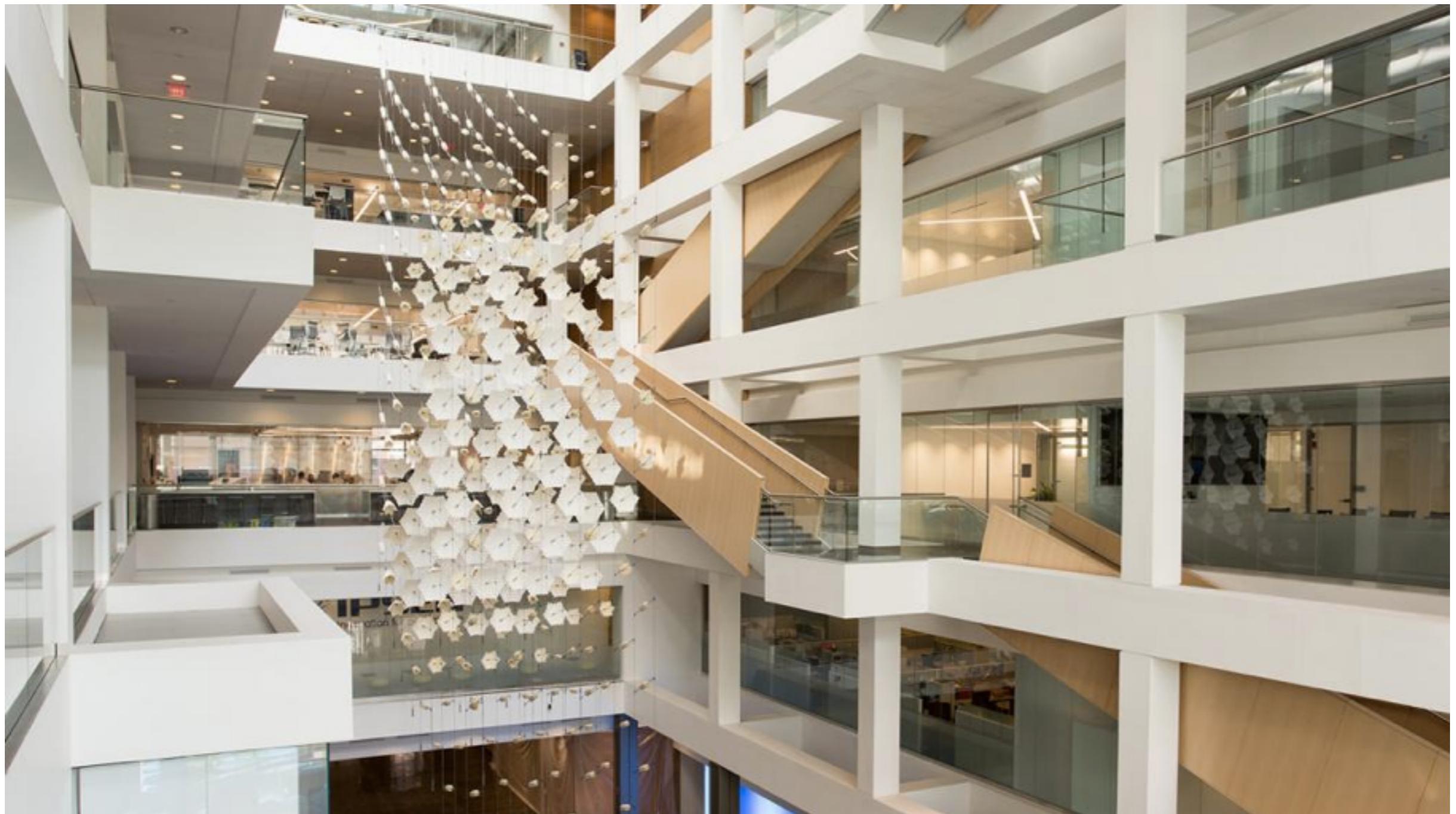


SFPC : Electronics

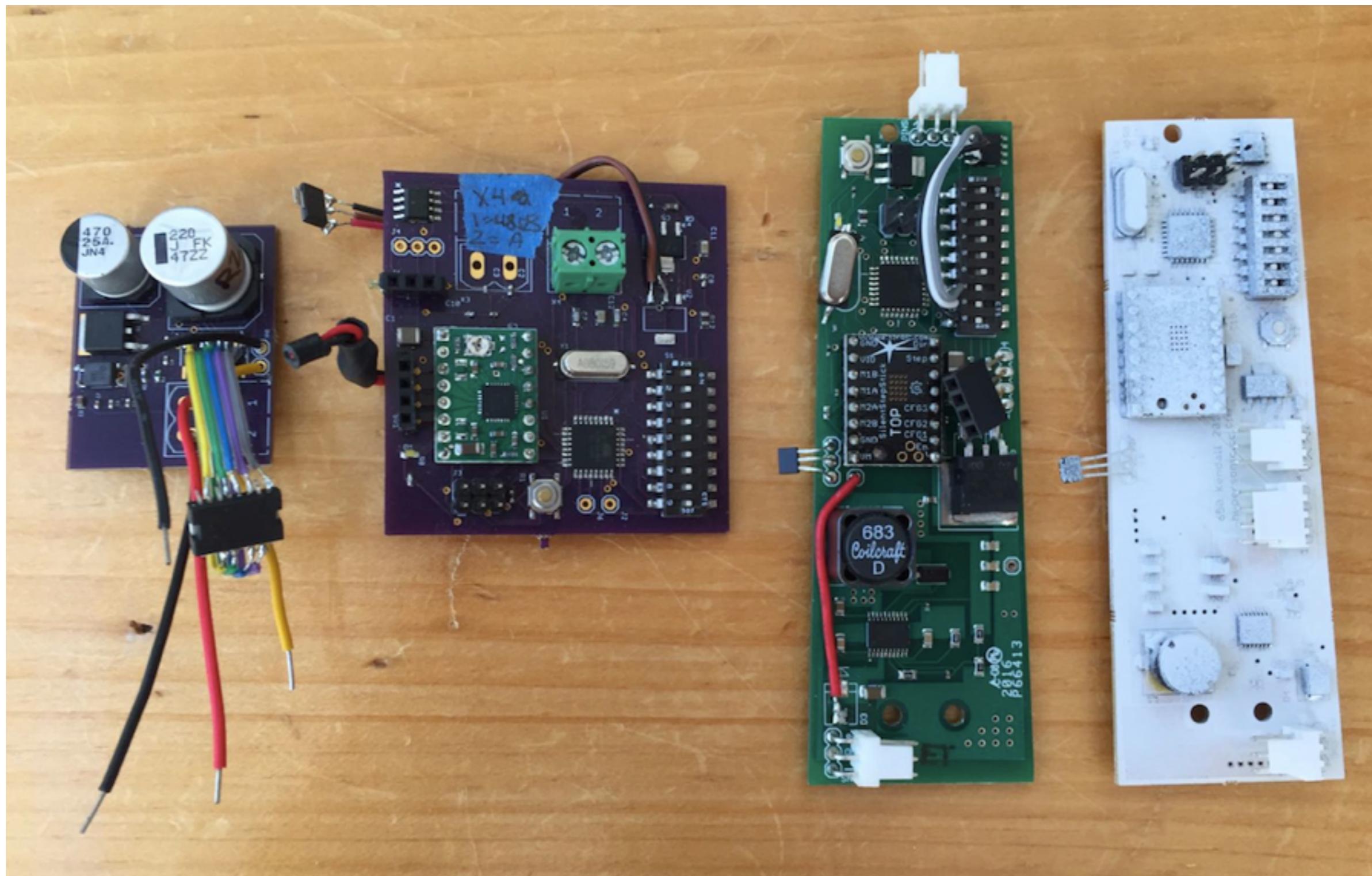
intro / background



intro / background



intro / background



intro / background



class structure

- Week 1: Intro to Arduino, digital inputs, outputs
- Week 2: Analog inputs, sensors, motors
- Week 3: Communication bridges I (Processing / oF)
- Week 4: Communication bridges II (OSC / Serial
cont'd)

things to consider

- Why choose one technology over another?
- What can you do in the built world that can't be done digitally / virtually, and vice versa?
- What are the benefits and/or dangers of the built environment?
- Technology as subject vs. technology as enabler

microcontrollers

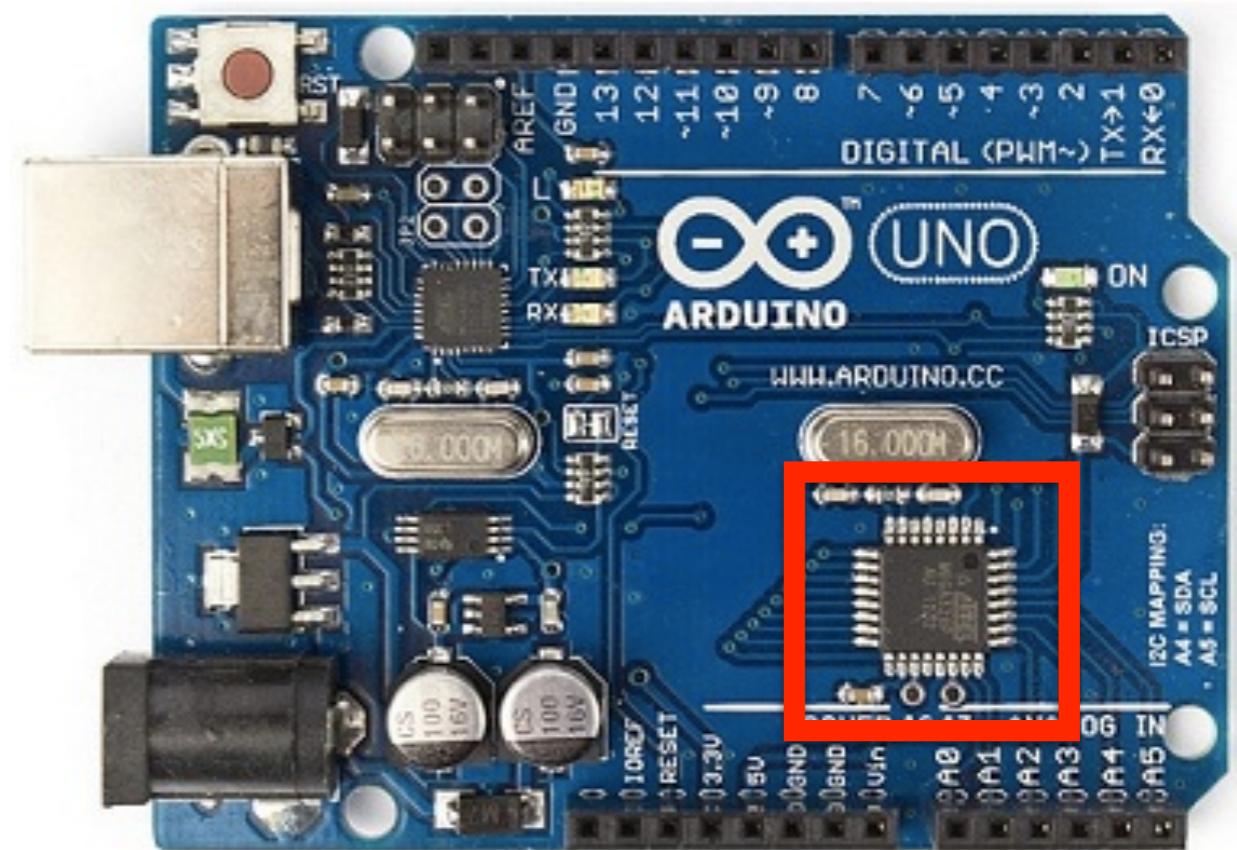
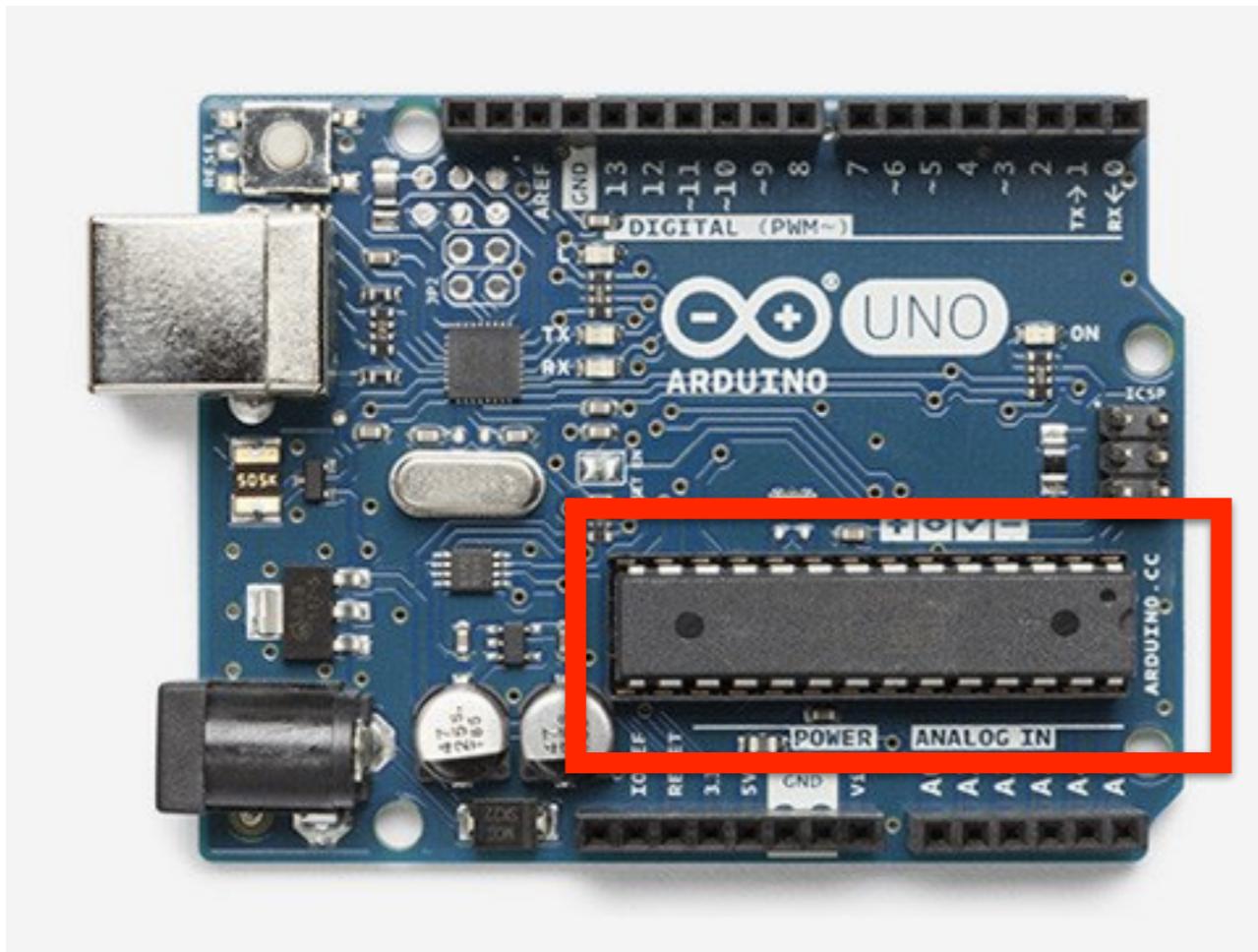
Microcontroller = mini-computer

IC (integrated circuit — many electronics on one chip)
with a processor, memory, and programmable input/
output.

anatomy of the arduino

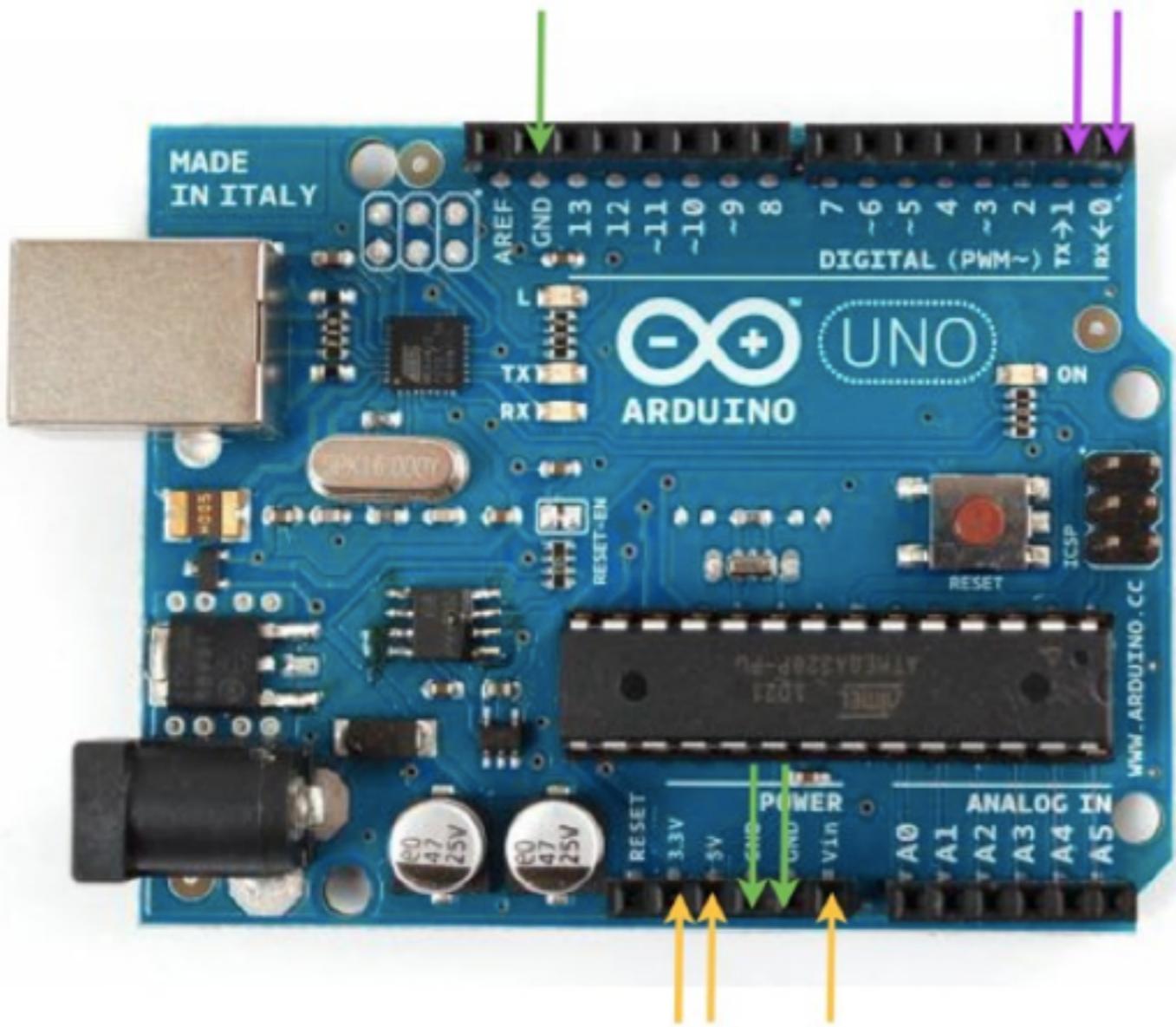


anatomy of the arduino



pins

A **pin** provides an input or output through which the controller can communicate with components.



TX/RX (serial -
transmit/receive)

3 ground pins

3 power pins

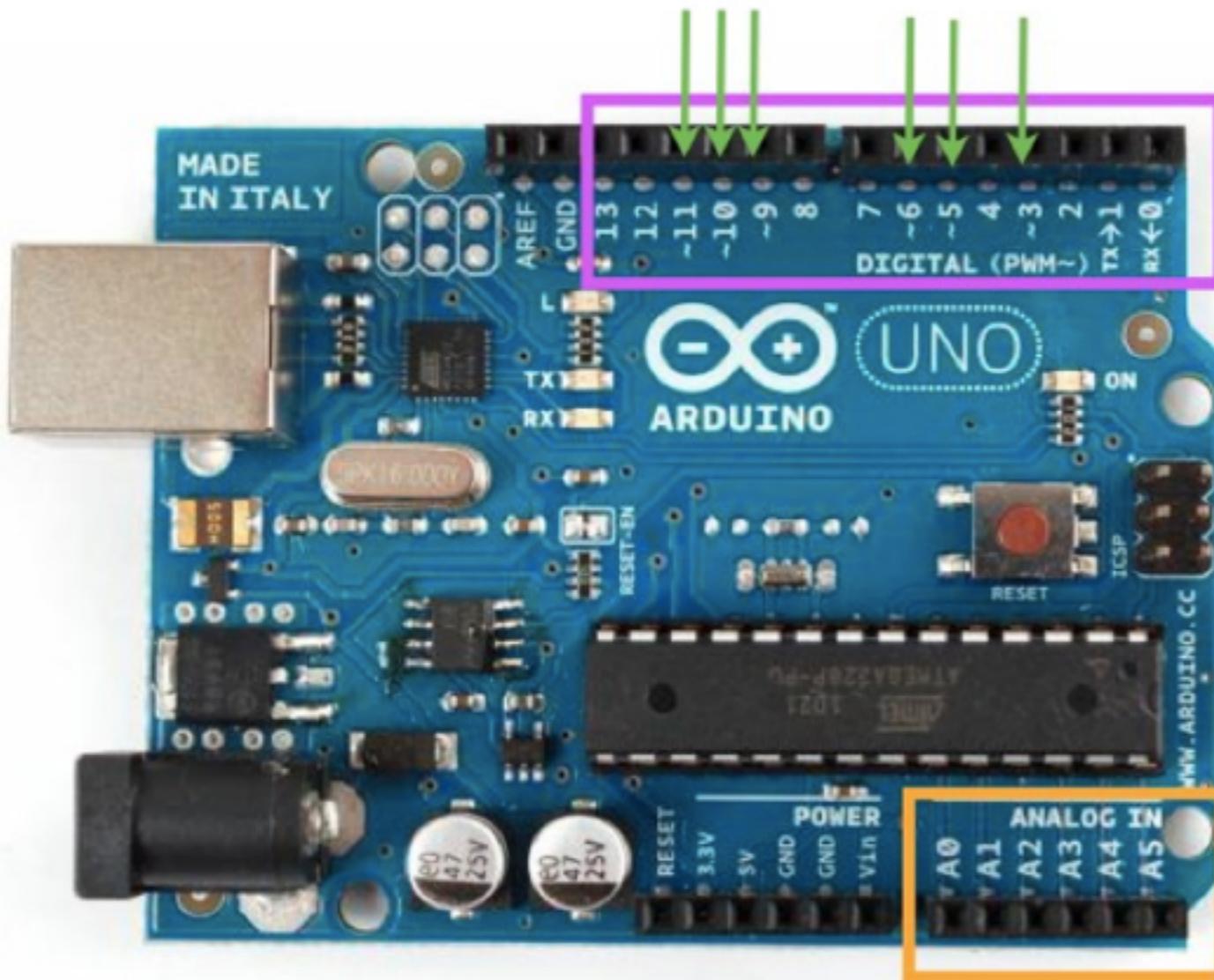
// 5 volts

// 3 volts

// VIN - can plug 9 volts here

pins

A **pin** provides an input or output through which the controller can communicate with components.



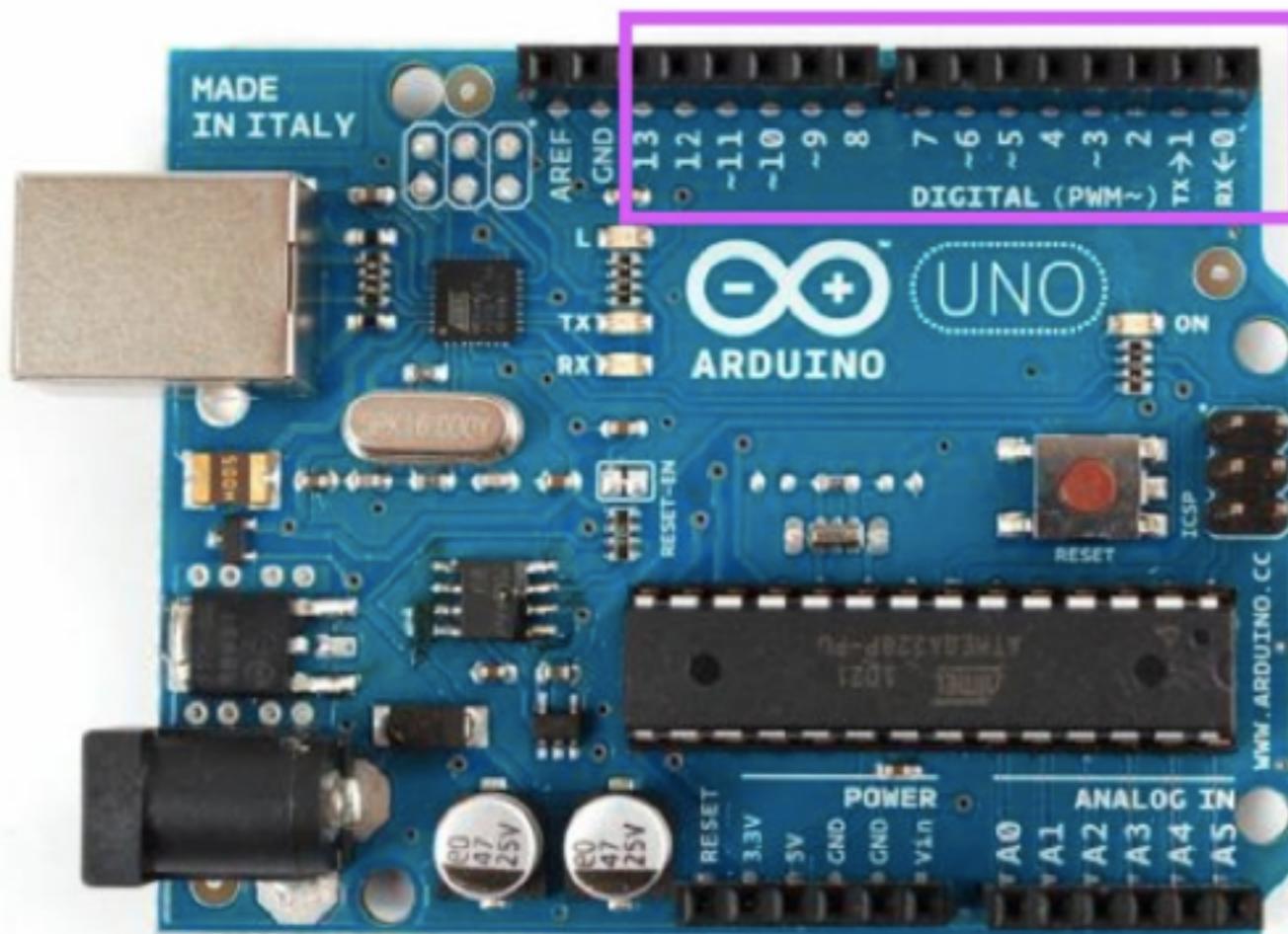
14 Digital pins

6 Pulse Width Modulation
enabled pins

6 Analog input pins

pins

Communicate with your components by sending different voltages to them.



14 Digital pins

You can read or write 2 different values to them:

HIGH

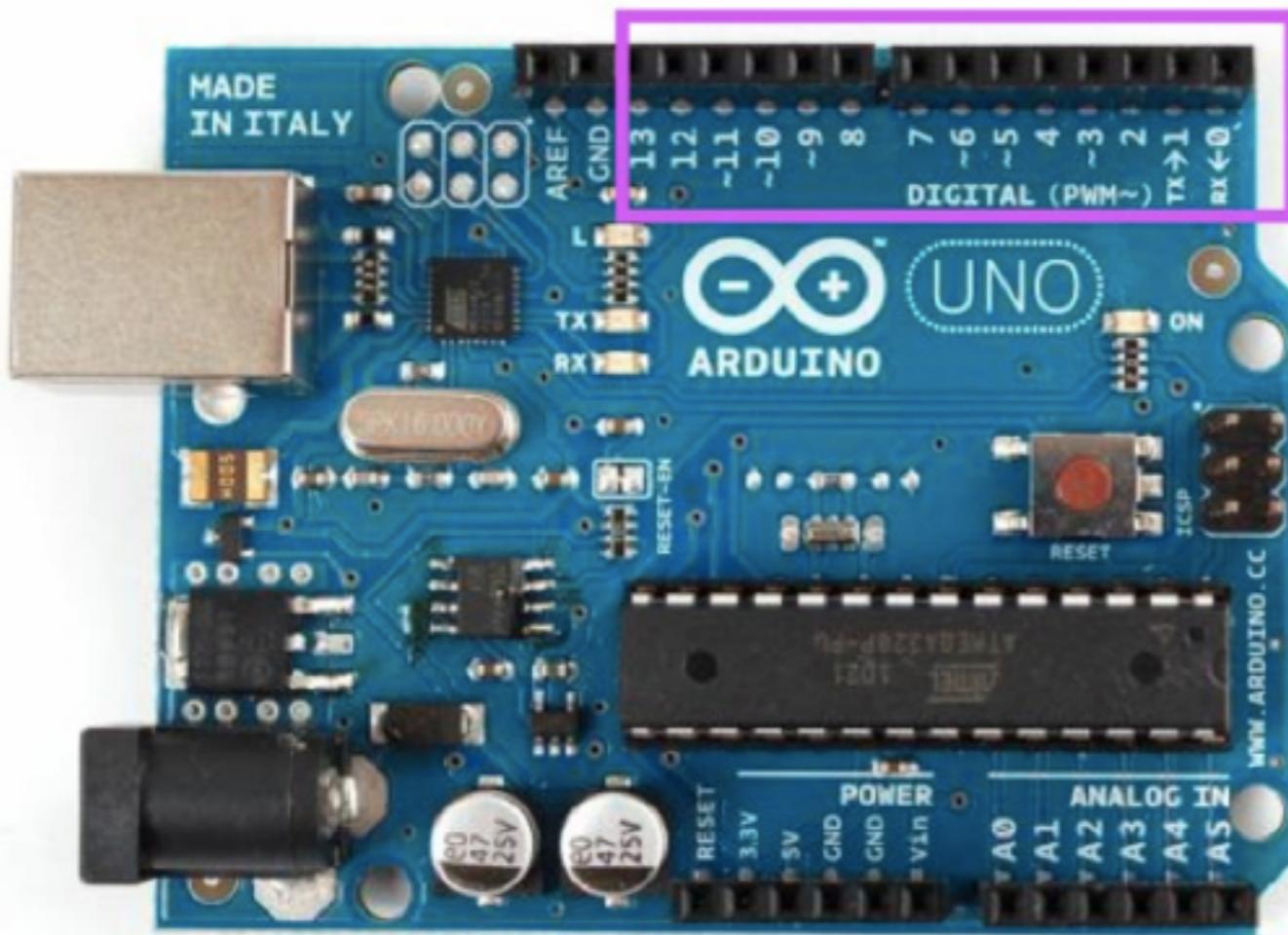
5 volts

LOW

0 volts

You can think of HIGH as on
and LOW as off

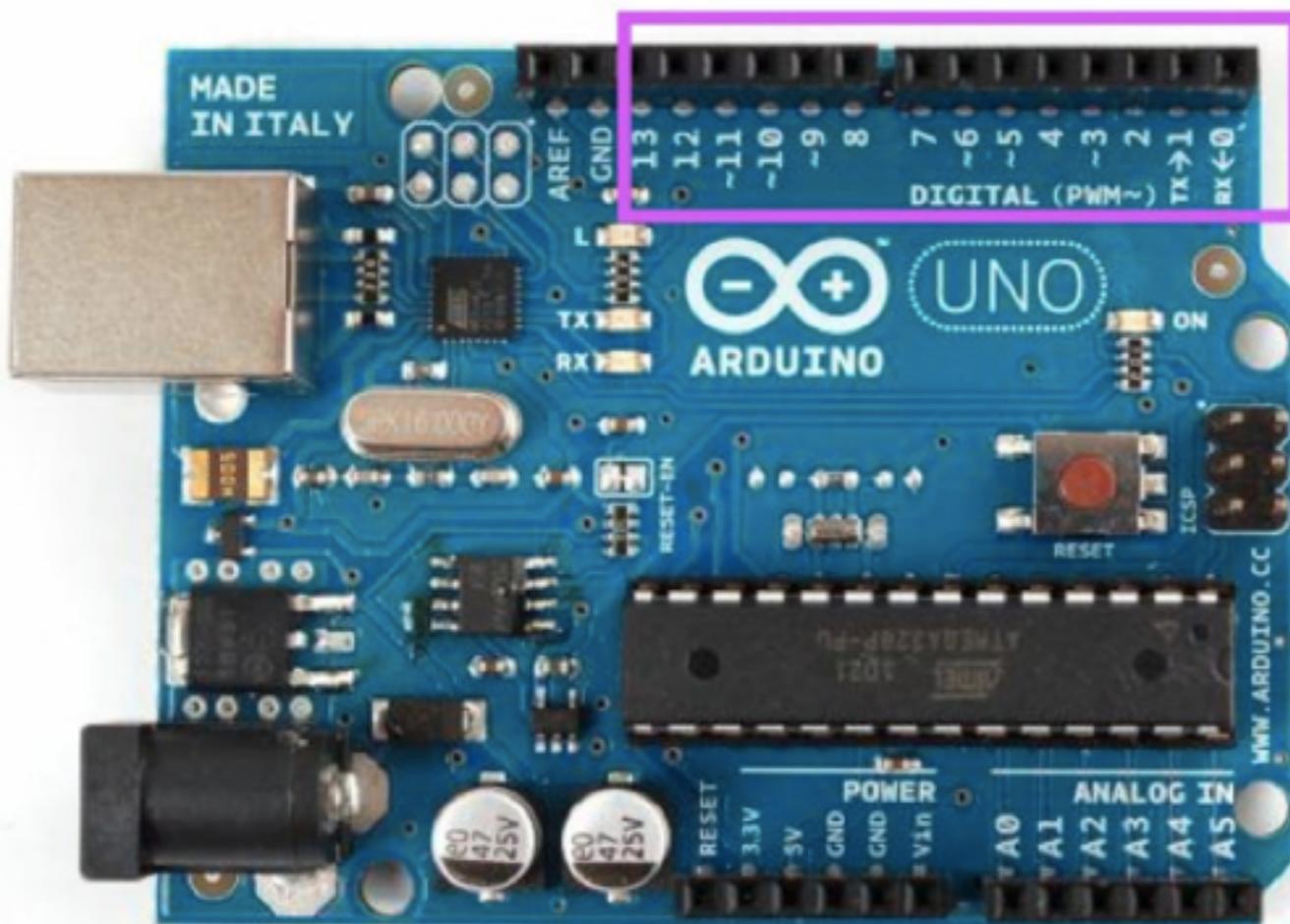
digital pin interactions



pinMode (pin, mode)

Sets the pin to be INPUT or OUTPUT
You don't have to do this every time but
it is GOOD PRACTICE

digital pin interactions



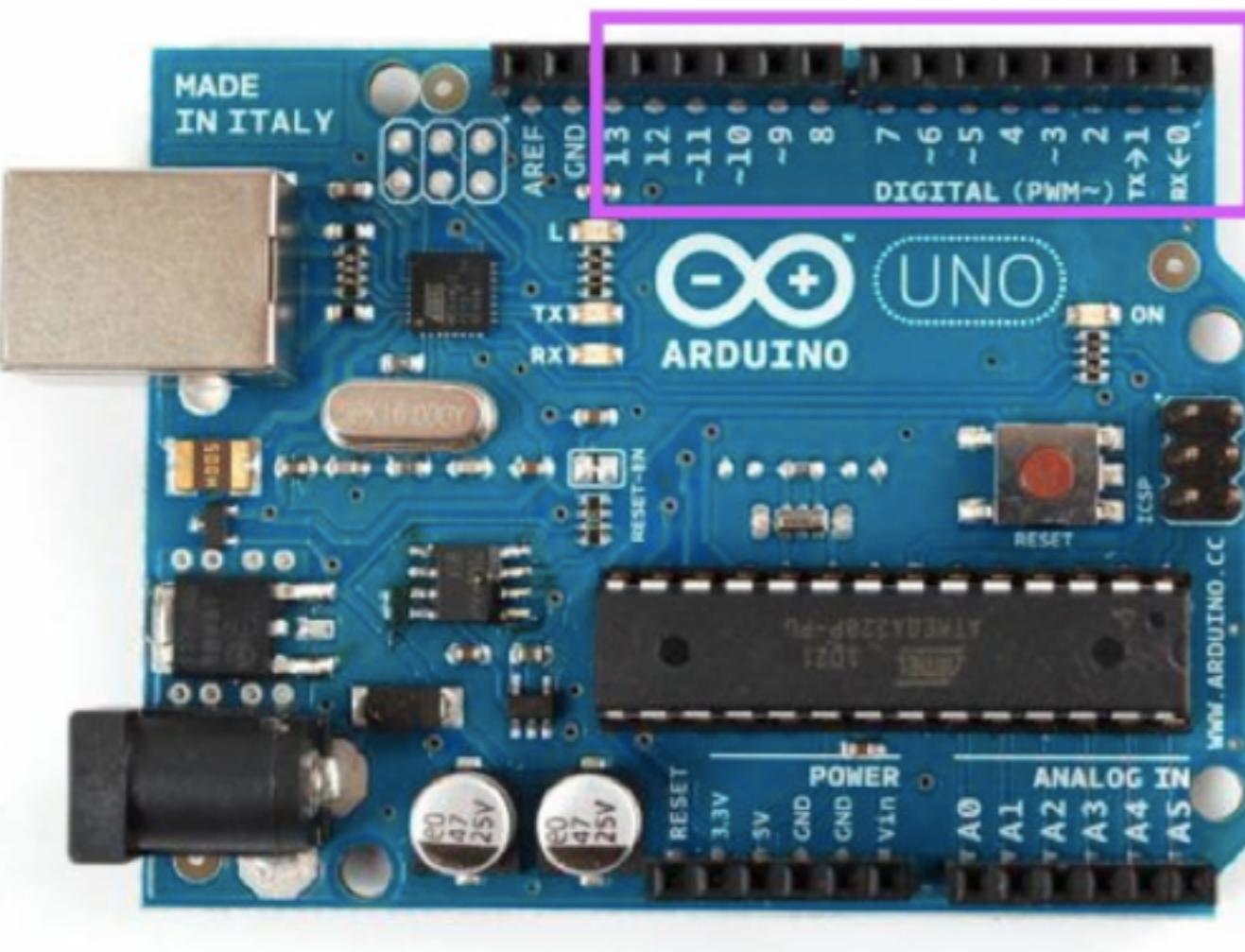
`pinMode(pin, mode)`

Sets the pin to be INPUT or OUTPUT
You don't have to do this every time but
it is GOOD PRACTICE

`digitalRead(pinNumber)`

Returns value from specified
For INPUT

digital pin interactions



`pinMode(pin, mode)`

Sets the pin to be INPUT or OUTPUT
You don't have to do this every time but
it is GOOD PRACTICE

`digitalRead(pinNumber)`

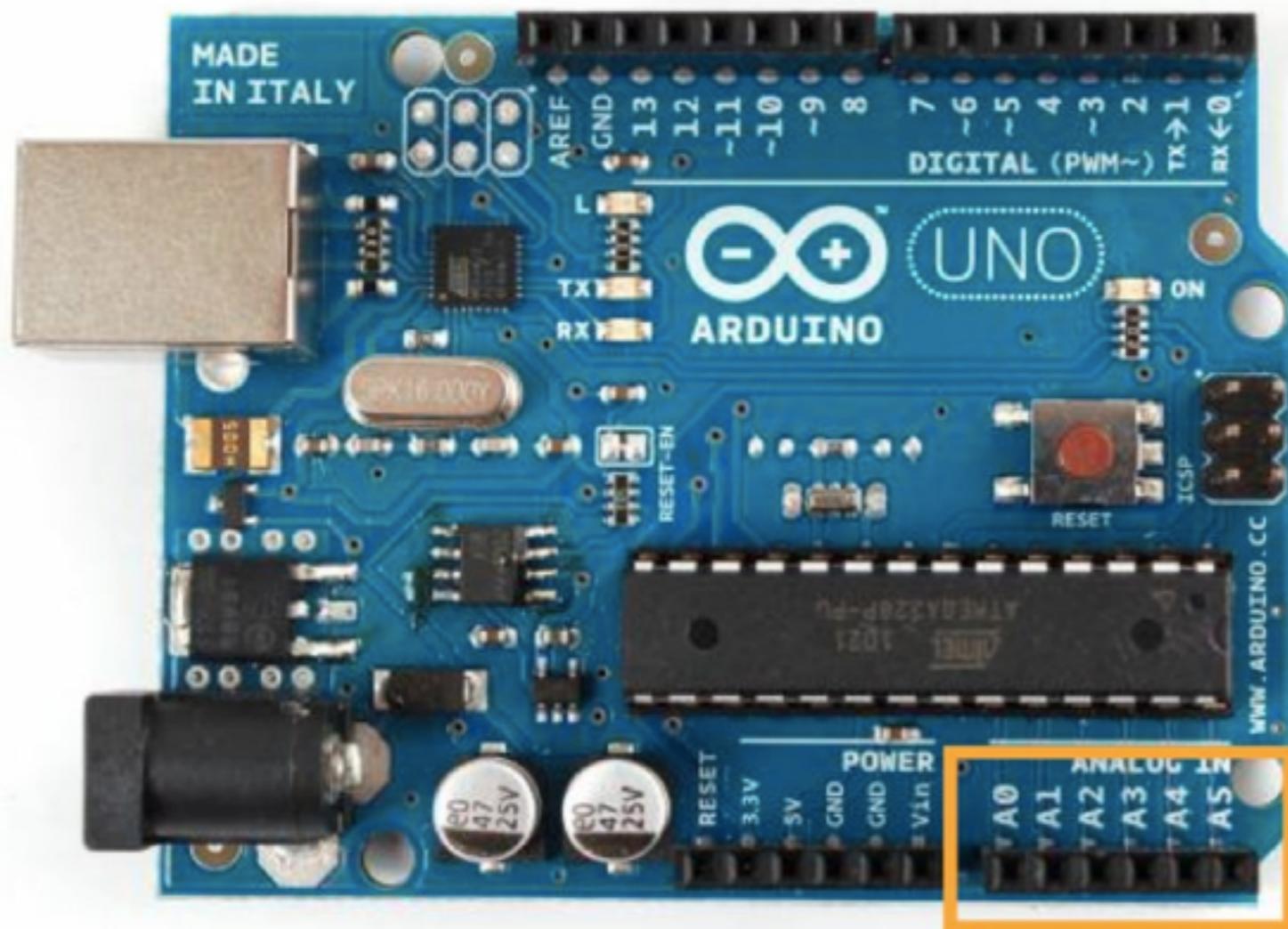
Returns value from specified
For INPUT

`digitalWrite(pinNumber, value)`

Writes a value to the pin. Here we
are talking HIGH (5V/on) or LOW
(0V/off)
For OUTPUT

blink blink

analog pin interactions



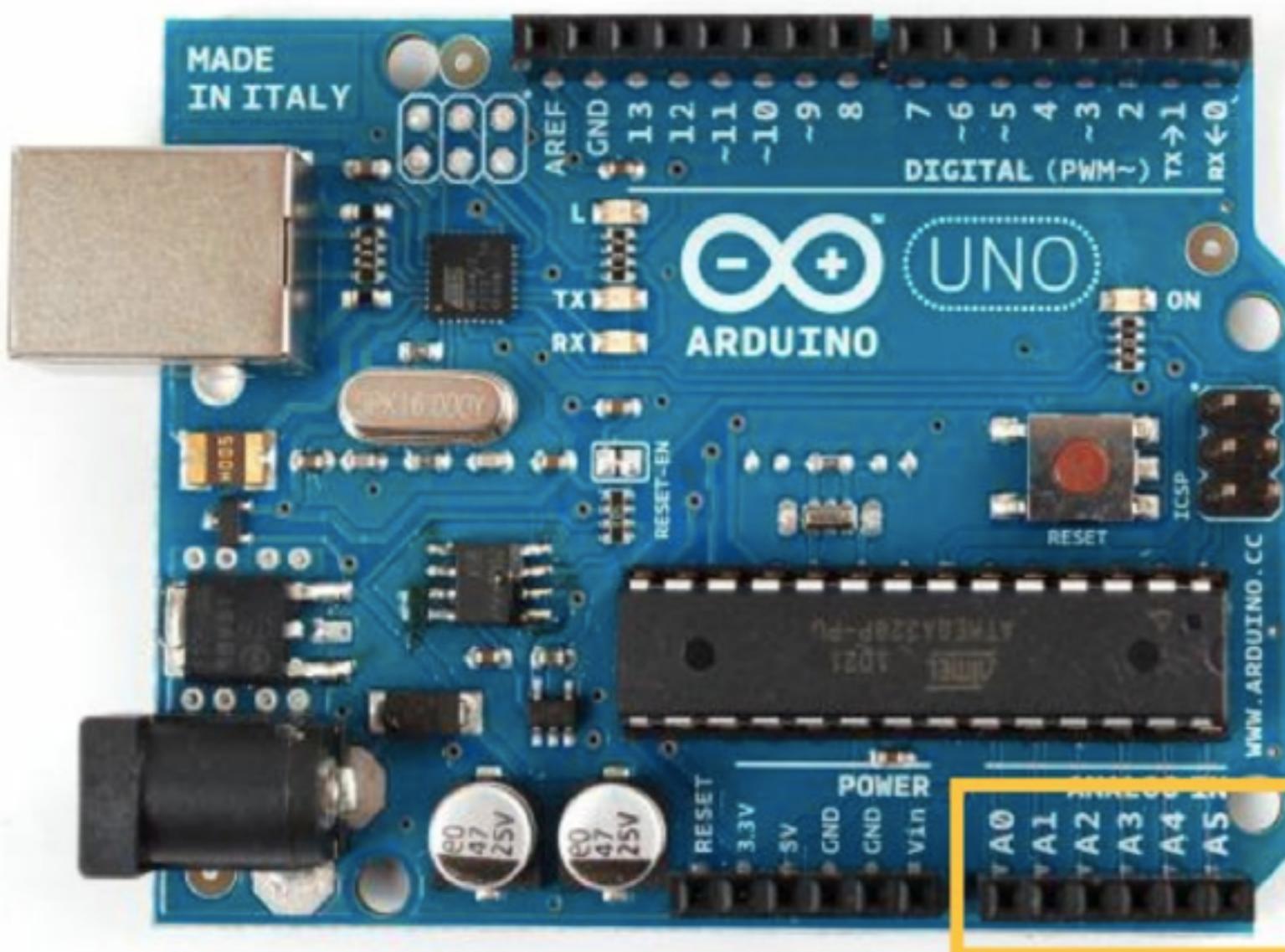
6 Analog Input pins

You can read or write a wide range of values

Read 0 - 1023

Written 0 - 255

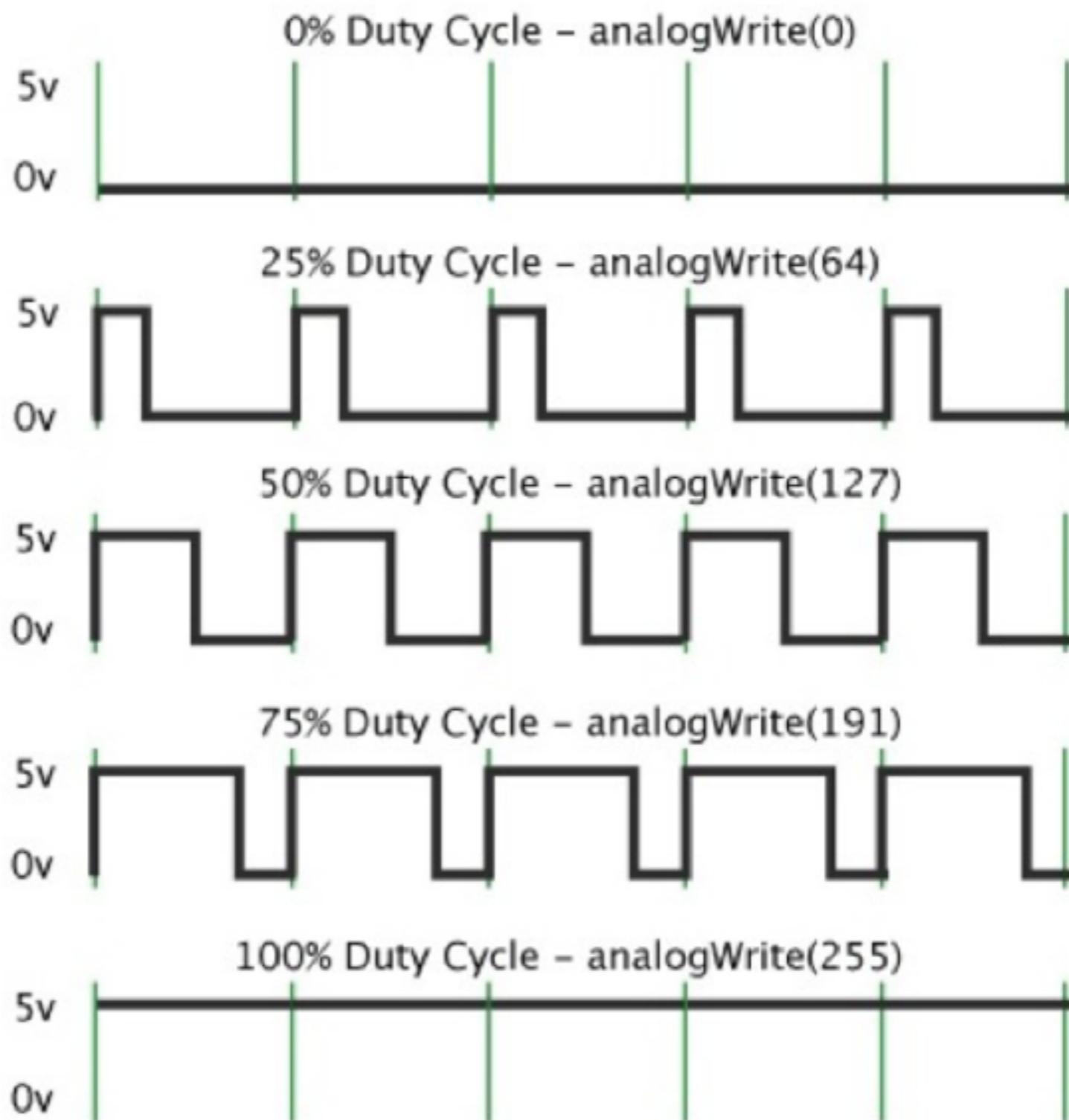
analog pin interactions



analogRead(pinNumber)

Reads value from specified
analog in pin
For INPUT

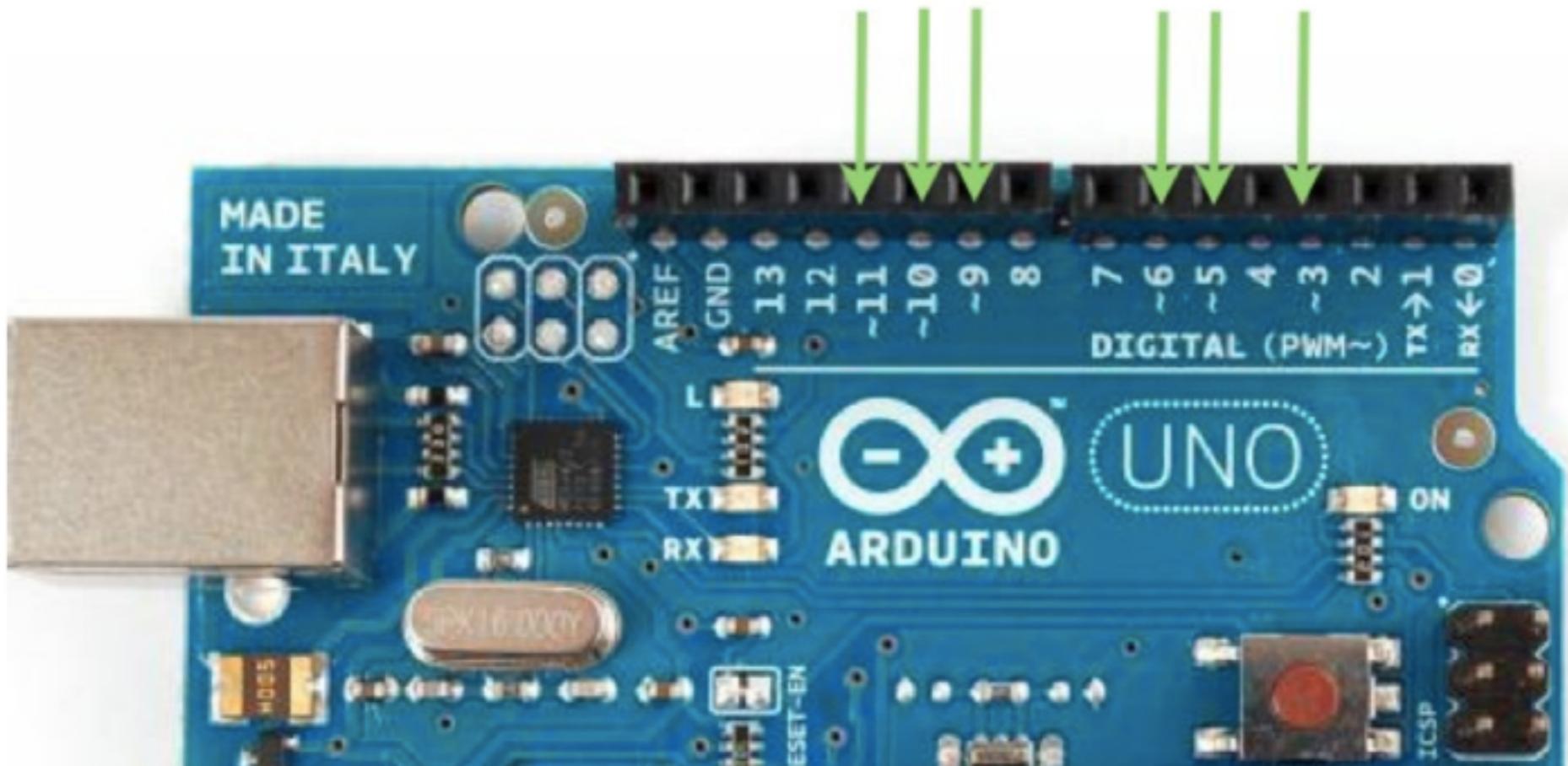
pulse width modulation



pulse width modulation

Only a few pins can execute this function:
3, 5, 6, 9, 10, and 11.

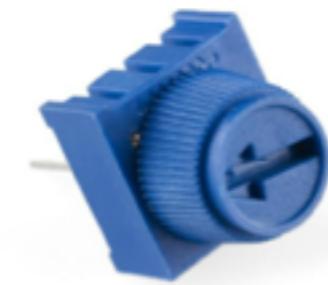
These are unique because they
can be **digital I/O** OR **analog out**.



inputs



Button



Potentiometer



Knob



Switch



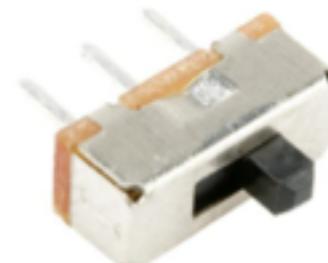
Knob



5 way switch



Arcade Button

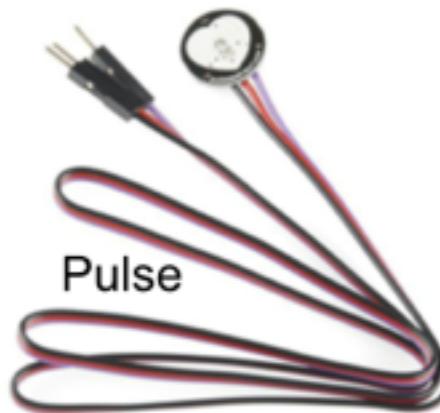


Switch



Switch

inputs



Pulse



Gas



Photocell



Range Finder



Color



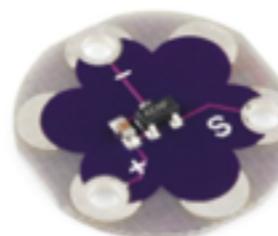
Humidity



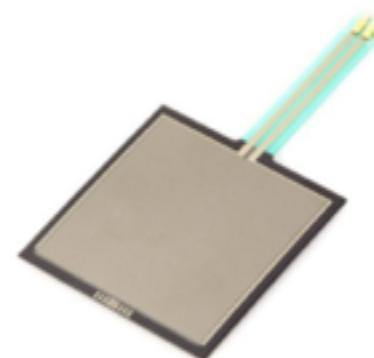
Motion



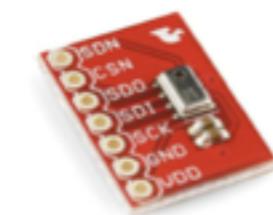
Pressure



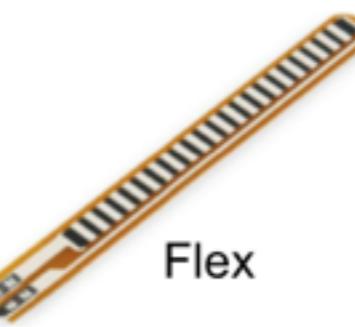
Temperature



Force



Barometric
Pressure



Light



pH

button push input

- Create a circuit that turns on an LED when a pushbutton is pressed.
- Use a pull-down resistor (10Kohm) — path of least resistance is to the pin when the button is pressed, but is pulled to ground instead of floating when the button is open

project: light language

- Design a simple light language. Build a system that communicates something in your designed language.
- Things to consider:
 - Color, speed, other properties
 - User interaction (buttons)
 - Tertiary devices (is it something only a camera / phone can see?)