

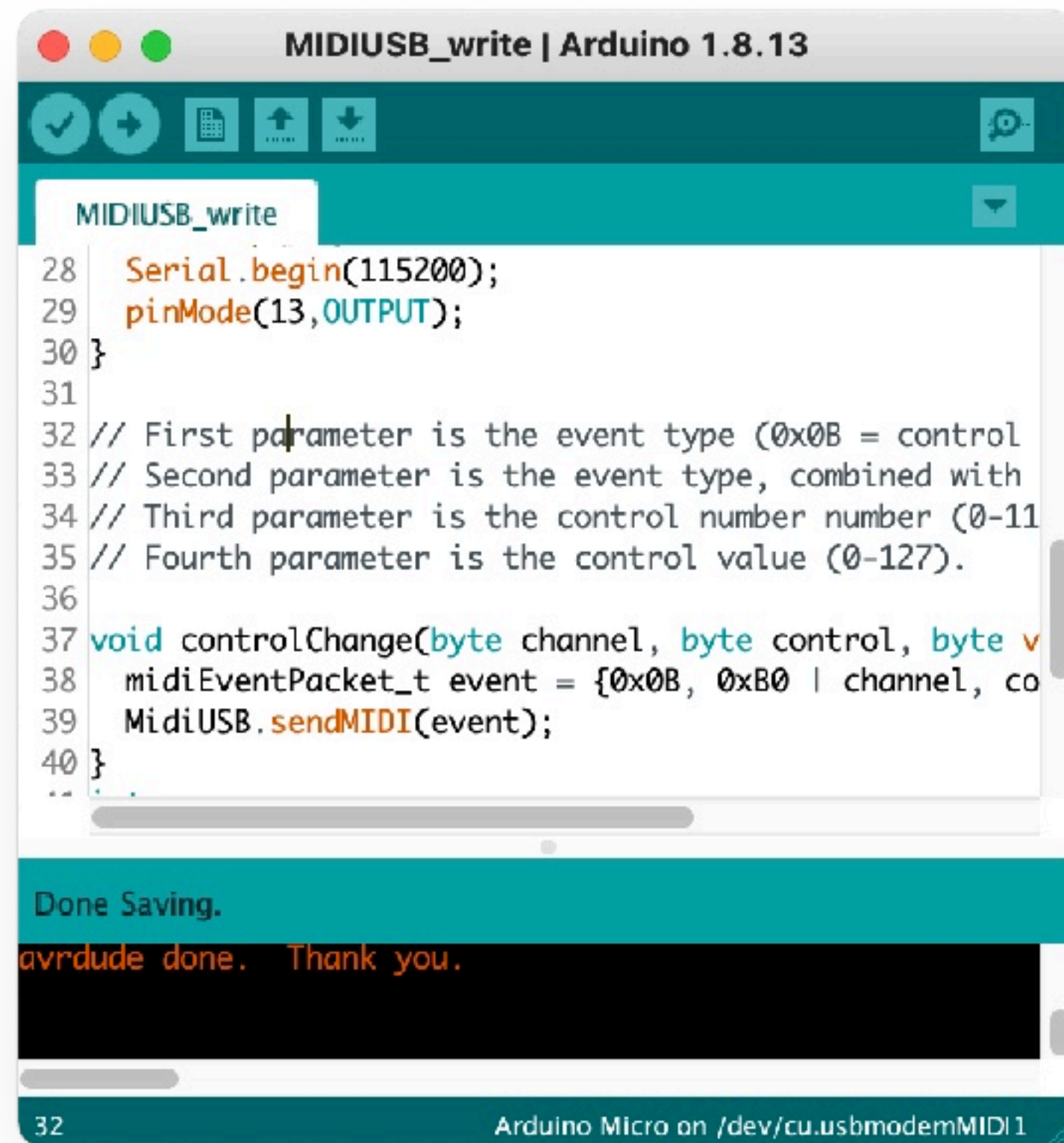
# **Workshop Arduino (MIDI) music**

**quick building blocks using USB-MIDI with Arduino Micro**



# Getting started with arduino

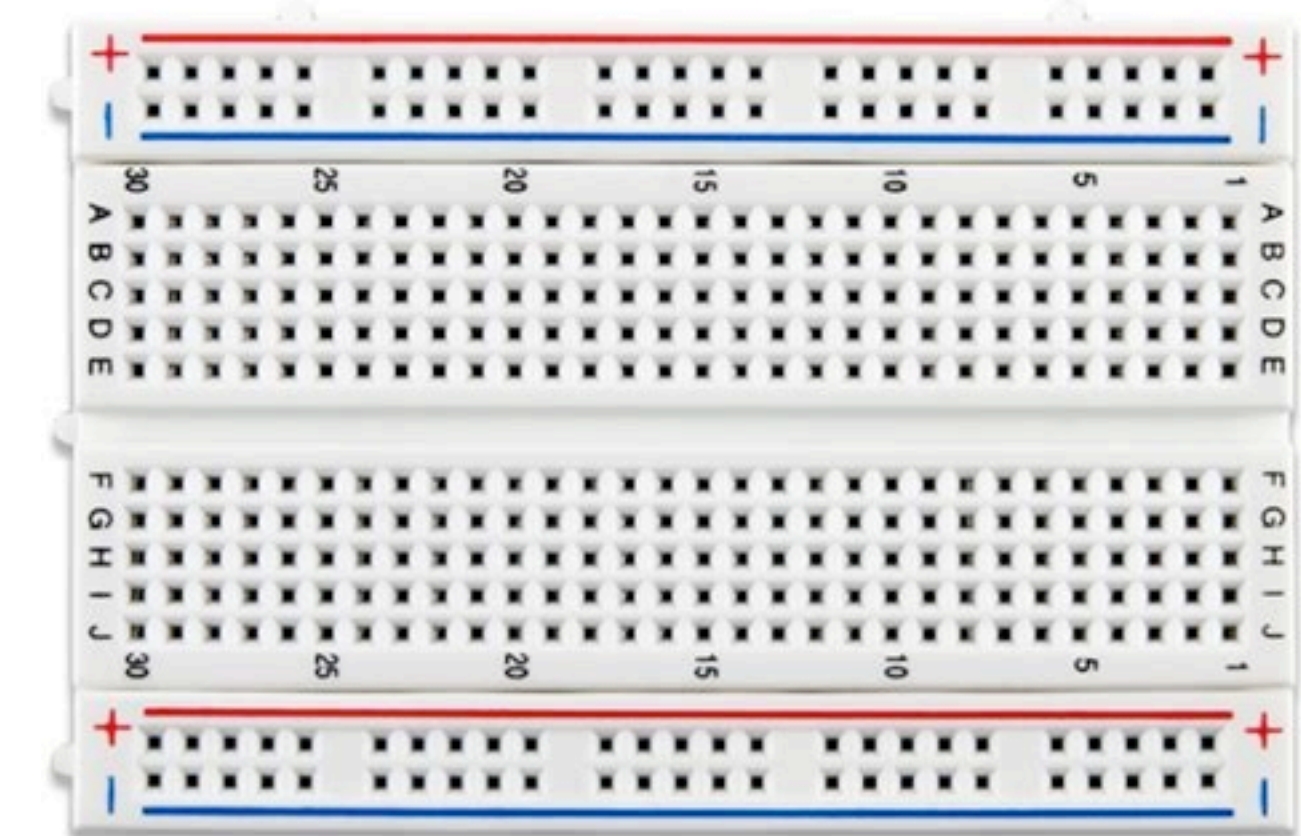
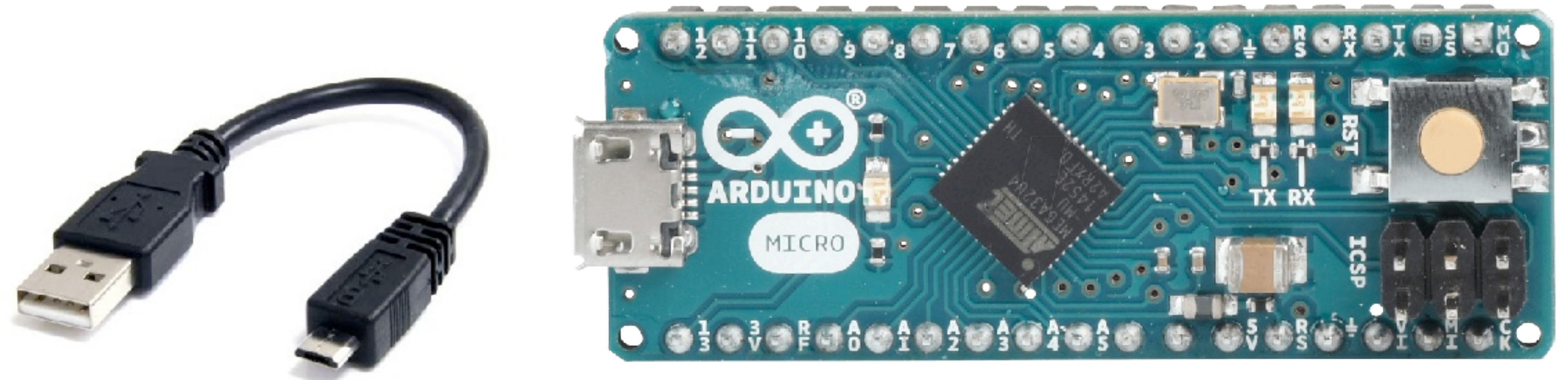
## bits and pieces



The screenshot shows the Arduino IDE interface with a sketch named "MIDIUSB\_write". The code is as follows:

```
28 Serial.begin(115200);
29 pinMode(13, OUTPUT);
30 }
31
32 // First parameter is the event type (0x0B = control
33 // Second parameter is the event type, combined with
34 // Third parameter is the control number number (0-11
35 // Fourth parameter is the control value (0-127).
36
37 void controlChange(byte channel, byte control, byte v
38   midiEventPacket_t event = {0x0B, 0xB0 | channel, co
39   MidiUSB.sendMIDI(event);
40 }
```

Below the code editor, a status bar indicates "Done Saving." and "avrdude done. Thank you." The bottom status bar shows "32" and "Arduino Micro on /dev/cu.usbmodemMIDI1".





# Get arduino up and running

the 'hello world' of physical computing: a blinking LED



- Install Arduino on your system (from [arduino.cc](https://arduino.cc) or our big-disk), start up the program
- connect the Arduino Micro board to your system. Select correct port and board in the 'tools' menu
- search in the examples directory (that comes with arduino) for a blink example. Check and Upload...
- install the MIDIUSB library through the library manager (or take our local copy)

# Get processing up and running

our free, open source audio tool and MIDI DAW for the day..

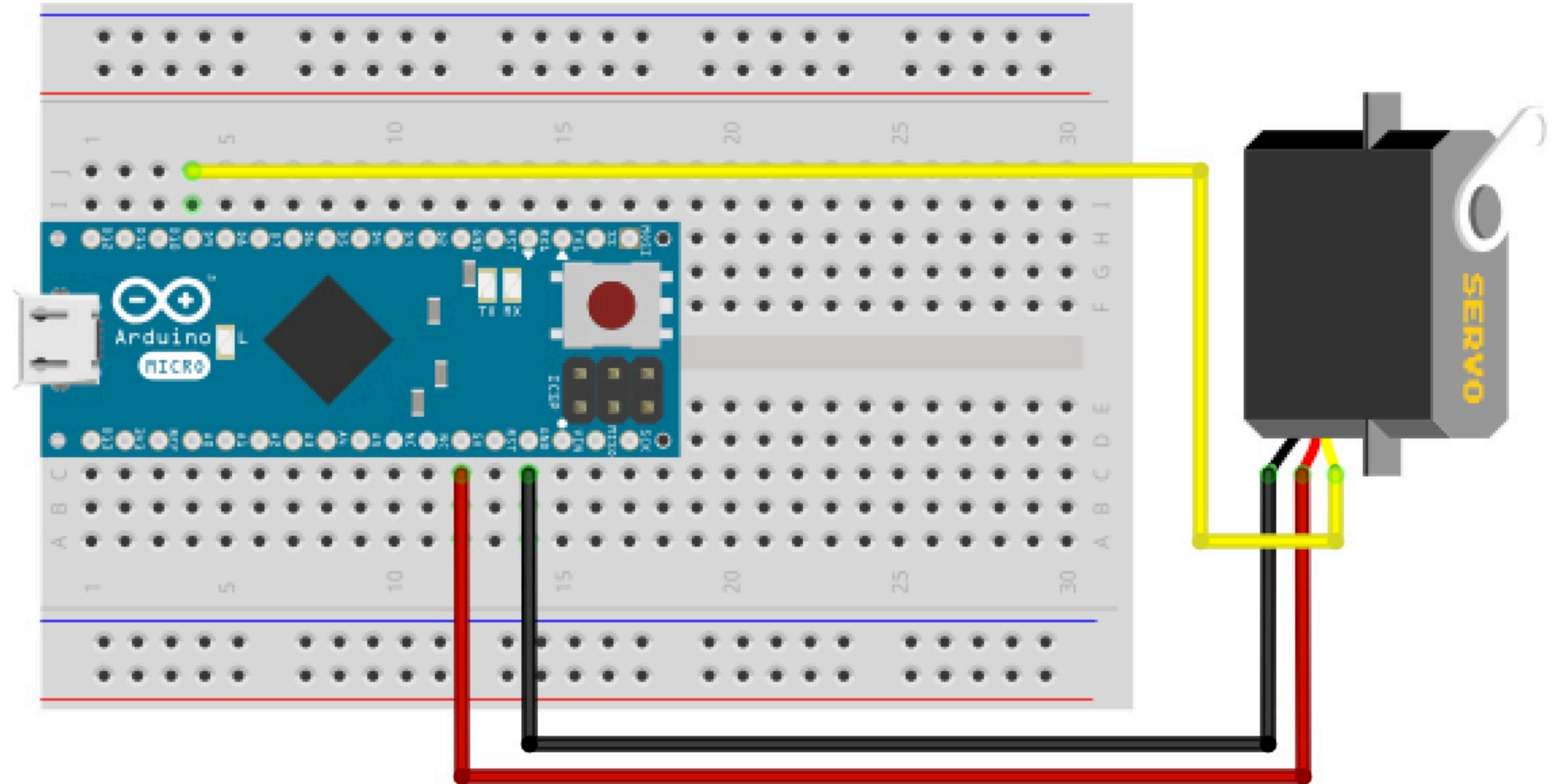


- download and install processing (from [processing.org](https://processing.org) or our big disk)
- install the 'midibus' library through the library manager
- collect the sources for today from <https://github.com/edwindertien/robottheaterlab/tree/main/workshop>
- you might want to try dimmer/fader panels, download akai config tools from (for example <https://www.akaipro.com/lpd8-lpd8>)



# RC servo

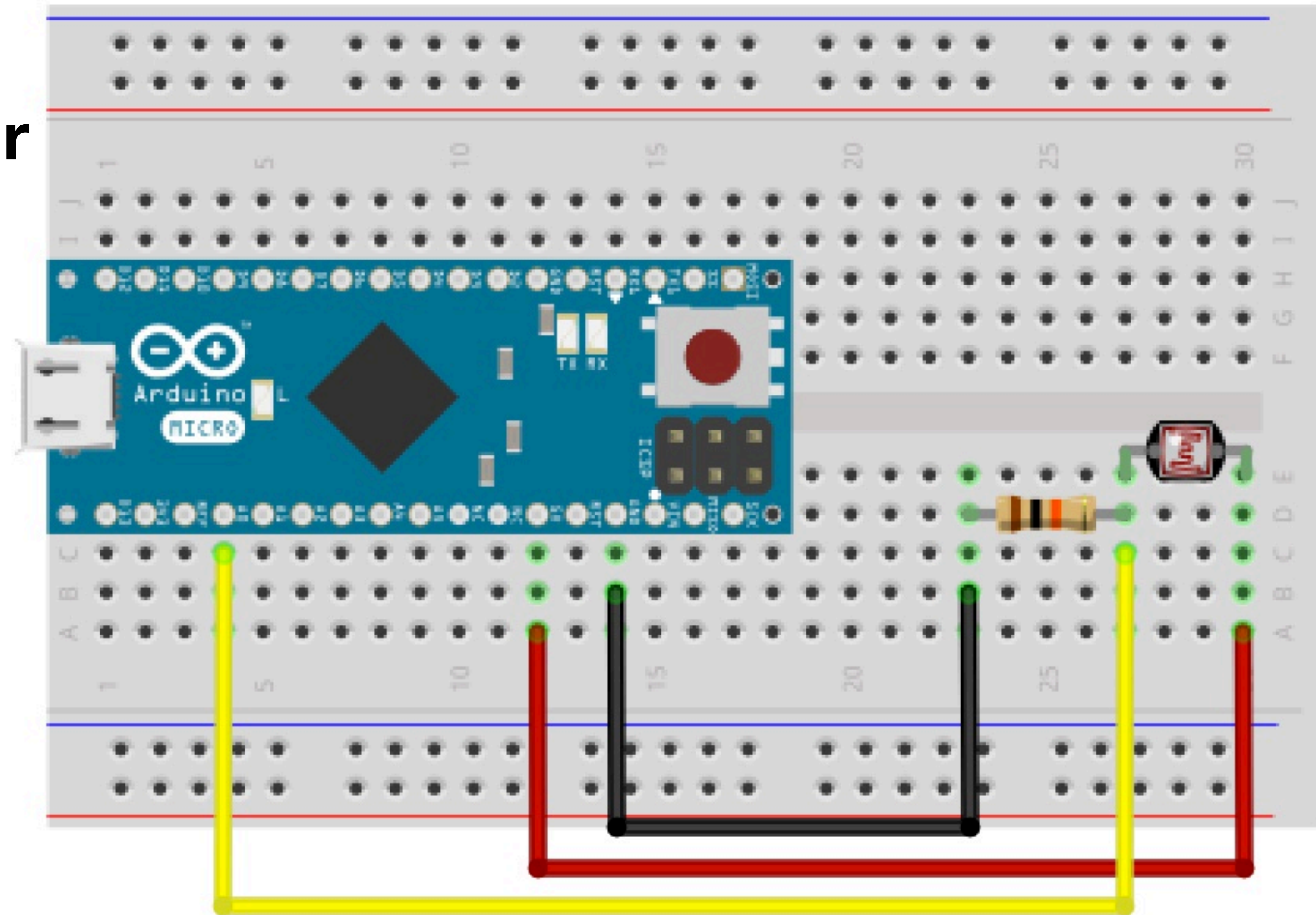
(signal from pin 9)





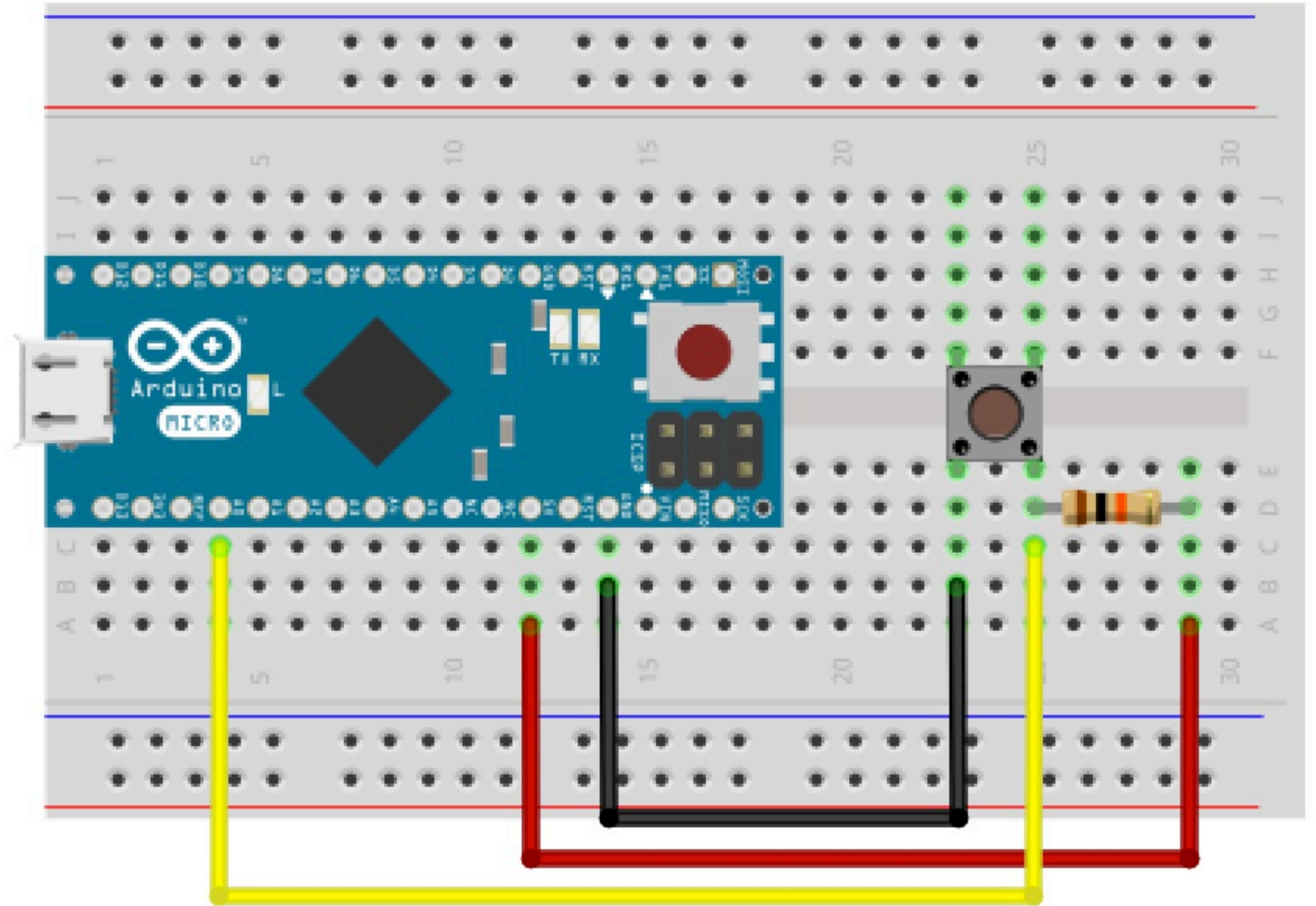
# Light sensor

## LDR with pull-down resistor



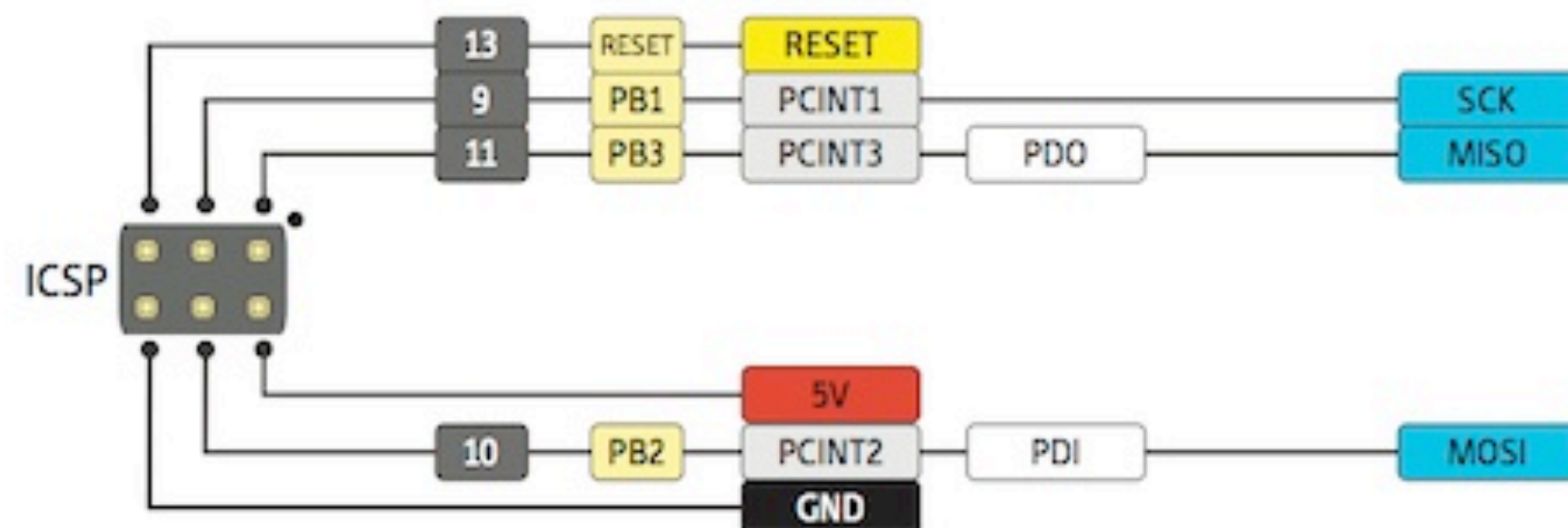
# pushbutton

(with pull-up resistor)





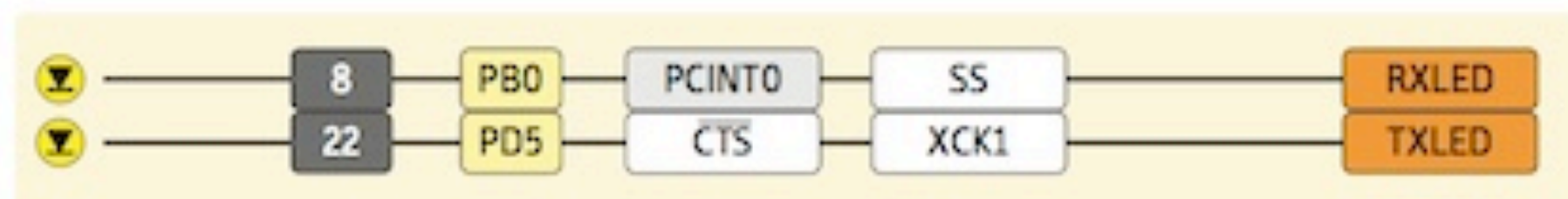
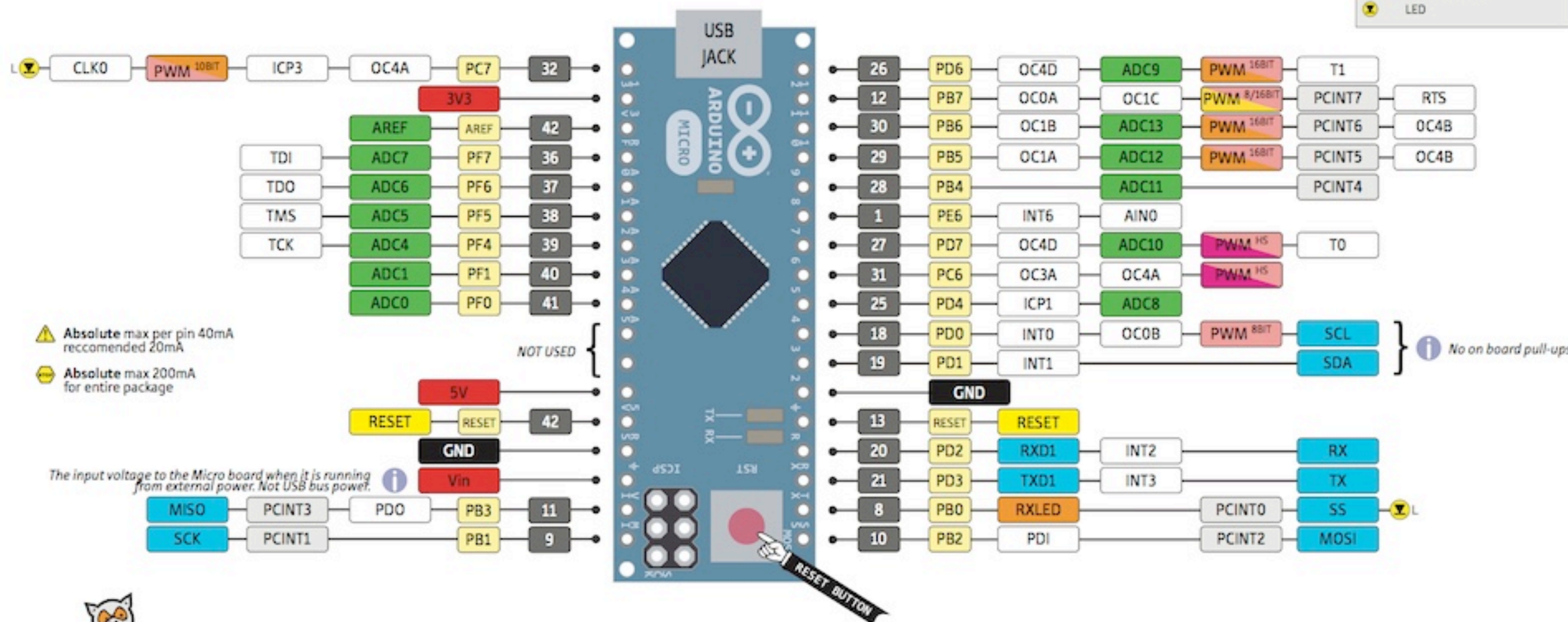
# THE UNOFFICIAL ARDUINO MICRO PINOUT DIAGRAM



**LEGEND**

GND
POWER
CONTROL
PHYSICAL PIN
PORT PIN
PIN FUNCTION
DIGITAL PIN
ANALOG-RELATED PIN
PWM PIN
SERIAL PIN

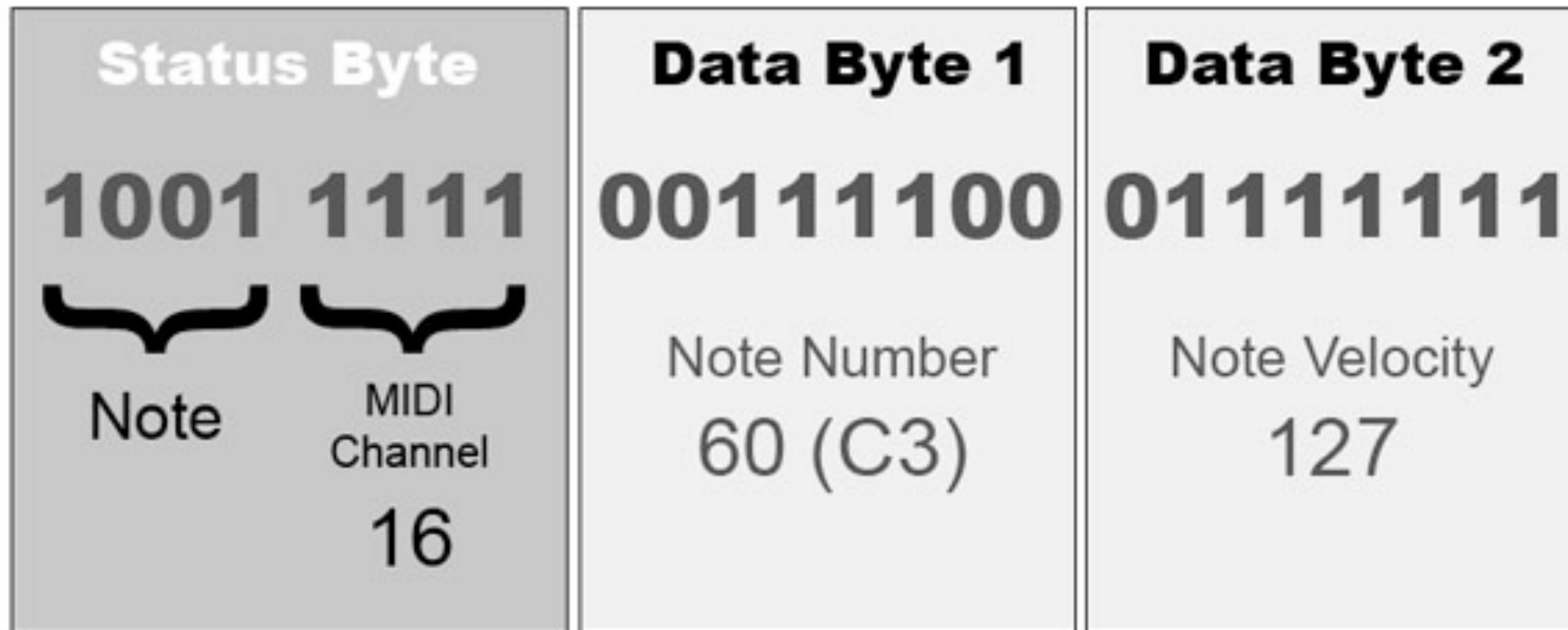
ⓘ General Information  
 ⚠ Pay Attention  
 ⚡ No Really PAY ATTENTION  
 💡 LED





# MIDI messages

(USB-MIDI or wired MIDI or virtual MIDI)



**Table 1: MIDI 1.0 Specification Message Summary**

<b>Status</b> <b>D7----D0</b>	<b>Data Byte(s)</b> <b>D7----D0</b>	<b>Description</b>
Channel Voice Messages [nnnn = 0-15 (MIDI Channel Number 1-16)]		
1000nnnn	0kkkkkkk  0wwwvv	Note Off event.  This message is sent when a note is released (ended). (kkkkkkk) is the key (note) number. (wwwvv) is the velocity.
1001nnnn	0kkkkkkk  0wwwvv	Note On event.  This message is sent when a note is depressed (start). (kkkkkkk) is the key (note) number. (wwwvv) is the velocity.
1011nnnn	0ccccccc  0wwwvv	Control Change.  This message is sent when a controller value changes. Controllers include devices such as pedals and levers. Controller numbers 120-127 are reserved as "Channel Mode Messages" (below). (ccccccc) is the controller number (0-119). (wwwvv) is the controller value (0-127).