

CLASS 3

DIGITAL I/O, ANALOG OUTPUT

PLAN FOR CLASS 3

- ▶ Blog review / Lab
- ▶ AMA from Class 2
- ▶ Partial quiz review
- ▶ Microcontrollers
- ▶ Sensors
- ▶ Programming terms and environment
- ▶ Digital Input and Output
- ▶ Analog Input

Bonus (if time) oscilloscope demos

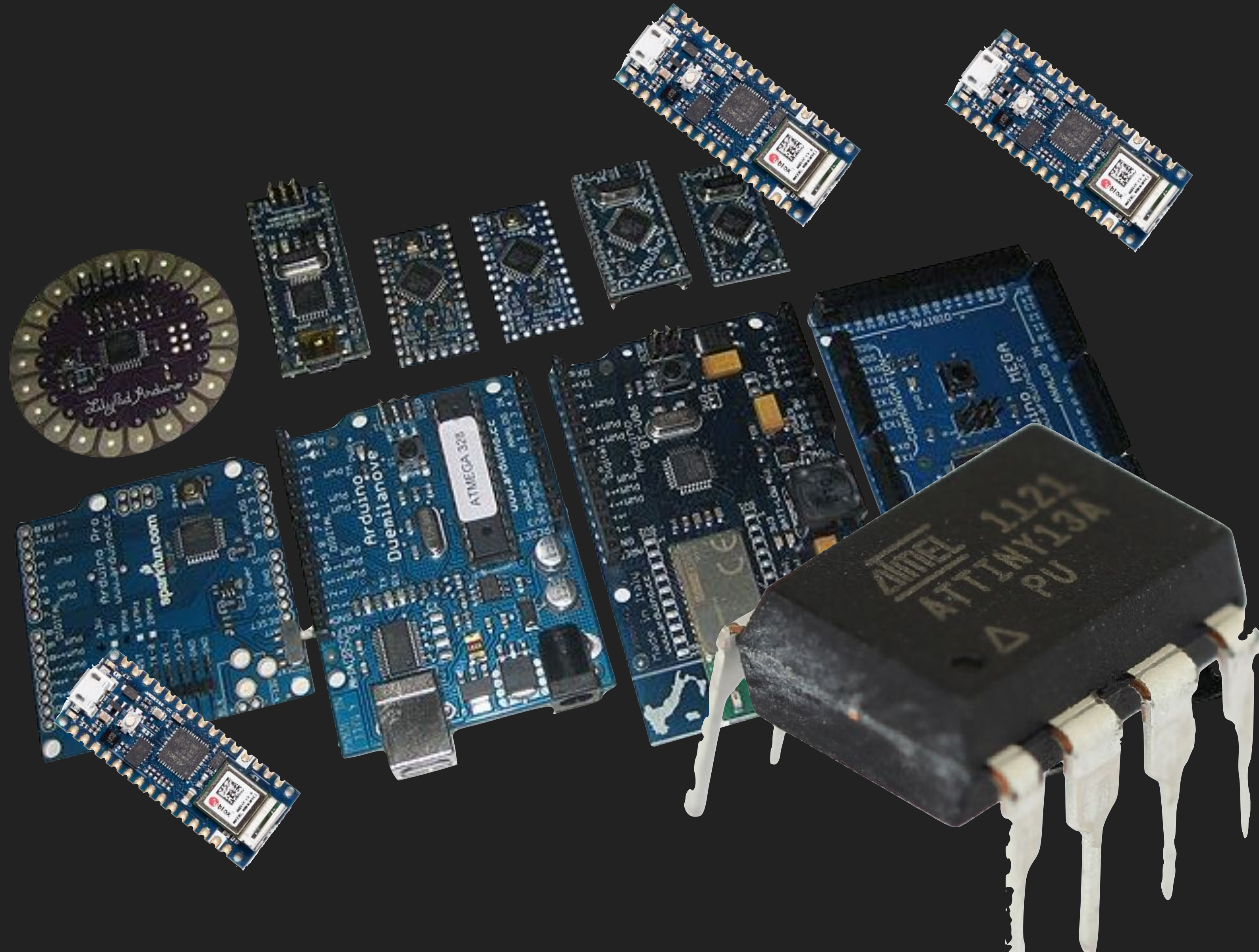
Class 4: Meets Monday
September 22

Room 411

3:10PM

MICROCONTROLLERS CAN BE LOTS OF DIFFERENT THINGS

(Reminder)



SENSORS

convert something in the world (smell, light, mass, motion, etc.) into something the microcontroller can read

- ▶ Voltage (usually)
 - ▶ digital = two states (1-bit) above or below a threshold
 - ▶ analog = many states (2+ bits) mapped to many levels
- ▶ Digital data (covered later)

PROGRAMMING TERMS AND ENVIRONMENT

An **IDE (Integrated Development Environment)** combines everything you need:

- text editor, compiler, libraries, uploader
- Arduino IDE has tools for specifying board, adding libraries, finding examples

Programming

- C / C++
- **Strongly typed** language (a big difference from JS)
- generally, since we're "closer" to the machine, we need to be aware a bit more how it works (e.g. bits and bytes)

LEARNING A LANGUAGE

Learn the **syntax** and **reserved words**

- case matters, semicolons matter, etc.

Learn how **data is handled**

- Variables, types

Learn how to **organize code**

- Functions (objects)
- Order of operations

Understand "scope"

Learn **flow control**

- for, if/then, while

Learn operators

- =, ==, !, &&, ||

LEARNING ^{ANY} A LANGUAGE

Type code in!!!

You'll learn faster than cut-n-paste

PCOMP ENVIRONMENT

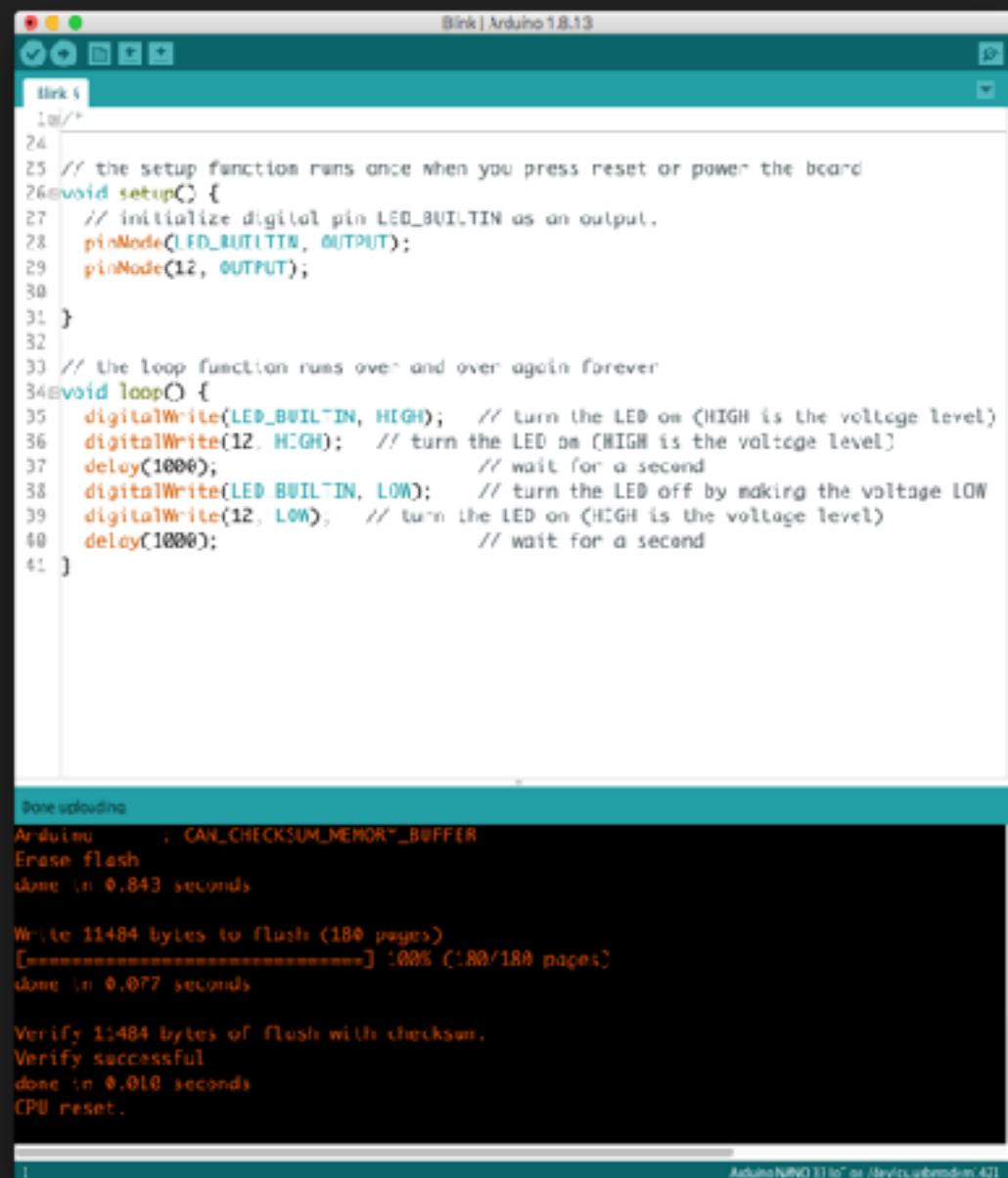
Most development environments will have similar elements

I.D.E.

LIBRARIES

HARDWARE

SUPPORT



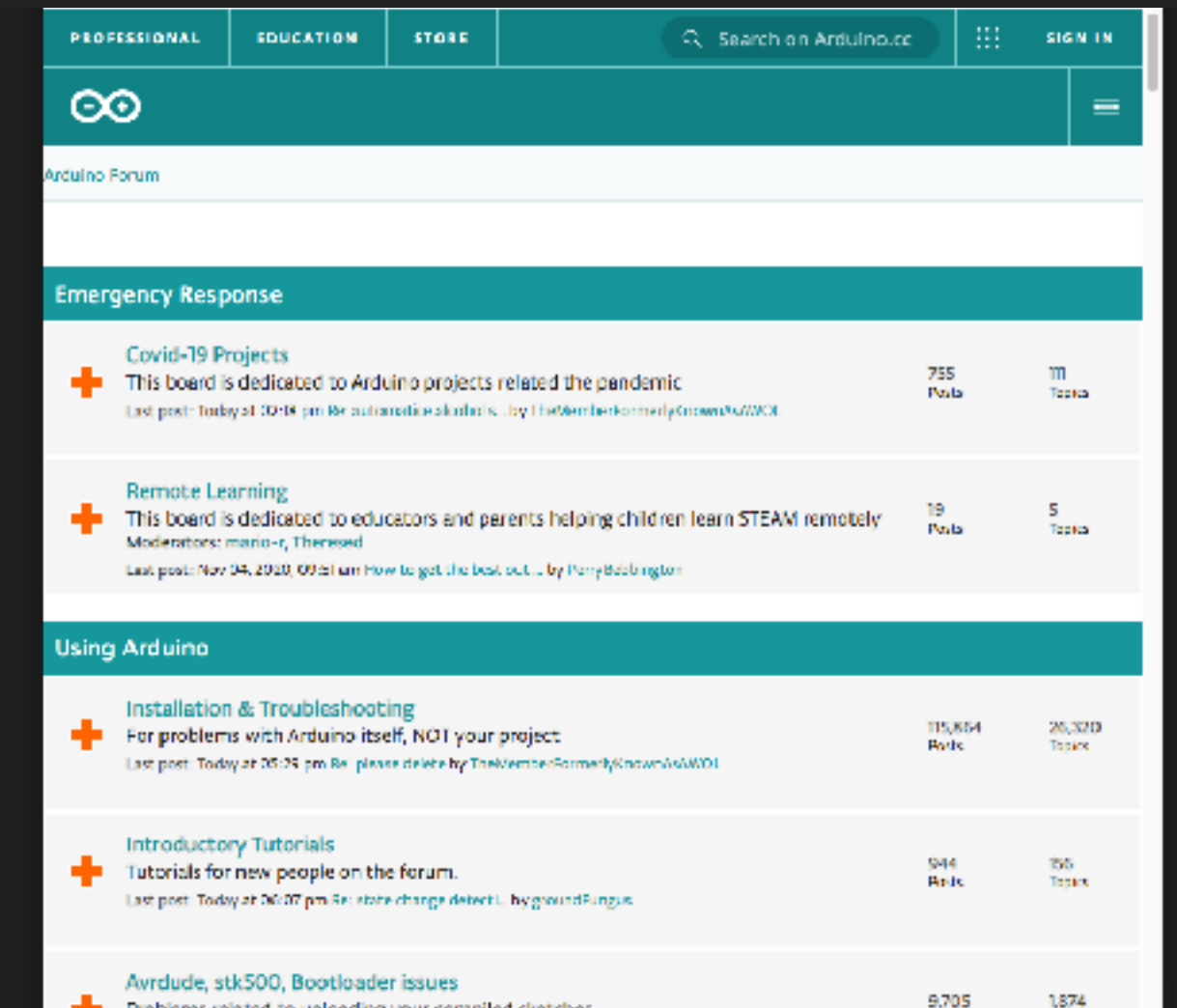
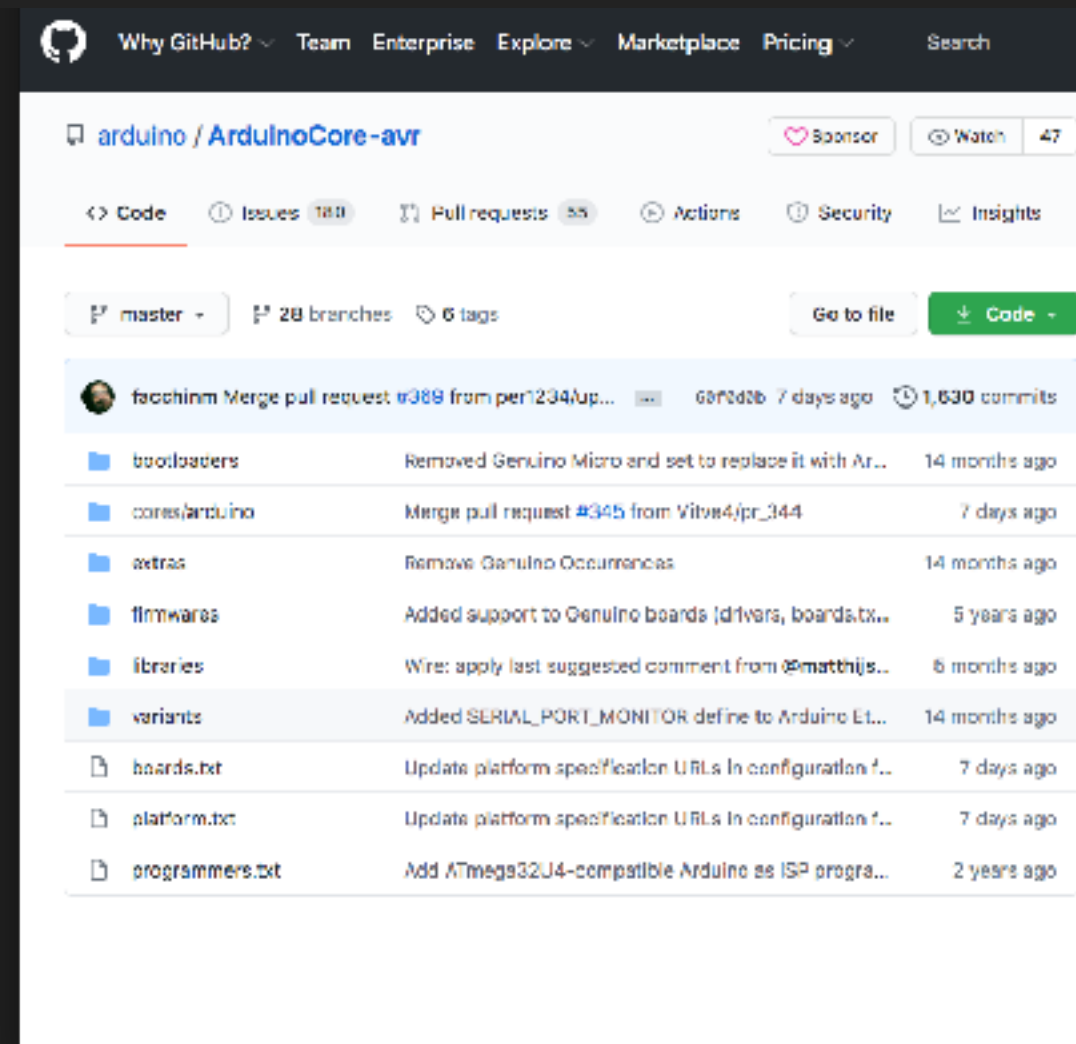
```
18/*
24
25 // the setup function runs once when you press reset or power the board
26void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29   pinMode(12, OUTPUT);
30
31 }
32
33 // the loop function runs over and over again forever
34void loop() {
35   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
36   digitalWrite(12, HIGH); // turn the LED on (HIGH is the voltage level)
37   delay(1000); // wait for a second
38   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
39   digitalWrite(12, LOW); // turn the LED on (HIGH is the voltage level)
40   delay(1000); // wait for a second
41 }
```

Serial Monitor Output:

```
Arduino: CAN_CHECKSUM_MEMORY_BUFFER
Erase flash
done in 0.843 seconds

Write 11484 bytes to flash (180 pages)
[=====] 100% (180/180 pages)
done in 0.077 seconds

Verify 11484 bytes of flash with checksum.
Verify successful
done in 0.010 seconds
CPU reset.
```



IDE

Compile +
upload to board

Code

Stats and status



```
Class 3 Demo | Arduino IDE 2.0.0
Arduino NANO 33 IoT

Class 3 Demo.ino
1  const int blueButton = 4, //it's a nice idea to label things semantically. Avoid magic numbers.
2      greenButton = 3,
3      yellowButton = 2,
4      pot = A7,
5      fsr = A6,
6      blueLED = 5;
7
8  void setup() {
9      // put your setup code here, to run once:
10     pinMode(blueButton, INPUT);
11     pinMode(yellowButton, INPUT);
12     pinMode(greenButton, INPUT);
13     pinMode(blueLED, OUTPUT);
14     Serial.begin(9600);
15
16     //flash the LED to show the program is starting
17     for (int i=0; i<10; i++) {
18         digitalWrite(blueLED, i%2==0); //Extra credit: what's happening here????
19         delay(100);
20     }
21 }
22
23 void loop() {
24     outputForPlotter();
25     delay(100);
26 }
27
28 //output data in a format that will plot nicely
29 void outputForPlotter() {
30     Serial.print("B:");
31     if (digitalRead(blueButton)) Serial.print("100,");
32     else Serial.print("0,");
33
34     Serial.print("G:");
35     if (digitalRead(greenButton)) Serial.print("300,");
36     else Serial.print("200,");
37
38     Serial.print("Y:");
39     if (digitalRead(yellowButton)) Serial.print("500,");
40     else Serial.print("400,");
41 }
```

Output Serial Monitor

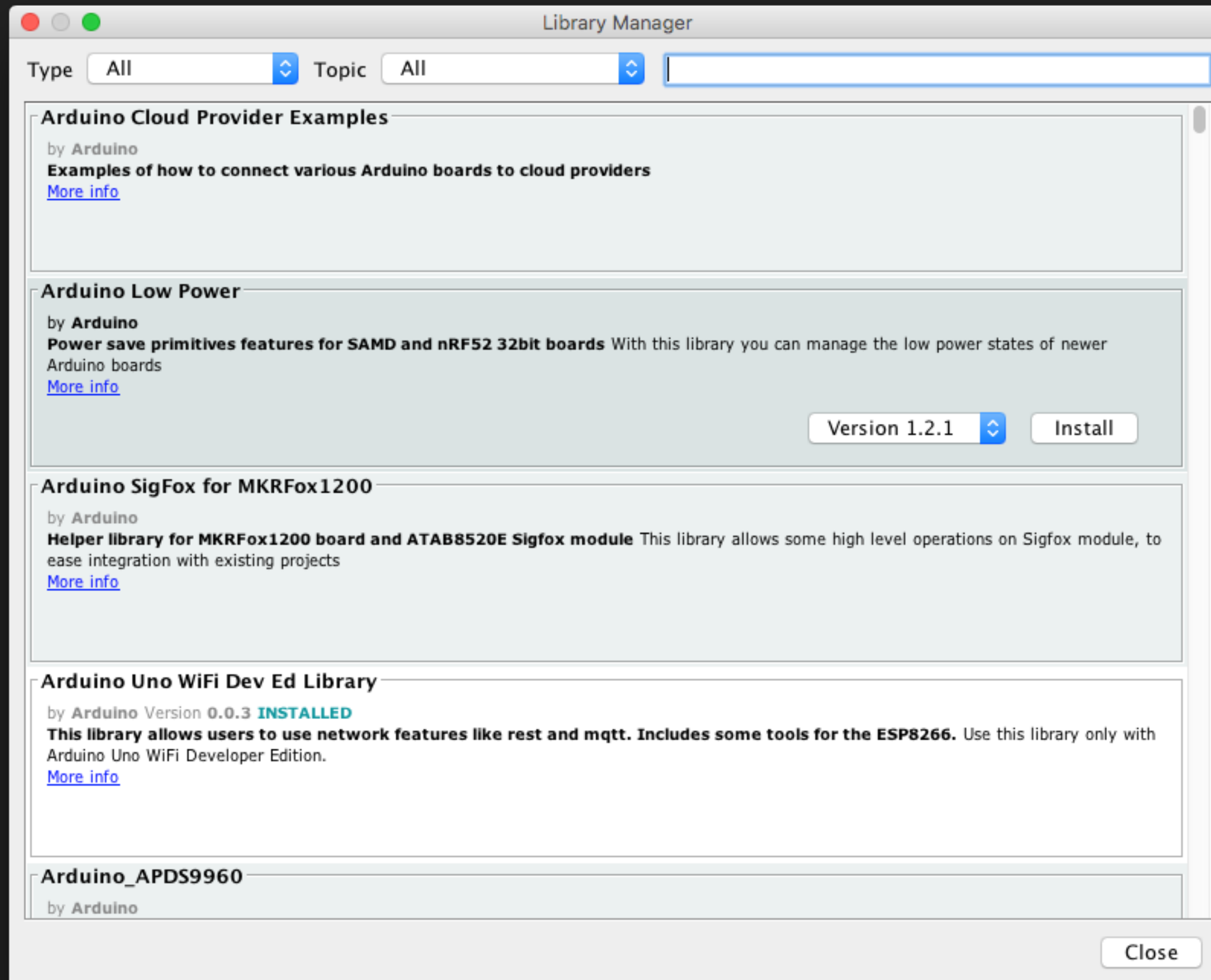
```
Write 12254 bytes to flash (192 pages)
[=====] 33% (64/192 pages)
[=====] 66% (128/192 pages)
[=====] 100% (192/192 pages)
done in 0.083 seconds

Verify 12254 bytes of flash with checksum.
Verify successful
done in 0.010 seconds
CPU reset.
```

Ln 50, Col 20 UTF-8 Arduino NANO 33 IoT on /dev/cu.usbmodem11401 2

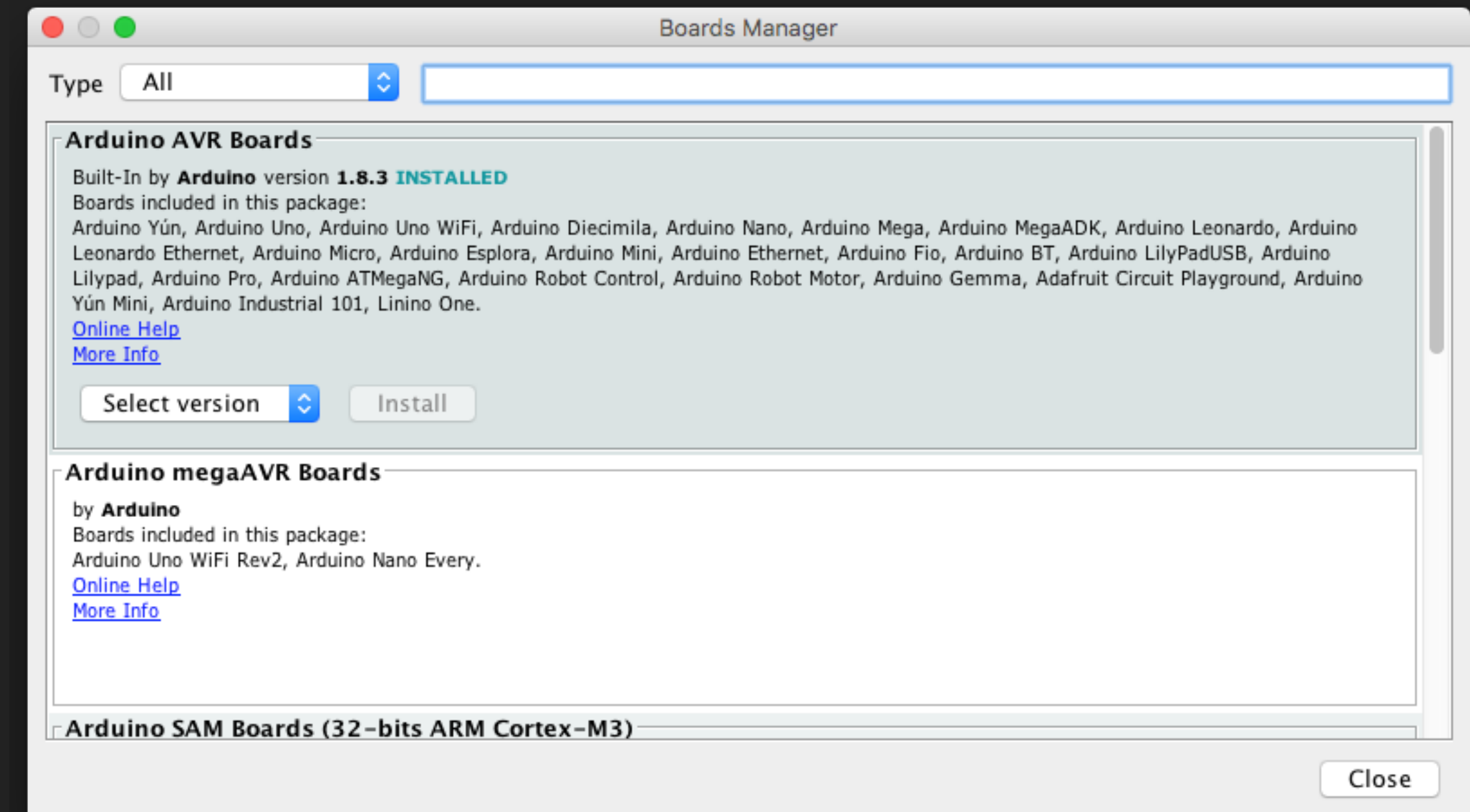
Integrated Development Environment links your code with core libraries, compiler, and uploading tool chains.

Library Manager*



Install software modules for additional functionality

Board Manager*



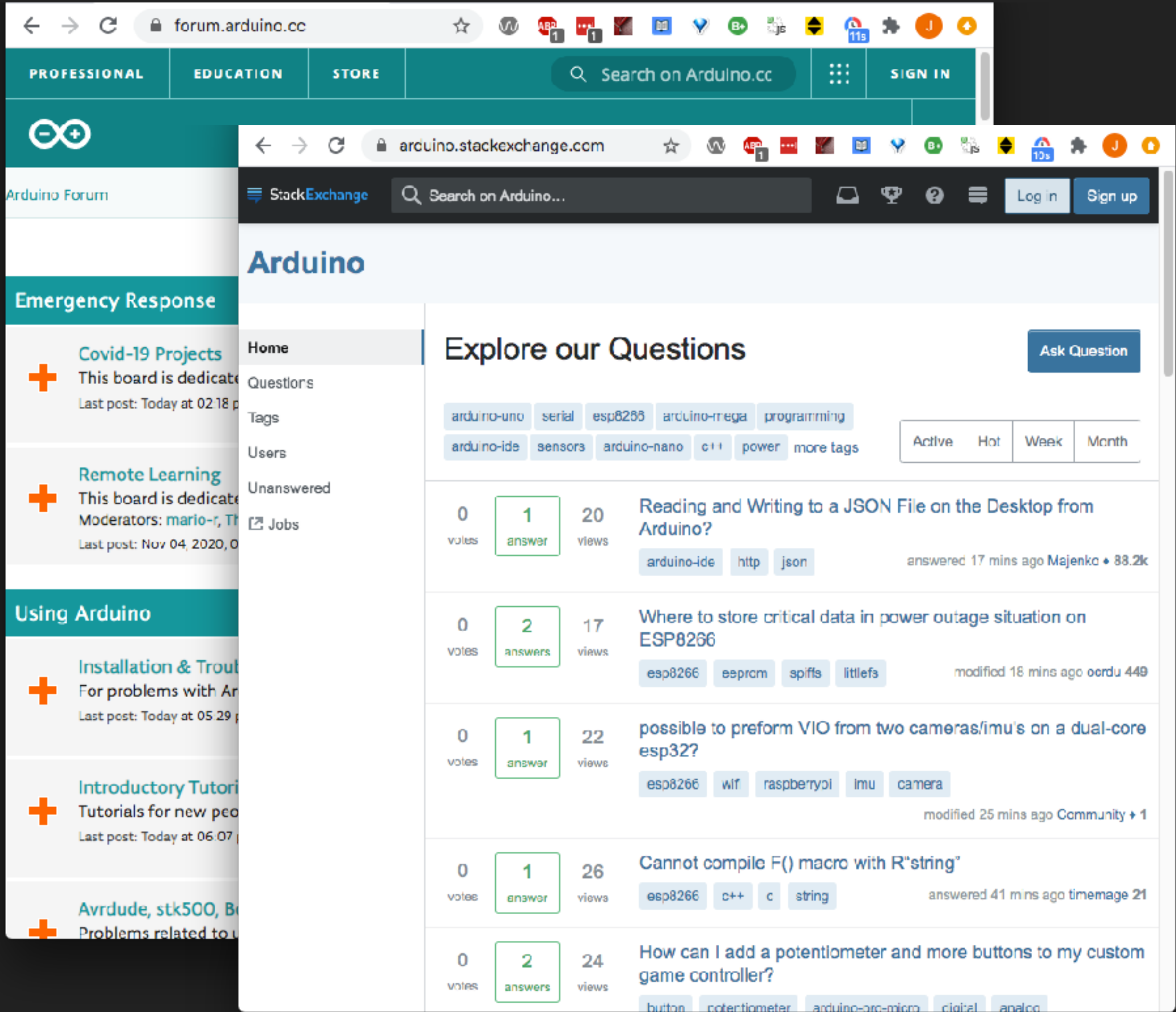
* IN SIDEBAR FOR ARDUINO IDE 2.0

Install hardware definitions to talk to other microcontrollers

SUPPORT

In addition to PCOMP syllabus

Forums (lots!)



Examples (lots!)

Built-in Examples	Examples from Custom Libraries
01.Basics	AccelStepper
02.Digital	Adafruit BNO055
03.Analog	Adafruit BusIO
04.Communication	Adafruit Fingerprint Sensor Library
05.Control	Adafruit GFX Library
06.Sensors	Adafruit HMC5883 Unified
07.Display	Adafruit INA219
08.Strings	Adafruit IO Arduino
09.USB	Adafruit LED Backpack Library
10.StarterKit_BasicKit	Adafruit Motor Shield V2 Library
11.ArduinoISP	Adafruit MQTT Library
Examples for any board	Adafruit NeoPixel
Adafruit Circuit Playground	Adafruit SHARP Memory Display
Arduino_LSM6DS3	Adafruit SleepyDog Library
ArduinoHttpClient	Adafruit SSD1306
Bridge	Adafruit TinyFlash
Ethernet	Adafruit TMP007 Library
Firmata	Adafruit VS1053 Library
LiquidCrystal	Adafruit-LED-Backpack
SD	ArdOSC
Servo	ArduinoBLE
Stepper	ArduinoMDNS
Temboo	ArduinoSound
WiFi101	Button
WiFININA	elapsedMillis
RETIRED	Encoder
Examples for Arduino NANO 33 IoT	EWMA
I2S	FastLED
SAMD_AnalogCorrection	FuGPS Library
SAMD_BootloaderUpdater	Gaussian
SBU	jeelib-master
SDU	LinkedList
SEU	Low-Power
SNU	MIDIUSB
SPI	MsTimer2
SSU	NewliquidCrystal
USBHost	OSC
Wire	Playtune
Examples from Custom Libraries	Ramp
AccelStepper	RTCLib
Adafruit BNO055	RTCZero
Adafruit BusIO	SimpleKalmanFilter
Adafruit Fingerprint Sensor Library	SparkFun Micro OLED Breakout
Adafruit GFX Library	Tasker
Adafruit HMC5883 Unified	TimeLord
Adafruit INA219	TinyWireM
Adafruit IO Arduino	U8g2
Adafruit LED Backpack Library	U8g2_for_Adafruit_GFX
Adafruit Motor Shield V2 Library	uTimerLib
Adafruit MQTT Library	WiFININA_Generic
Adafruit NeoPixel	WiFiWebServer
	INCOMPATIBLE

SUPPORT

New 2.0 IDE adds:

- Code completion (Yay!)
- In-circuit debugging (to be explored...)
- More serial plotting features



The screenshot displays the Arduino IDE 2.0.0 interface. The top bar shows the project name 'Class 3 Demo' and the board 'Arduino NANO 33 IoT'. The main editor window contains the following C++ code:

```
1  const int blueButton = 4, //it's a nice idea to label things semantically. Avoid magic numbers.
2      greenButton = 3,
3      yellowButton = 2,
4      pot = A7,
5      fsr = A6,
6      blueLED = 5;
7
8  void setup() {
9      // put your setup code here, to run once:
10     pinMode(blueButton, INPUT);
11     pinMode(yellowButton, INPUT);
12     pinMode(greenButton, INPUT);
13     pinMode(blueLED, OUTPUT);
14     Serial.begin(9600);
15
16     //flash the LED to show the program is starting
17     for (int i=0; i<10; i++) {
18         digitalWrite(blueLED, i%2==0); //Extra credit: what's happening here????
19         delay(100);
20     }
21 }
22
23 void loop() {
24     outputForPlotter();
25     delay(100);
26 }
27
28 //output data in a format that will plot nicely
29 void outputForPlotter() {
30     Serial.print("B:");
31     if (digitalRead(blueButton)) Serial.print("100,");
32     else Serial.print("0,");
33
34     Serial.print("G:");
35     if (digitalRead(greenButton)) Serial.print("300,");
36     else Serial.print("200,");
37
38     Serial.print("Y:");
39     if (digitalRead(yellowButton)) Serial.print("500,");
40     else Serial.print("400,");
41 }
```

The bottom panel shows the 'Serial Monitor' output, which displays the progress of writing the code to the flash memory:

```
Write 12264 bytes to flash (192 pages)

[=====] 33% (64/192 pages)
[=====] 66% (128/192 pages)
[=====] 100% (192/192 pages)
done in 0.083 seconds

Verify 12264 bytes of flash with checksum.
Verify successful
done in 0.010 seconds
CPU reset.
```

The status bar at the bottom indicates the current position in the code: 'Ln 50, Col 20' and the board: 'Arduino NANO 33 IoT on /dev/cu.usbmodem11401'.

BINARY

BINARY DECIMAL

Review

“Place-value” number systems

You have some number of symbols (e.g. ‘0’-‘9’)

You assemble those symbols to represent a value. The place of the symbol determines its contribution to the total

“Thousands” place — **5723** — “Ones” place

“Hundreds” place “Tens” place

$$=$$
$$5 * 1000 + 7 * 100 + 2 * 10 + 3 * 1$$
$$=$$
$$5 * 10^3 + 7 * 10^2 + 2 * 10^1 + 3 * 10^0$$

In other words, the “place value” of the symbol is:

the number of possible symbols raised to the power of it’s place in the string of symbols

BINARY

1 COIN, 2 STATES:



Heads



Tails

BINARY

2 COIN, 4 STATES:



HH



HT



TH



TT

BINARY

Each additional coin doubles the number of possible states.

With 3 coins there are 8 states:



Previous states, plus Heads



Previous states, plus Tails

BINARY

Put another way, the number of states is:

$2^{\text{number of coins}}$



Previous states, plus Heads



Previous states, plus Tails

$$2^3 = 8$$

BINARY

Instead of coins, computers use bits, but the idea is the same.

Decimal	Binary	
0	0 0 0 0	
1	0 0 0 1	1 bit, $2^1 = 2$ combos
2	0 0 1 0	
3	0 0 1 1	2 bits, $2^2 = 4$ combos
4	0 1 0 0	
5	0 1 0 1	
6	0 1 1 0	
7	0 1 1 1	3 bits, $2^3 = 8$ combos
8	1 0 0 0	
9	1 0 0 1	
10	1 0 1 0	
11	1 0 1 1	
12	1 1 0 0	
13	1 1 0 1	
14	1 1 1 0	
15	1 1 1 1	4 bits, $2^4 = 16$ combos

BINARY

Microcontrollers like the Nano often have 8-bit PWM output resolution*:

$$8 \text{ bits, } 2^8 = 256 \text{ combos} = [0...255]$$

...and 10-bit analog input resolution*:

$$10 \text{ bits, } 2^{10} = 1024 \text{ combos} = [0...1023]$$

BOOLEAN / BINARY LOGIC

1 == HIGH == TRUE

0 == LOW == FALSE

HOW A MICROCONTROLLER TOUCHES THE WORLD

