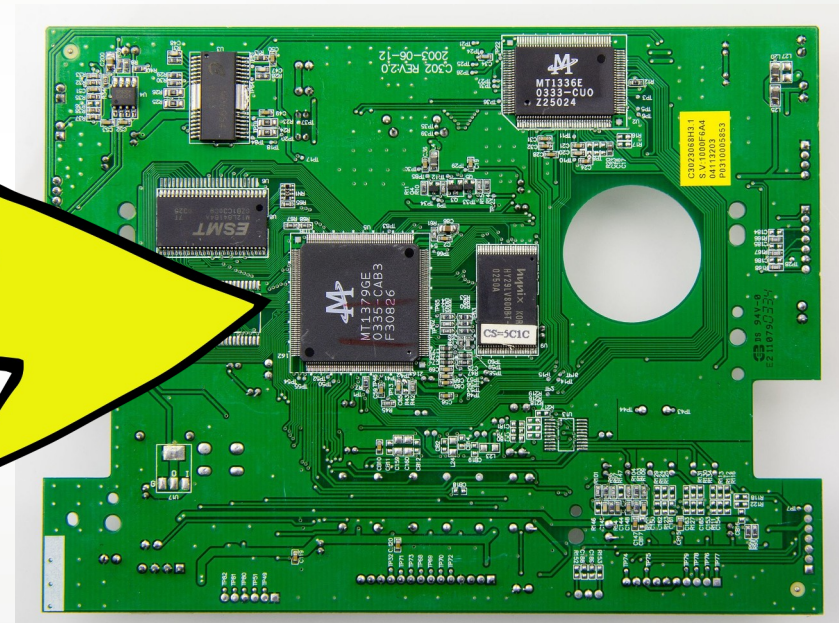
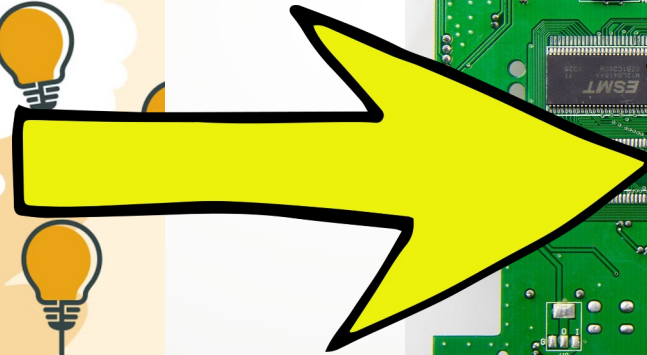


# Cursus Kicad

How to go from idea to product

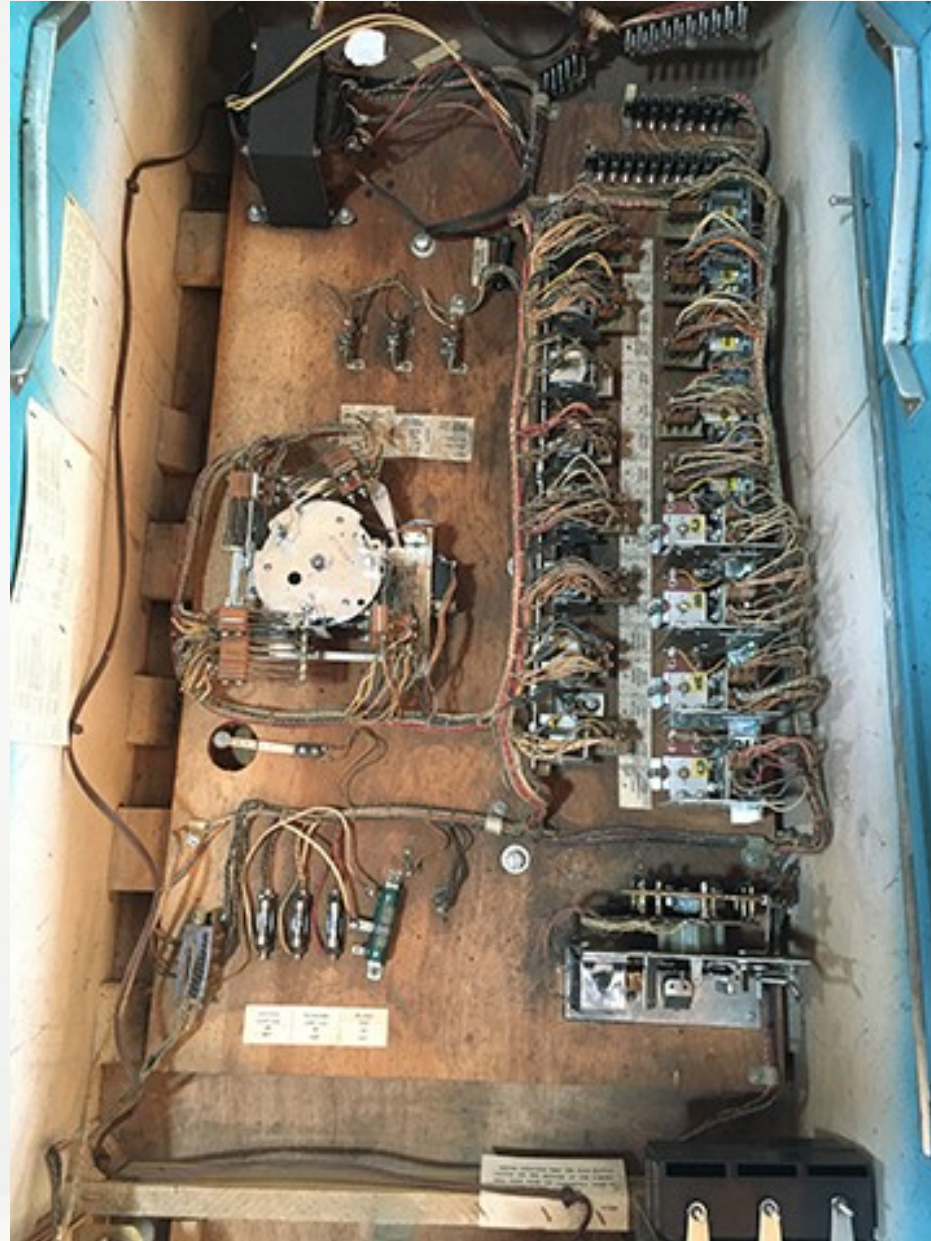


# A bit of history





# A bit of history



This  
contains  
logic

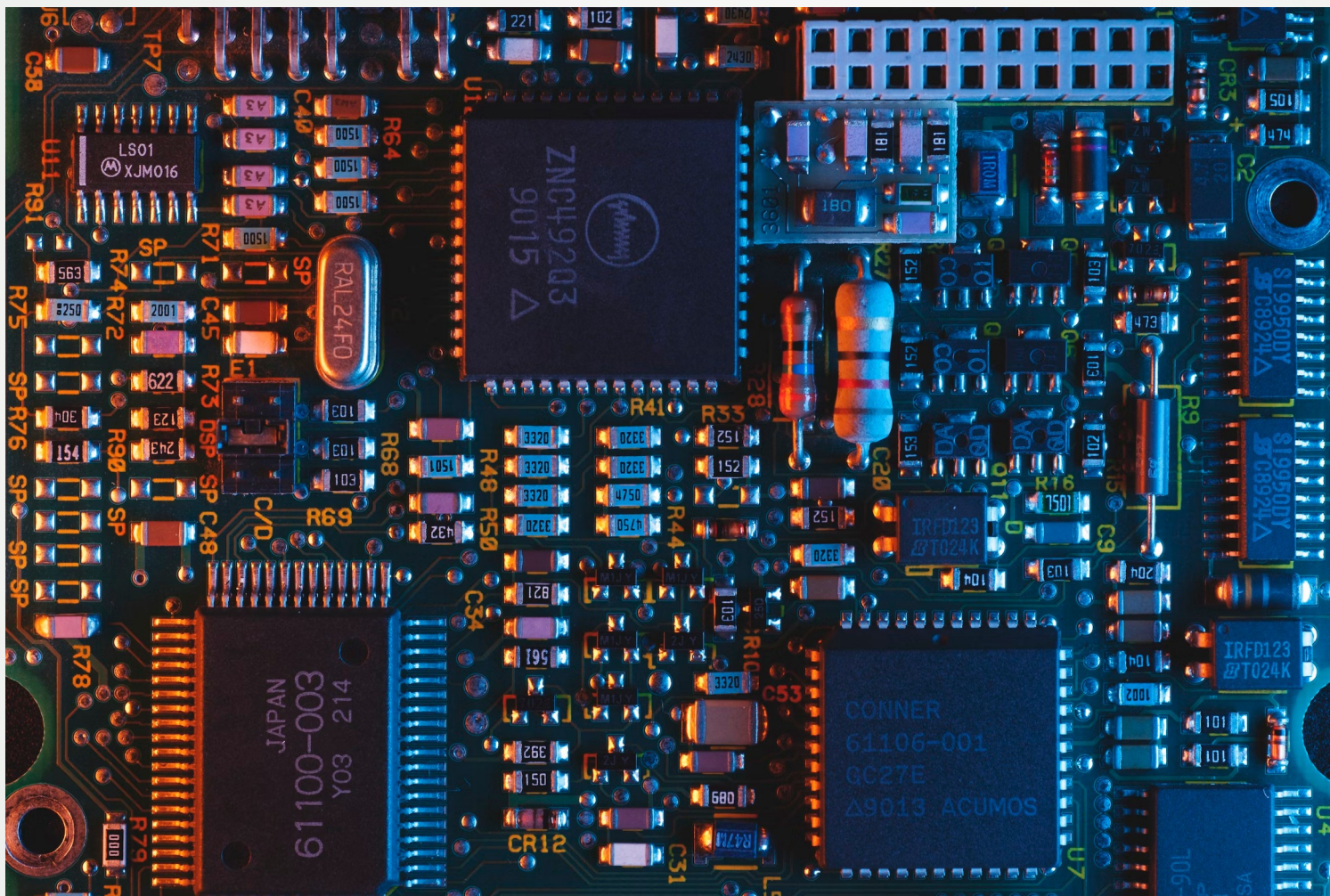
# A bit of history



That  
includes  
these



# Chips replaced the logic



# Let start with your idea



# Let start with your idea



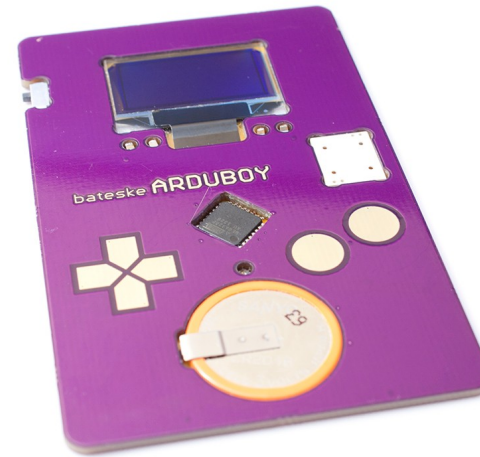
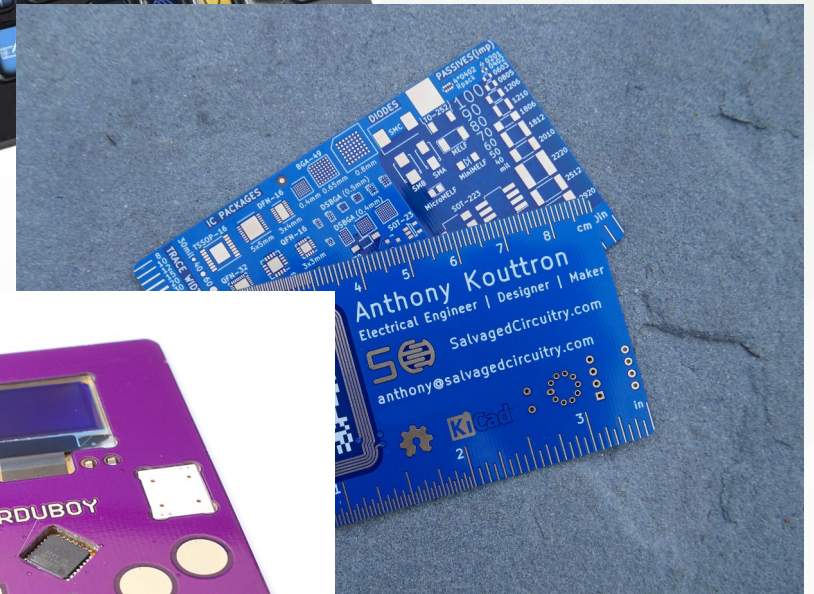


# Let start with your idea





# Let start with your idea



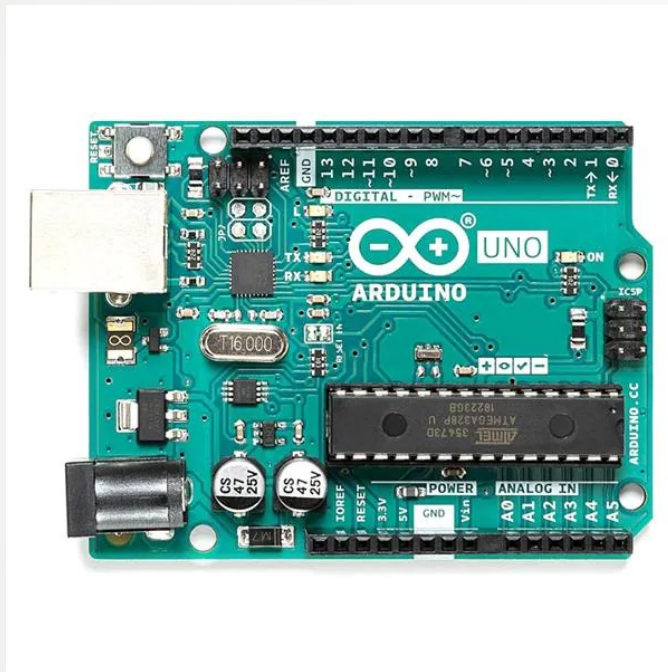
# Let start with your idea



- What does it needs?
- Usb?
- Leds?
- Sensors?
- Motors?
- Touch sensors?

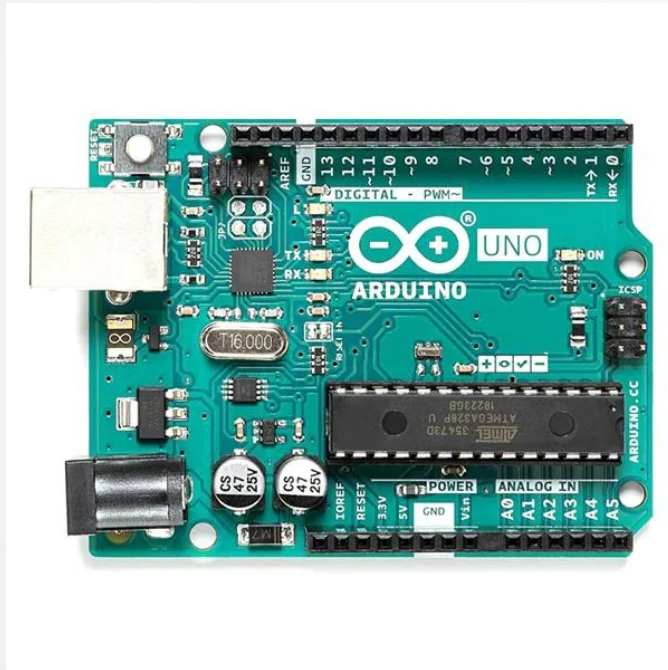


# Find a microcontroller that suits your needs



- What does it needs?
- Usb?
- Leds?
- Sensors?
- Motors?
- Touch sensors?

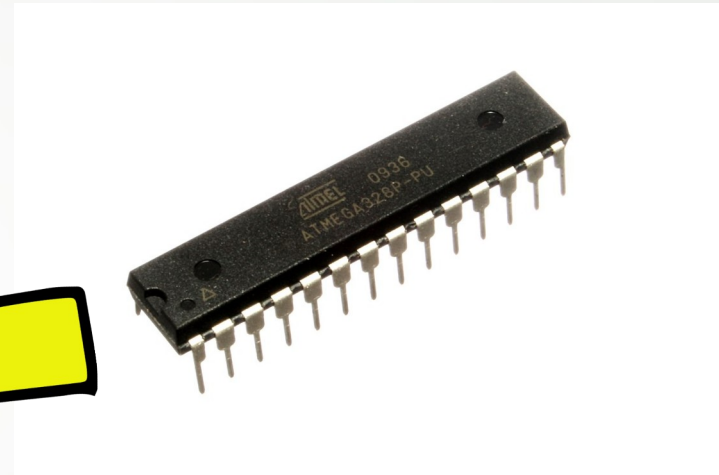
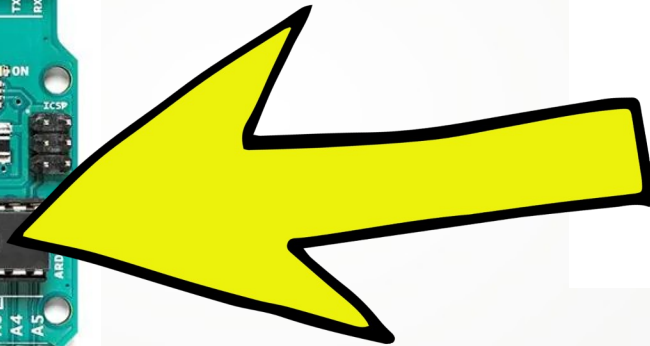
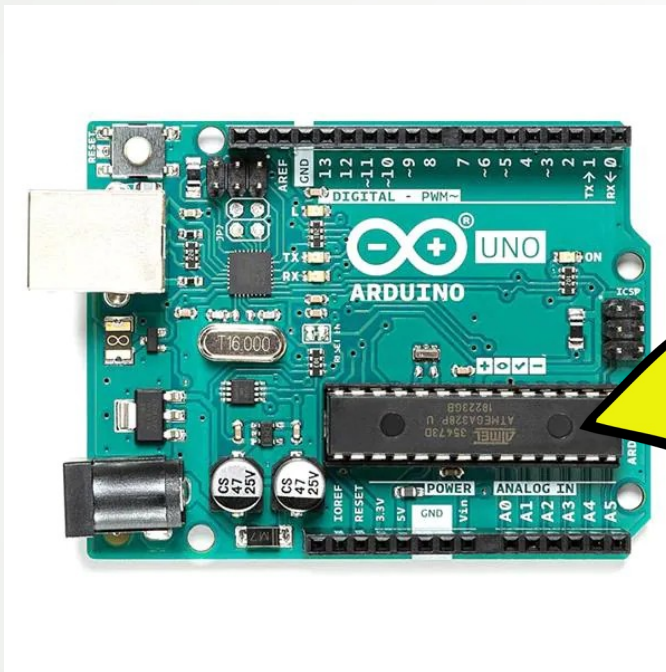
# Find a microcontroller that suits your needs



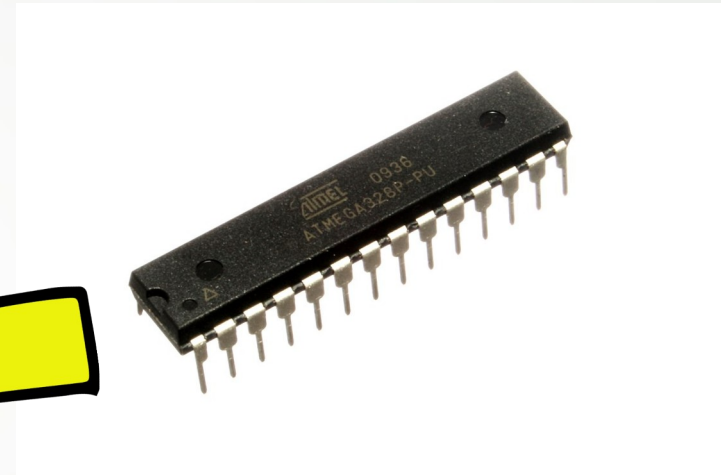
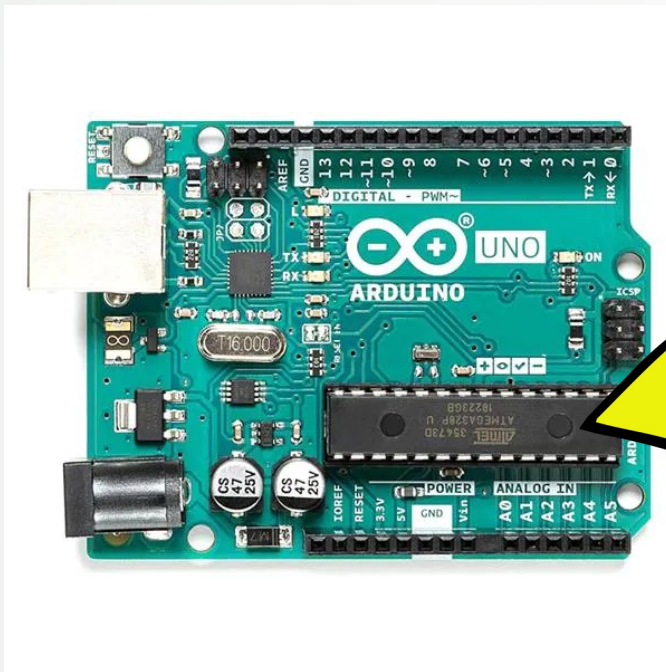
- What does it need?
- Usb → USB compatible
- Leds → GPIO / PINout
- Sensors → ADC / digital in
- Motors → motor controller
- Touch sensors → Pin input



# Find a microcontroller that suits your needs

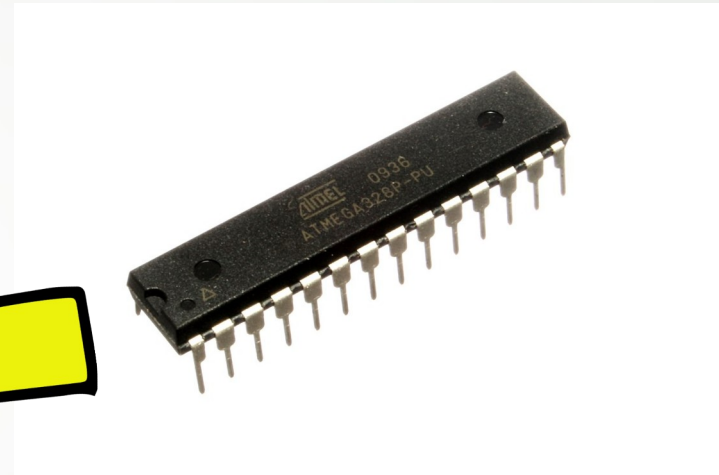
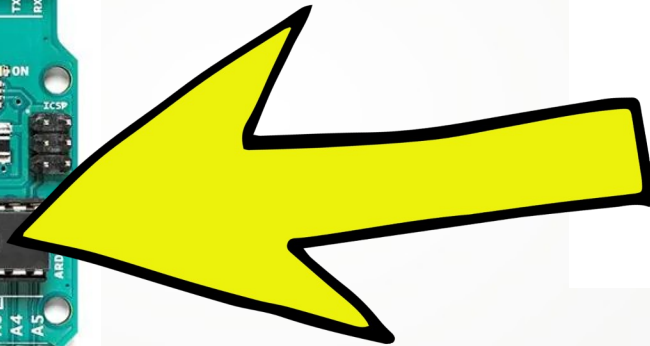
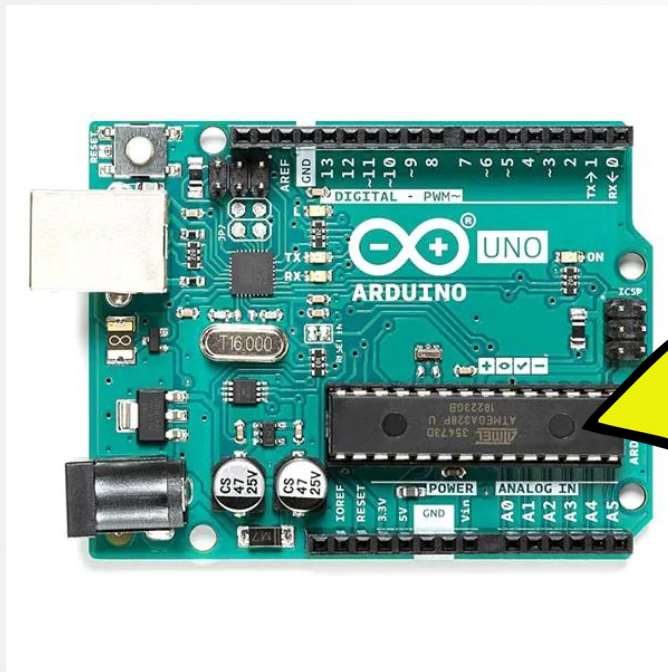


# Find a microcontroller that suits your needs





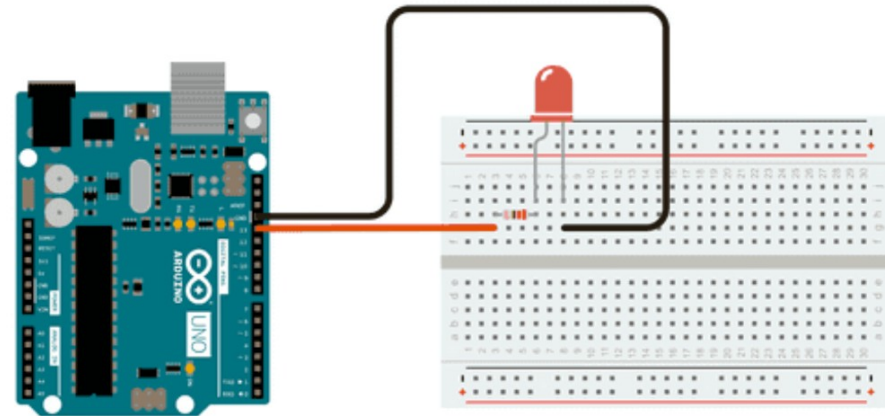
# Step 1: prototyping



- Easy to program
- Easy to prototype with

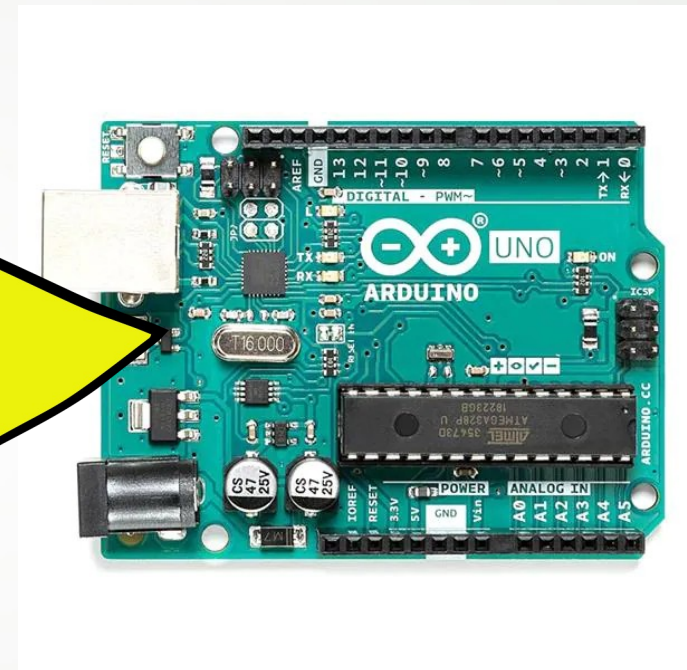
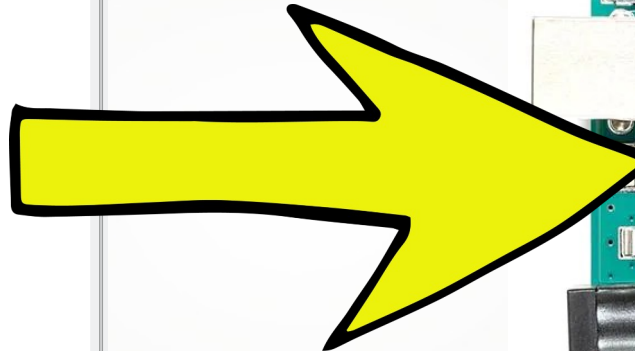
# Step 1: prototyping

```
Blink | Arduino 1.8.19
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * Turns an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
 * the correct LED pin independent of which board is used.
 * If you want to know what pin the on-board LED is connected to on your Arduino
 * model, check the Technical Specs of your board at:
 * https://www.arduino.cc/en/Main/Products
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 * modified 2 Sep 2016
 * by Arturo Guadalupi
 * modified 8 Sep 2016
 * by Colby Newman
 *
 * This example code is in the public domain.
 *
 * https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
 */
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
G:\Disabled, Disabled, Disabled, UART0 / Hardware CDC, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi), 921600, None
```



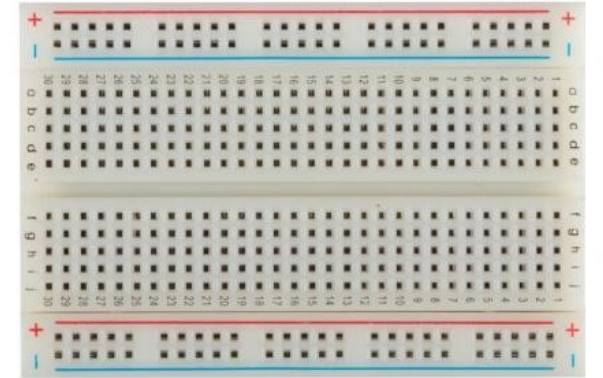
# Step 1: prototyping

```
Blink | Arduino 1.8.19
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * Turns an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
 * the correct LED pin independent of which board is used.
 * If you want to know what pin the on-board LED is connected to on your Arduino
 * model, check the Technical Specs of your board at:
 * https://www.arduino.cc/en/Main/Products
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 * modified 2 Sep 2016
 * by Arturo Guadalupi
 * modified 8 Sep 2016
 * by Colby Newman
 *
 * This example code is in the public domain.
 *
 * https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
 */
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
G:\Disabled, Disabled, Disabled, UART0 / Hardware CDC, Default 4MB with spiifs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi), 921600, None
```



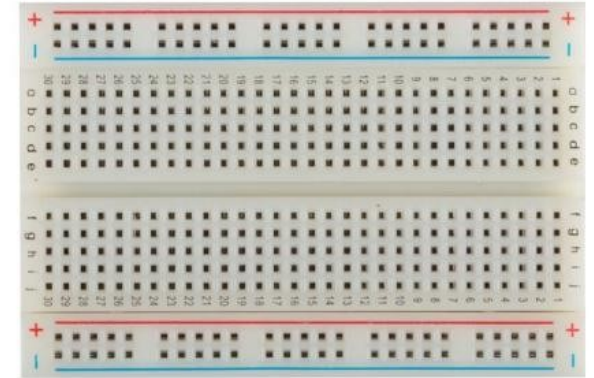
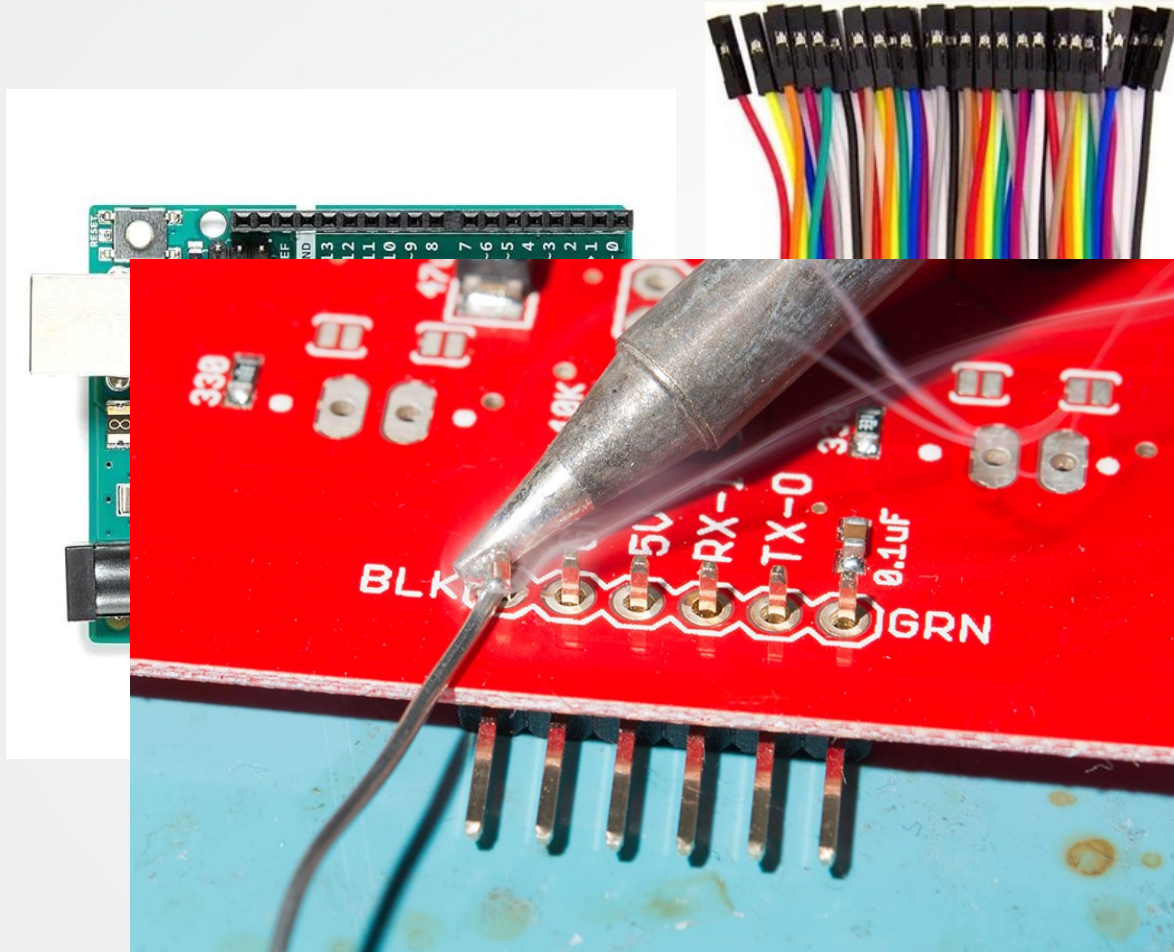


# Step 1: prototyping



MALE TO FEMALE

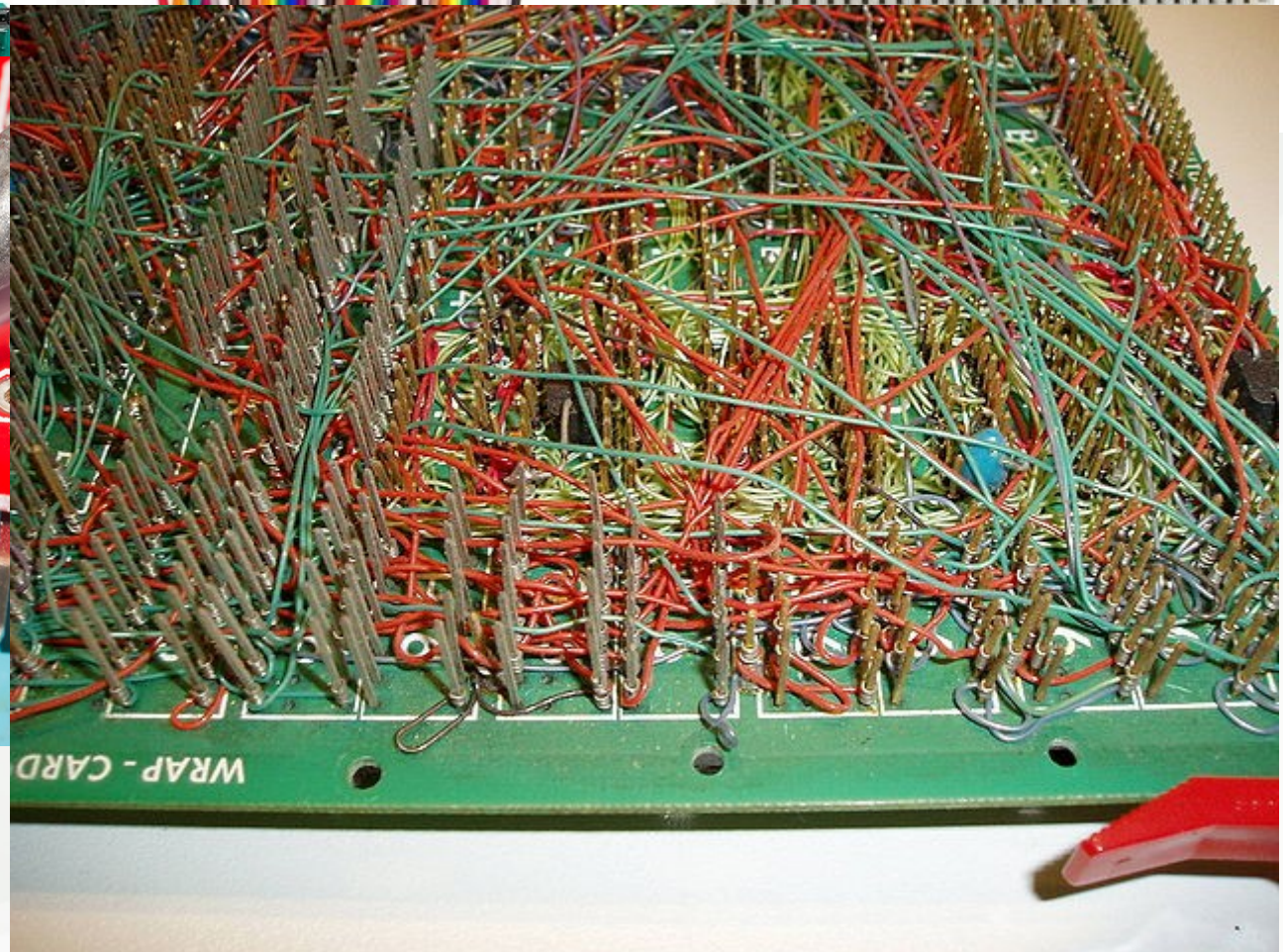
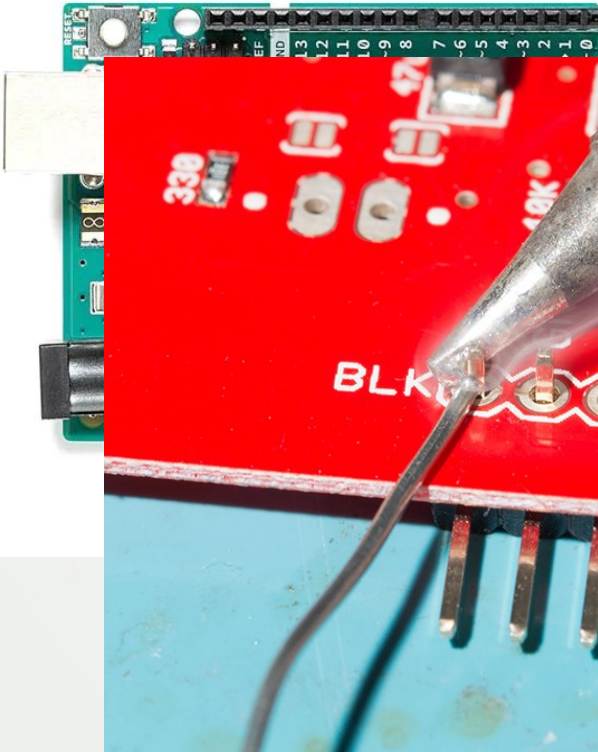
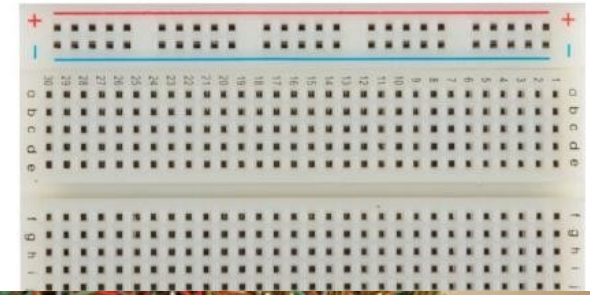
# Step 1: prototyping



MALE TO FEMALE

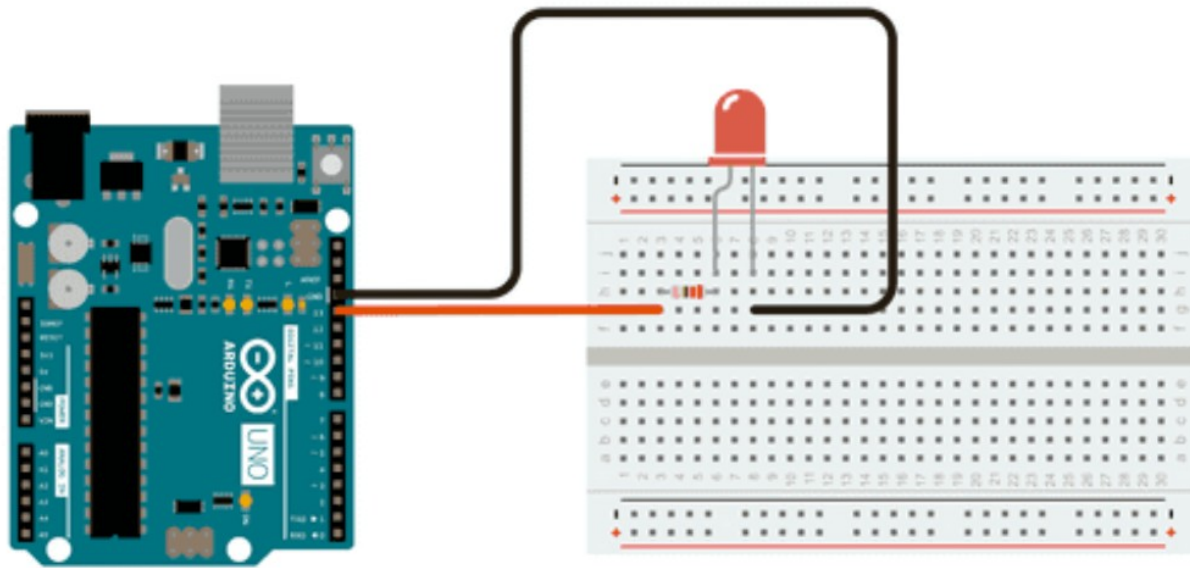


# Step 1: prototyping

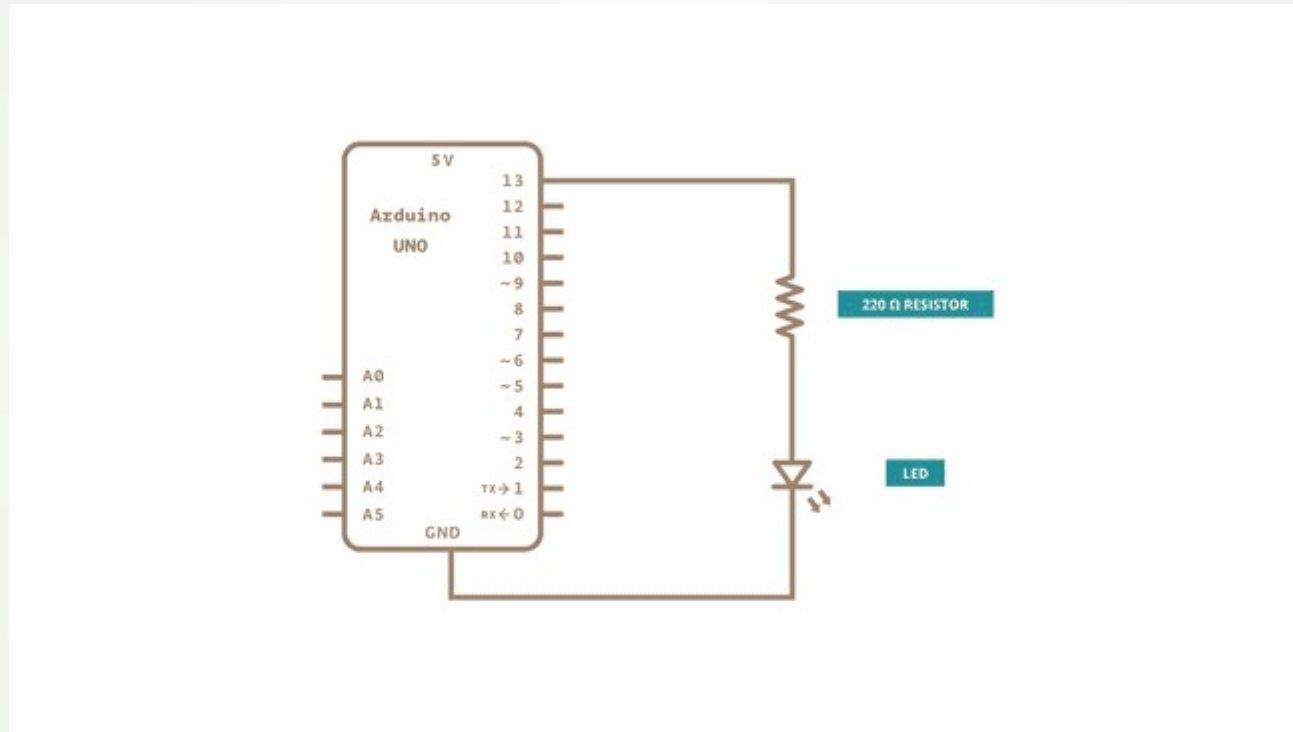




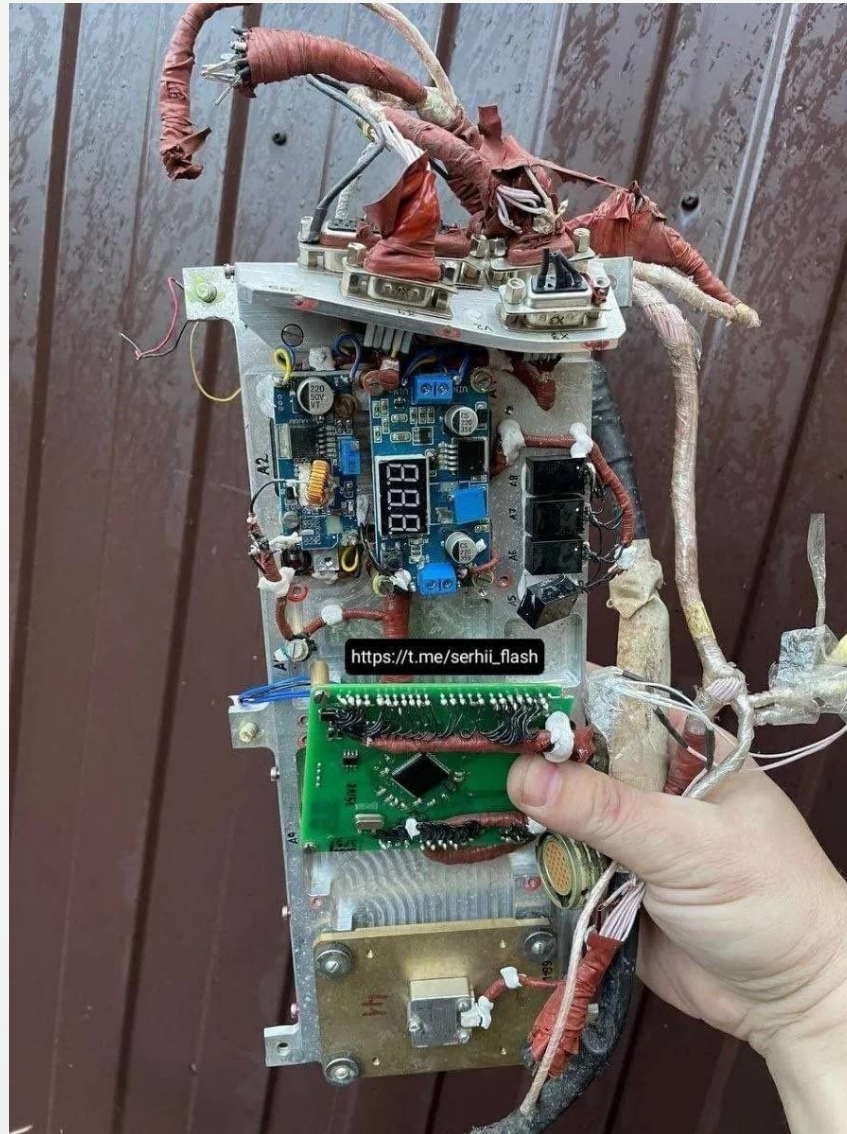
# Step 1: prototyping



# Step 1: prototyping



# Product of prototyping



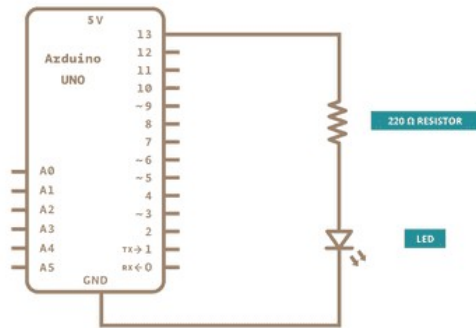


# Product of prototyping

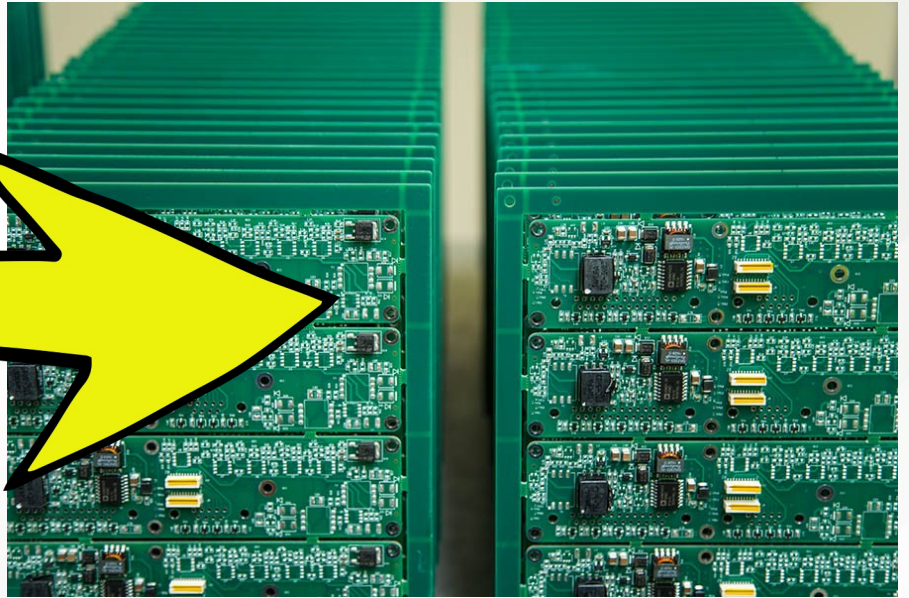
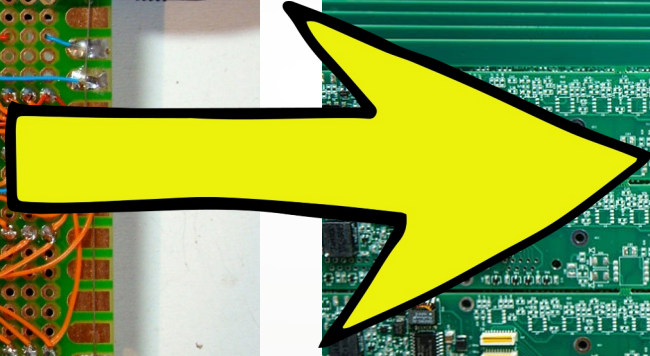
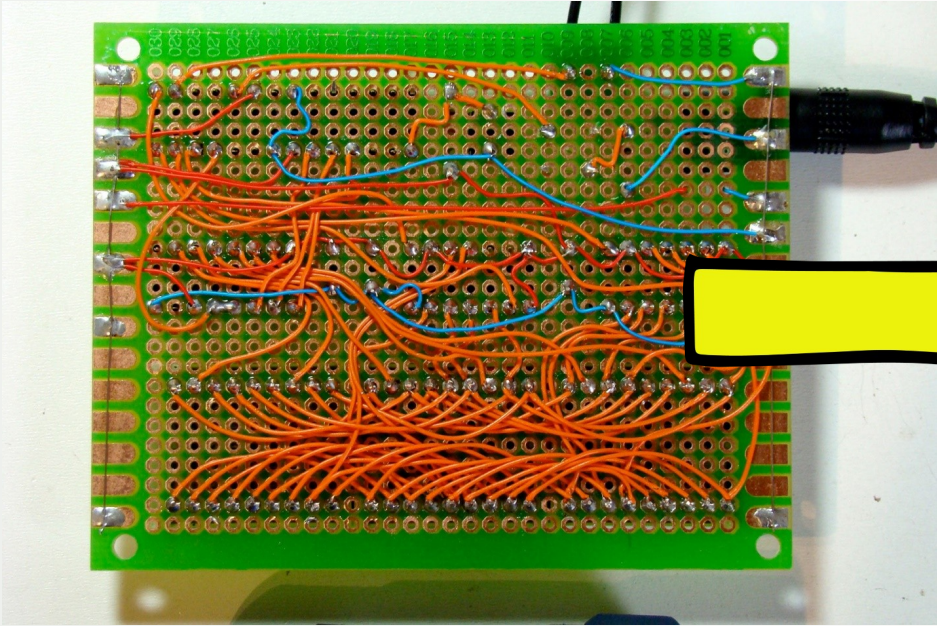


# Product of prototyping

- Working schematic
- Working code

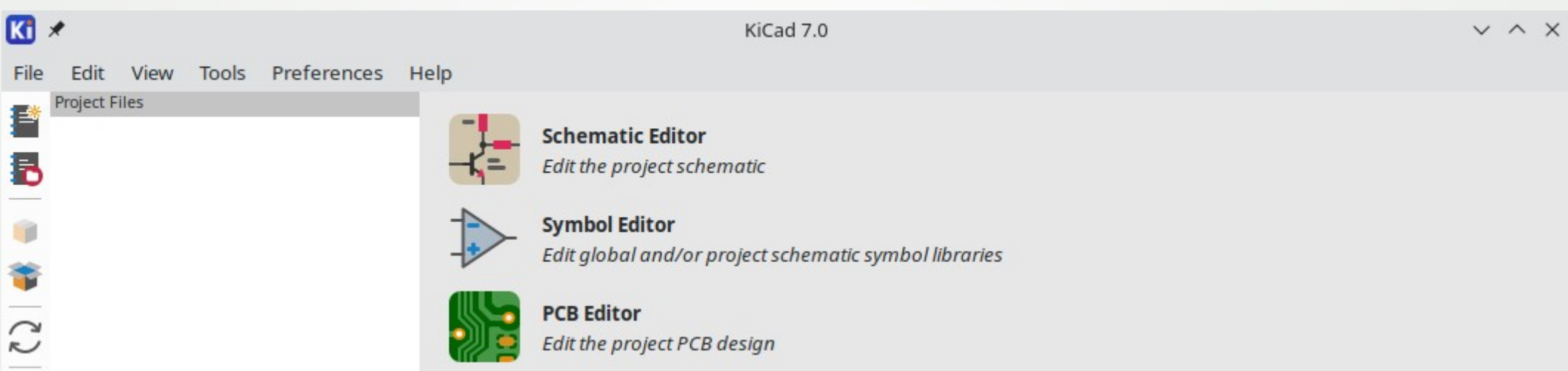


```
Blink | Arduino 1.8.19
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * Turns an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
 * the correct LED pin independent of which board is used.
 * If you want to know what pin the on-board LED is connected to on your Arduino
 * model, check the Technical Specs of your board at:
 * https://www.arduino.cc/en/Main/Products
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 * modified 2 Sep 2016
 * by Arturo Guadalupi
 * modified 8 Sep 2016
 * by Colby Newman
 *
 * This example code is in the public domain.
 *
 * https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
 */
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
G Disabled, Disabled, Disabled, UART0 / Hardware CDC, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi), 921600, None
```





# Step 2: Kicad



# Step 2: Schemetic editor

# Step 2: Symbol / Footprint Editor



# Step 2: PCB Editor

- Import from schematic
- 
- Layers(enter schrek meme here)

# Step 3: Export

- Gerberfiles
- Pick and place
- Bom
  
- Kicad PCB file

# Order

- JLC PCB
- Handsolder vs Let it be produced for you



# Beautify your pcb

## SVG Magic

- As layer
- Layer used for other designs
- Use the back of PCB
- Gold plating
-

# Features

- Text based, só git compatible
- Kibot