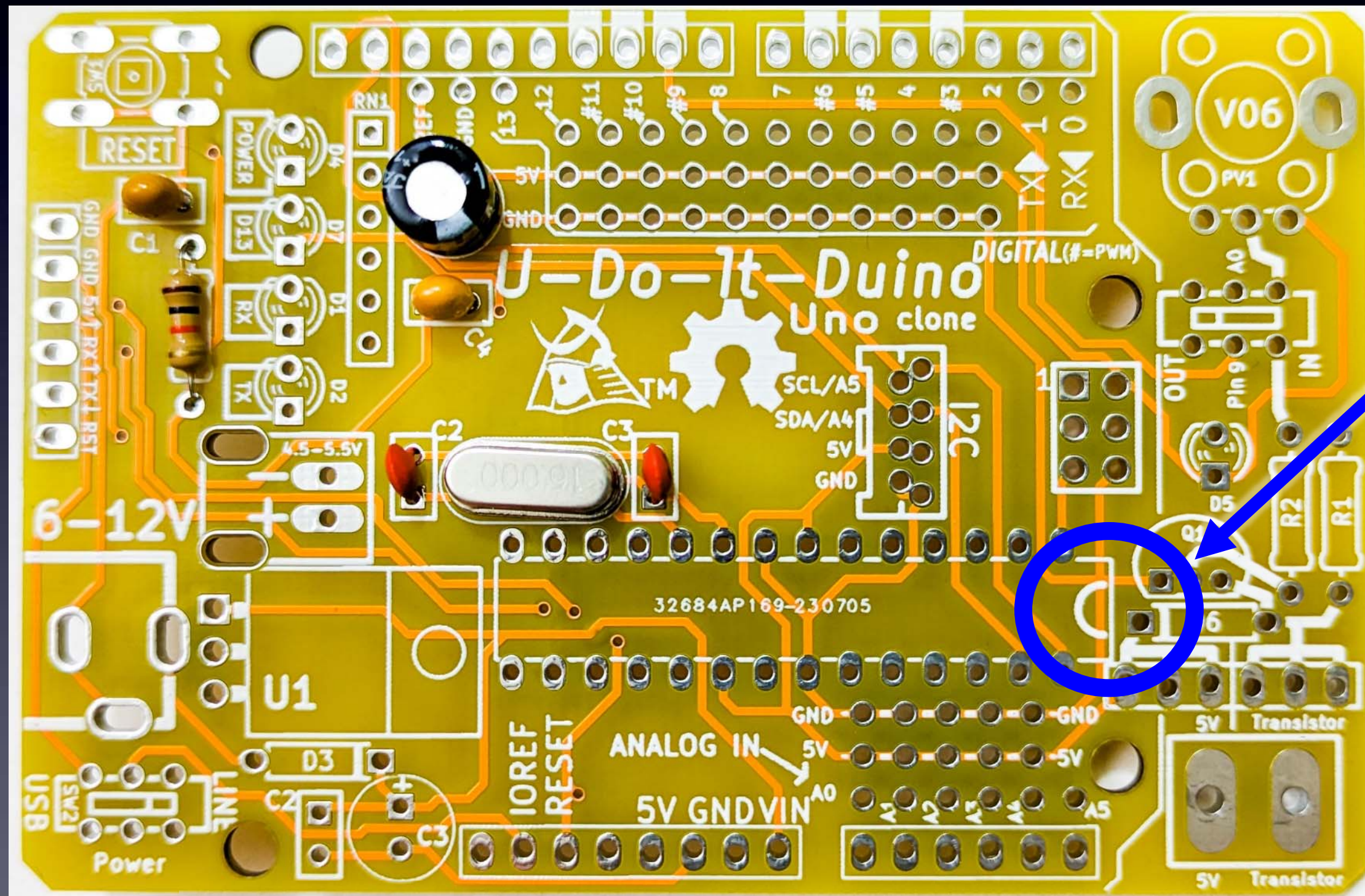
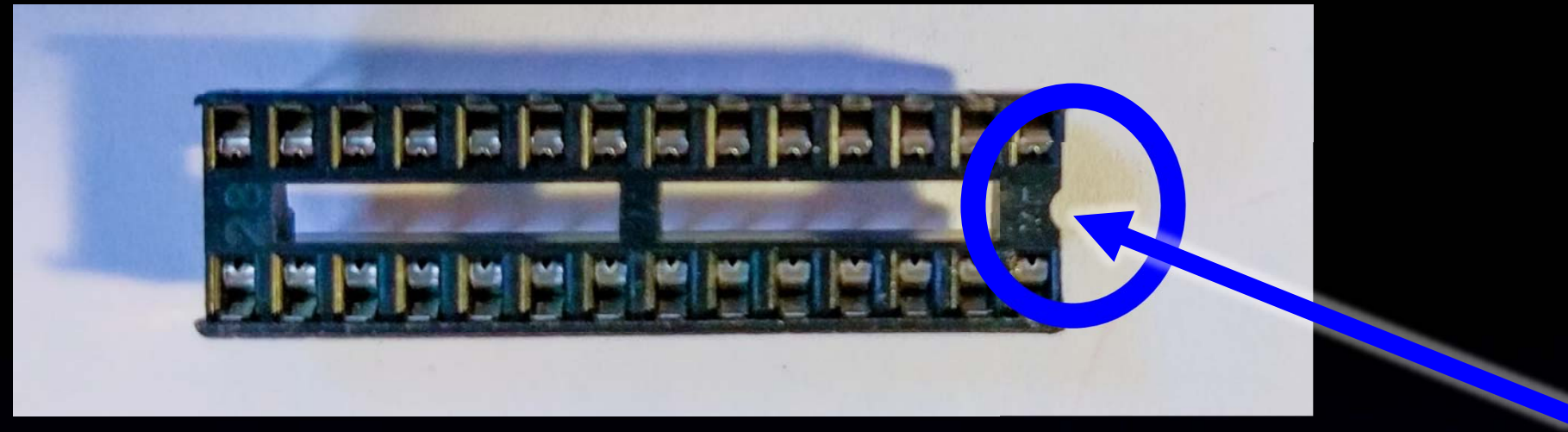
Y1





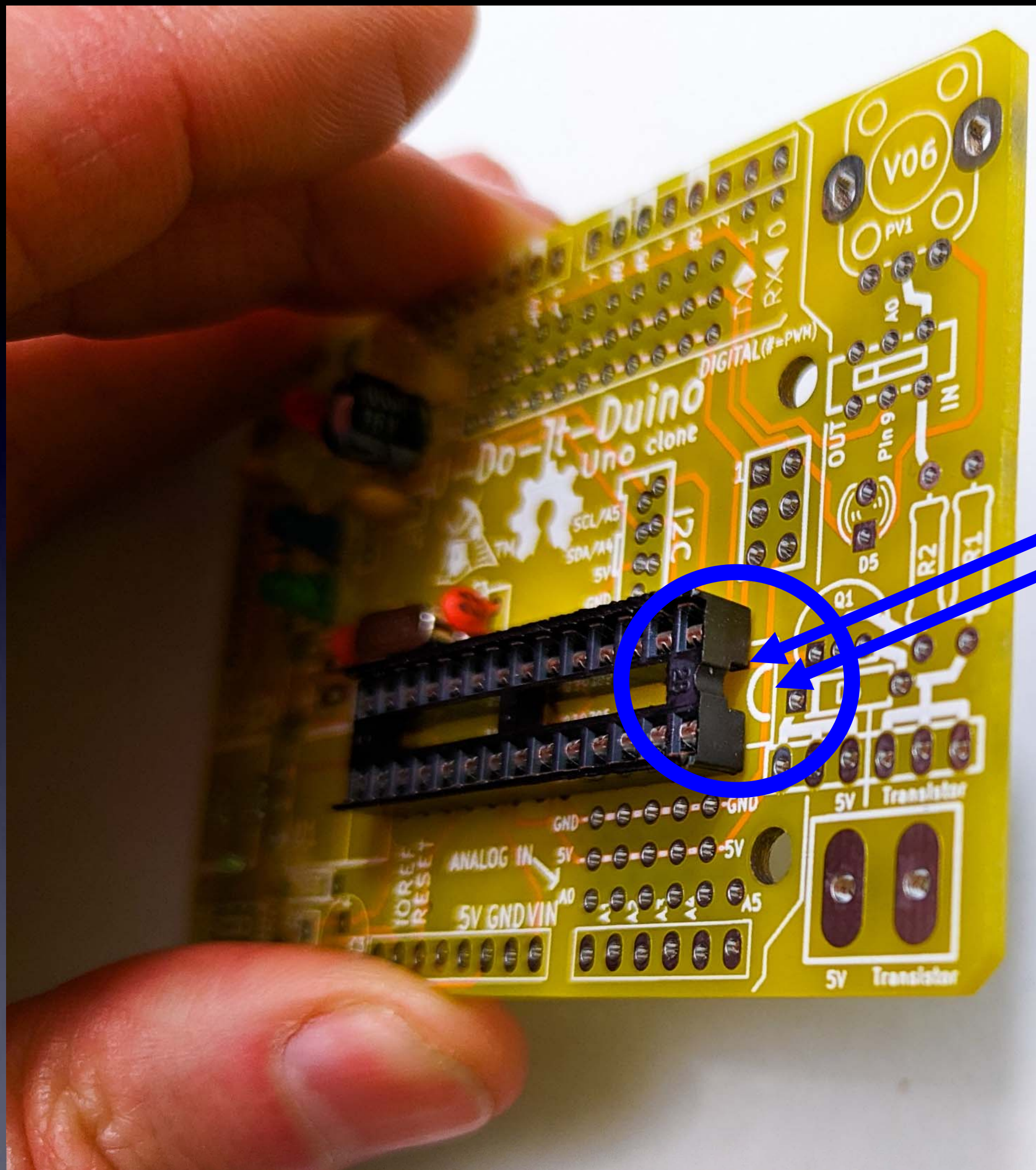
C2 and C3





**Orientation of Microcontroller Socket**





**Orientation of Microcontroller Socket**

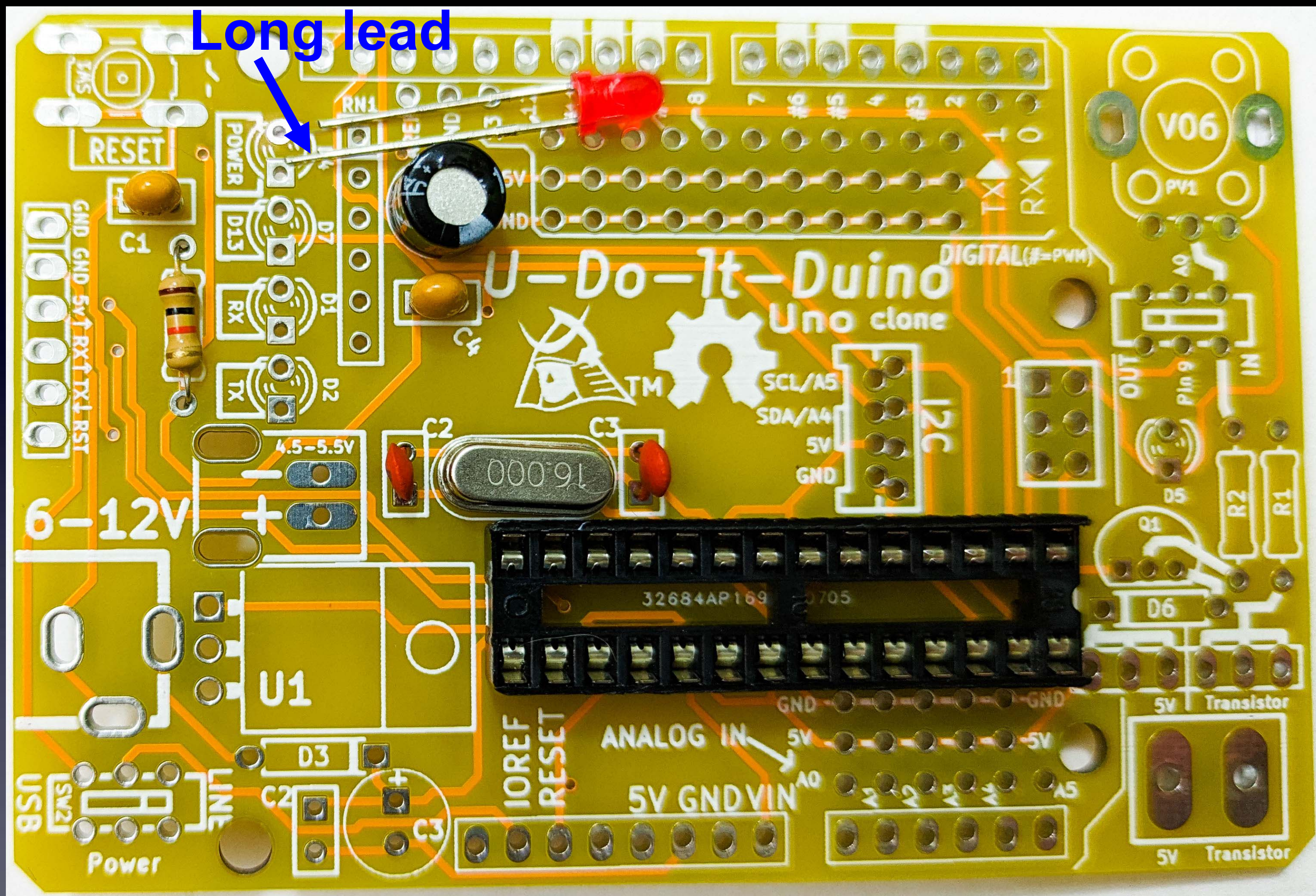


# Bend pins down on two opposite corners



- Solder all 28 pins.
- Only need to clean the tip after it gets dirty.
- No need to cut the pins short after soldering.





**LEDs: Long lead “+” (square pad)**

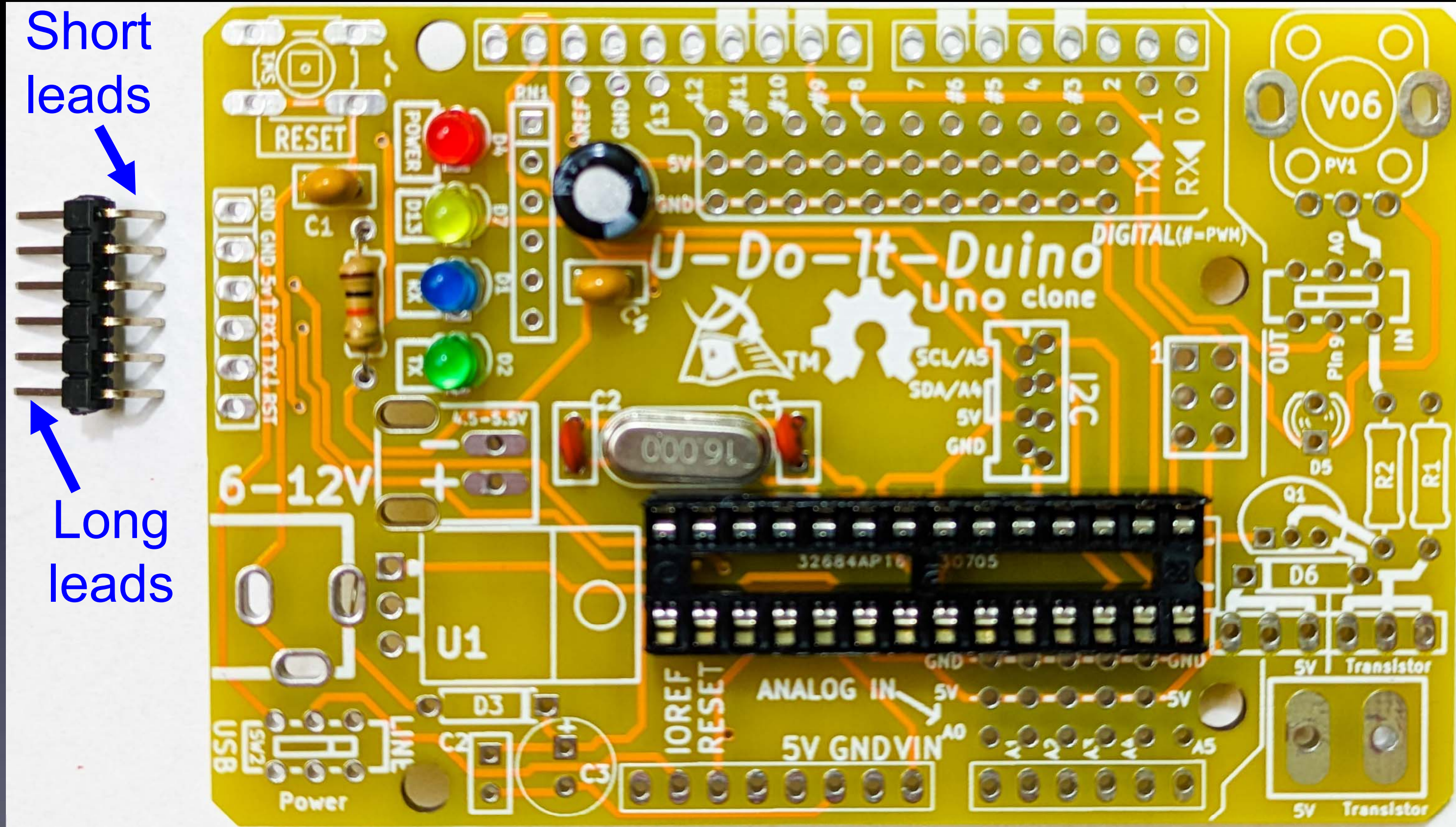




# LEDs: Long lead “+” (square pad)



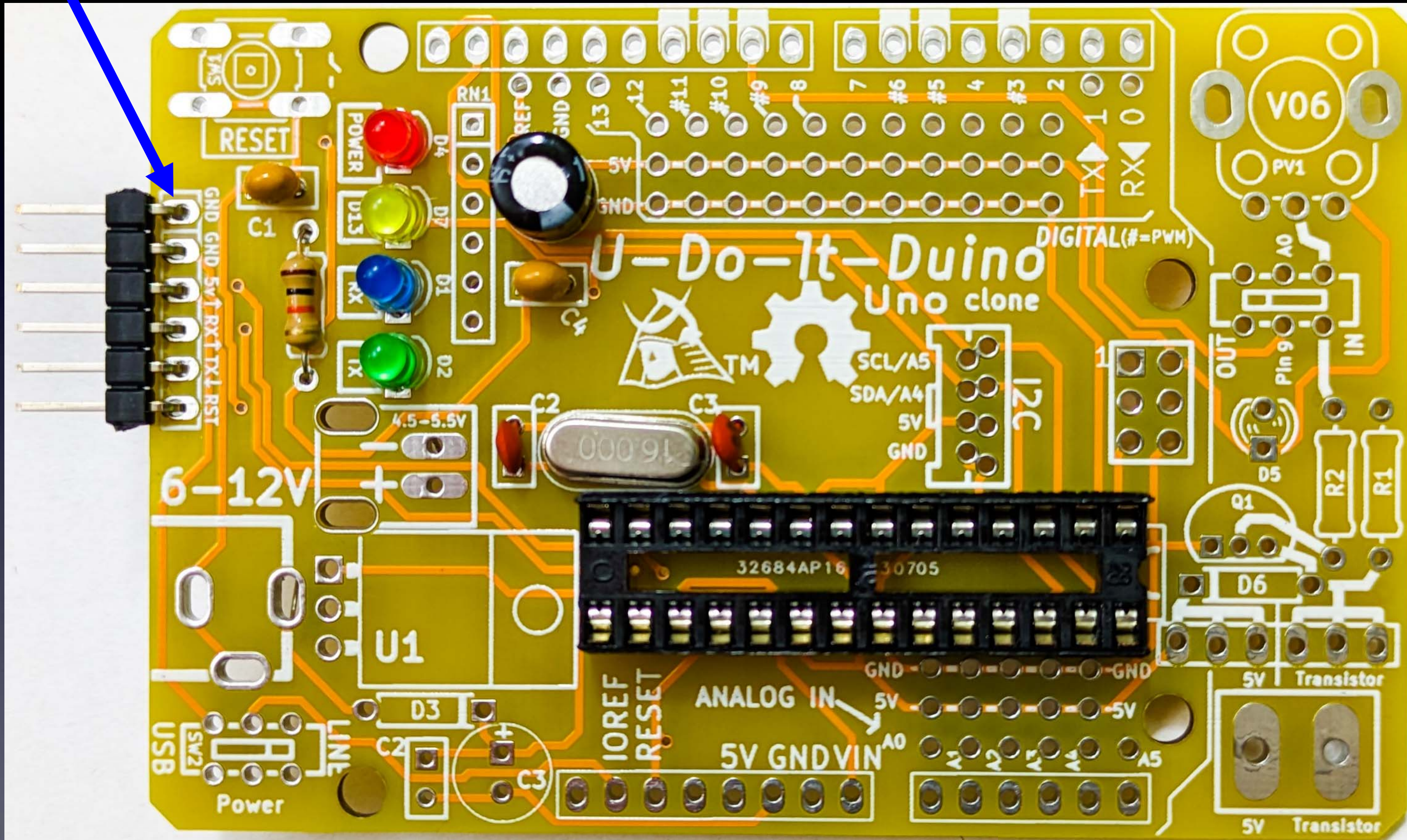
**Short leads of the Serial Port Connector go into the board**



**Serial Port Connector: Long leads point to the left**



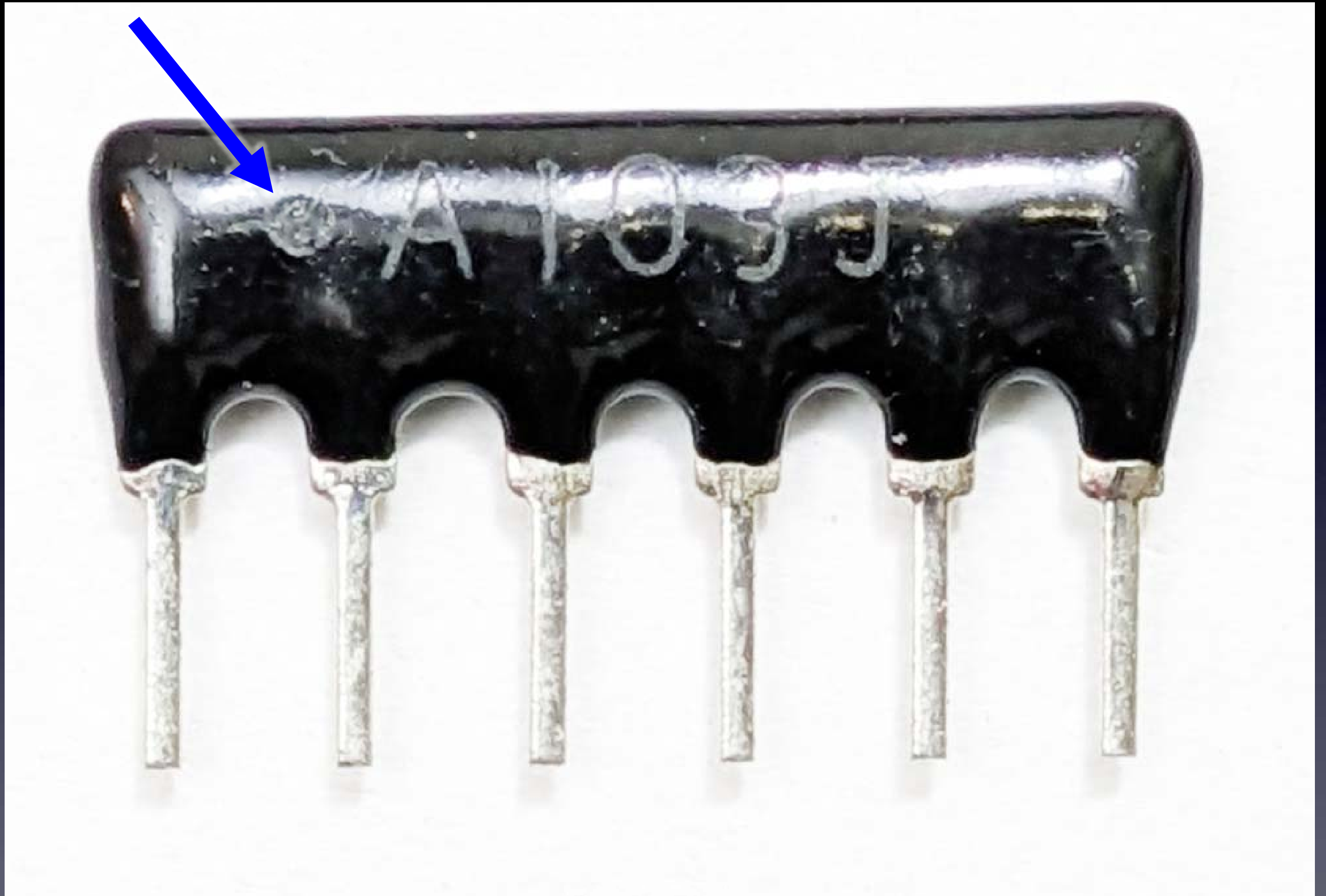
**Solder on top of board if it falls out upside down**



**Serial Port Connector**



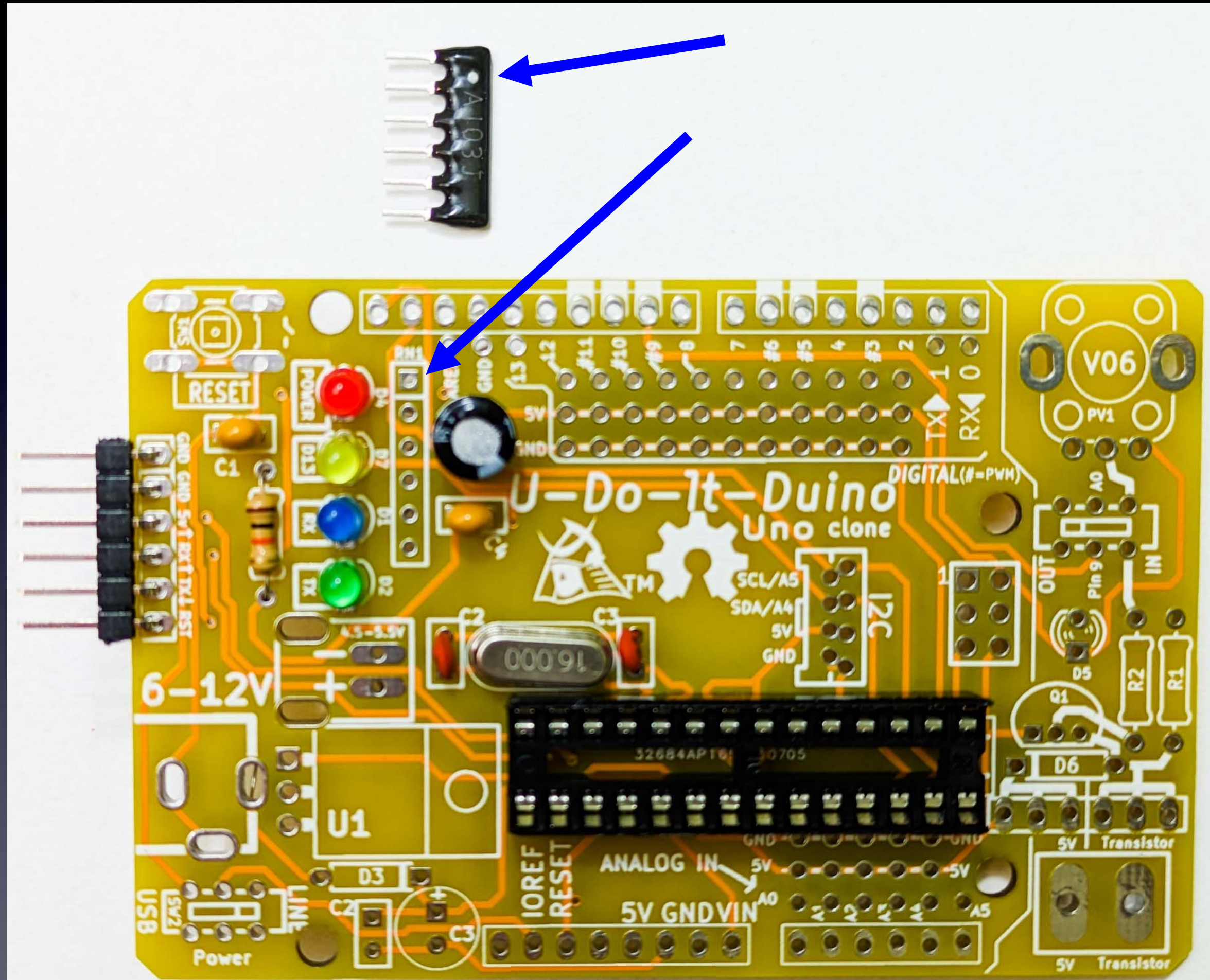
**The white circle (or diamond)**



**RN1**



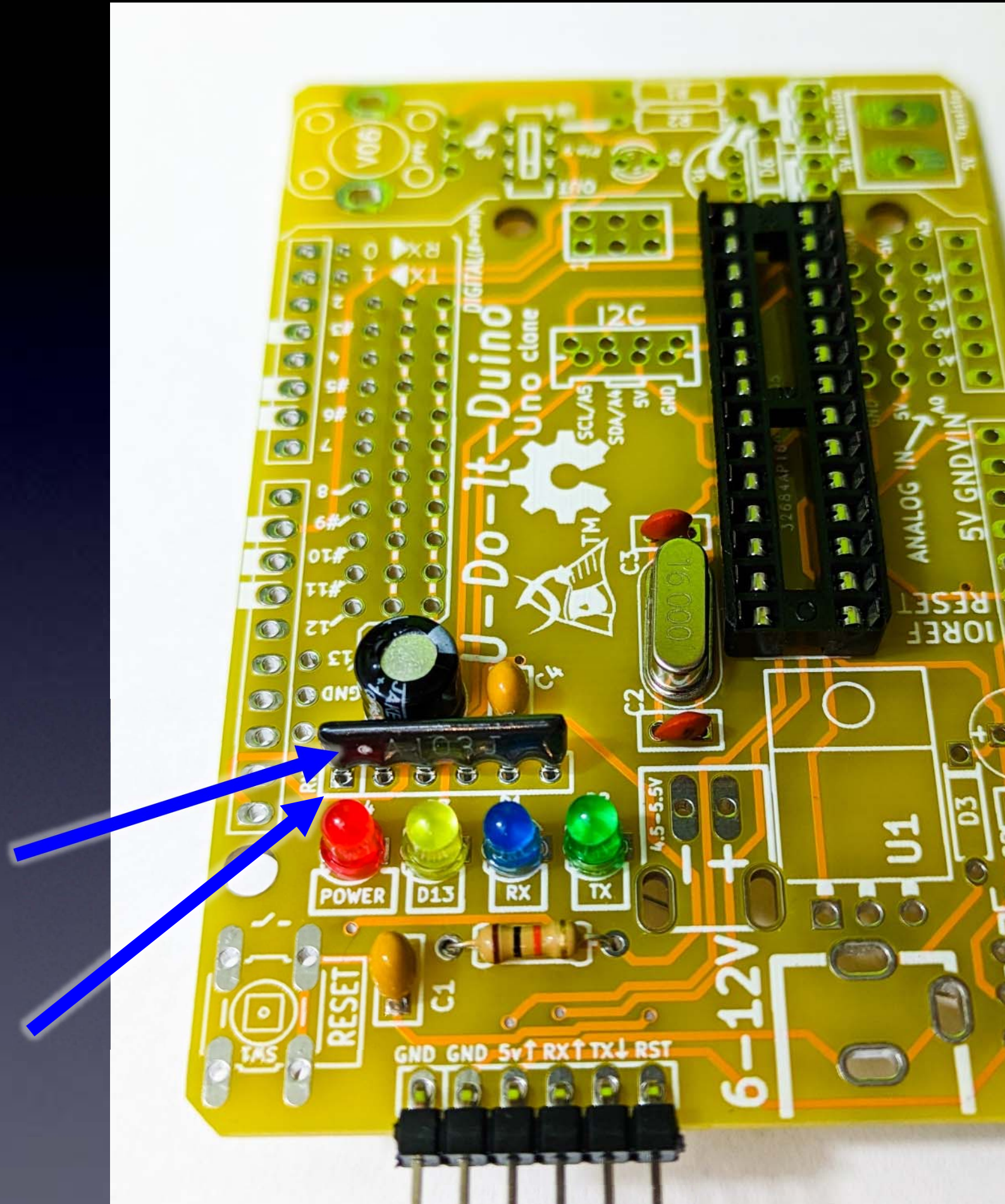
The white circle (or diamond) goes in the square pad



RN1



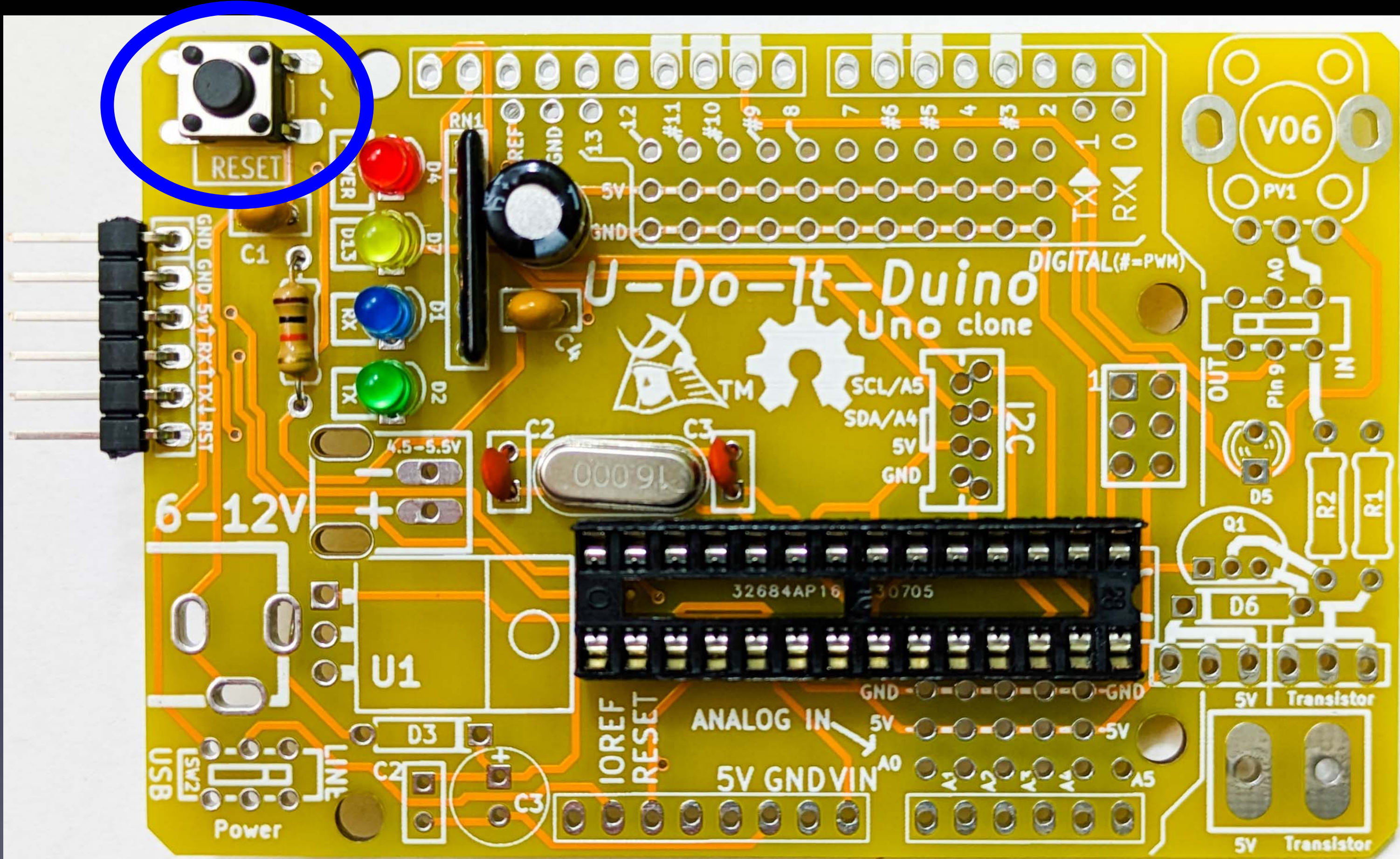
The white circle (or diamond) goes in the square pad



RN1

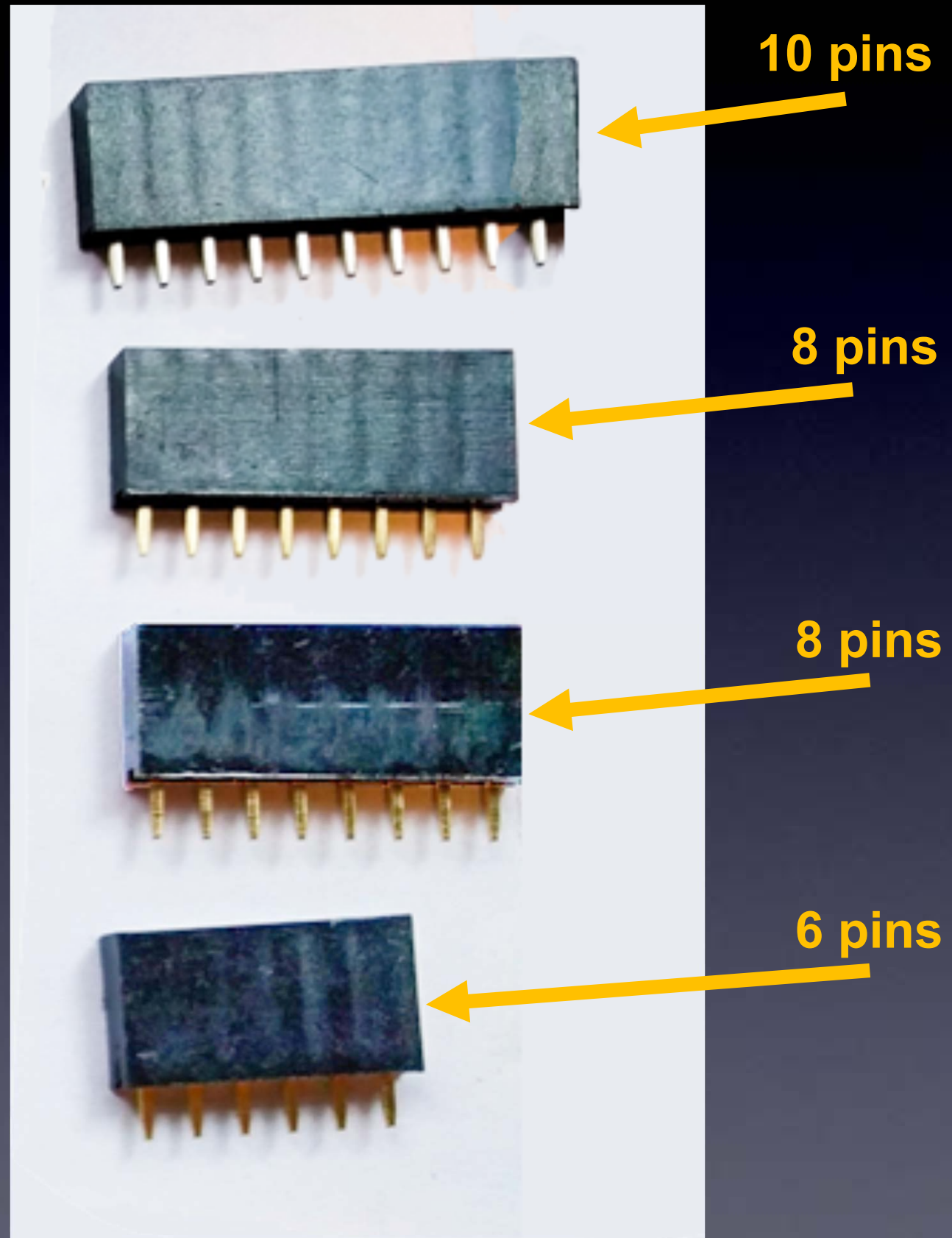


The Reset Switch fits in 2 ways – either way is fine



Reset Switch

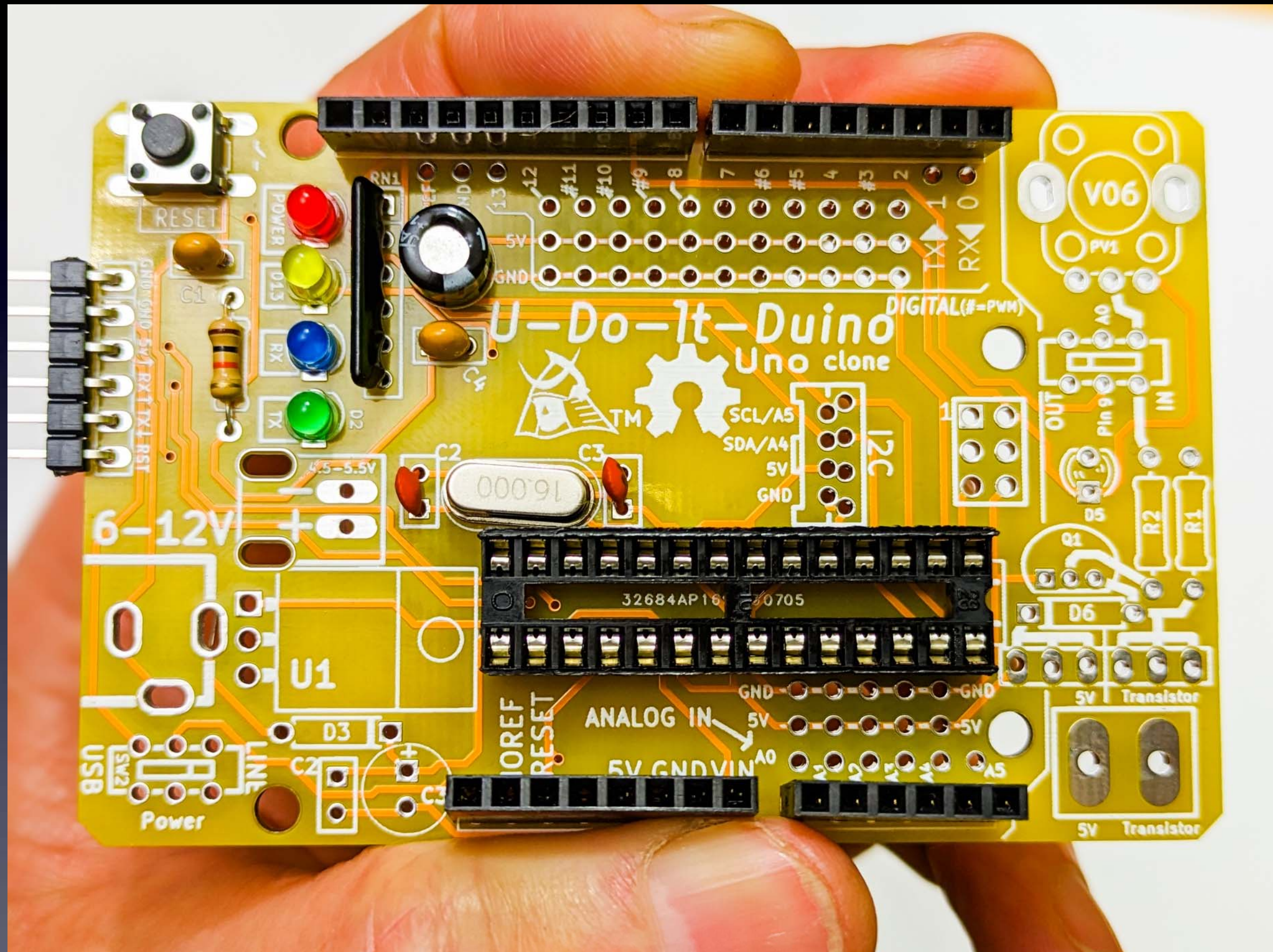




**Headers**



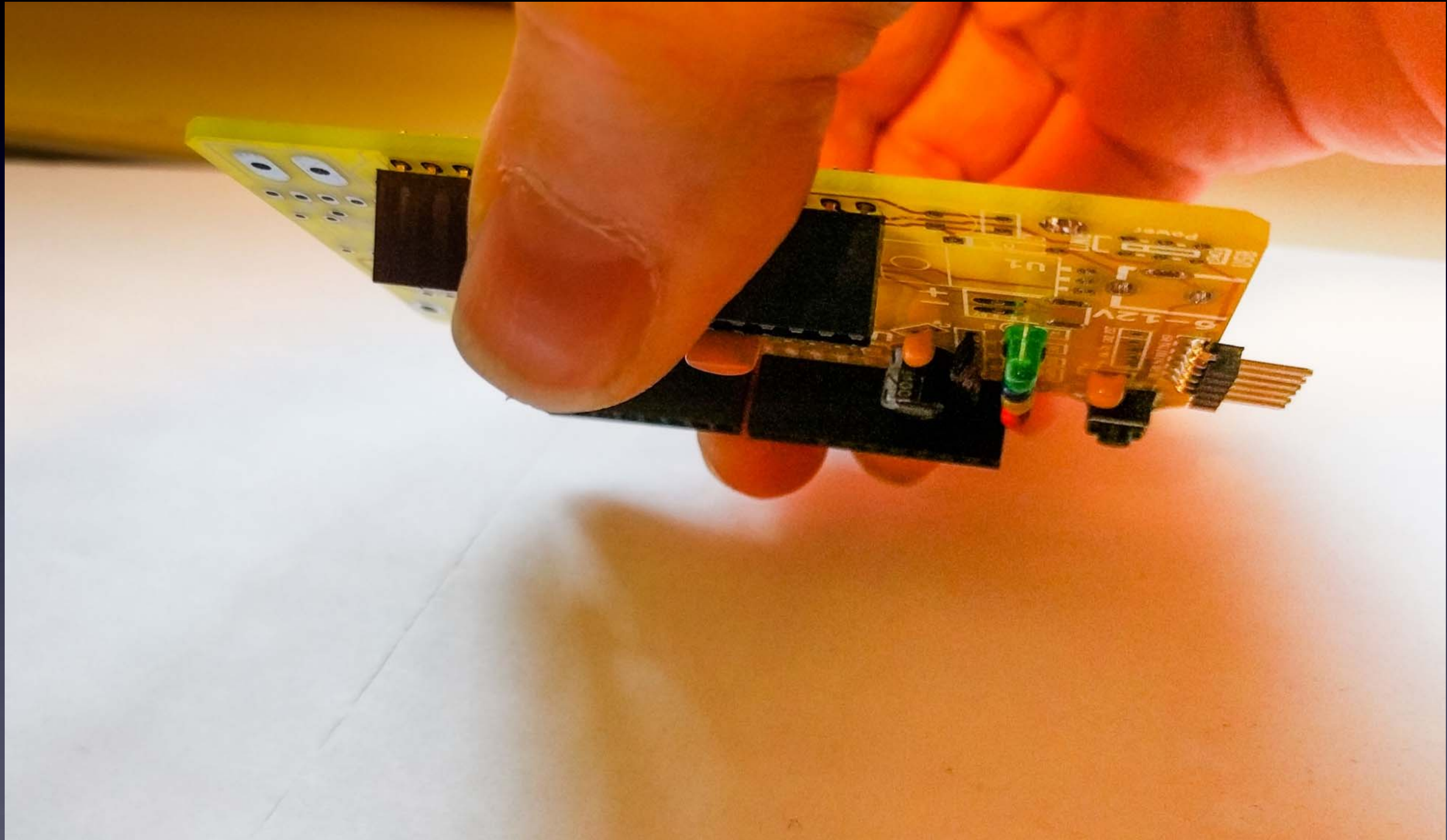
It's easiest to solder all Headers at once.  
Insert, and hold them like this:



**Headers**

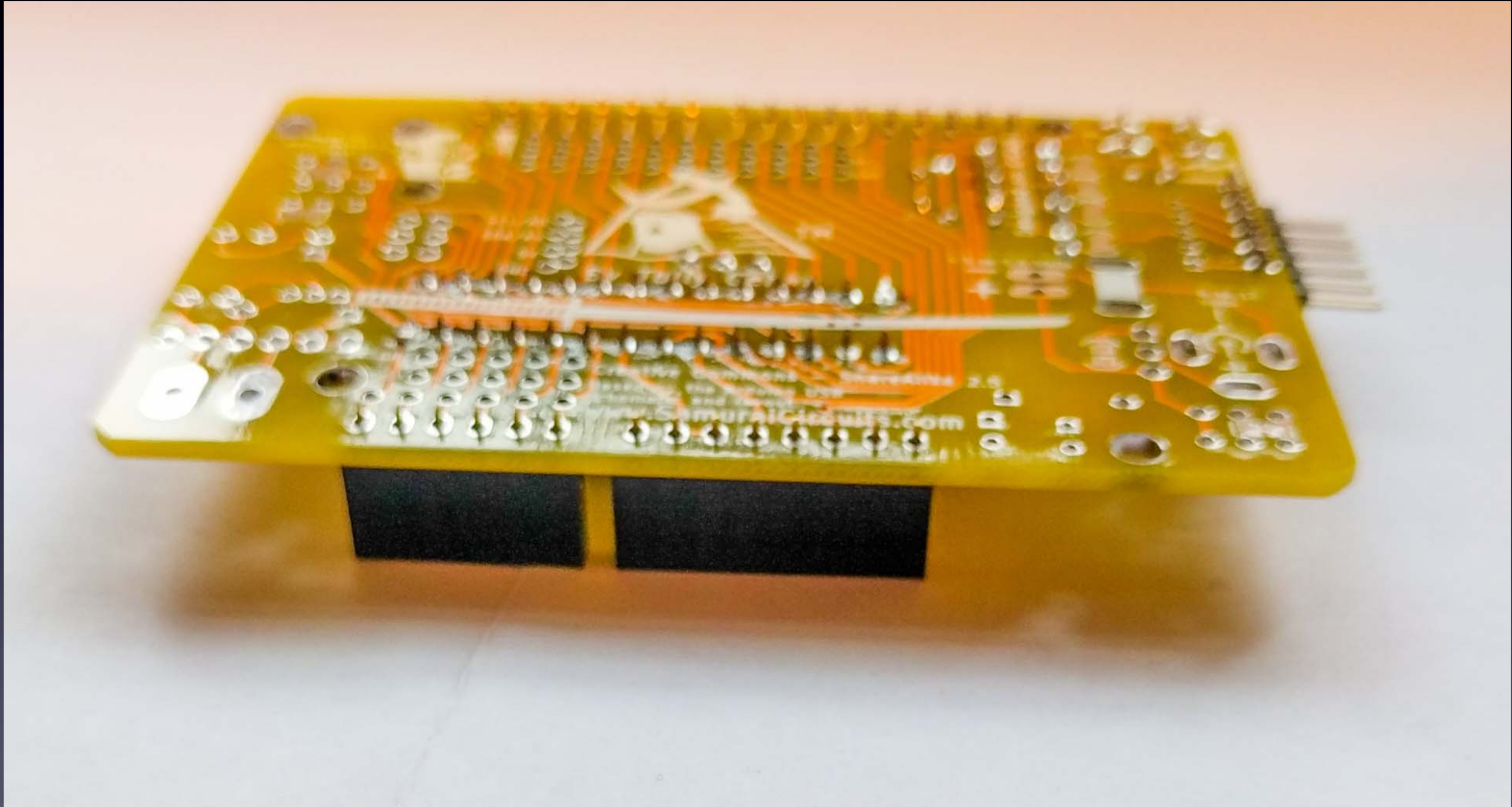


**It's easiest to solder all Headers at once.  
Lay it all on the table:**



**Headers**

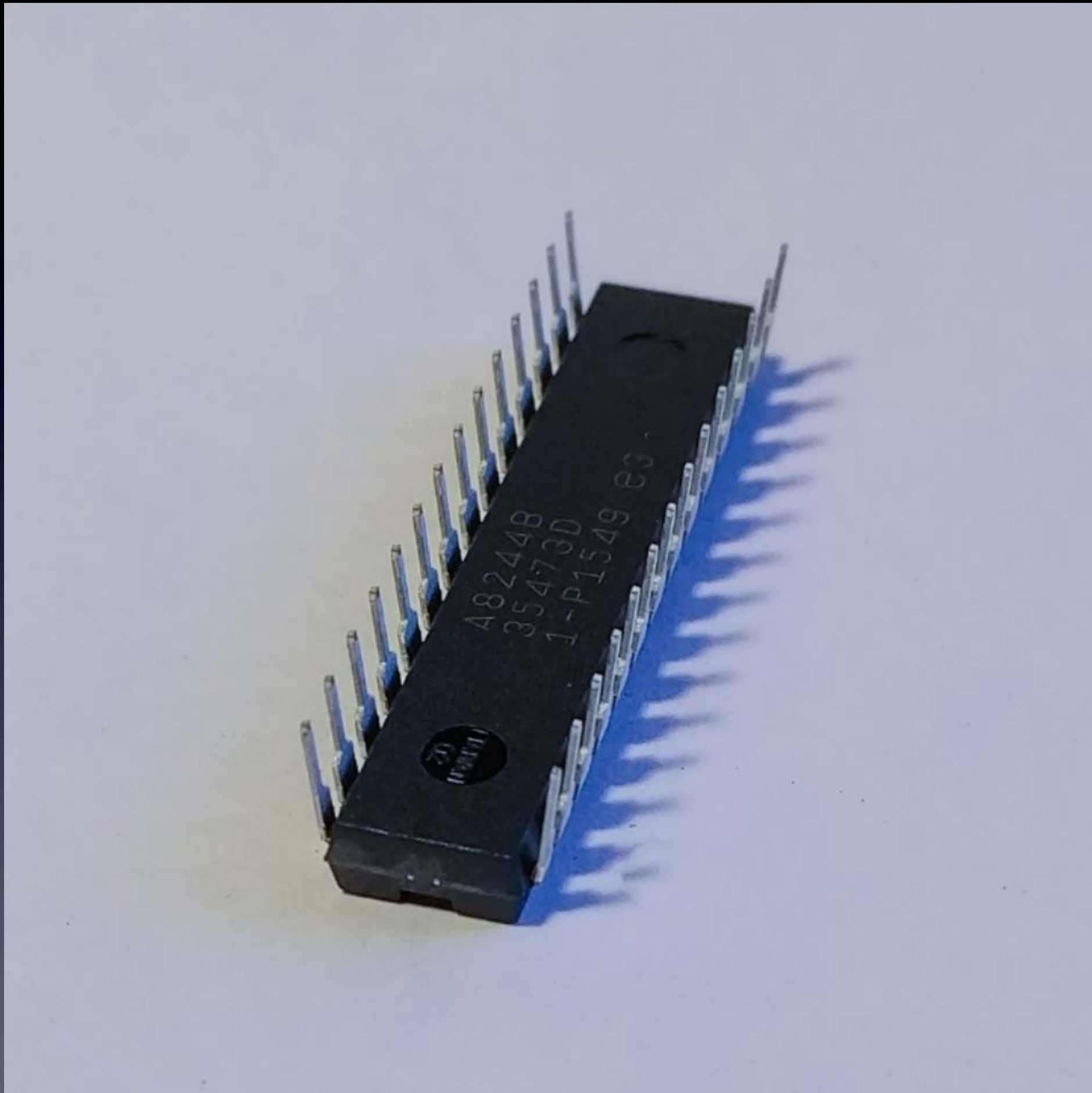
**Now it is easy to solder all of the Header pads at once.**



**Headers**

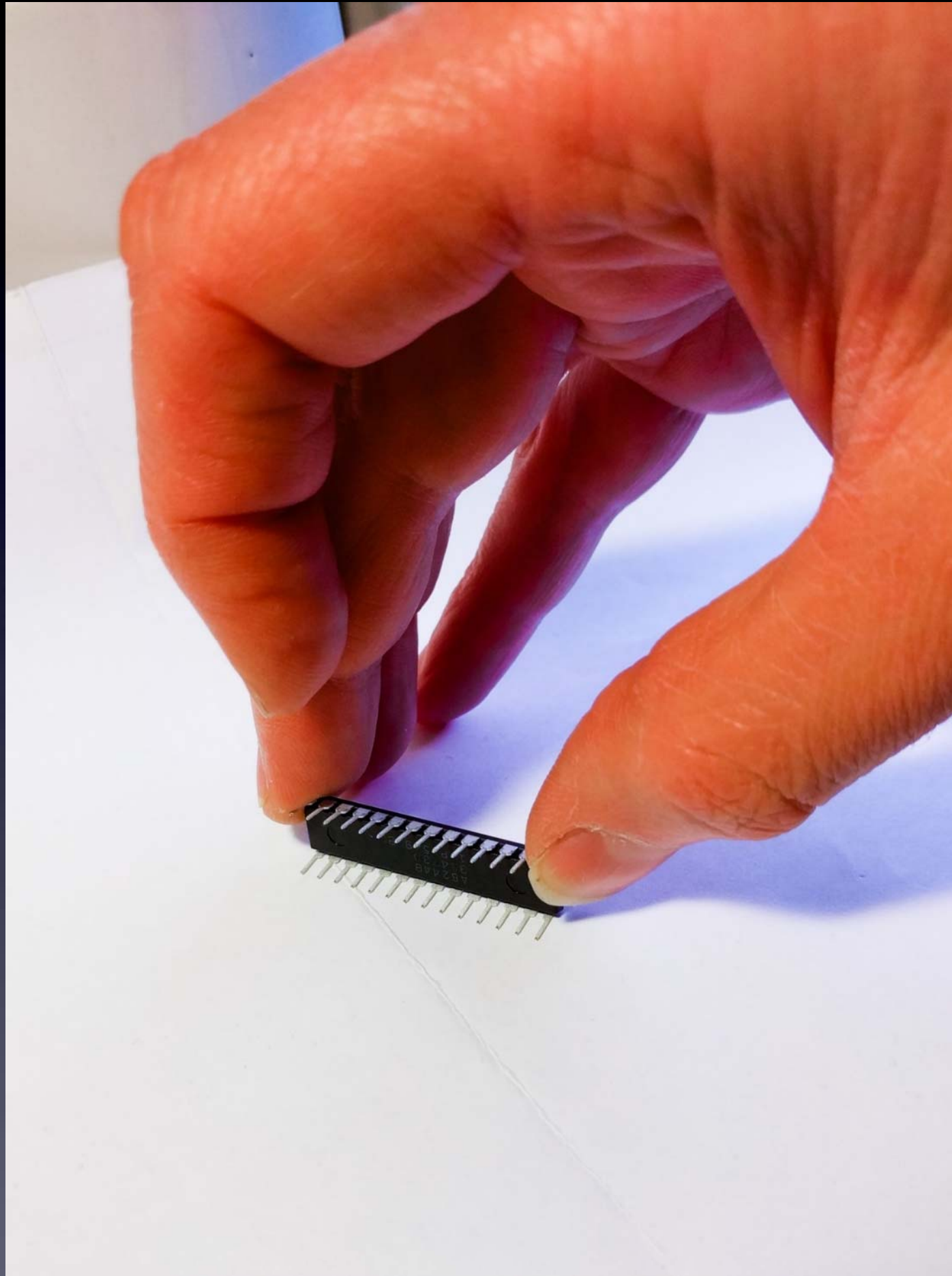


**You probably need to bend the pins so they're parallel**



**Microcontroller**

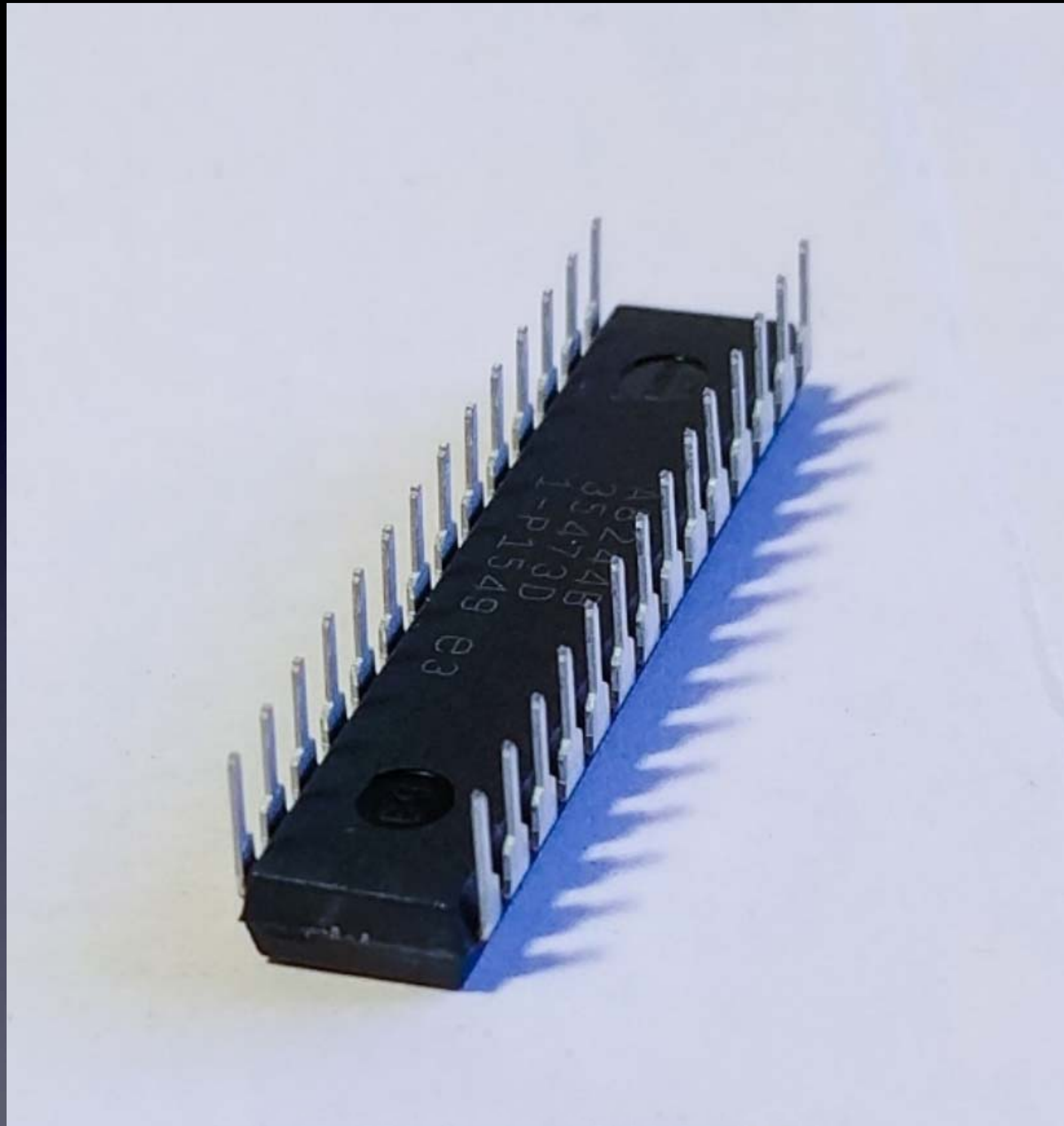
**You probably need to bend the pins so they're parallel**



**Microcontroller**

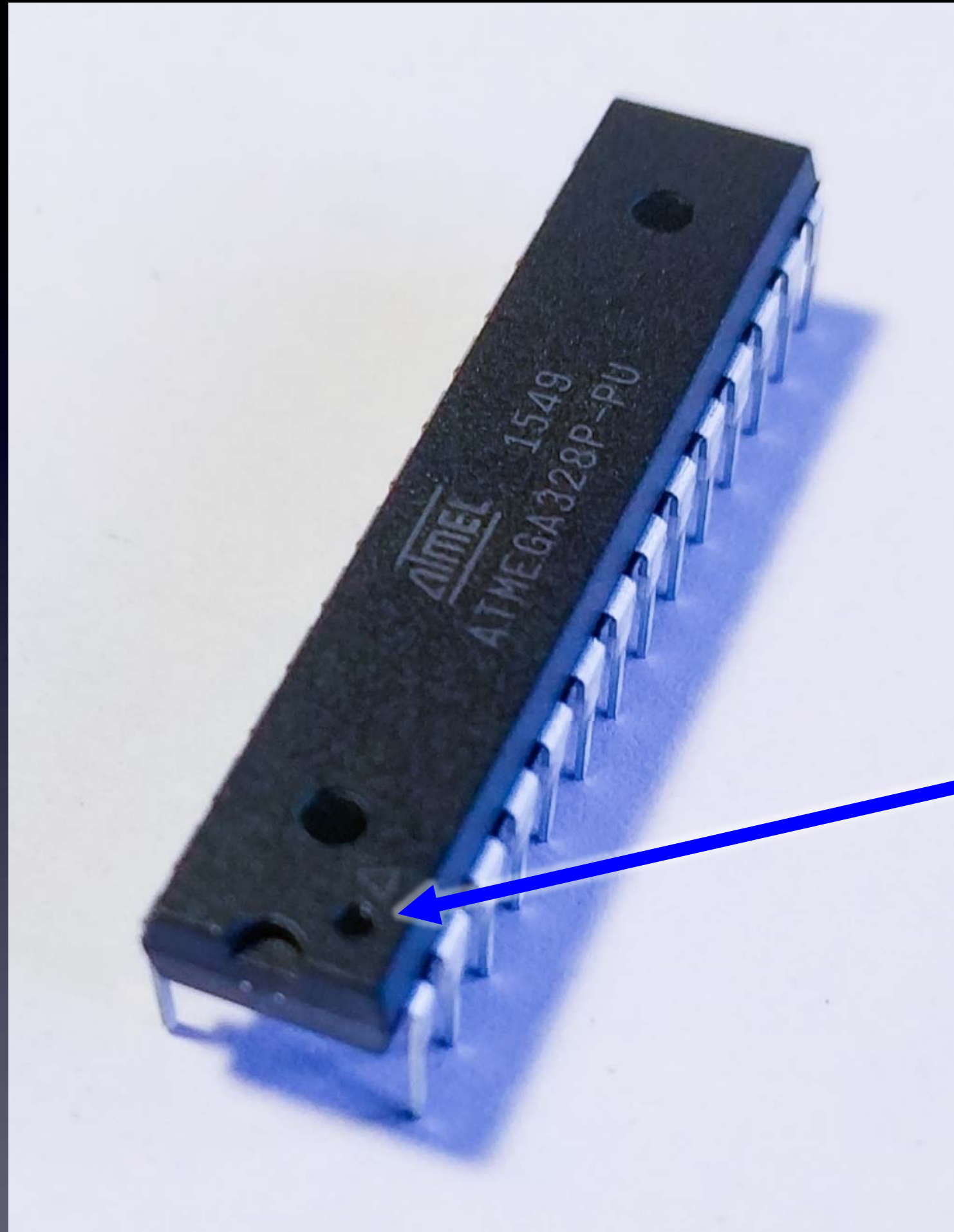


**You probably need to bend the pins so they're parallel**



**Microcontroller**

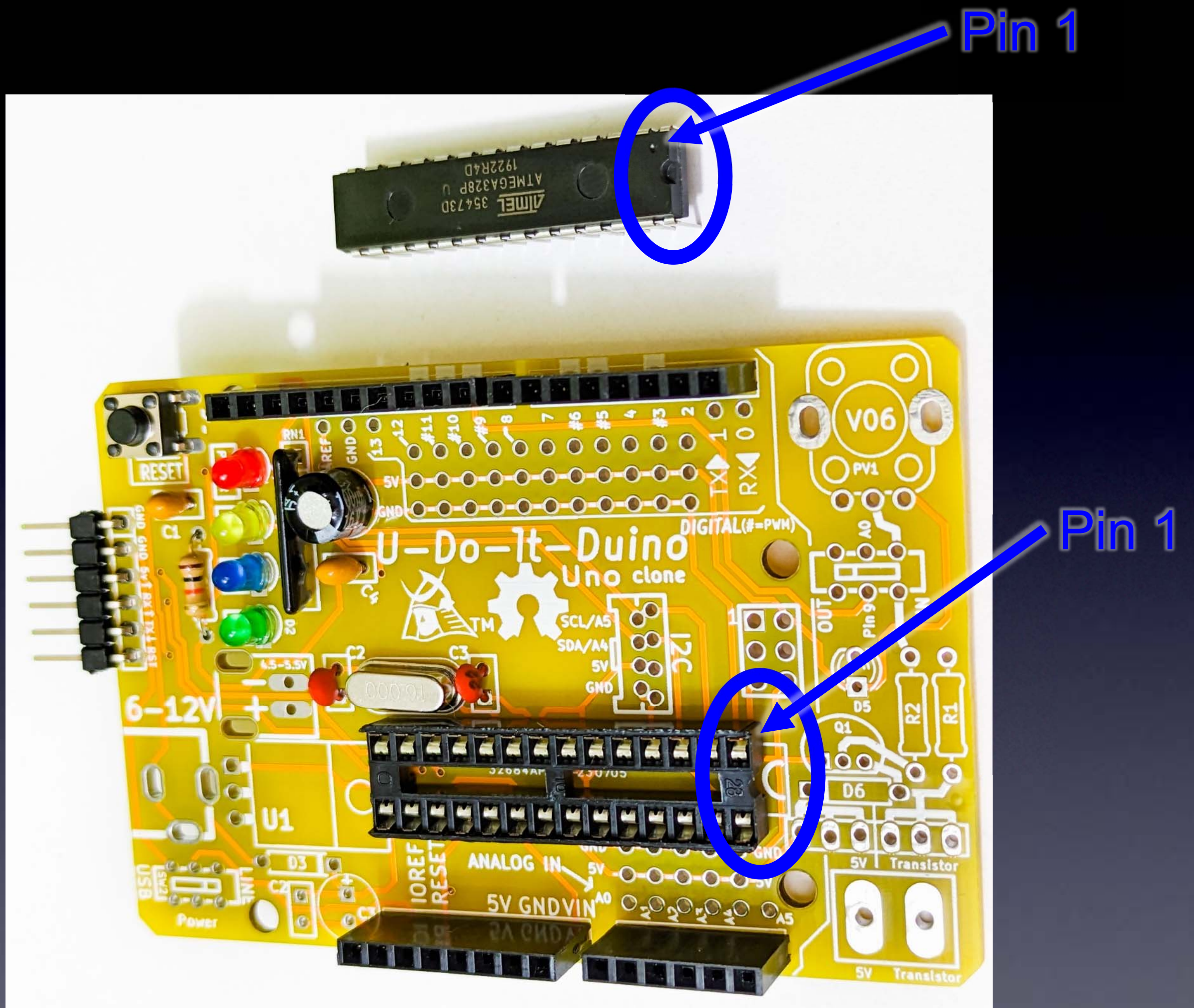
**Ready to insert into its socket**



**Pin 1**

**Microcontroller**

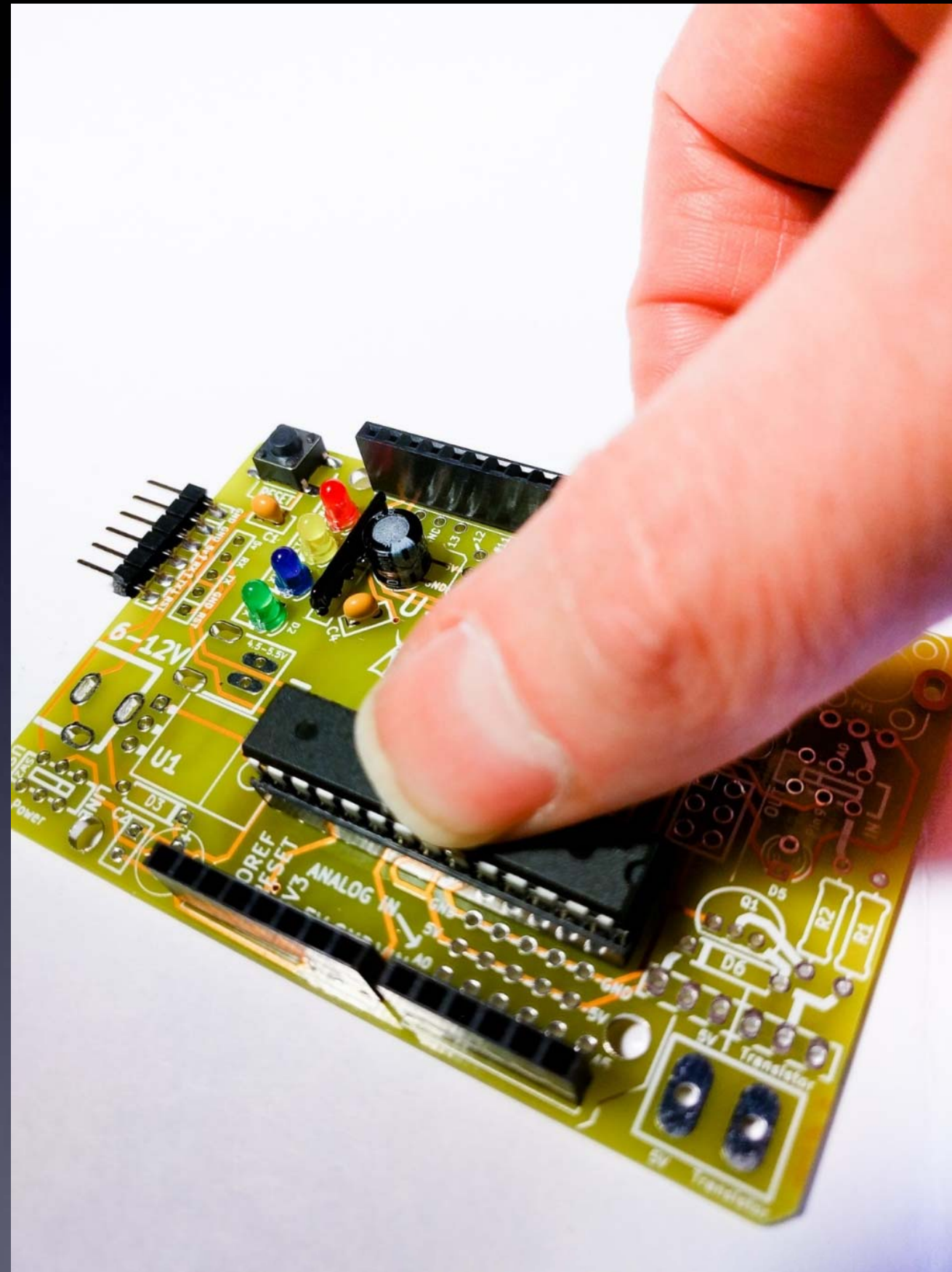




**Microcontroller**



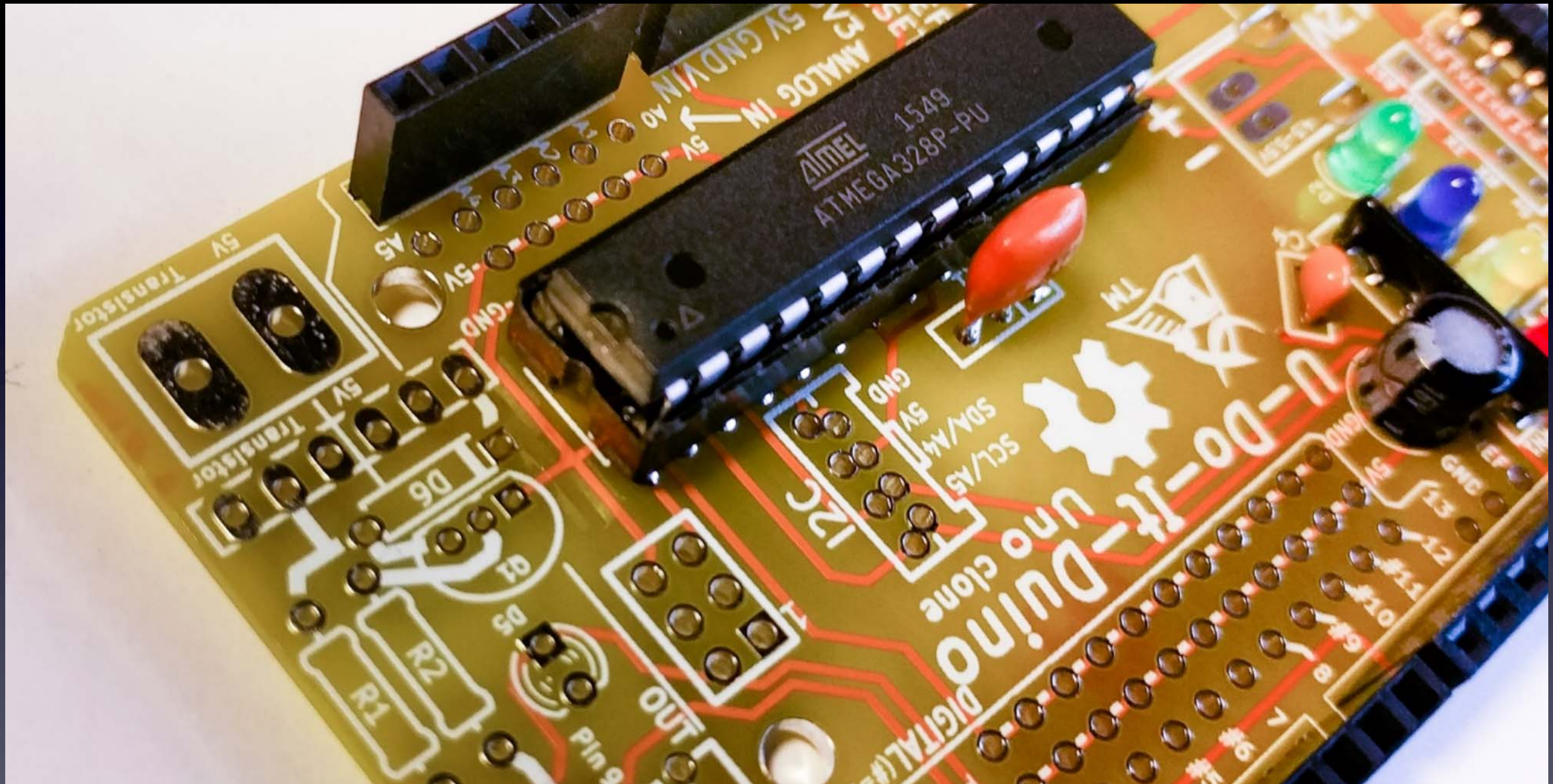
**When all pins fit in their holes,  
use both thumbs to push chip into socket**



**Microcontroller**



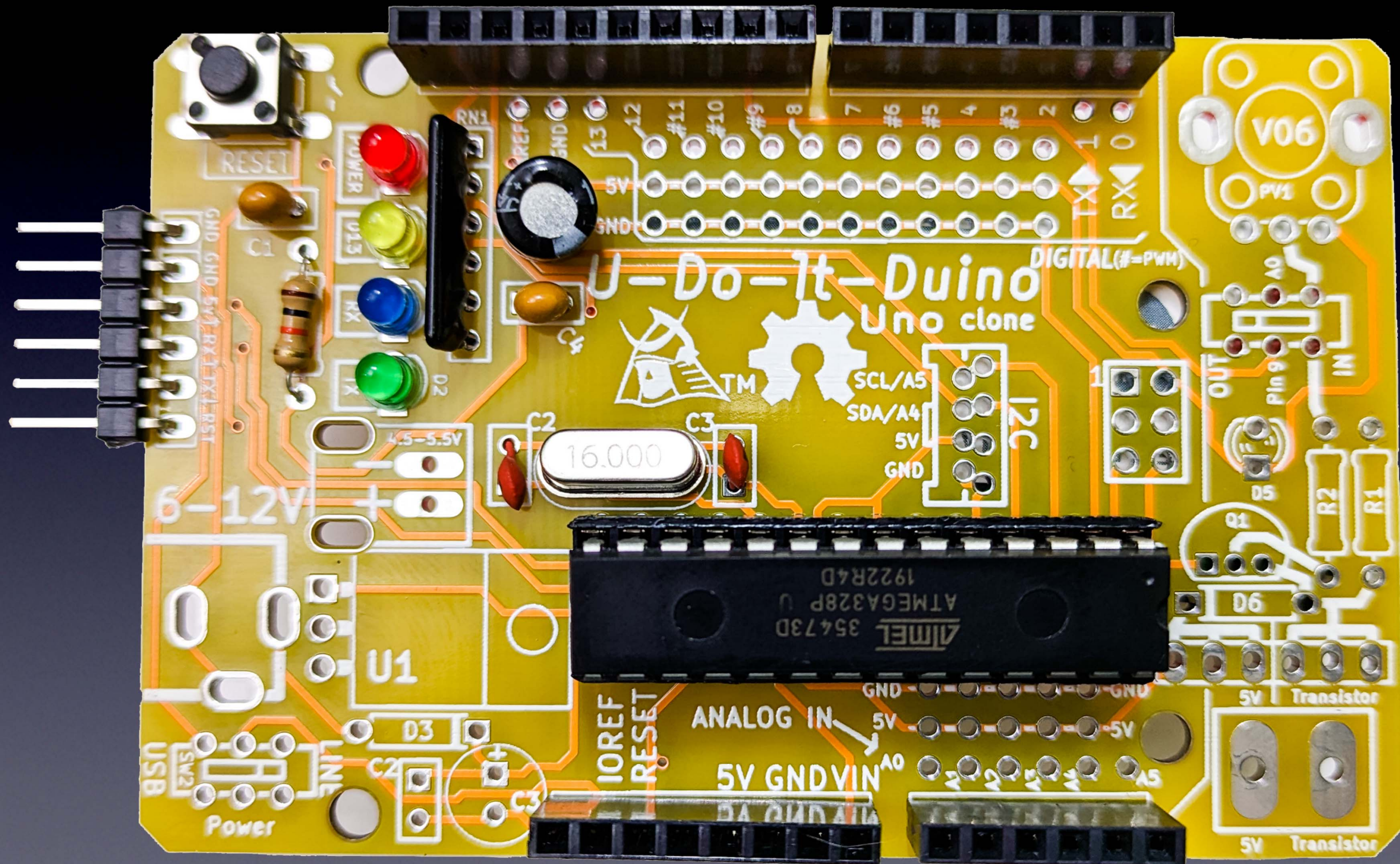
**Each pin in its hole, chip pushed down all the way.  
Pin 1 is oriented correctly.**



**Microcontroller**

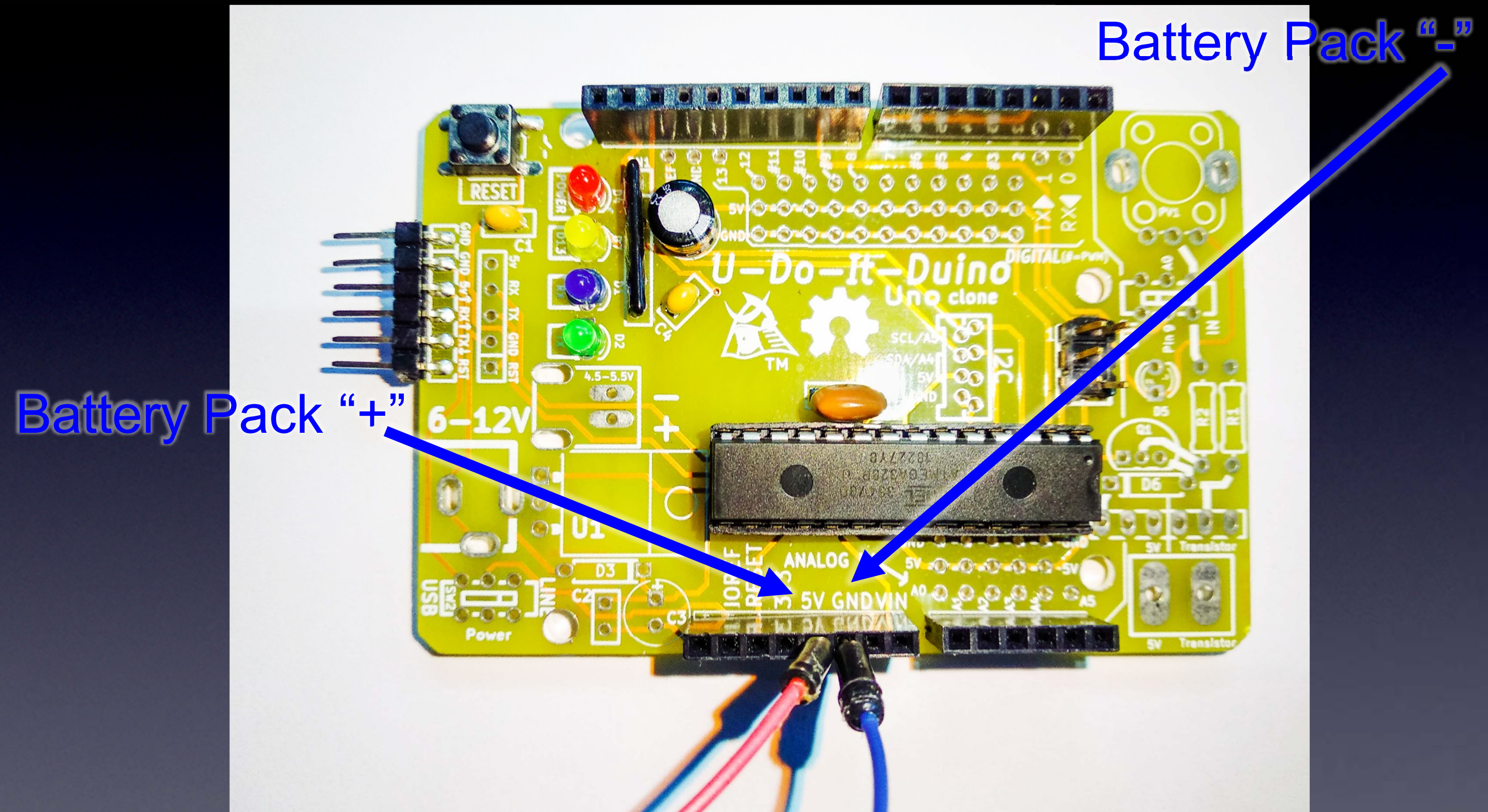


# We're done!





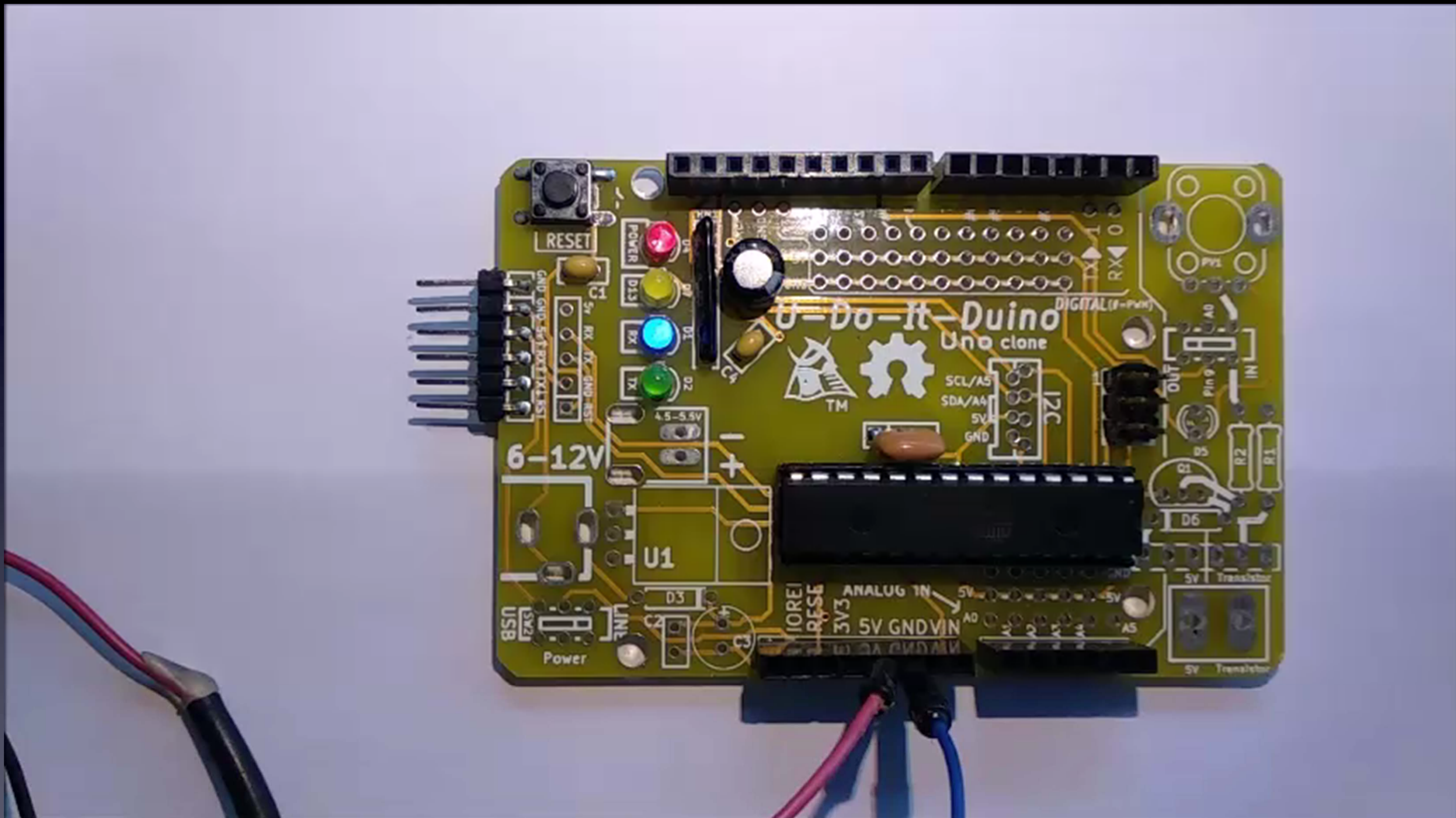
# First test:



Connect power with a battery pack...



# First test:

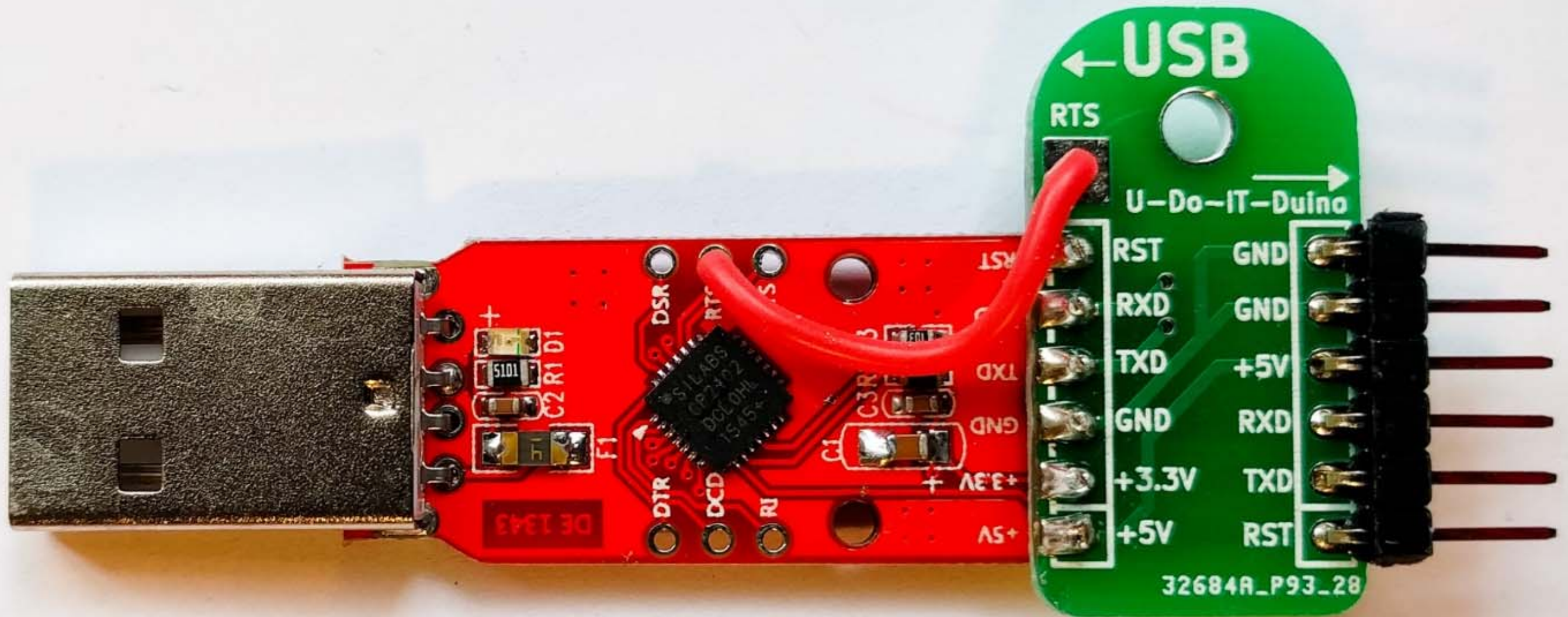


... and it blinks !



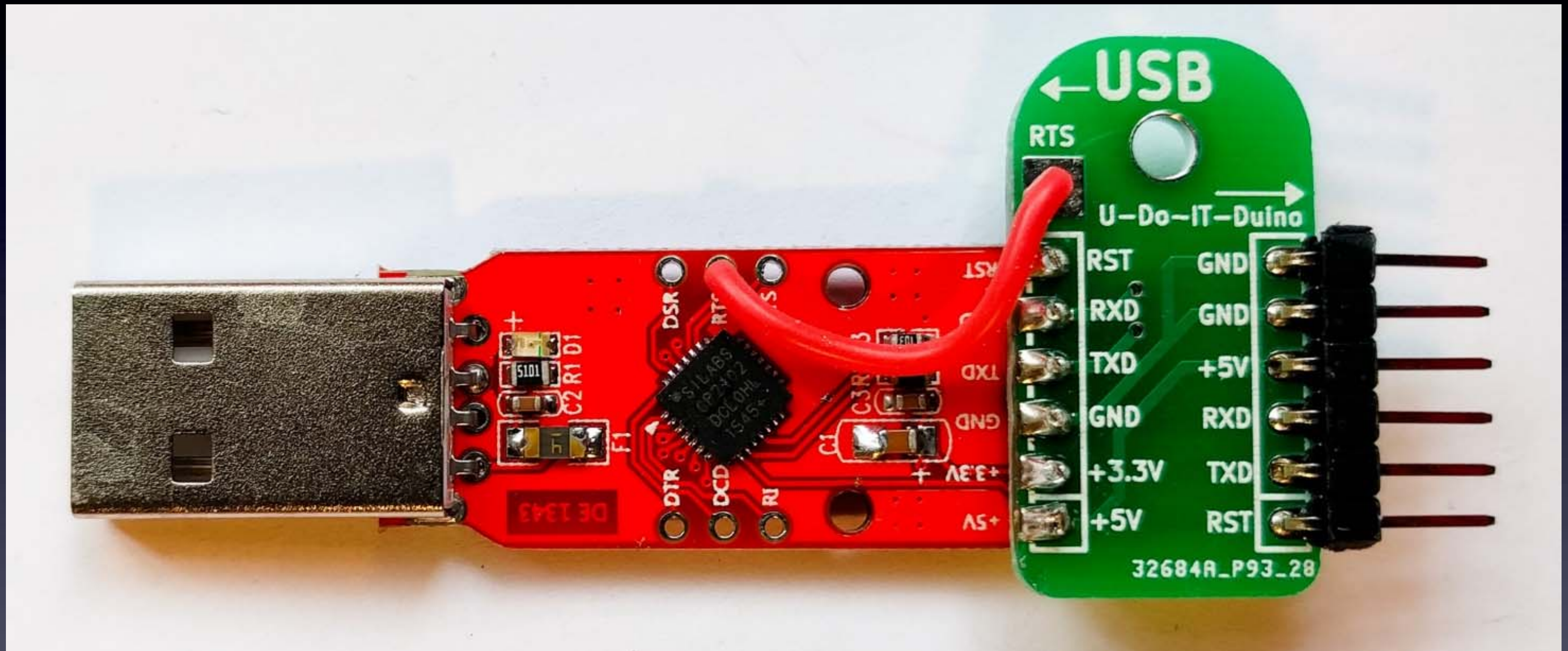
Now we can  
connect parts to our Arduino,  
and program it!

# USB-Serial Cable





# USB-Serial Cable Driver



**You may need to download and install a driver  
for your Operating System (Windows, MacOS, or Linux):**

[<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers)

*Helpful info*  
on the  
Arduino for(4) Total Newbies

---

**workshop**  
**web-page:**

[https://cornfieldelectronics.com/cfe/projects/tvbg\\_arduino/tvbg\\_arduino\\_workshop.php](https://cornfieldelectronics.com/cfe/projects/tvbg_arduino/tvbg_arduino_workshop.php)





cornFIELD electronics

useful electronics for a better world

home

buy

about us

press

distributors

projects

show cart



# Take CONTROL

At Cornfield Electronics we create devices that give people opportunities for effective choices in their lives. Each of us can decide whether to watch TV monitors, and when to watch. Each of us can decide when to get the rest we want, and how we dream. Everyone can learn to make cool things with our kits. Please explore our [products](#), make your own choices, and see how *your* life can be enhanced.

[join our mailing list](#)

Love it or hate it, TV screens are all around us. [TV-B-Gone®](#) universal remote control is the first fruit of our technical savvy, embodying our belief in empowerment, and sense of humor. This universal remote control fits in your pocket and allows you to discreetly turn TVs off wherever you go. TV-B-Gone fans around the world are using it for a variety of practical, philosophical, and humorous purposes. Imagine the possibilities...

Years in the making [NeuroDreamer](#) sleep mask is another of our personal empowerment inventions. We all need rest, but we don't always get it in our busy lives. NeuroDreamer sleep mask lets you use your own brainwaves to

bring you the rest you need. And with the [lucid dreaming model](#), you can take control of your dreams.

Want to learn electronics? We make way cool, fun, intriguing, educational [kits](#) that **anyone can make!** Our most **POPULAR** kits are: [ArduTouch music synthesizer kit](#) and [TV-B-Gone kit!](#)

We make truly useful technological solutions that put you in charge.

Welcome to our better world!

**NOTE: As of 14-Feb-2023 Cornfield Electronics is a sole proprietorship of Mitch Altman.**



## DO-IT-YOURSELF PROJECTS

by **Mitch Altman**, and friends.

Last modified: 5-Oct-2022

### You Can Make Cool Things With Electronics!

The projects on this page were all created for total beginners, with no experience, so everyone can complete them successfully at my workshops, or at home, or anywhere!

All you need is:

a desire, a handful of parts, a soldering iron (with stand and sponge), a wire-cutter, a wire-stripper, solder, and an afternoon.



[Here](#) is a really nice tutorial on how to solder -- for total beginners!

[Soldering Tutorial for total beginners](#)

### Open Hardware!

Everything on this page (and everything I do) is free and open source!

(That's *free* as in *freedom*.)

(But everything here is free to download -- and that is *free* as in *beer*.)

If you have any questions on anything, please feel free to email me:

*mitch AT CornfieldElectronics DOT com*



### Soldering!

Soldering is fun! And it is easy! Really, it is!

I have taught tens of thousands of people around the world how to solder.

Everyone can do it! All ages, all skill levels.

People who have never even sewn a button can easily learn to solder. Even you!

Once you learn how to make one good solder connection, you can make anything on this page.

And if you can make anything on this page, you can learn to make anything with electronics and microcontrollers.



---

**Project: Make your own open source TV-B-Gone Kit (developed with Ladayada)**

The TV-B-Gone Kit was originally developed from a MiniPOV3 hack (see below) (which, of course, I hacked from my original [TV-B-Gone](#).)

For excellent **assembly instructions**, please go to the [TV-B-Gone Kit page](#) of the of the Adafruit.com website.

For **questions** about the TV-B-Gone Kit, please go to the [TV-B-Gone Kit user forum](#).  
To see the **schematic, firmware, and board layout**, please go to [TV-B-Gone Kit downloads](#).

TV-B-Gone Kits are available for **purchase** from the [TVBGone.com](#) website.

---

**Project: Arduino For Total Newbies workshop**  
-- Learn Arduino, and make your own TV-B-Gone!

This workshop covers lots of ground -- all you need to learn how to play with Arduinos. As an example project, you can make your own TV-B-Gone using [Arduino](#).  
*Many thanks to Ken Shirriff for the original [TV-B-Gone for Arduino project](#)!*  
For documentation on this workshop, please see the:  
[Arduino For Total Newbies Workshop](#) page.

scroll down



### Project: Make your own open source TV-B-Gone Kit (developed with Ladayada)



The TV-B-Gone Kit was originally developed from a MiniPOV3 hack (see below) (which, of course, I hacked from my original [TV-B-Gone](#).)

For excellent **assembly instructions**, please go to the [TV-B-Gone Kit page](#) of the of the Adafruit.com website.

For **questions** about the TV-B-Gone Kit, please go to the [TV-B-Gone Kit user forum](#).  
To see the **schematic, firmware, and board layout**, please go to [TV-B-Gone Kit downloads](#).

TV-B-Gone Kits are available for **purchase** from the [TVBGone.com](#) website.

### Project: Arduino For Total Newbies workshop -- Learn Arduino, and make your own TV-B-Gone!



This workshop covers lots of ground -- all you need to learn how to play with [Arduinos](#).

As an example project, you can make your own TV-B-Gone using [Arduino](#).

Many thanks to Ken Shirriff for the original [TV-B-Gone for Arduino Project](#)!

For documentation on this workshop, please see the:

[Arduino For Total Newbies Workshop page](#).





## Arduino For Total Newbies Workshop

*last updated: 1-May-2023*

Learn how to make your own way cool projects with Arduino,  
using TV-B-Gone as an example project to learn from.



I've given this workshop at [Noisebridge](#) hackerspace in San Francisco (several times), at [27C3](#) and [28C3](#) in Berlin, and [29C3](#), [30C3](#), [31C3](#), [32C3](#), and [33C3](#) in Hamburg, and at [34C3](#), [35C3](#), [36C3](#) in Leipzig, and [RC3](#) online, at [CCCamp2011](#) [CCCamp2015](#) and [CCCamp2019](#) outside of Berlin, at [HeatSync Labs](#) hackerspace in Phoenix, AZ, at [Fabelier](#) hackerspace in Paris, at Unit One in [2012](#), [2014](#), [2016](#), and [2018](#) as Hacker In Residence at the University of Illinois, in Urbana, IL, at Makerspace Urbana in Urbana, IL in [2012](#) and in [2016](#), at [Workshop Weekend](#) in Oakland, CA (twice), at [XinCheJian](#) hackerspace in Shanghai, at [Maker Carnival](#) in Beijing (twice) at several conferences and hackerspaces on my [Hackers on a Train Workshop Tour 2012](#) including at [HOPE Number 9](#) in New York City, at [ToorCamp 2012](#) in Neah Bay, WA, at [OHM 2013](#) and [SHA 2017](#) outside of Amsterdam, at [RockIT CoLabs](#) in San Francisco, at [BalcCon2k14](#) in Novi Sad, Serbia, at [HOPE X](#), [The Eleventh HOPE](#), [The Circle of HOPE](#) [A New HOPE](#) in New York City, and [HOPE 2020](#) online, at [the iCenter](#) as Hacker In Residence at Tsingua University in Beijing, at [EMF Camp 2016](#), [EMF Camp 2018](#), and [EMF Camp 2022](#), outside of London, at [Tami](#) hackerspace in Tel Aviv, at [Le Wagon](#) and [Zhongxi](#) in Chengdu, at [Astralship](#) hackerspace in North Wales (three times), at [Open Source Microfactory Build Camp](#) online, at [Newline](#) in Ghent, at [GPN20](#) in Karlsruhe, at [Fri3d Camp 2022](#) near Sint-Joris-Weert, Belgium, at [Maker Faire Brno 2022](#) in Brno, Czech Republic, at [HiP-Berlin](#) in Berlin, Germany, and lots of other places.

Each time 10 to 50 people show up. (Folks seem to like it.)

**Itinerary for Arduino For Total Newbies Workshop:**

### Here is what is available for downloading for the Arduino For Total Newbies Workshop:

1) If you don't already have **Arduino software** you need to download it for your computer (Windows, Mac OS, or Linux):

[Arduino download page](#)

2) You need a **driver for your USB communications/programming cable**. Several different ones are available. Choose the driver for the cable you have and the operating system for your computer.

**Samurai Circuits board (SiLabs CP210x USB-to-Serial TTL) drivers:**

[The latest drivers from SiLabs' website](#)

*The SiLabs driver is installed by default on most Linux systems.*

**Adafruit FTDI Friend drivers:**

[The latest drivers from FTDI's website](#)

*The FTDI driver is installed by default on most Linux systems.*

**FTDI Cable drivers:**

[The latest drivers from FTDI's website](#)

*The FTDI driver is installed by default on most Linux systems.*

3) You also need the **TV-B-Gone Arduino Sketch** (download this, unzip it, and copy it to your computer in the "examples" folder inside your "arduino" folder that you downloaded):

[TV-B-Gone Arduino sketch\(22KB\)](#)

4) **Schematic Diagram for Arduino TV-B-Gone remote control:**

[Schematic Diagram\(449KB\)](#)

**Parts List for Arduino TV-B-Gone remote control:**

[Parts List \(Open Office\)\(12KB\)](#)

[Parts List \(MS Office\)\(9KB\)](#)

**Schematics for the U-Do-It-Duino Arduino clone kit:**

[U-Do-It-Duino schematic \(110KB\)](#)

**Complete assembly instructions for the U-Do-It-Duino Arduino clone kit:**

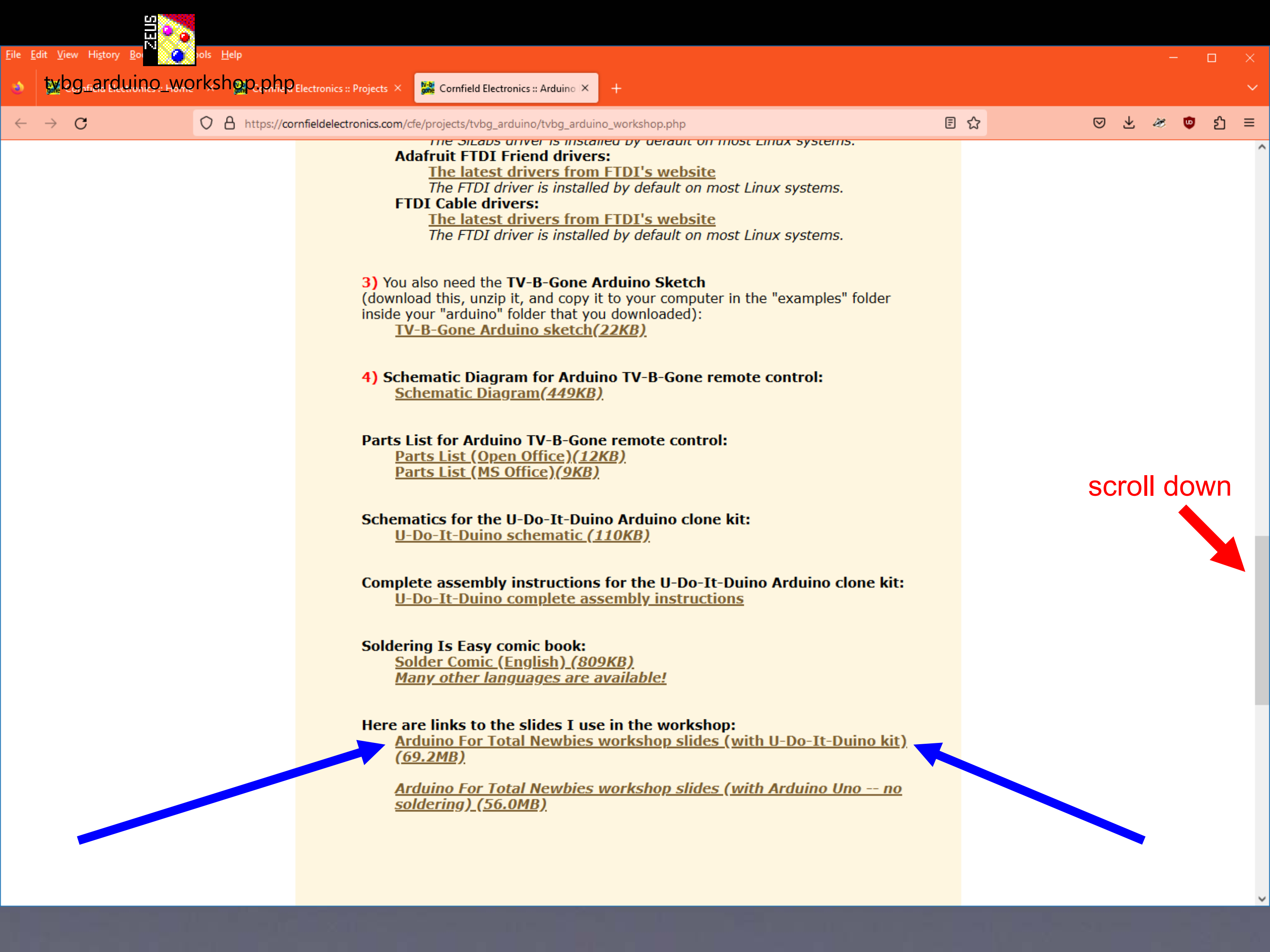
[U-Do-It-Duino complete assembly instructions](#)

Stuff to download

scroll down







*The SiLabs driver is installed by default on most Linux systems.*

**Adafruit FTDI Friend drivers:**

[The latest drivers from FTDI's website](#)

*The FTDI driver is installed by default on most Linux systems.*

**FTDI Cable drivers:**

[The latest drivers from FTDI's website](#)

*The FTDI driver is installed by default on most Linux systems.*

**3) You also need the TV-B-Gone Arduino Sketch**

(download this, unzip it, and copy it to your computer in the "examples" folder inside your "arduino" folder that you downloaded):

[TV-B-Gone Arduino sketch\(22KB\)](#)

**4) Schematic Diagram for Arduino TV-B-Gone remote control:**

[Schematic Diagram\(449KB\)](#)

**Parts List for Arduino TV-B-Gone remote control:**

[Parts List \(Open Office\)\(12KB\)](#)

[Parts List \(MS Office\)\(9KB\)](#)

**Schematics for the U-Do-It-Duino Arduino clone kit:**

[U-Do-It-Duino schematic \(110KB\)](#)

**Complete assembly instructions for the U-Do-It-Duino Arduino clone kit:**

[U-Do-It-Duino complete assembly instructions](#)

**Soldering Is Easy comic book:**

[Solder Comic \(English\)\(809KB\)](#)

[Many other languages are available!](#)

**Here are links to the slides I use in the workshop:**

[Arduino For Total Newbies workshop slides \(with U-Do-It-Duino kit\)\(69.2MB\)](#)

[Arduino For Total Newbies workshop slides \(with Arduino Uno -- no soldering\)\(56.0MB\)](#)



*The SiLabs driver is installed by default on most Linux systems.*

**Adafruit FTDI Friend drivers:**

[The latest drivers from FTDI's website](#)

*The FTDI driver is installed by default on most Linux systems.*

**FTDI Cable drivers:**

[The latest drivers from FTDI's website](#)

*The FTDI driver is installed by default on most Linux systems.*

**3) You also need the TV-B-Gone Arduino Sketch**

(download this, unzip it, and copy it to your computer in the "examples" folder inside your "arduino" folder that you downloaded):

[TV-B-Gone Arduino sketch\(22KB\)](#)

**4) Schematic Diagram for Arduino TV-B-Gone remote control:**

[Schematic Diagram\(449KB\)](#)

**Parts List for Arduino TV-B-Gone remote control:**

[Parts List \(Open Office\)\(12KB\)](#)

[Parts List \(MS Office\)\(9KB\)](#)

**Schematics for the U-Do-It-Duino Arduino clone kit:**

[U-Do-It-Duino schematic \(110KB\)](#)

**Complete assembly instructions for the U-Do-It-Duino Arduino clone kit:**

[U-Do-It-Duino complete assembly instructions](#)

**Soldering Is Easy comic book:**

[Solder Comic \(English\)\(809KB\)](#)

[Many other languages are available!](#)

**Here are links to the slides I use in the workshop:**

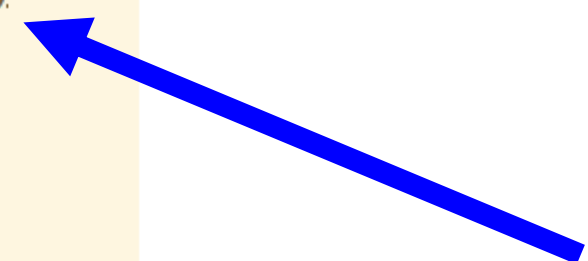
[Arduino For Total Newbies workshop slides \(with U-Do-It-Duino kit\)\(69.2MB\)](#)

[Arduino For Total Newbies workshop slides \(with Arduino Uno -- no soldering\)\(56.0MB\)](#)

scroll down



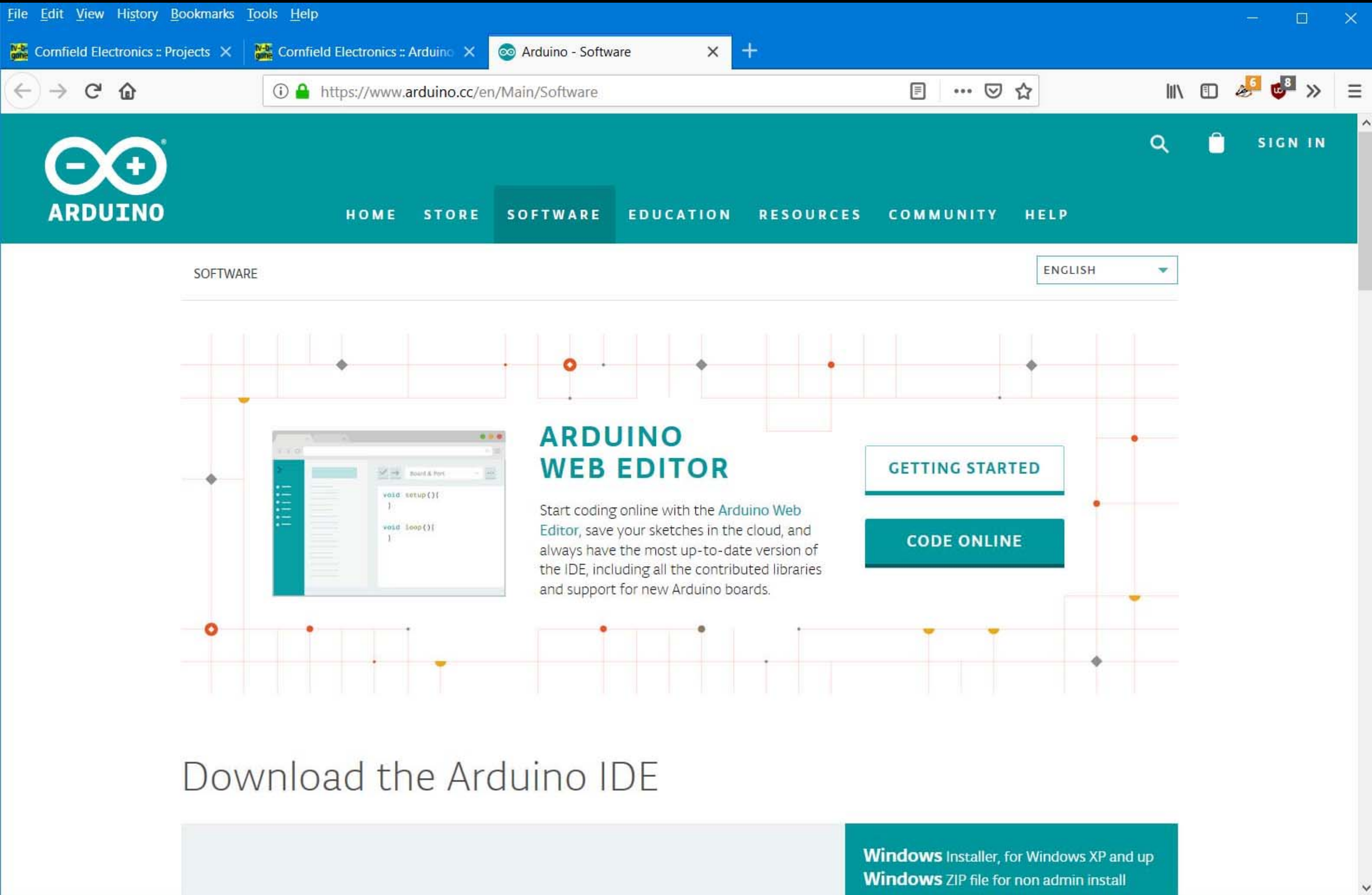
Slides for this workshop





1) If you don't already have **Arduino software** you need to download it for your computer (Windows, Mac OS, or Linux):

[Arduino download page](#)



The screenshot shows a web browser window with the Arduino Software page. The browser's address bar displays the URL <https://www.arduino.cc/en/Main/Software>. The page features the Arduino logo on the left and a navigation menu with links to HOME, STORE, SOFTWARE (which is highlighted), EDUCATION, RESOURCES, COMMUNITY, and HELP. On the right side of the header, there are icons for search, a shopping cart, and a 'SIGN IN' button. Below the navigation bar, the word 'SOFTWARE' is displayed on the left, and a language dropdown menu is set to 'ENGLISH'. The main content area is decorated with a background of circuit board traces. It includes a section for the 'ARDUINO WEB EDITOR' with a description: 'Start coding online with the [Arduino Web Editor](#), save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards.' To the left of this text is a small image of the web editor interface showing a code editor with the following code:

```
void setup(){  
  }  
  
void loop(){  
  }
```

To the right of the text are two buttons: 'GETTING STARTED' and 'CODE ONLINE'. At the bottom of the page, there is a section titled 'Download the Arduino IDE' with two download options: 'Windows Installer, for Windows XP and up' and 'Windows ZIP file for non admin install'.

File Edit View History Bookmarks Tools Help

Cornfield Electronics :: Projects Cornfield Electronics :: Arduino Arduino - Software

<https://www.arduino.cc/en/Main/Software>

ARDUINO

HOME STORE SOFTWARE EDUCATION RESOURCES COMMUNITY HELP

SOFTWARE

ENGLISH

## ARDUINO WEB EDITOR

Start coding online with the [Arduino Web Editor](#), save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards.

GETTING STARTED

CODE ONLINE

## Download the Arduino IDE

Windows Installer, for Windows XP and up

Windows ZIP file for non admin install



**2)** You need a **driver for your USB communications/programming cable**. Several different ones are available. Choose the driver for the cable you have and the operating system for your computer.

**Samurai Circuits board (SiLabs CP210x USB-to-Serial TTL) drivers:**

**The latest drivers from SiLabs' website**

*The SiLabs driver is installed by default on most Linux systems.*



About ▾ Products ▾ Solutions ▾ Community & Support ▾

简体中文 日本語

Log In | Register

Parametric Search | Cross-Reference Search

Search silabs.com



Silicon Labs » Products » Development Tools » Software » USB to UART Bridge VCP Drivers

# CP210x USB to UART Bridge VCP Drivers

The CP210x USB to UART Bridge Virtual COM Port (VCP) drivers are required for device operation as a Virtual COM Port to facilitate host communication with CP210x products. These devices can also interface to a host using the [direct access driver](#). These drivers are static examples detailed in application note 197: The Serial Communications Guide for the CP210x, download an example below:

[AN197: The Serial Communications Guide for the CP210x](#)

## Download Software

The CP210x Manufacturing DLL and Runtime DLL have been updated and must be used with v6.0 and later of the CP210x Windows VCP Driver. Application Note Software downloads affected are AN144SW.zip, AN205SW.zip and AN223SW.zip. If you are using a 5.x driver and need support you can download archived Application Note Software.

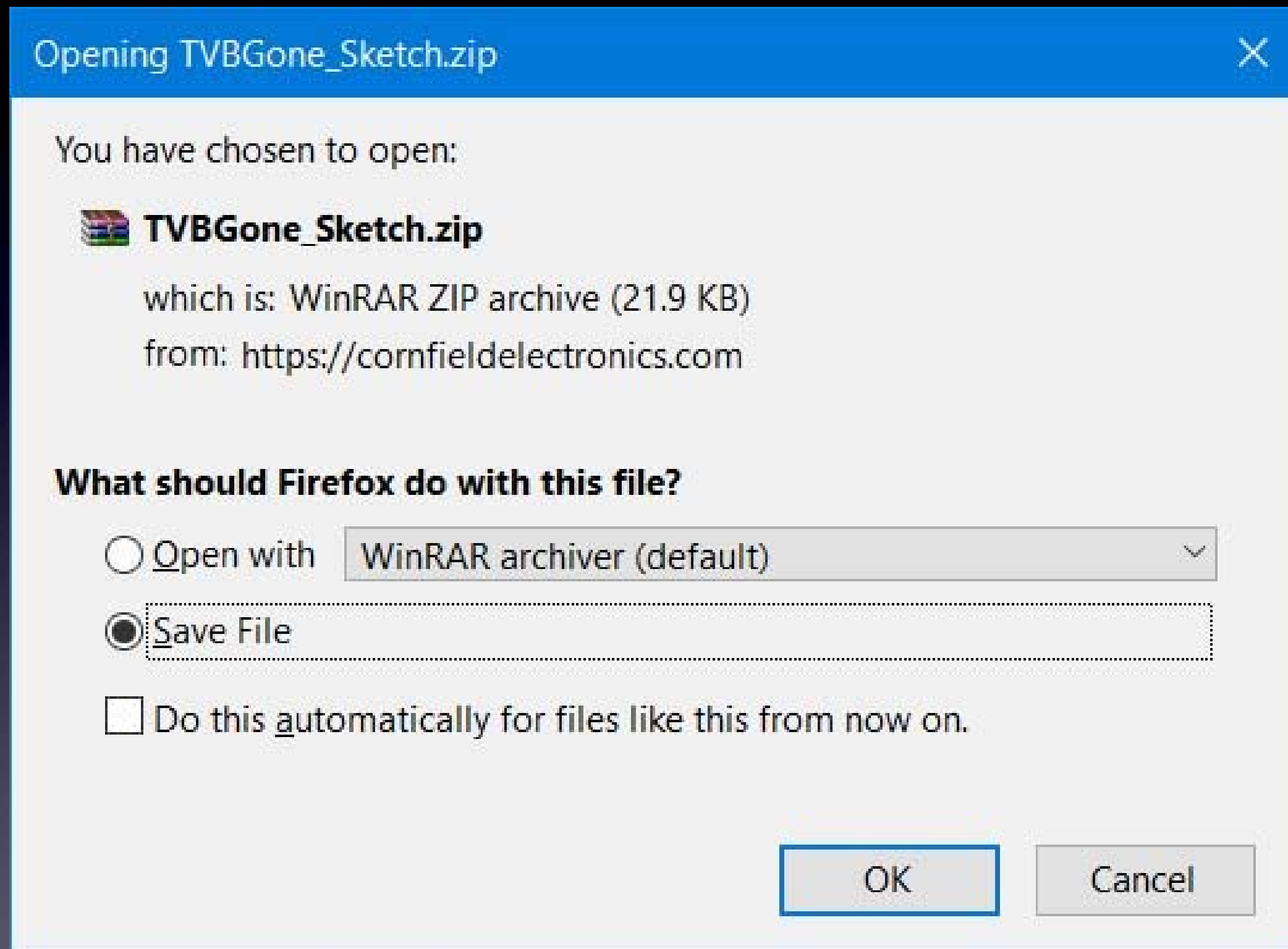
[Legacy OS software and driver package download links and support information](#) >

## Download for Windows 10 Universal (v10.1.4)



**3)** You also need the **TV-B-Gone Arduino Sketch**  
(download this, unzip it, and copy it to your computer in the "examples" folder  
inside your "arduino" folder that you downloaded):

**[TV-B-Gone Arduino sketch\(22KB\)](#)**



#### 4) Schematic Diagram for Arduino TV-B-Gone remote control: Schematic Diagram(449KB)

File Edit View History Bookmarks Tools Help

Cornfield Electronics :: Projects X Cornfield Electronics :: Arduino X arduino\_tvbgone\_schematic.pdf X

https://cornfeldelectronics.com/cfe/projects/tvbg\_arduino/arduino\_tvbgone\_schematic.pdf 70%

### Arduino For Total Newbies

4-Sep-2015

Mitch Altman (original TV-B-Gone hardware and firmware, modified TV-B-Gone Arduino design)  
Limore Fried (firmware modifications, kit design)  
Ken Shiffert (original modifications for Arduino)  
Johannes Schneemann (documentation)

4.5V red wire  
power switch  
battery pack with switch (3 AA batteries)  
black wire

5V / Vcc  
D3 (out)  
47  
B  
C  
E  
2N3904  
4.5V  
IR LED  
10  
gnd  
black wire

Arduino

D2 (in)  
D5 (in)  
1K  
EU: use 1K  
NA: no resistor  
Visible LED (on Arduino board)  
1K  
D13 (out)

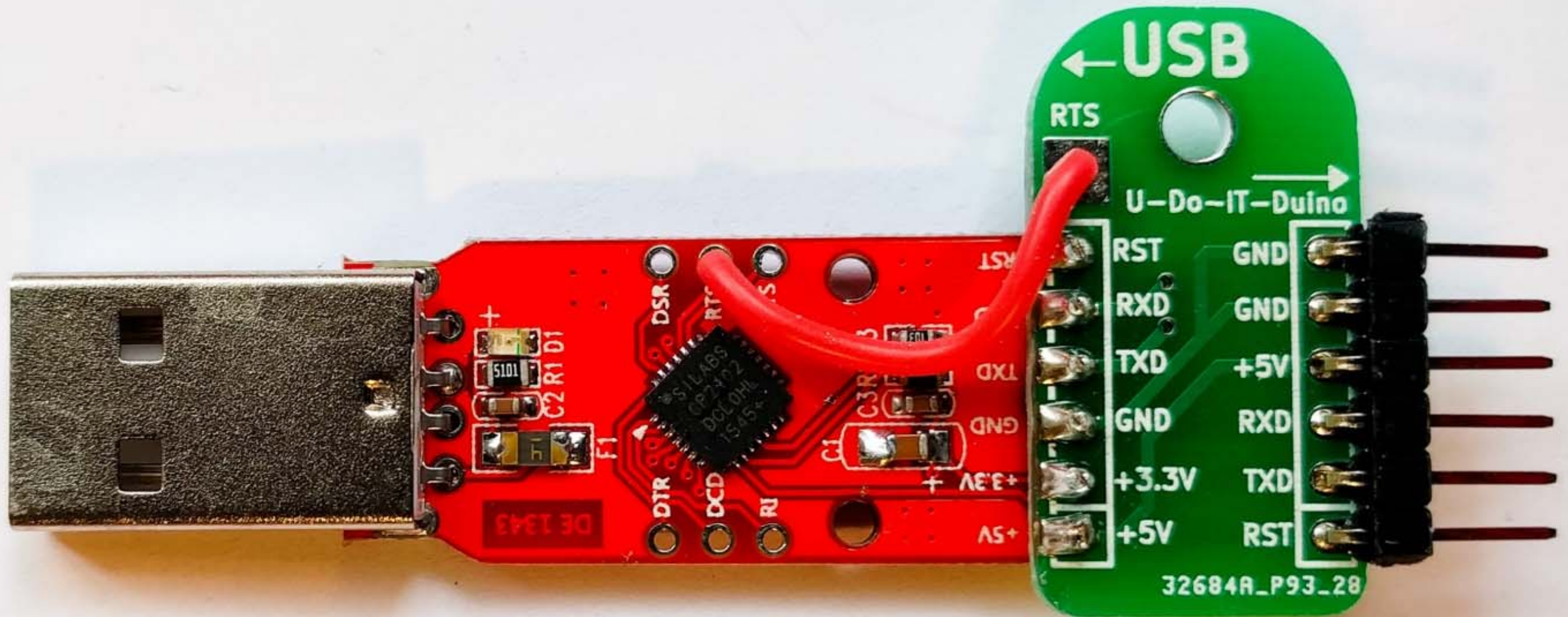
TO-92 package  
2N3904  
C  
B  
E

10 Ohm  
47 Ohm

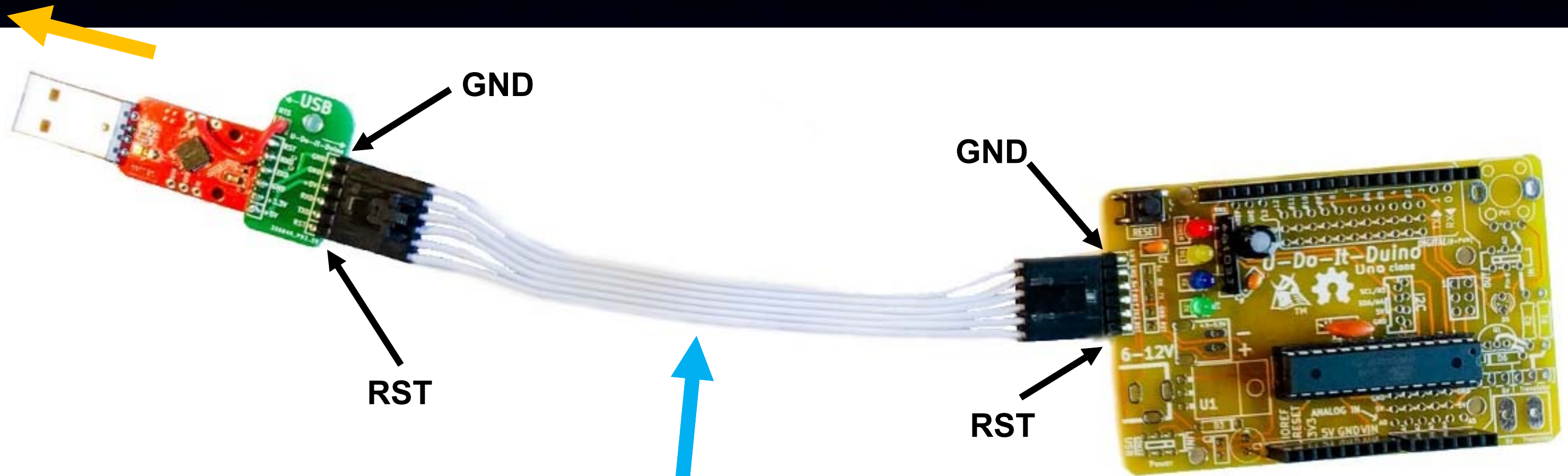
Use either one of these two terminals for one side of the switch  
Use either one of these two terminals for the other side of the switch



# USB-Serial Cable



To computer's USB

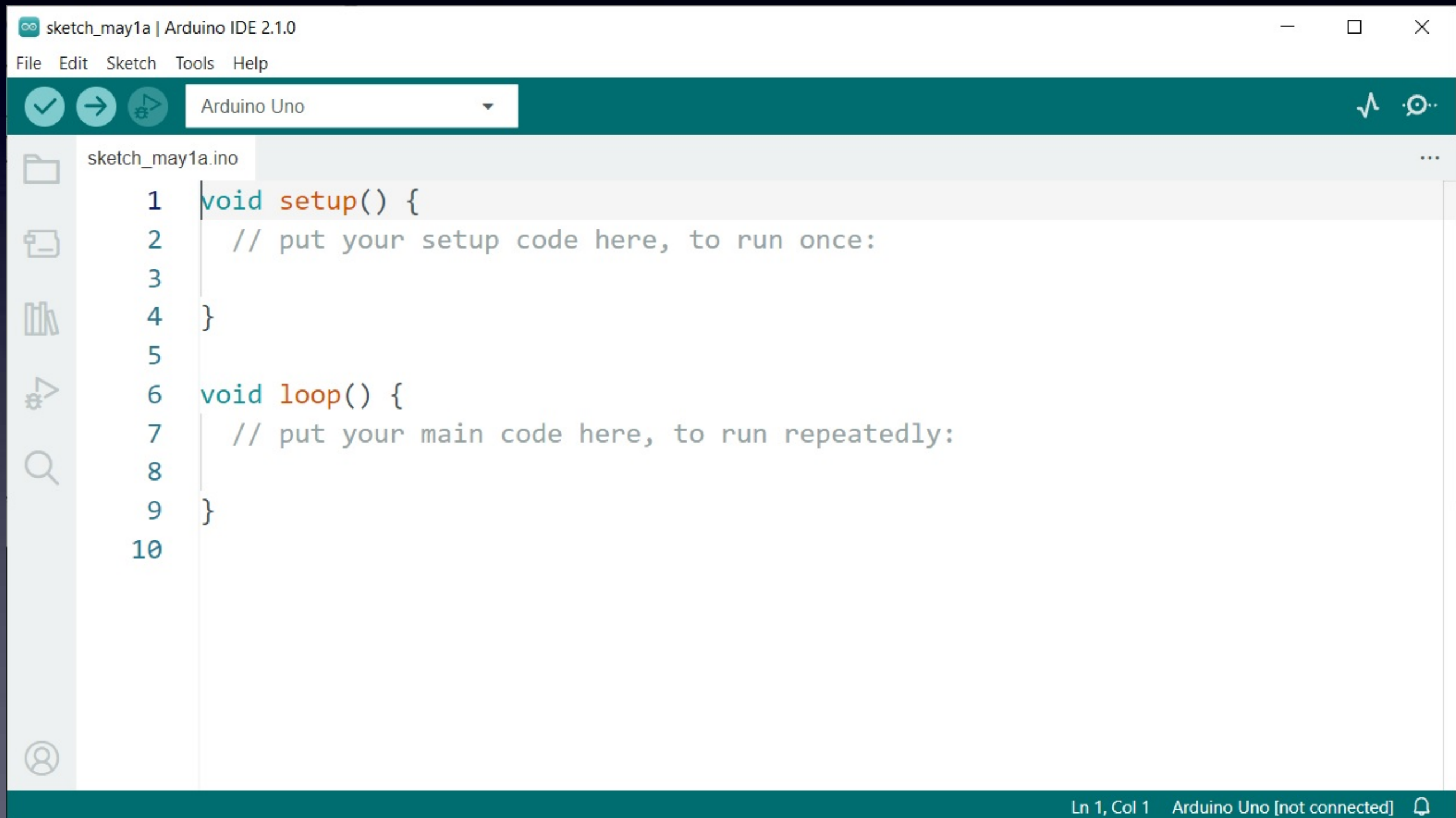


no twists



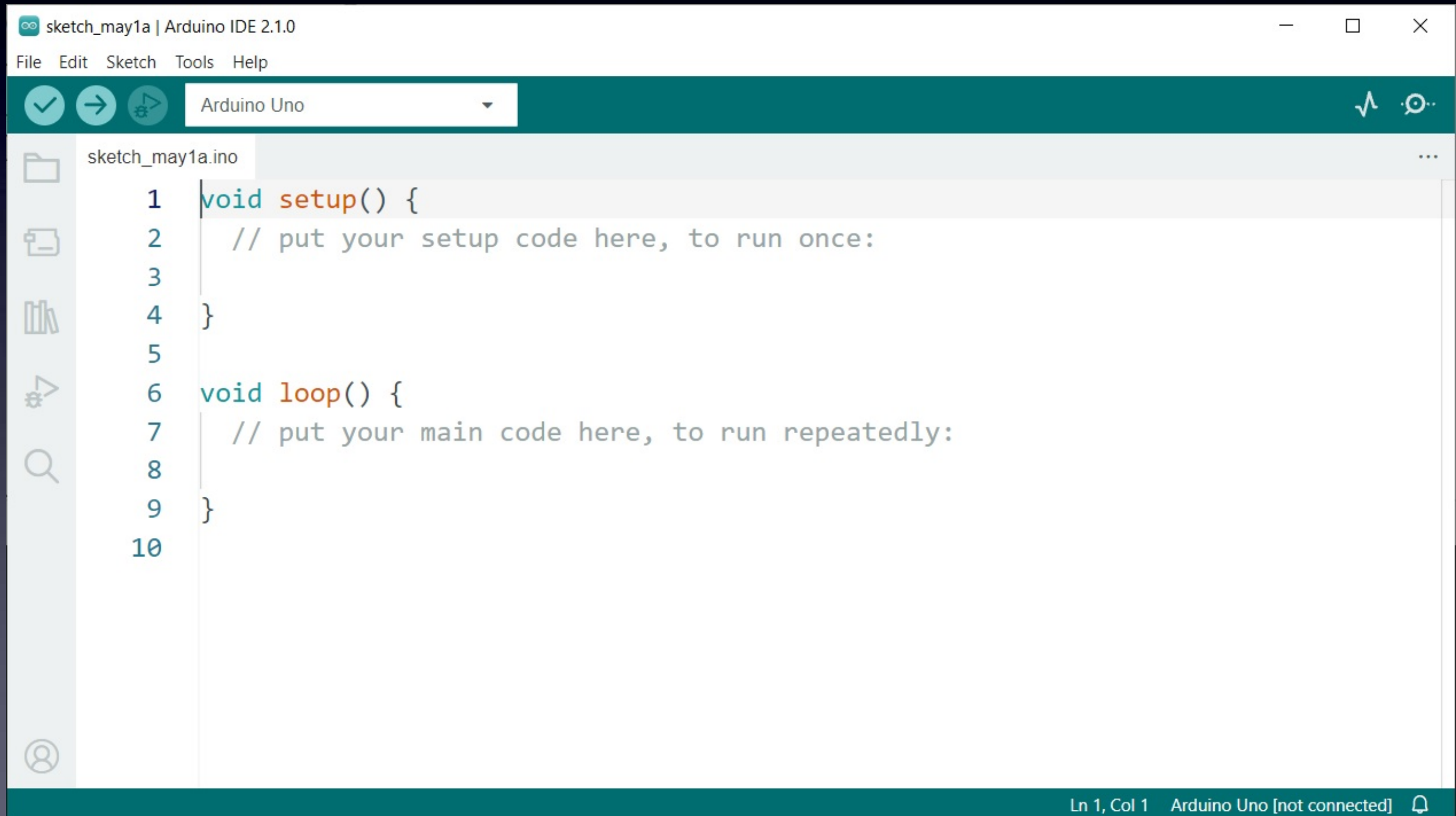
# Arduino

**After you download and install the Arduino software start it, and you will see a screen that looks like this:**



# Arduino

## How to Set Up and Use the Arduino Software



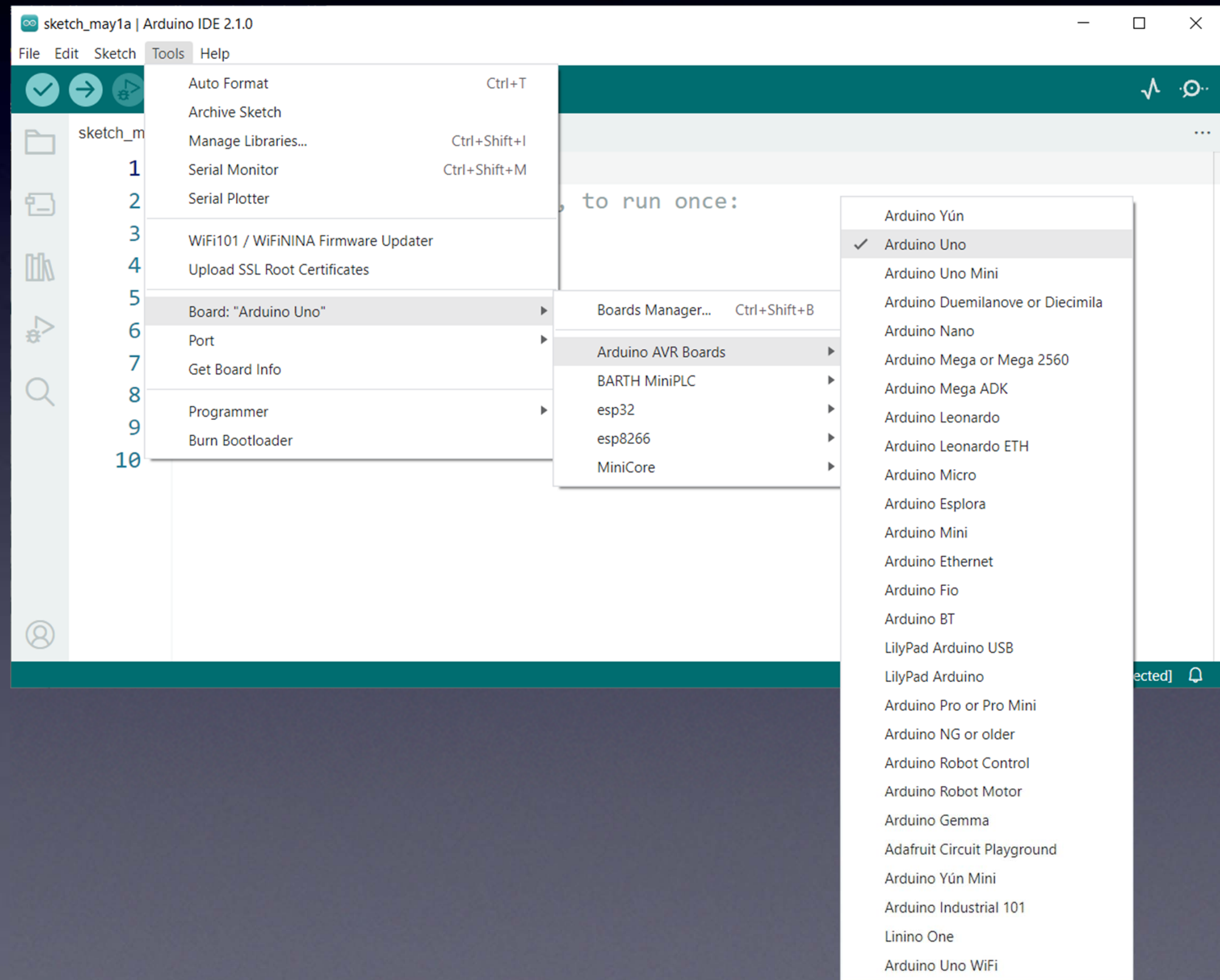


# Arduino

The **first time** you start your Arduino software you need to do **two things** to set things up:

(1)  
Choose “Uno”  
as the Board

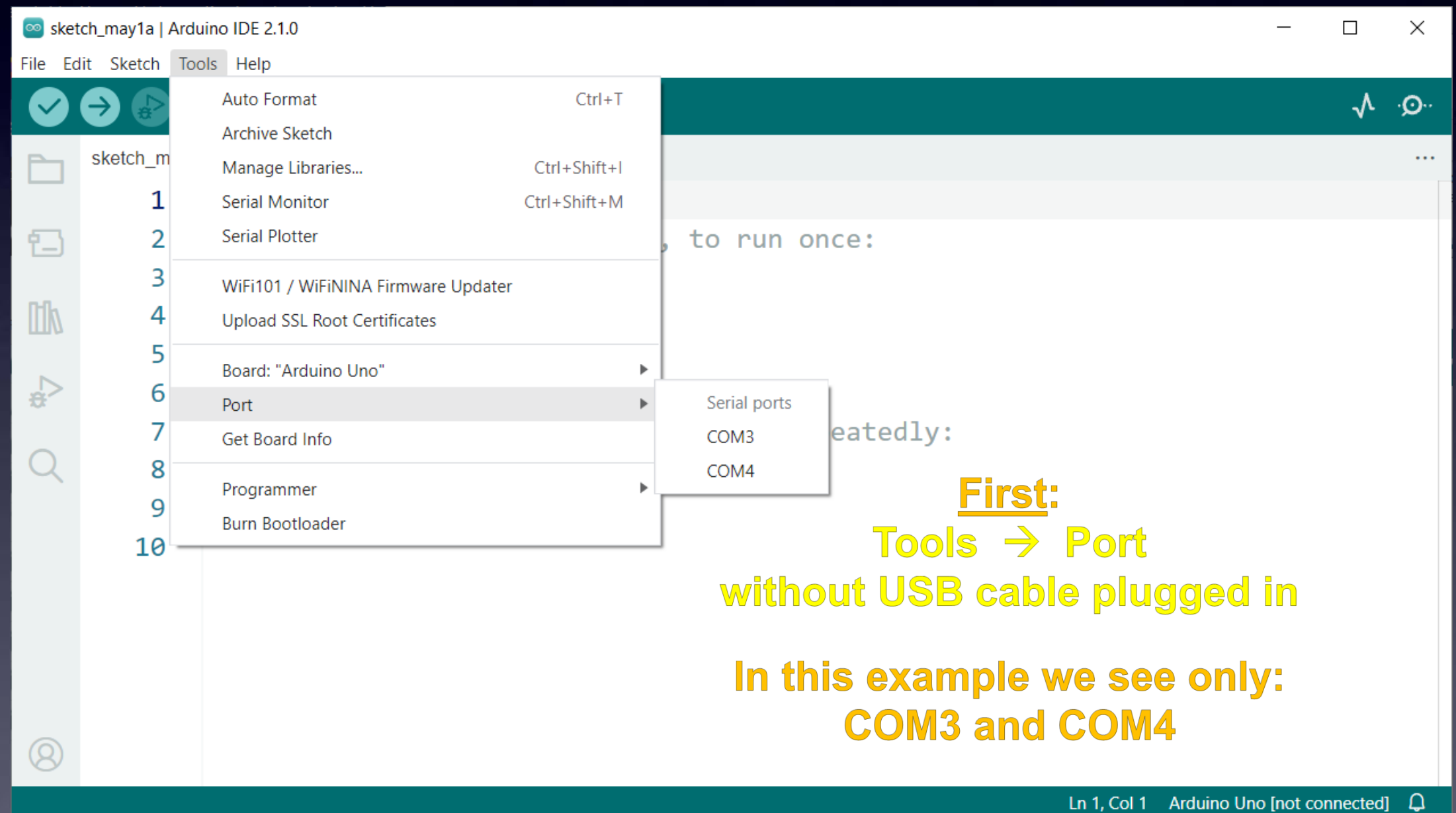
(Your  
U-Do-It-Duino  
acts  
just like  
an  
Arduino Uno board)



# Arduino

The **first time** you start your Arduino software you need to do **two things** to set things up:

(2)  
**Choose the Port**  
(this will be different depending on your Operating System)



**First:**  
**Tools → Port**  
**without USB cable plugged in**

**In this example we see only:**  
**COM3 and COM4**



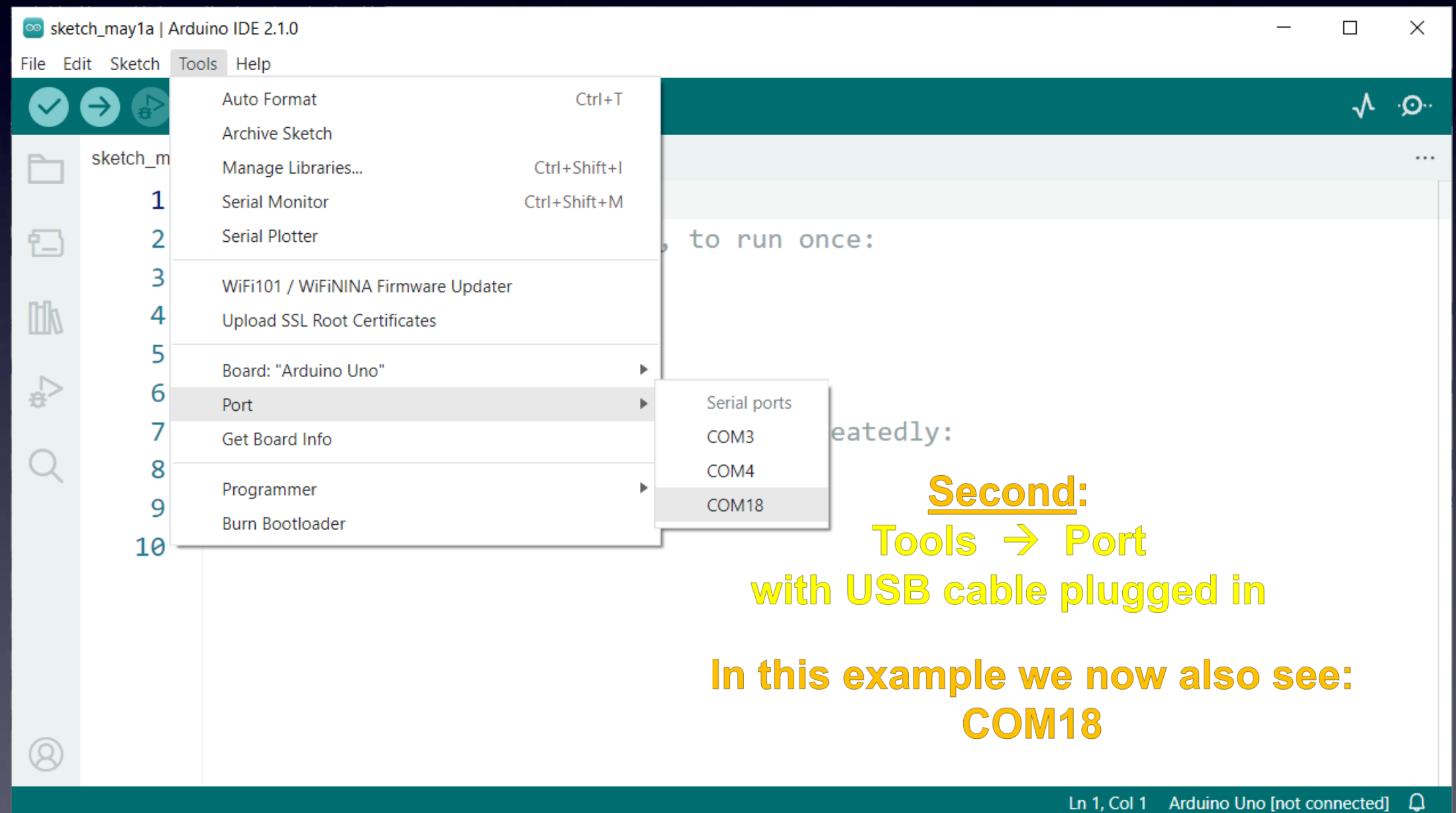
# Arduino

The **first time** you start your Arduino software you need to do **two things** to set things up:

(2)

**Choose the Port**  
(this will be different depending on your Operating System)

(After installing the driver for your USB-Serial cable and plugging it in your operating system will see a serial port and it appears here.)

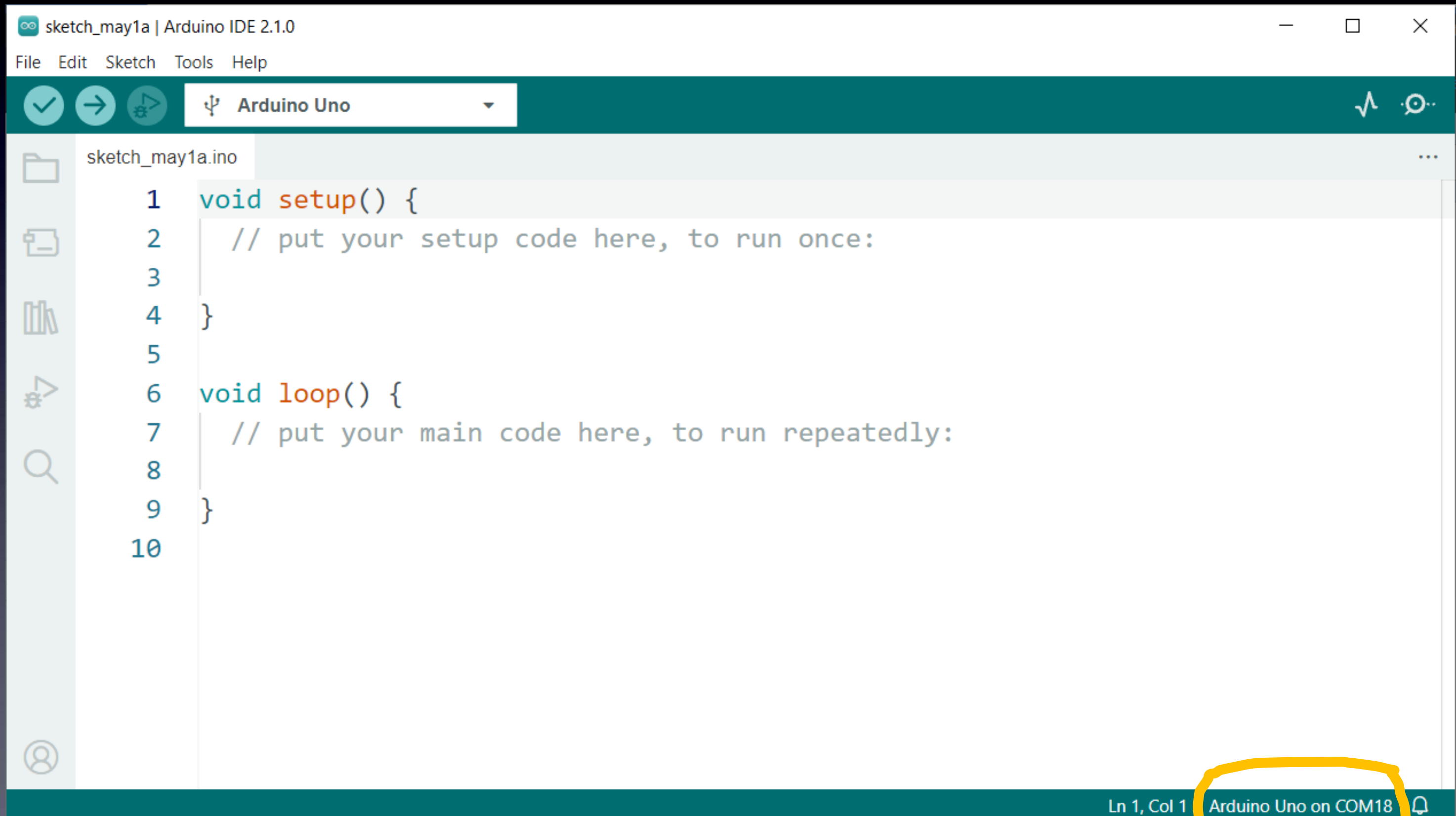


**Second:**  
**Tools → Port**  
with USB cable plugged in

In this example we now also see:  
**COM18**

# Arduino

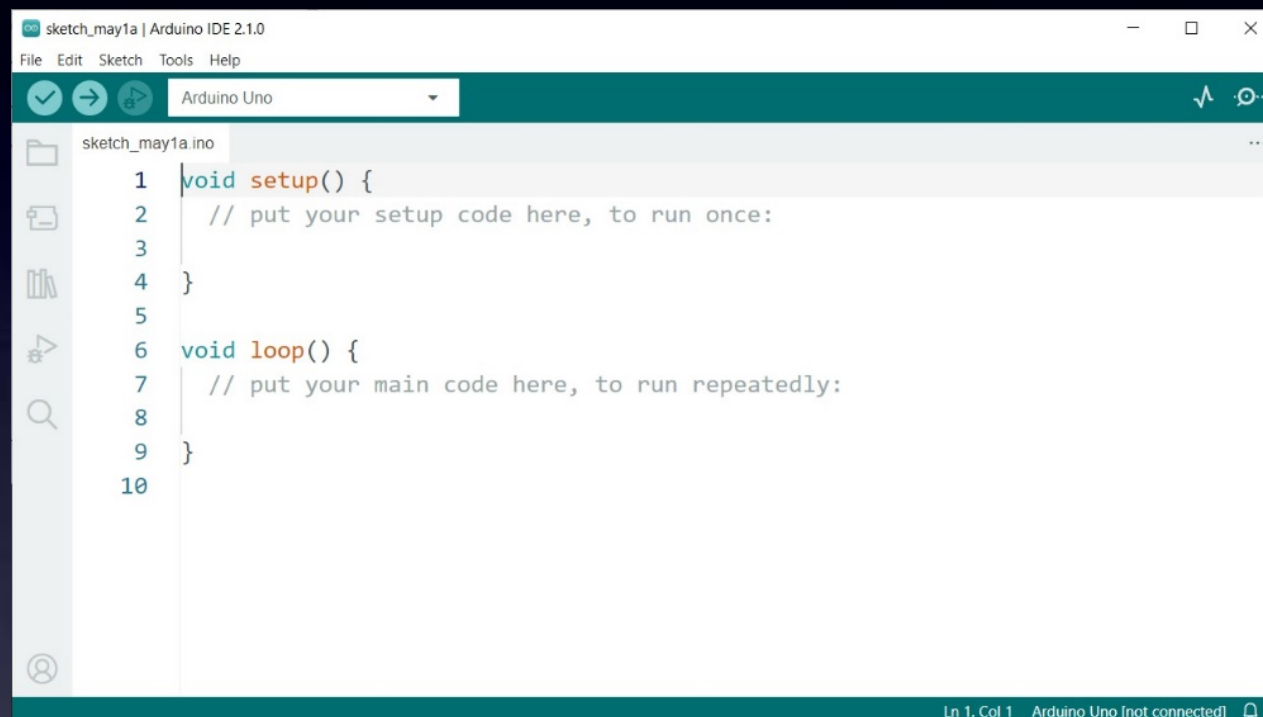
**Your Arduino software is now ready to program your U-Do-It-Duino !**





# Arduino

Designed for non-geeky artists



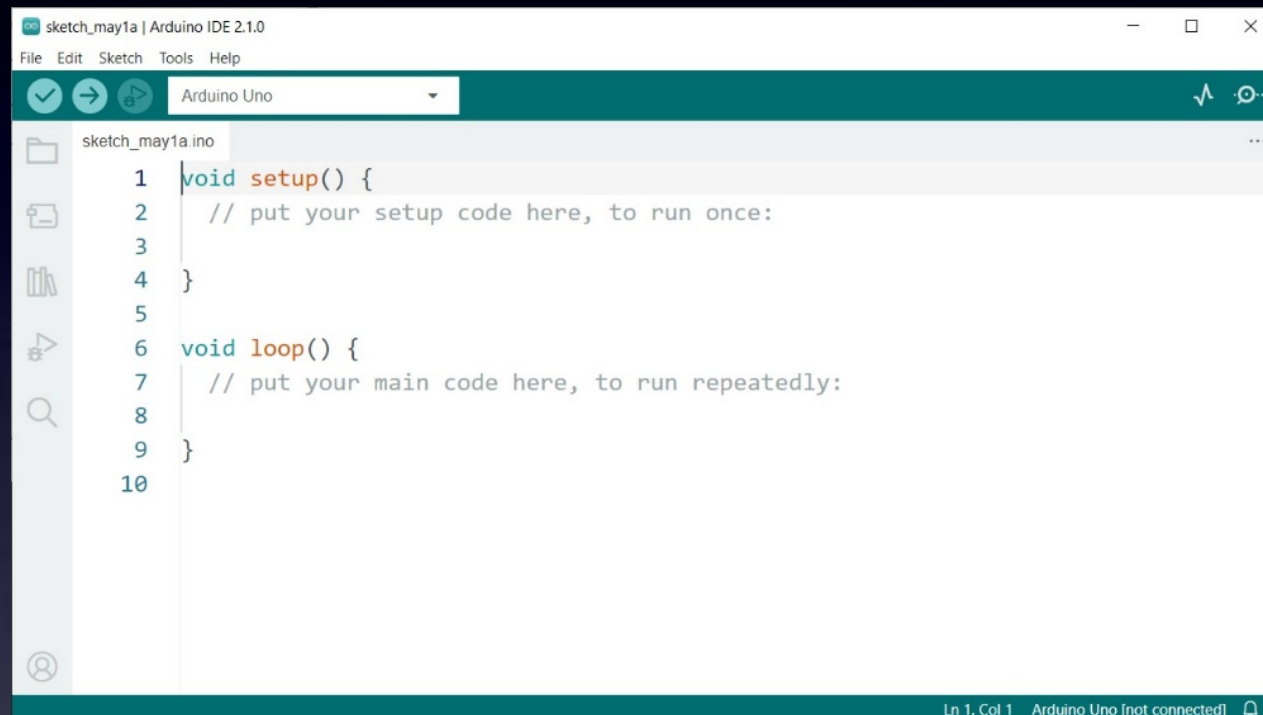
```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }  
10
```

“Sketch” :

an Arduino program

# Arduino

Designed for non-geeky artists



```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }  
10
```

The Arduino language :

“Wiring”

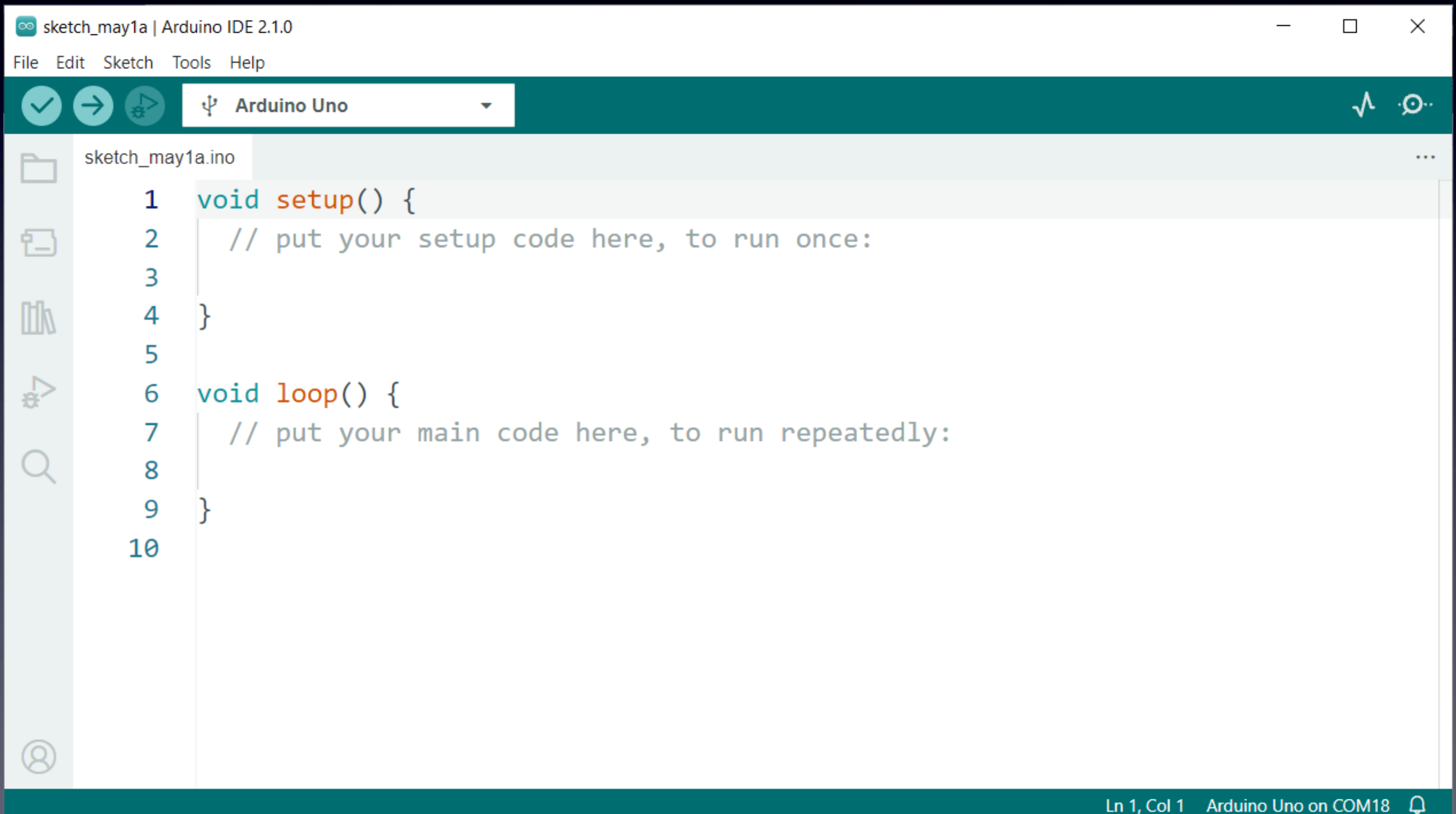
(actually C/C++)



# Arduino

**Your Arduino software is now ready to program your U-Do-It-Duino !**

**Let's make an LED blink! Hello World**

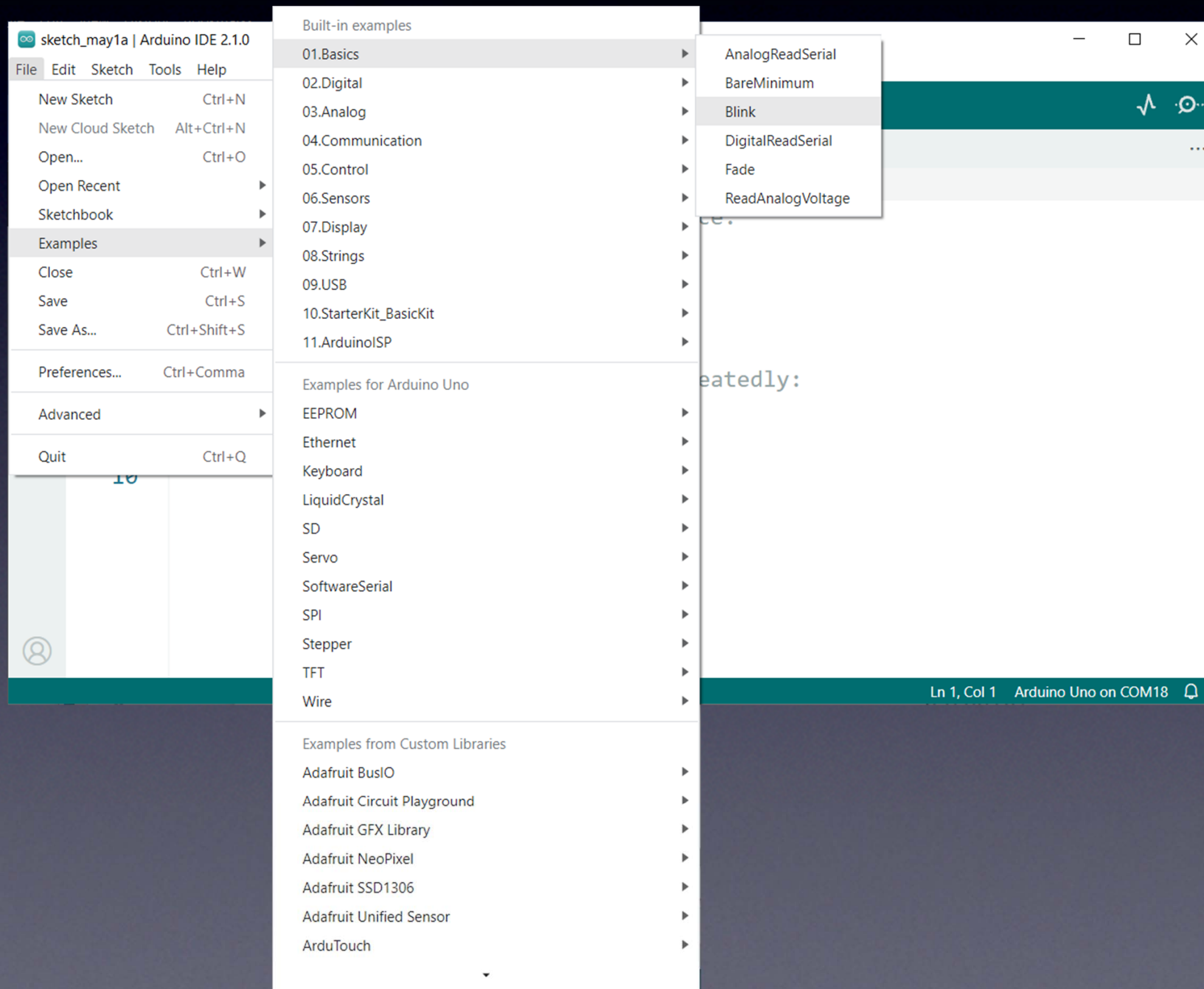


```
sketch_may1a | Arduino IDE 2.1.0
File Edit Sketch Tools Help
[Checkmark] [Right Arrow] [Upload] Arduino Uno [Waveform] [Serial Monitor]
sketch_may1a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
Ln 1, Col 1 Arduino Uno on COM18
```

# Arduino

**Your Arduino software is now ready to program your U-Do-It-Duino !**

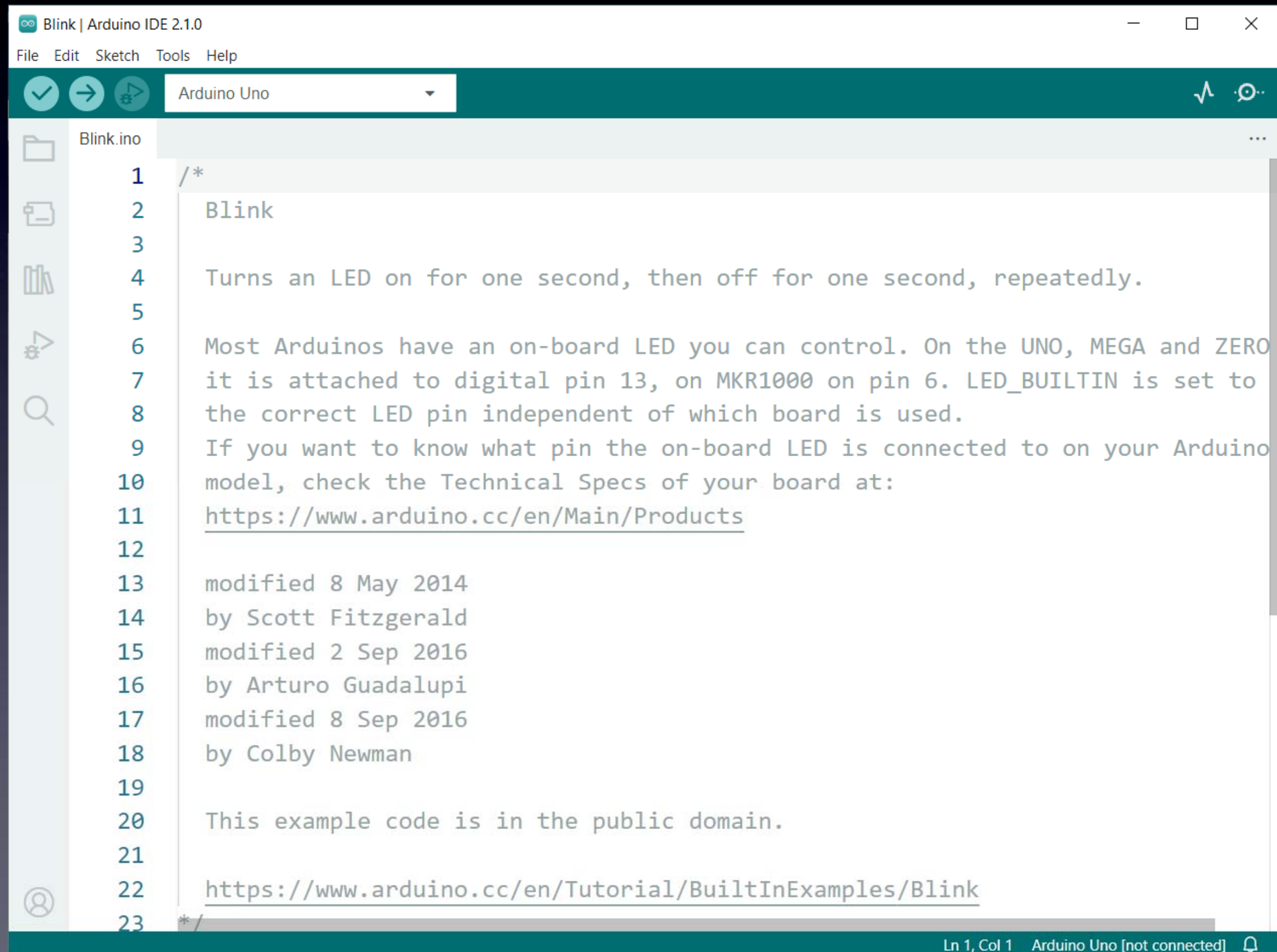
**Let's make an LED blink! Hello World**





# Arduino

Let's make an LED blink! Hello World



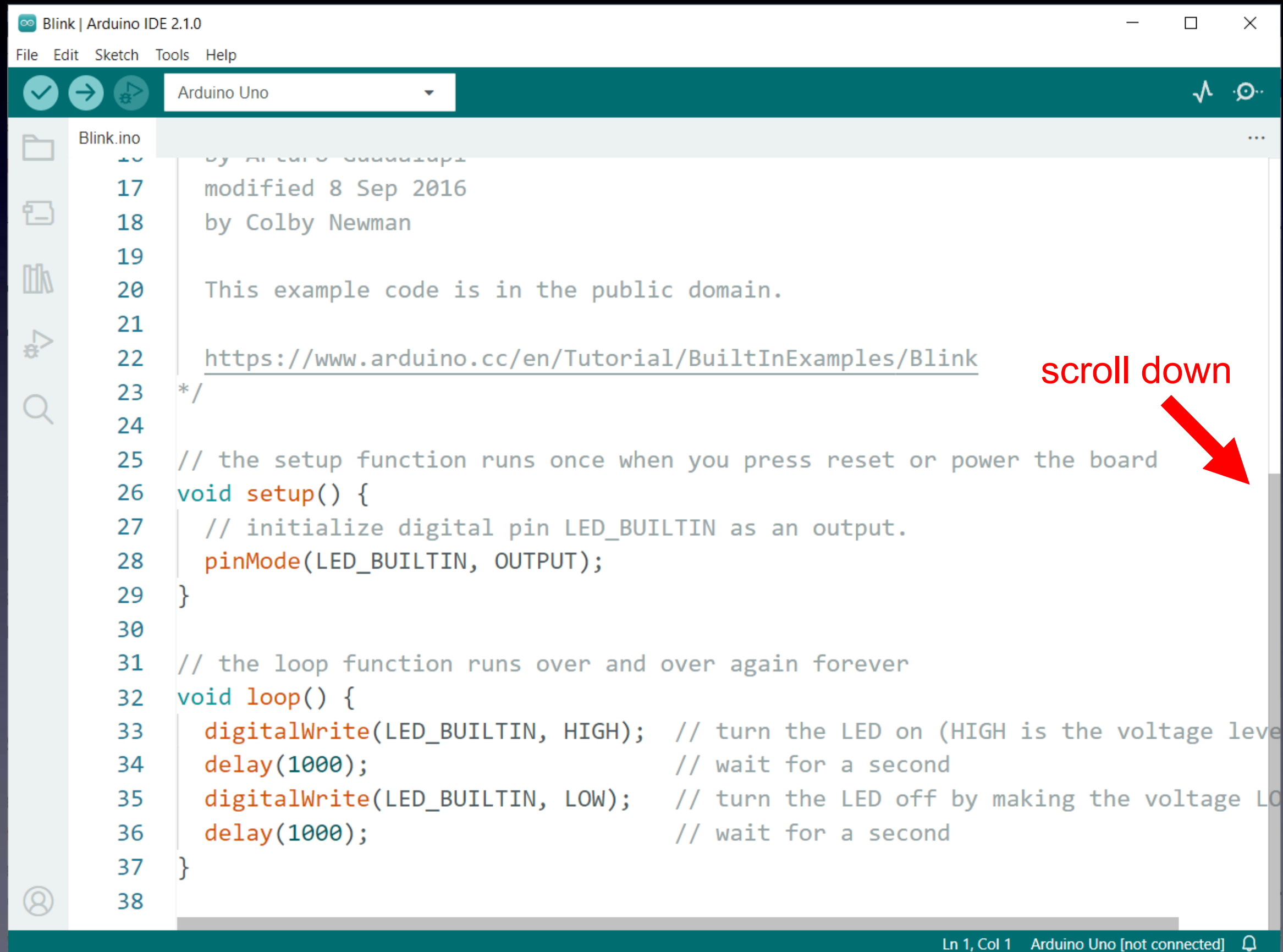
```
Blink | Arduino IDE 2.1.0
File Edit Sketch Tools Help
[Check] [Run] [Upload] Arduino Uno [Waveform] [Serial]

Blink.ino
1  /*
2  Blink
3
4  Turns an LED on for one second, then off for one second, repeatedly.
5
6  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
7  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
8  the correct LED pin independent of which board is used.
9  If you want to know what pin the on-board LED is connected to on your Arduino
10 model, check the Technical Specs of your board at:
11 https://www.arduino.cc/en/Main/Products
12
13 modified 8 May 2014
14 by Scott Fitzgerald
15 modified 2 Sep 2016
16 by Arturo Guadalupi
17 modified 8 Sep 2016
18 by Colby Newman
19
20 This example code is in the public domain.
21
22 https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23 */
```

Ln 1, Col 1 Arduino Uno [not connected]

# Arduino

Let's make an LED blink! Hello World



Blink | Arduino IDE 2.1.0

File Edit Sketch Tools Help

Arduino Uno

Blink.ino

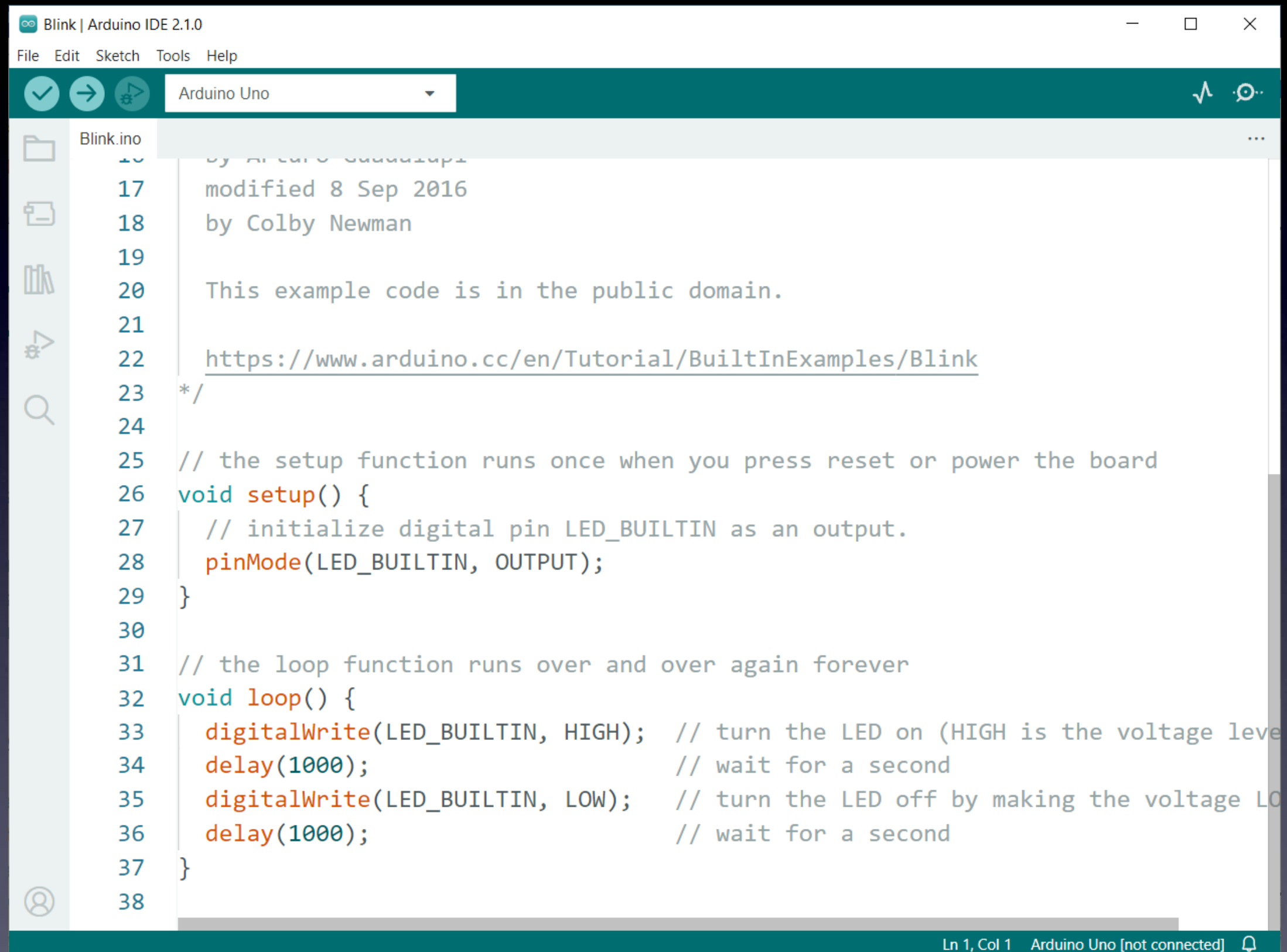
```
16  by Arduino.cc  
17  modified 8 Sep 2016  
18  by Colby Newman  
19  
20  This example code is in the public domain.  
21  
22  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink  
23  */  
24  
25  // the setup function runs once when you press reset or power the board  
26  void setup() {  
27    // initialize digital pin LED_BUILTIN as an output.  
28    pinMode(LED_BUILTIN, OUTPUT);  
29  }  
30  
31  // the loop function runs over and over again forever  
32  void loop() {  
33    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
34    delay(1000); // wait for a second  
35    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
36    delay(1000); // wait for a second  
37  }  
38
```

scroll down

Ln 1, Col 1 Arduino Uno [not connected]



# How to Hack Arduino Programs (“Sketches”)



```
Blink | Arduino IDE 2.1.0
File Edit Sketch Tools Help
[Icons] Arduino Uno

Blink.ino
16  by Arduino Corporation
17  modified 8 Sep 2016
18  by Colby Newman
19
20  This example code is in the public domain.
21
22  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23  */
24
25  // the setup function runs once when you press reset or power the board
26  void setup() {
27    // initialize digital pin LED_BUILTIN as an output.
28    pinMode(LED_BUILTIN, OUTPUT);
29  }
30
31  // the loop function runs over and over again forever
32  void loop() {
33    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34    delay(1000);                      // wait for a second
35    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
36    delay(1000);                      // wait for a second
37  }
38
```

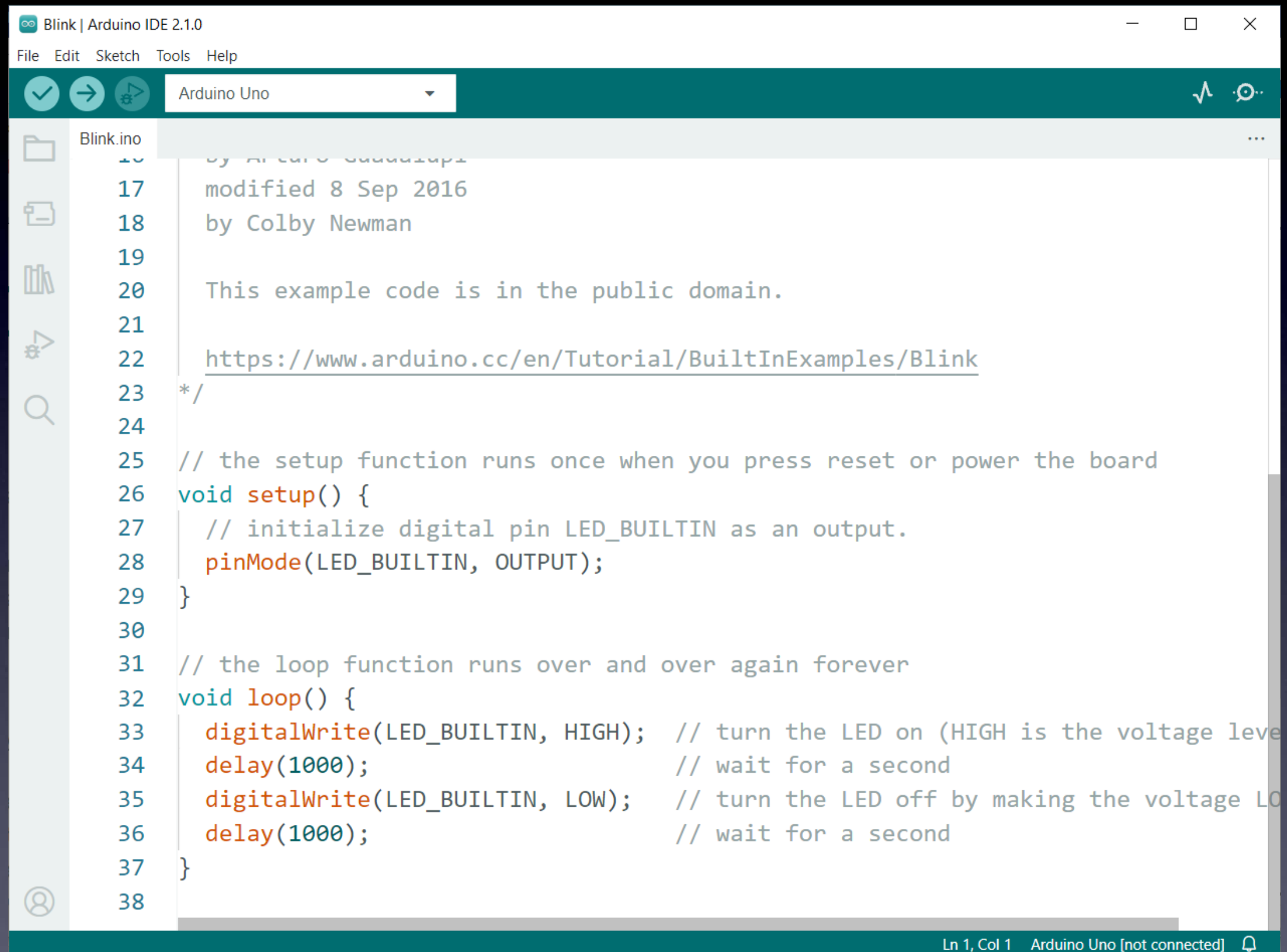
Ln 1, Col 1 Arduino Uno [not connected]

# How to Hack Arduino Programs (“Sketches”)

Many ways!

Here are just a few:

- Change blink rates
- External LED
- External motor
- External speaker
- External LED on Solderless breadboard
- More complex projects on Solderless breadboard



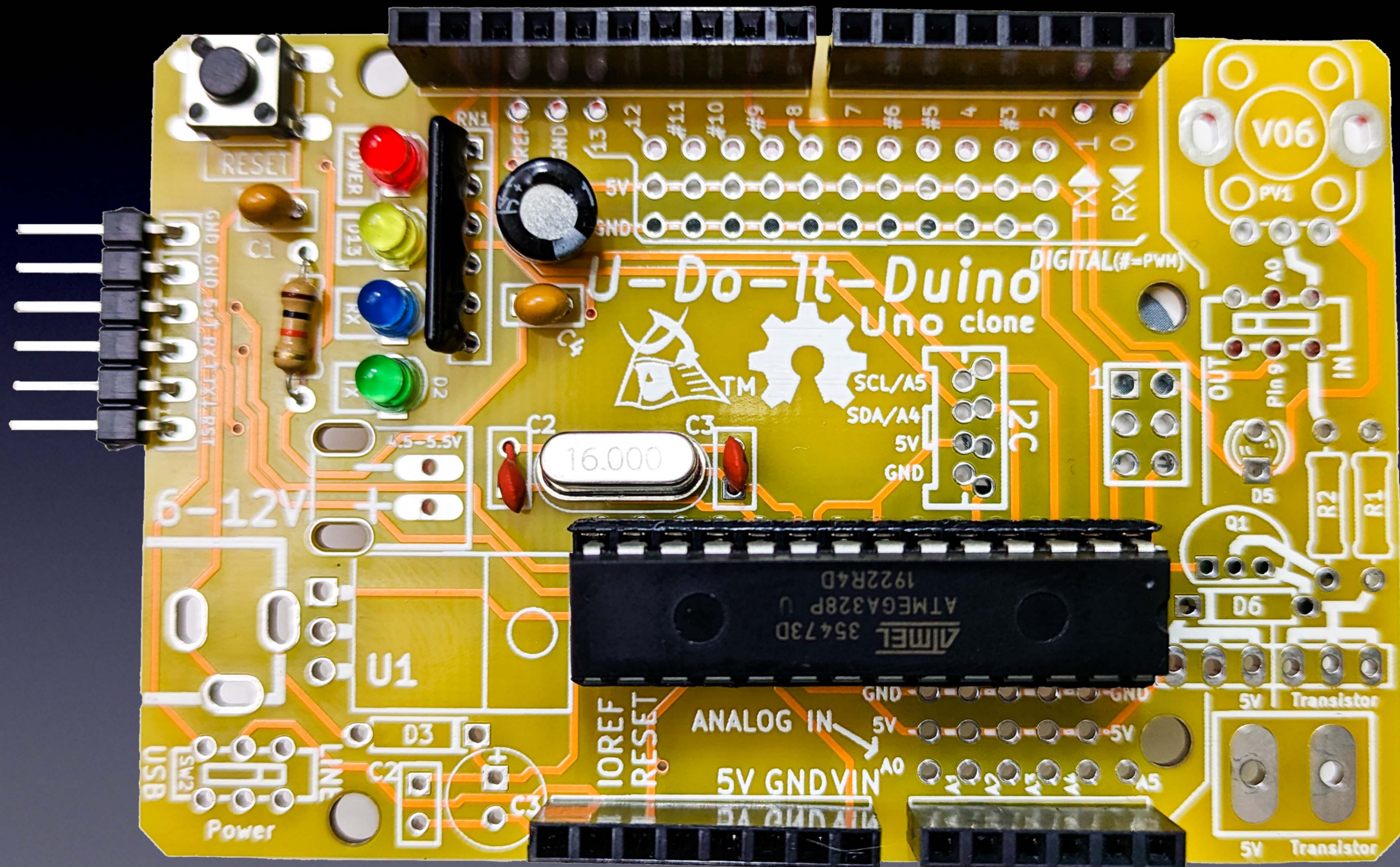
```
Blink | Arduino IDE 2.1.0
File Edit Sketch Tools Help
[Icons] Arduino Uno

Blink.ino
16 by Arduino Staff
17 modified 8 Sep 2016
18 by Colby Newman
19
20 This example code is in the public domain.
21
22 https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
38

Ln 1, Col 1 Arduino Uno [not connected]
```



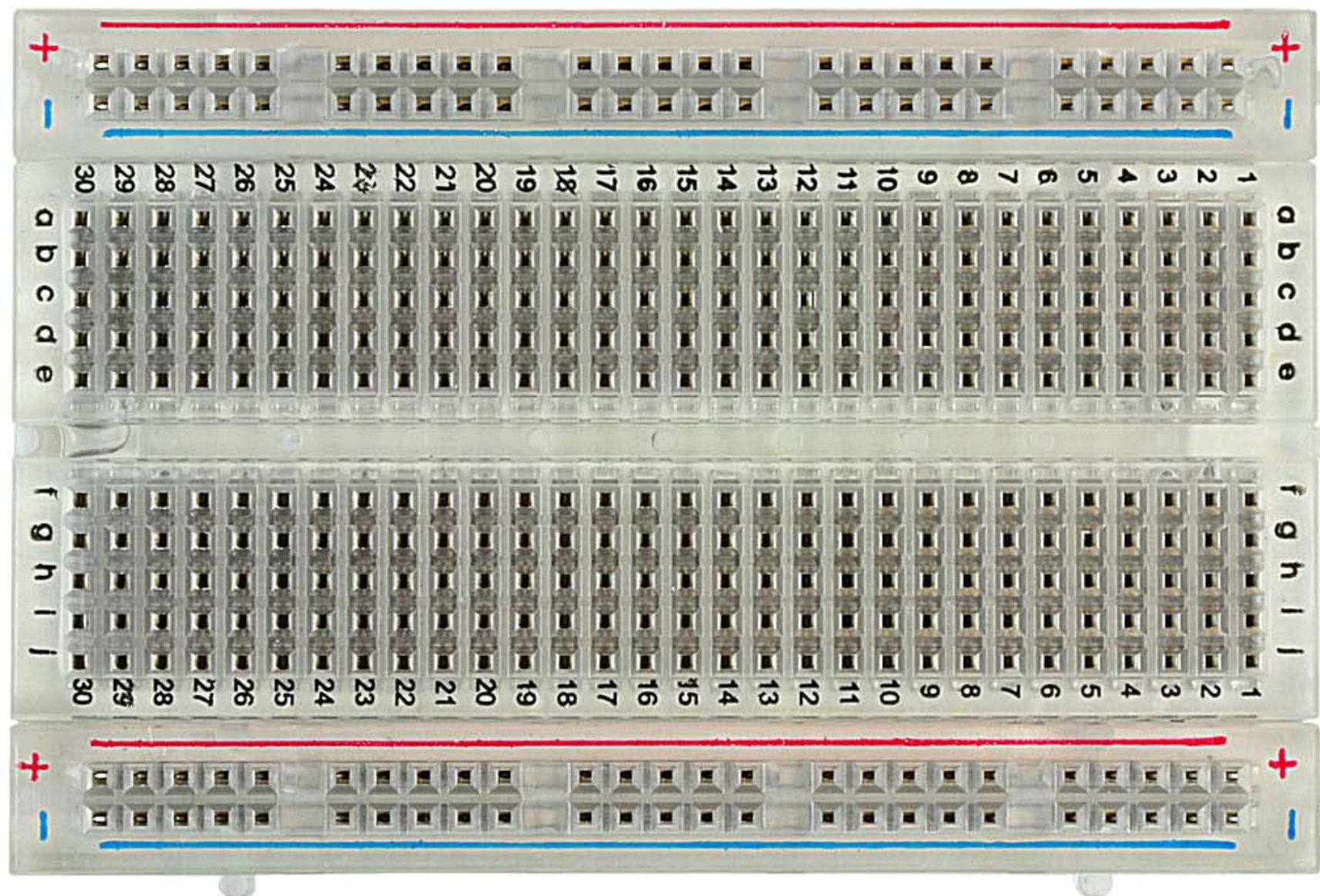
# How to Hack Arduino Programs (“Sketches”)





# How to Use Solderless Breadboards

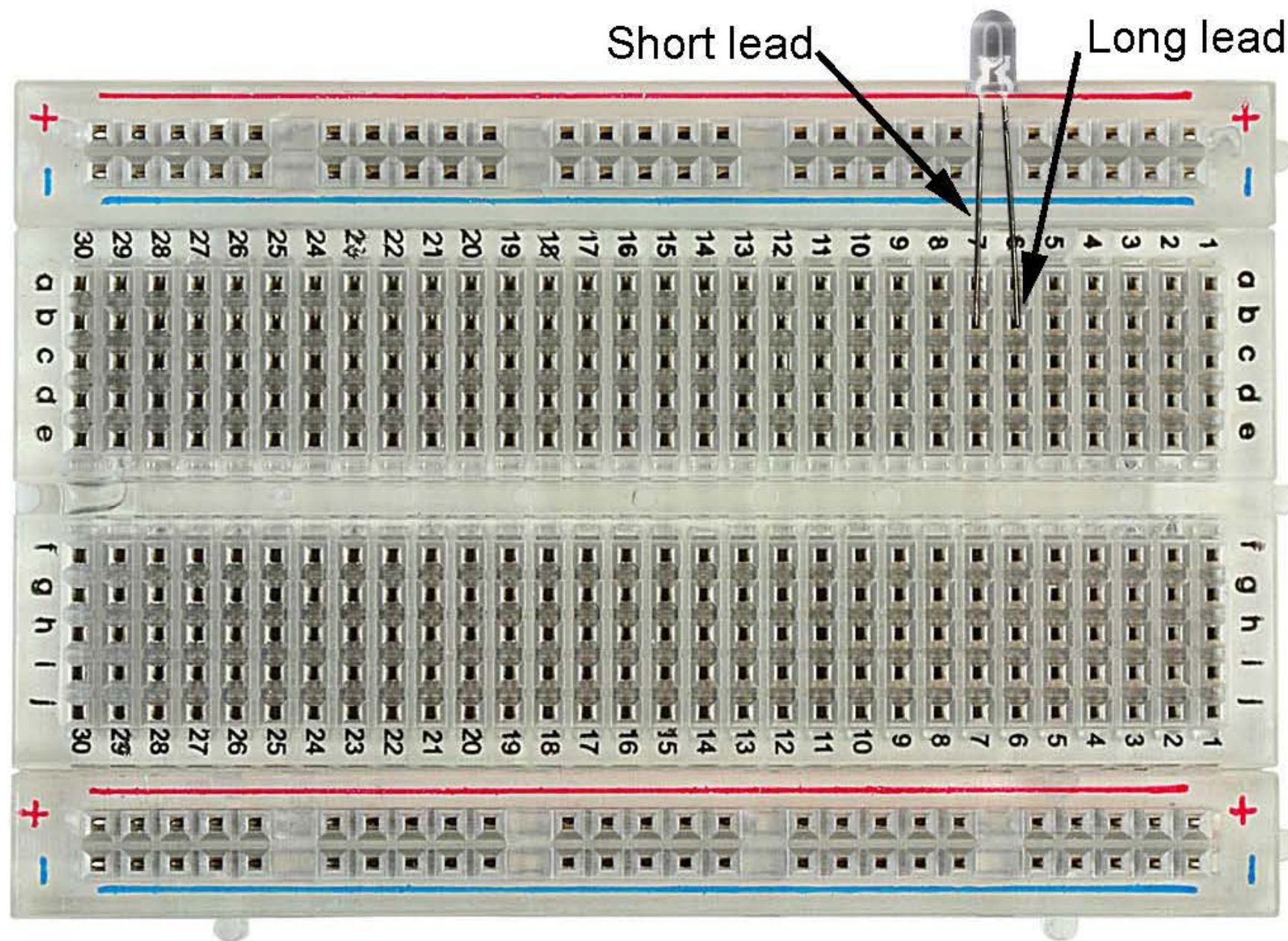
## Solderless Breadboard





# How to Use Solderless Breadboards

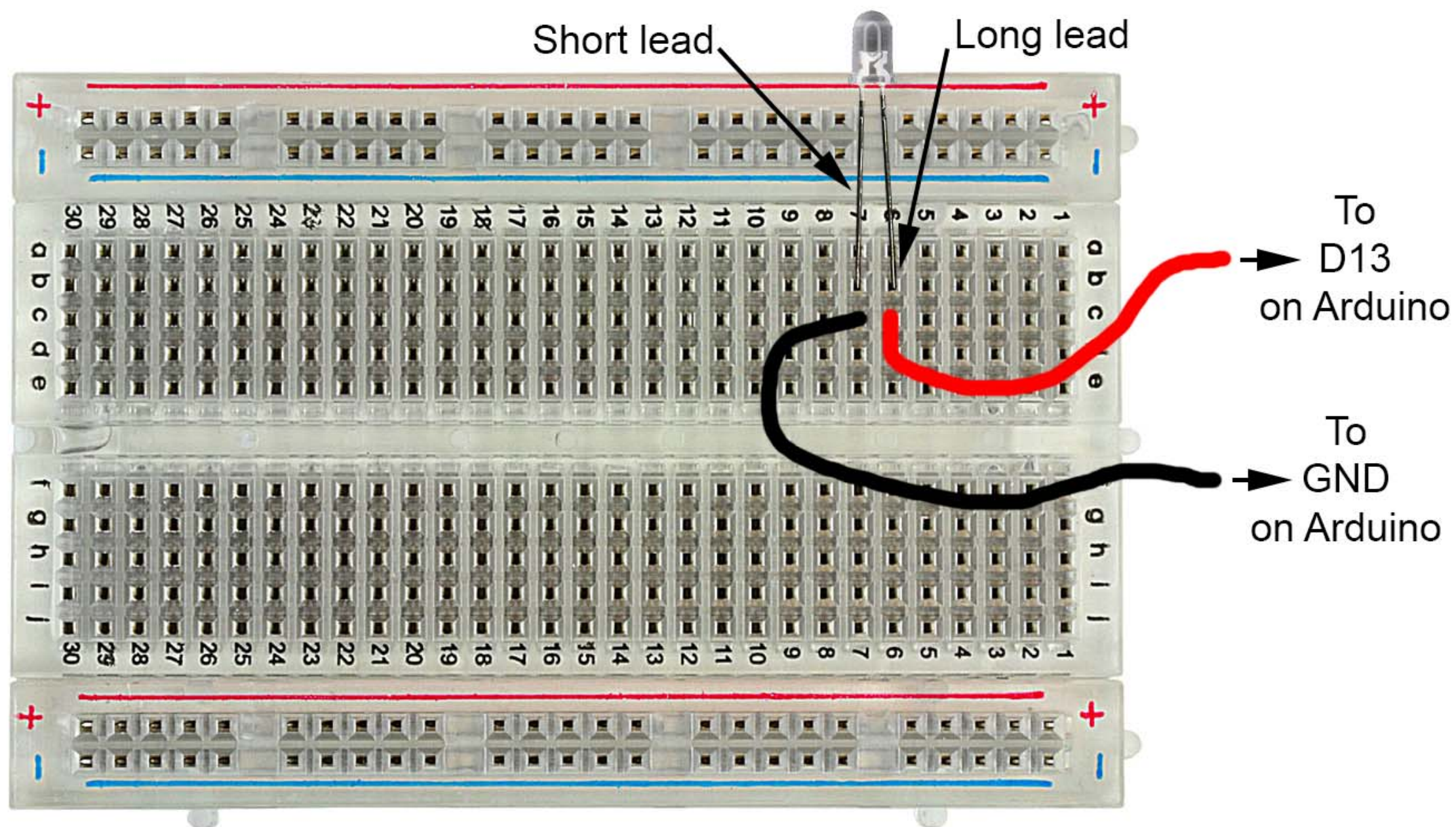
## Solderless Breadboard with LED





# How to Use Solderless Breadboards

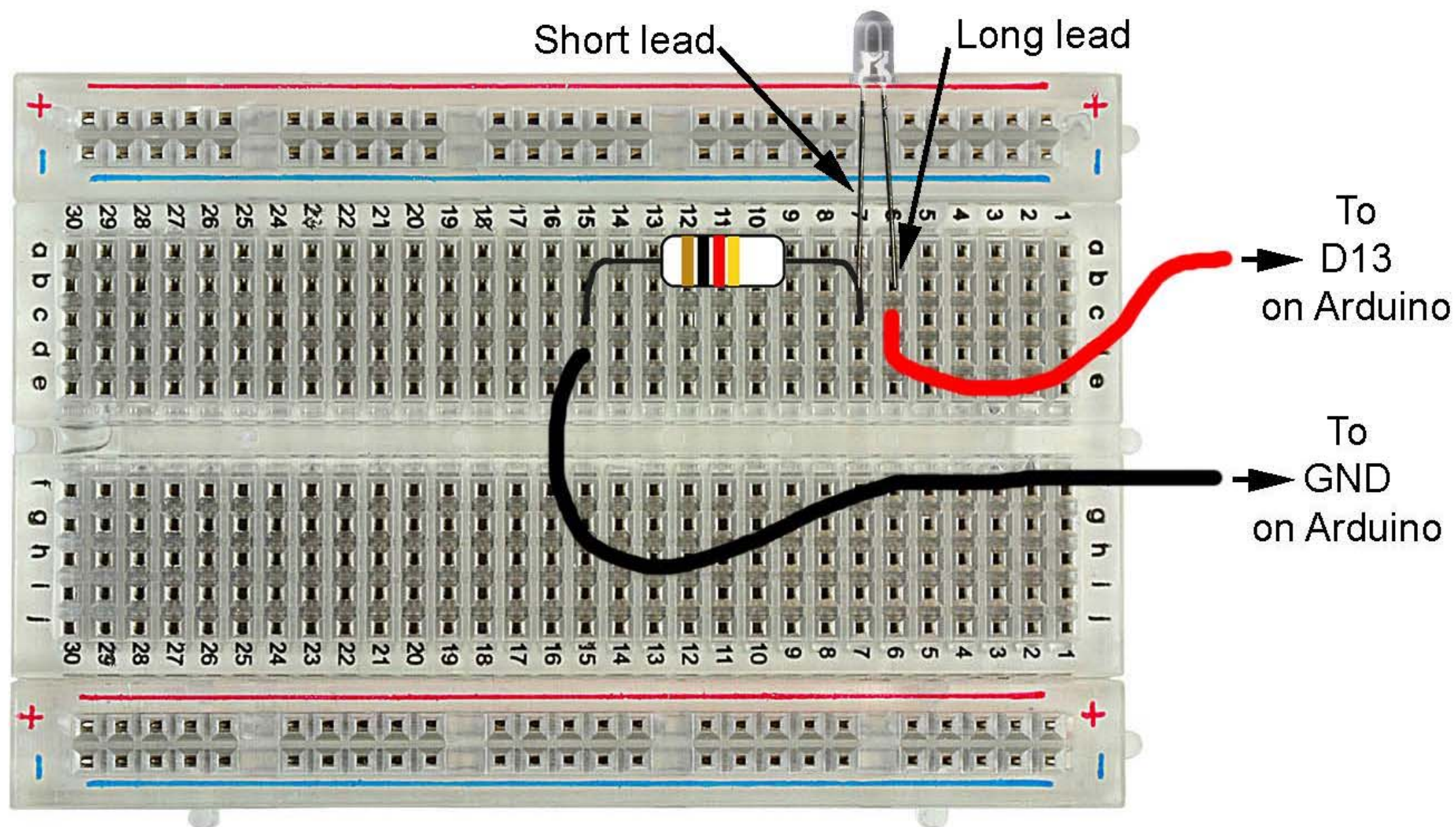
## Solderless Breadboard with LED and wires



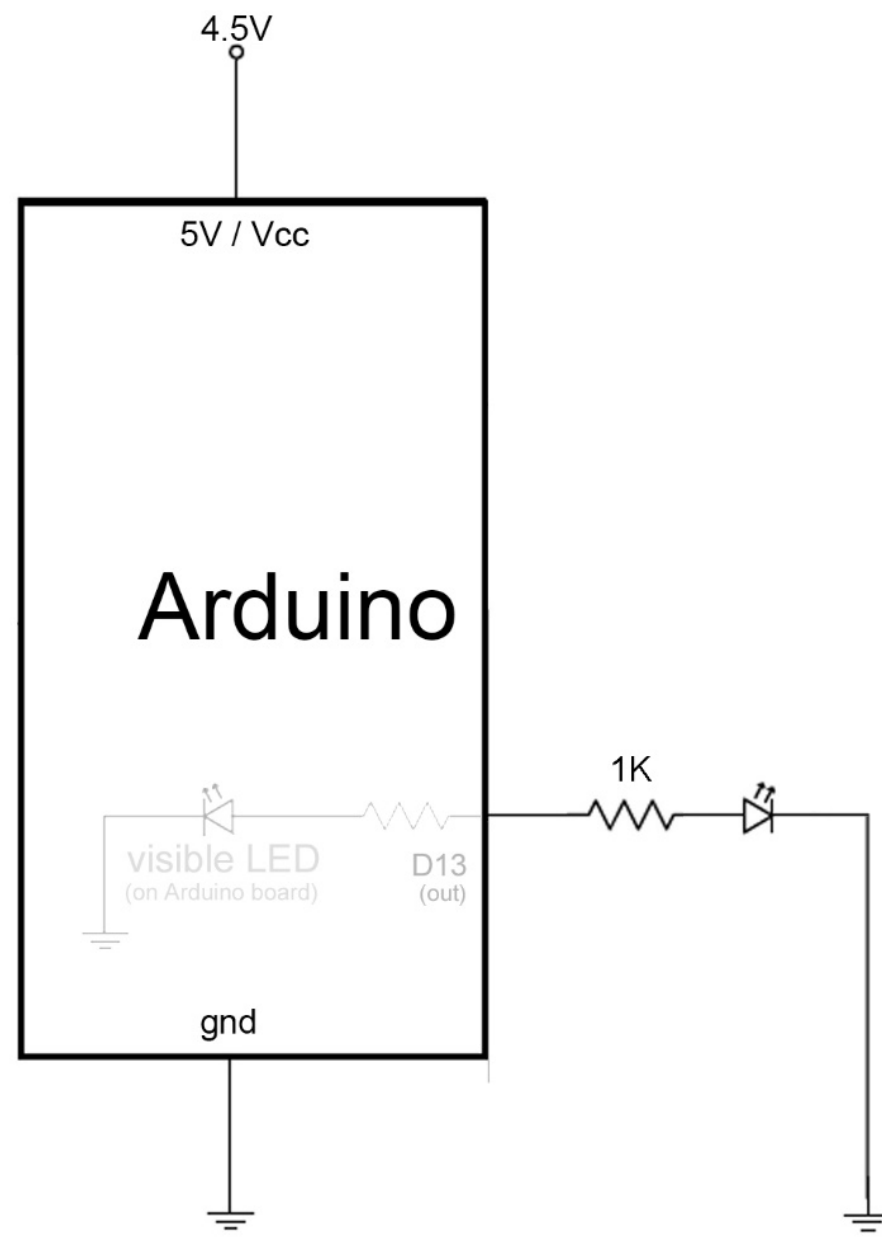


# How to Use Solderless Breadboards

## Solderless Breadboard with LED and Resistor and wires



# How to Read a Schematic



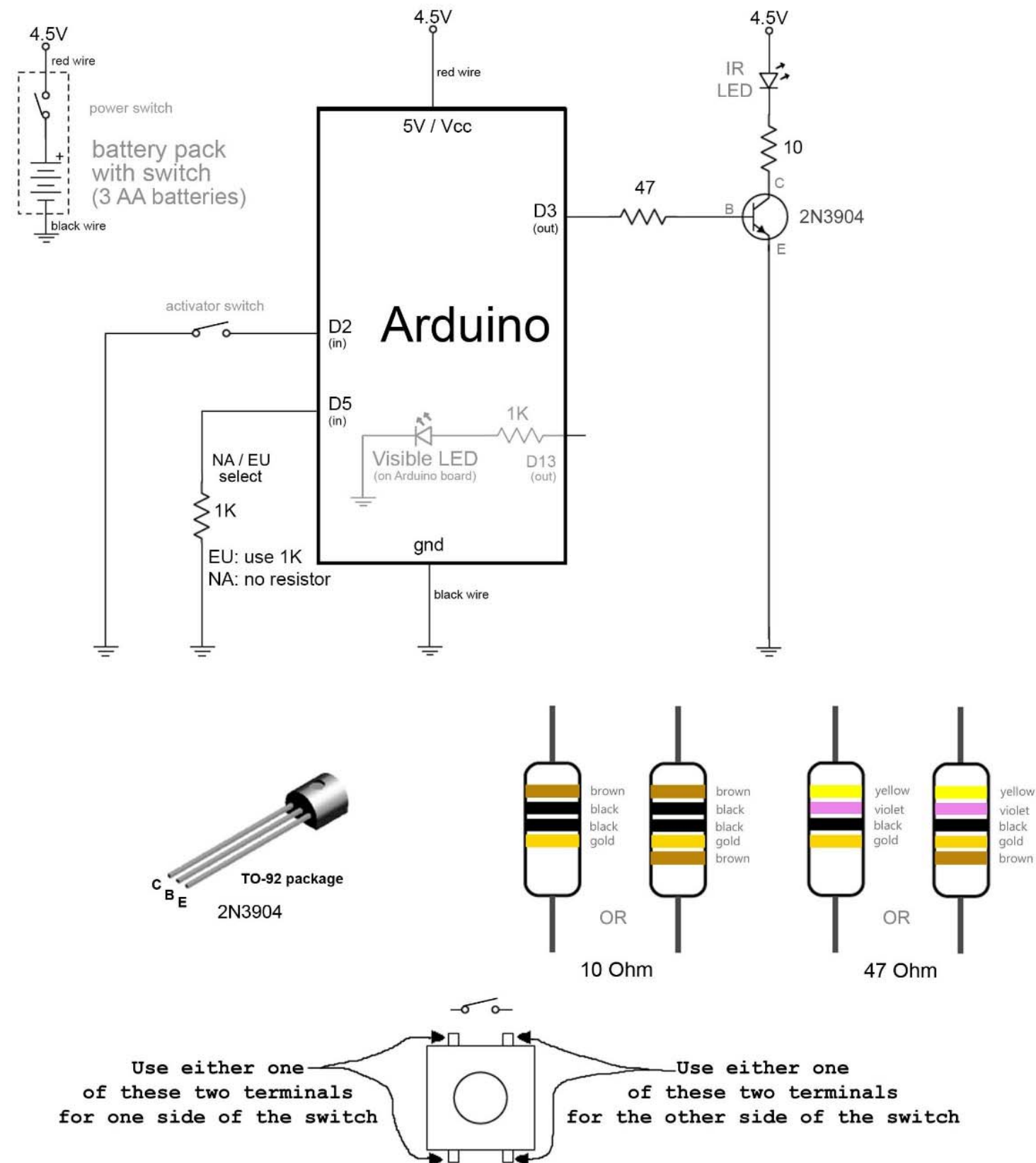


# How to Read a Schematic

## Arduino For Total Newbies

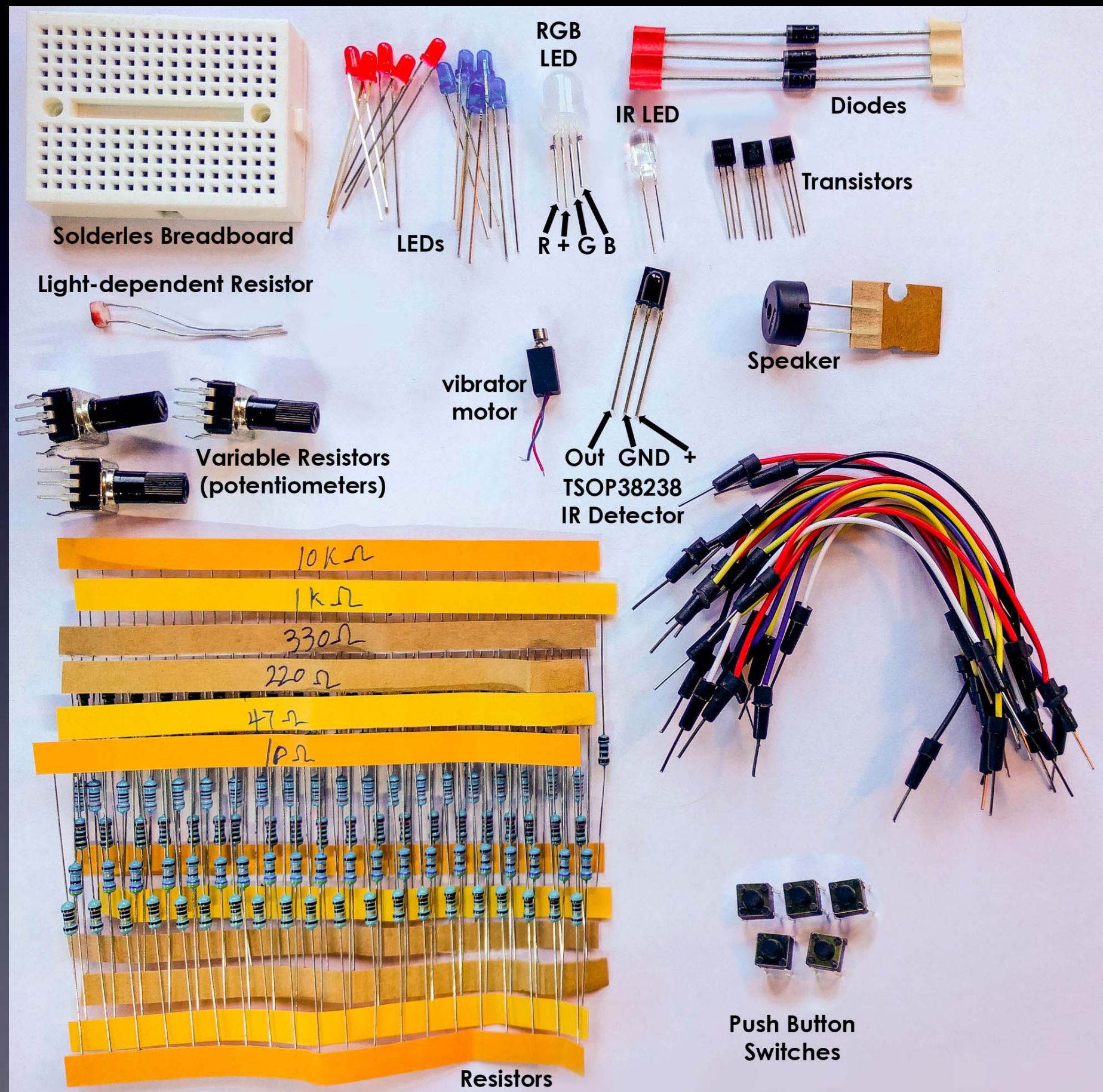
4-Sep-2015

Mitch Altman (original TV-B-Gone hardware and firmware, modified TV-B-Gone Arduino design)  
Limore Fried (firmware modifications, kit design)  
Ken Shirriff (original modifications for Arduino)  
Johannes Schneemann (documentation)



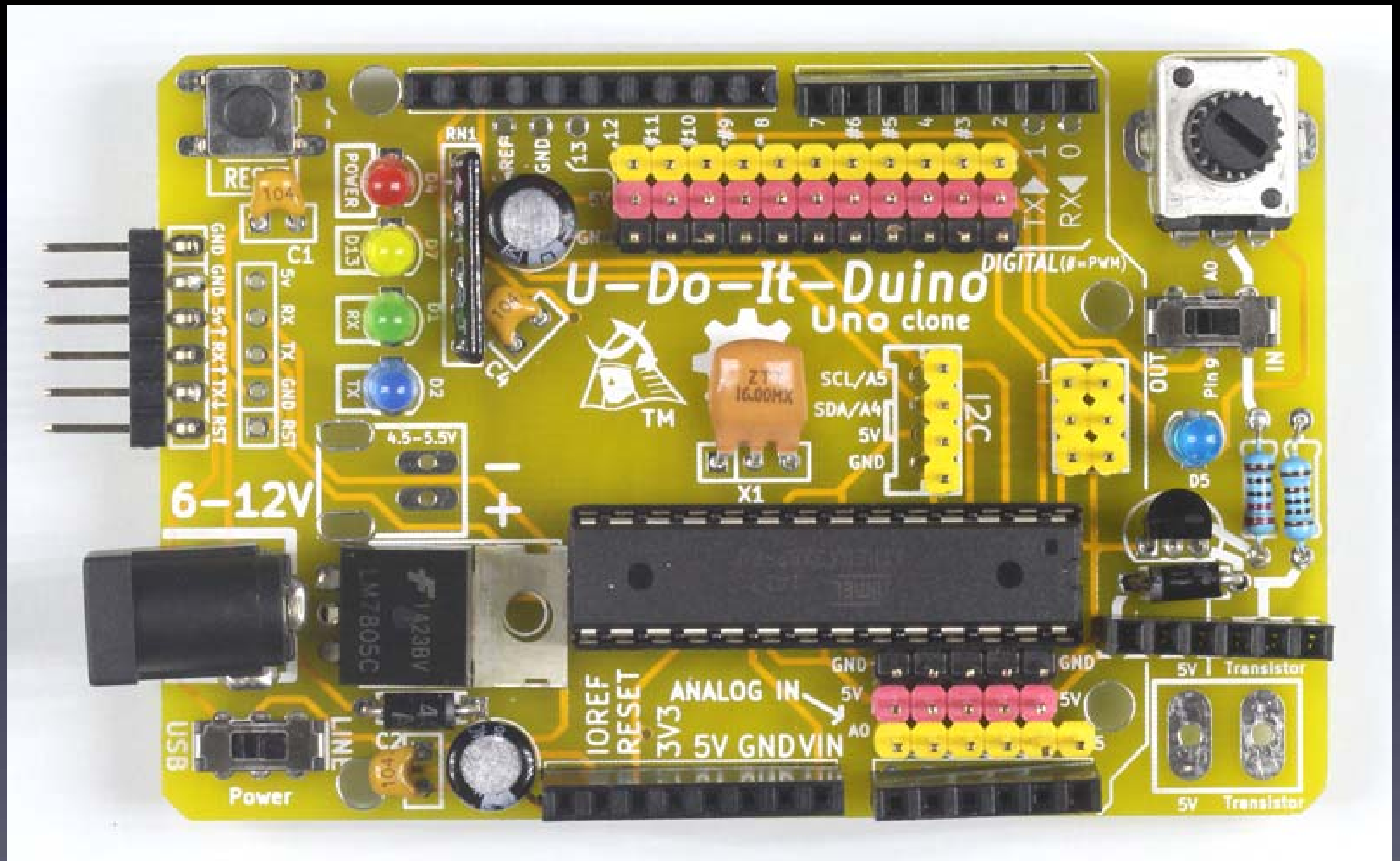


# Parts Pack Contents





# If you want to:



[https://cornfieldelectronics.com/cfe/u do it duino complete.php](https://cornfieldelectronics.com/cfe/u%20do%20it%20duino%20complete.php)

Please Remember:

to

**Wash your hands**



**I have these  
Toolkits  
for sale**

# Tools



# Arduino For Total Newbies

## *w/ TV-B-Gone as example project*

Mitch Altman

Chief Scientist, **Cornfield Electronics**, San Francisco, CA

Inventor of **TV-B-Gone** universal remote controls

Co-founder of **3Ware** (successful Silicon Valley startup)

Pioneer of **VR** (in the mid-1980s)

Founding mentor at **HAX** (1<sup>st</sup> and biggest hardware accelerator)

Co-founder of **Noisebridge** (San Francisco hackerspace)

email: [mitch@CornfieldElectronics.com](mailto:mitch@CornfieldElectronics.com)

site: [www.CornfieldElectronics.com](http://www.CornfieldElectronics.com)

twitter: [@maltman23](https://twitter.com/maltman23)

flickr: [maltman23](https://www.flickr.com/photos/maltman23/)

WeChat: [mitchaltman](#)



CORNFIELD ELECTRONICS