Arduino 101

Presented to PCR Extra 2026 North

Seth Neumann

May 1, 2026

What can we do to help you get started?

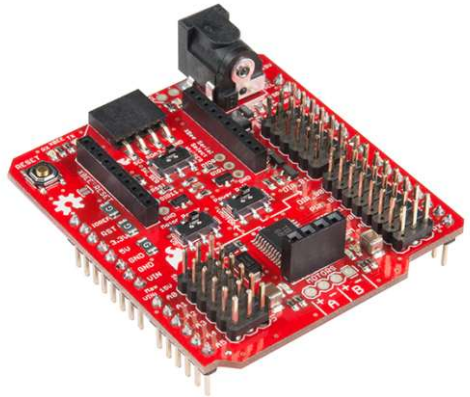
- Introduce the Arduino integrated development environment (IDE)
- Show you how to select the proper board and COM port, and download a program
- Basic programming introduction
- Provide sample programs you can reuse or build upon
- Minimal theory, maximum hands-on

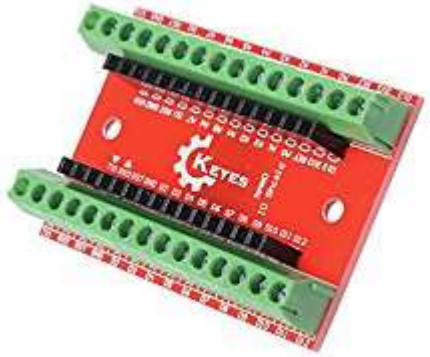
What is Arduino?

- Programmable controller
- Analog and Digital I/O
- Program written in simple language, compiled, and uploaded into the Arduino
- Arduino board then runs the program, computer connection no longer needed
- Don’t worry, Arduino was designed in Italy for use by kinetic artists. If they can do it, so can you!

Hardware

- Many boards out there, basic functionality the same
- Terminal blocks and headers allow expansion and connections
- Boards can be powered from a 9V wall wart or stepped down 12V layout power



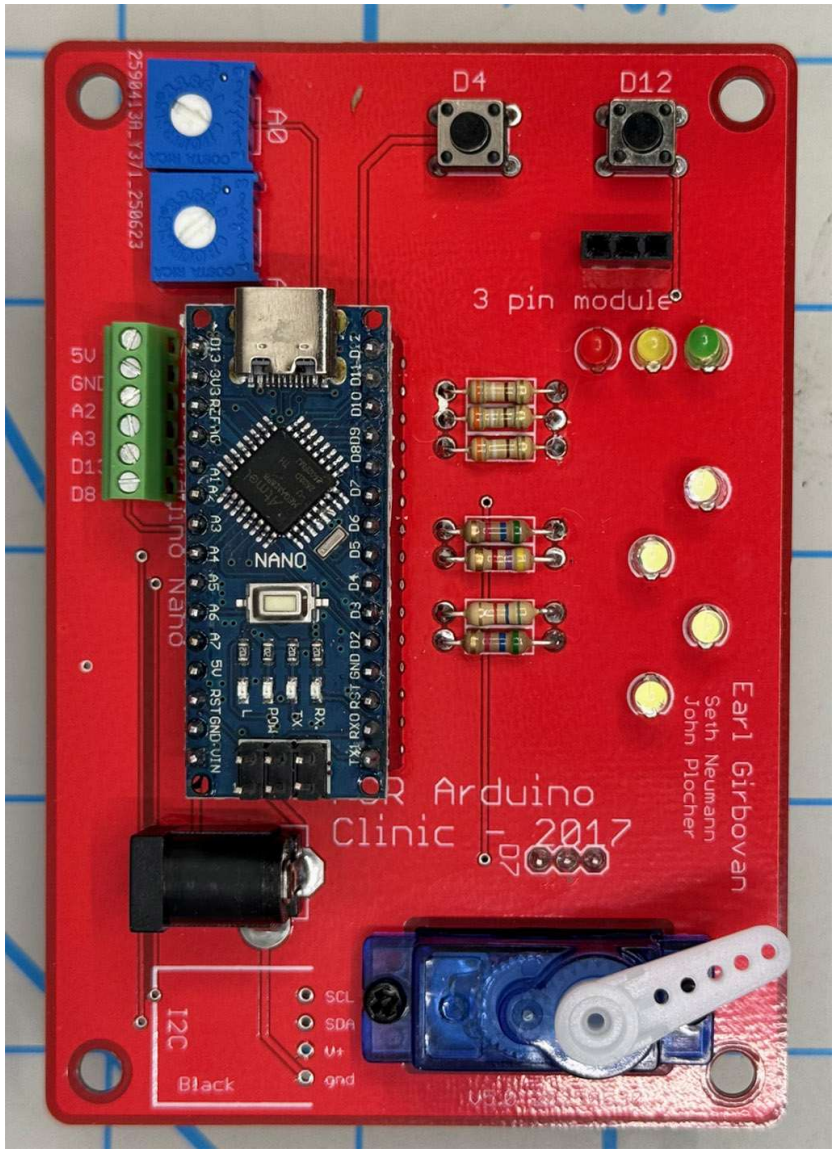


Where to Get It All

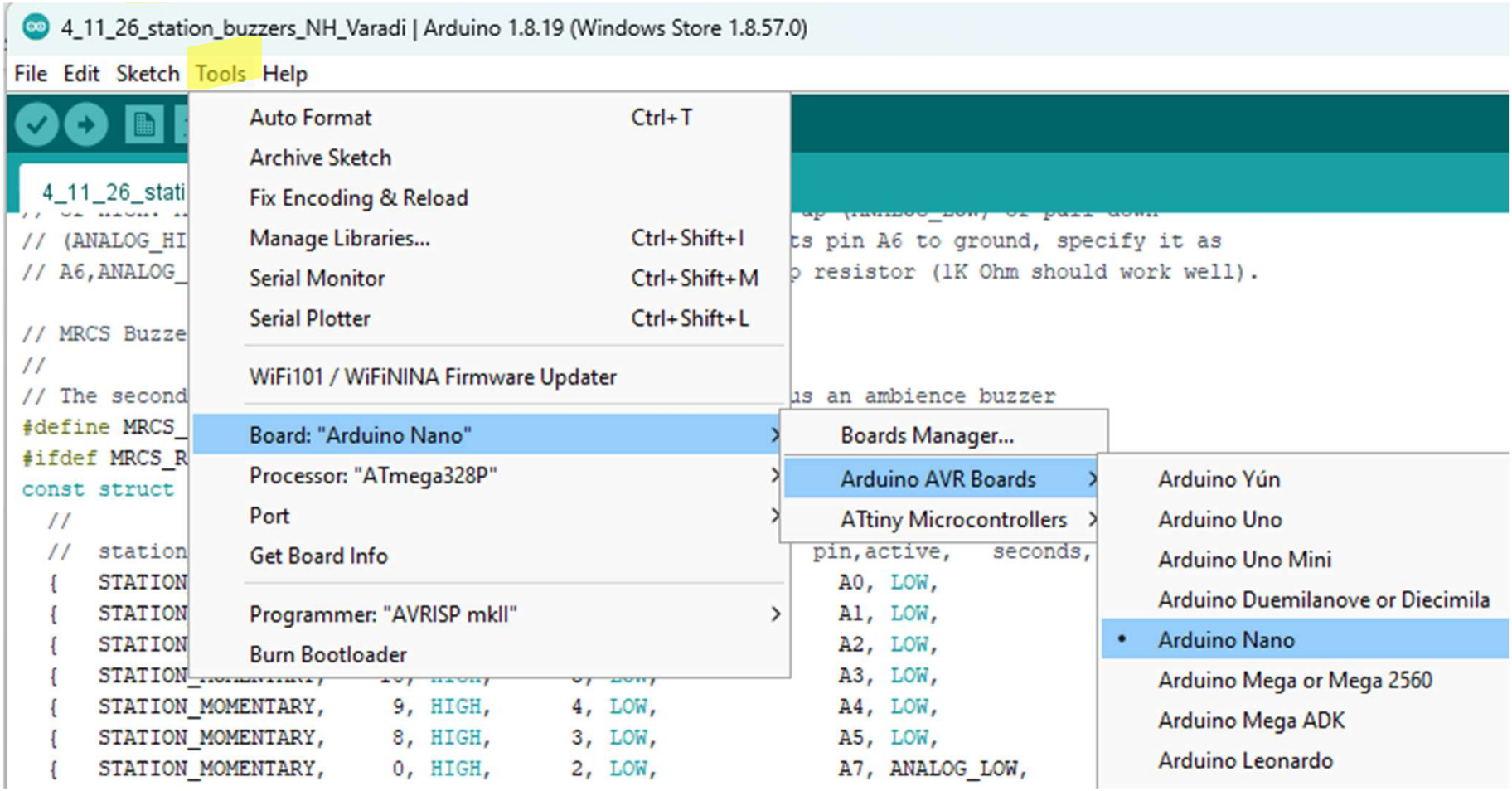
- Amazon
- Asian suppliers like AliExpress
- eBay sellers
- Jameco
- DigiKey, Newark, Mouser
- Sparkfun
- Adafruit

Clinic Circuit Board

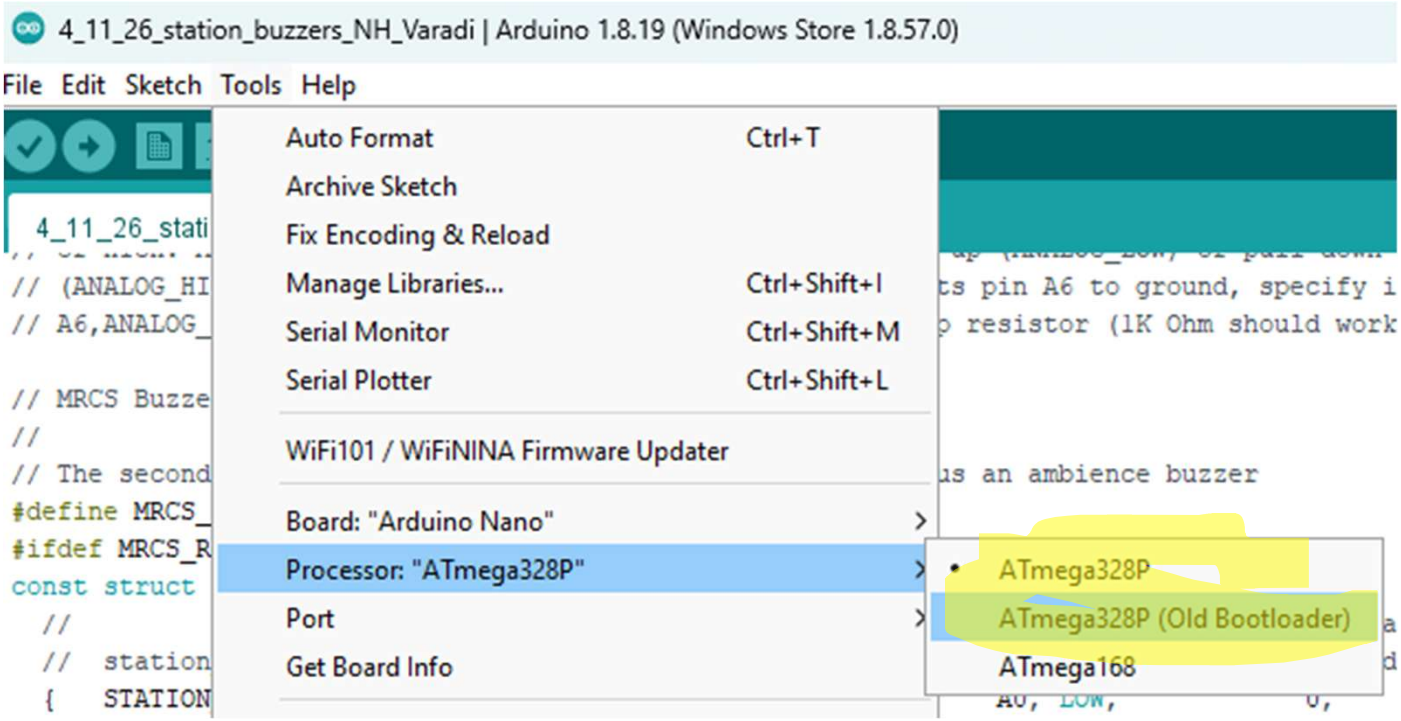
- Arduino Nano based PCB
- Nano has:
  - 14 Digital I/Os
  - 8 Analog inputs (sort of)
  - I2C Header (used 2 Analog lines)
- Clinic PCB adds an assortment of LEDs, pots, pushbuttons and a servo motor
- Jack for power so you can run without the USB cable *
- Unused pins brought out to a terminal block
- Schematic is in your workbook
- Most components can be interfaced directly to the Arduino, just need current limiting resistors for LEDs



Talking to your Nano

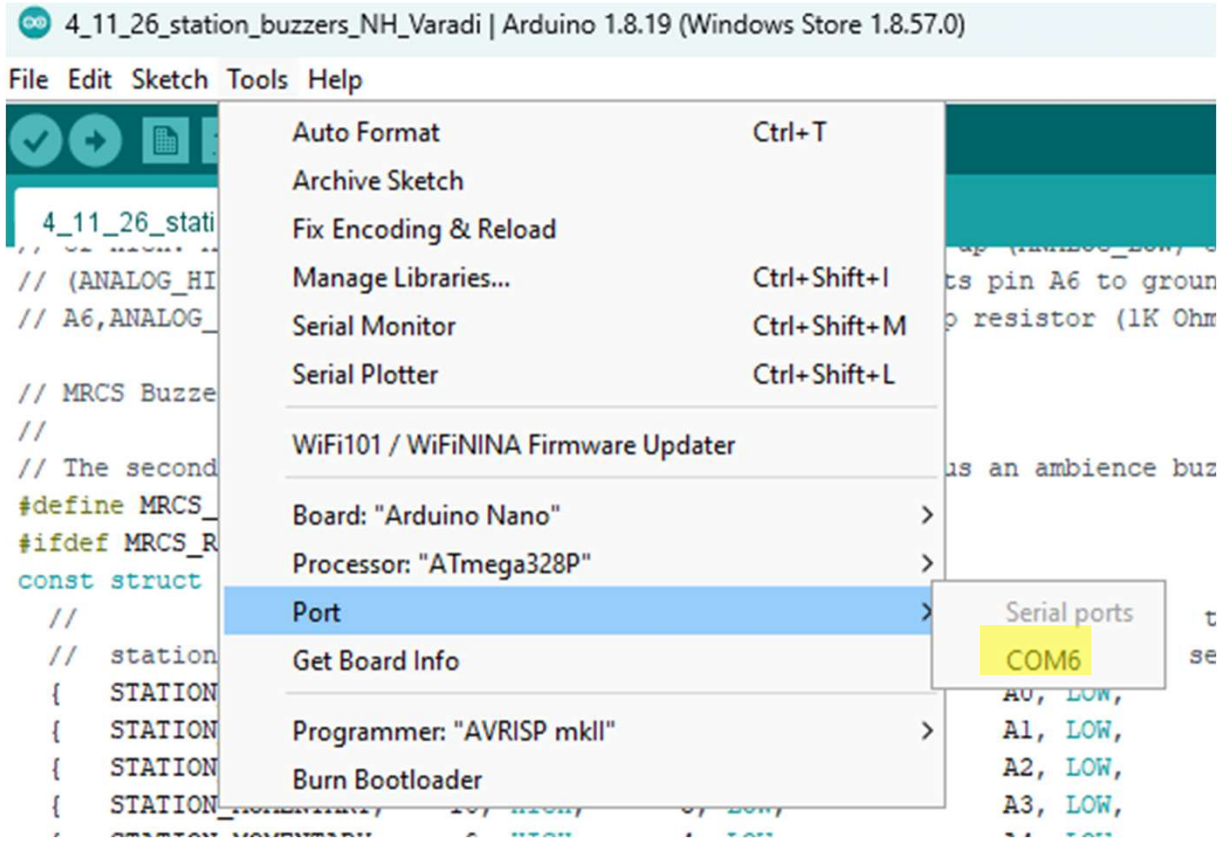


Selecting the Processor



It's one of these! You often don't know which, so you may need to try both!
I purchased 20 for this clinic and I got 13 new 7 old bootloaders! Don't let this Throw you off!

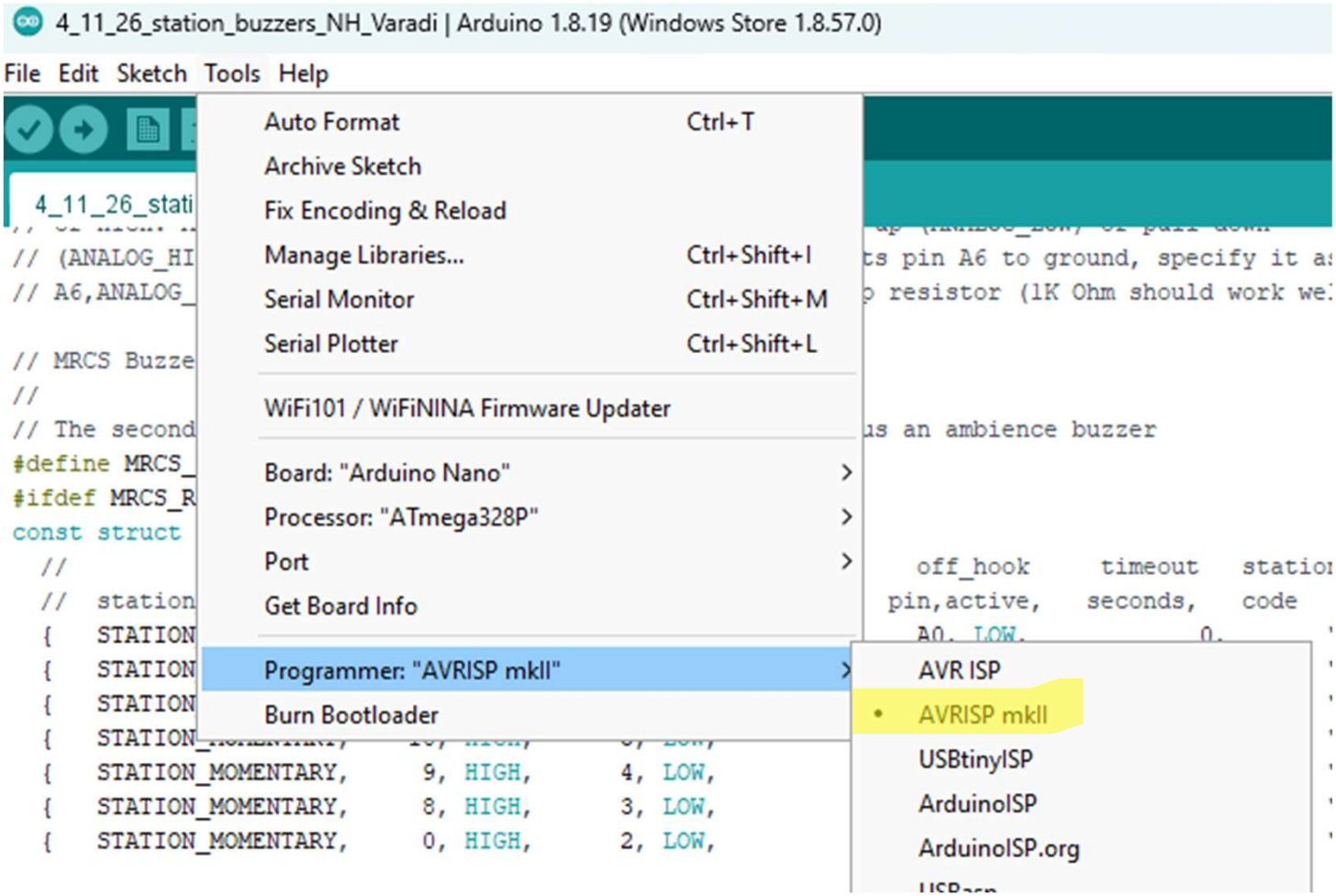
Select the COM Port



If you're not sure, look at this with the Nano unplugged and then plug it in and see What's new. It's the new one. If you don't see anything you either have a charging Cable or you are missing a driver or you have a dead Nano.

Check the “Programmer”

This is software in the IDE, not hardware you need to buy



Writing Software

- Often no need to write from scratch. Find a program that's similar and modify it. Many programs are available on the web, try an AI search.
- Compile often as you write, it's easier to find mistakes
- Test code in small modules
- Comment your code
- Language Reference on the Arduino.cc web site

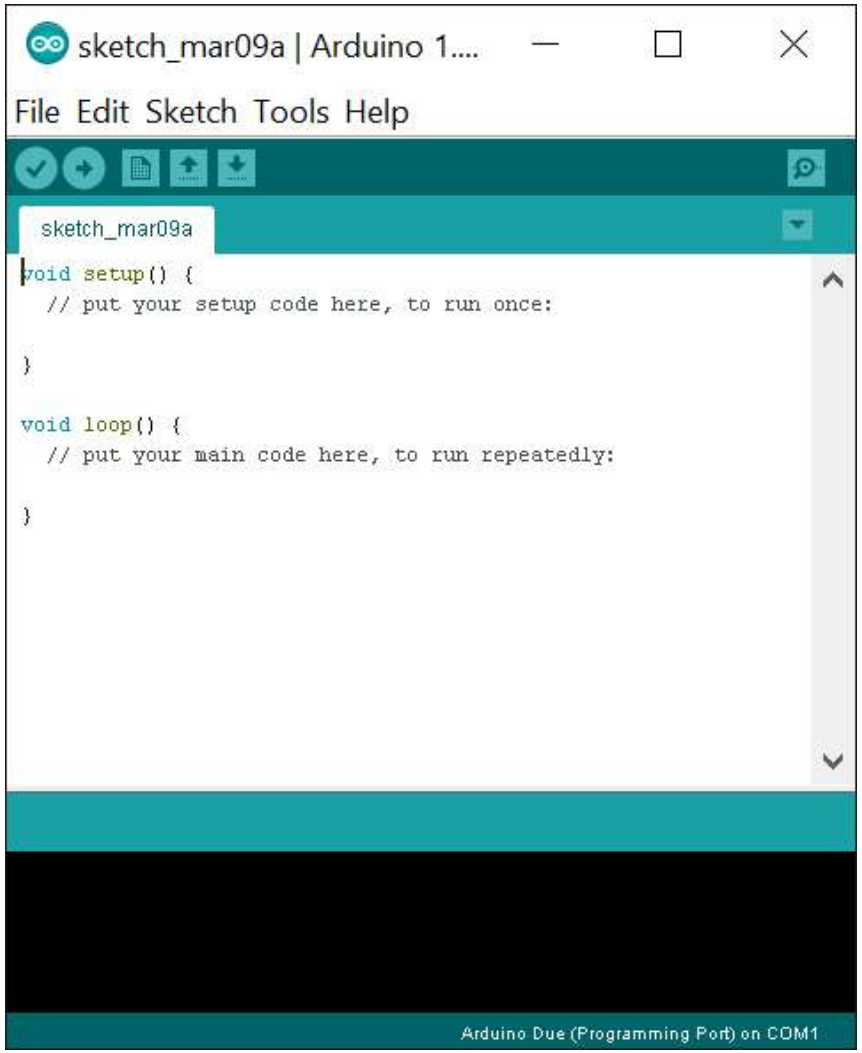
Syntax & The Compiler

This isn't Burger King, you can't have it your way

- Syntax – the 'grammar' the compiler expects
- Most lines terminate with a semicolon
- Variables need to be defined prior to use
- Capitalization and punctuation matter a lot!
- Compiler error messages don't always point at the error. Could be much earlier in the code.
- Make sure your brackets are balanced {}
- Did I mention comment your code?

Arduino IDE

Integrated Development Environment



Commonly Used Functions:

- **Sketch – Verify/Compile or checkmark**, allows you to check your program as you go
- **Tools – Board**, selects the proper Arduino for the compiler. For this clinic, Nano w/ATmega328.
- **Tools – COM Port**, port the Arduino is connected to for downloading code
- **File – Upload or right arrow**, compiles and uploads your code to the board. The right arrow key in the bar below it performs the same function.

Sample Code – Blink an LED

```
/* Blink
 This program blinks an LED on and off
 In the TOOLS BOARD tab in the Arduino IDE, ensure 'Arduino Nano w/ATmega328' is selected
 */

const int LED2 = 2;  //D2 output port

void setup()
{
  pinMode( LED2, OUTPUT );
 }

void loop()
{
  digitalWrite( LED2, HIGH );
  delay( 1000);
  digitalWrite( LED2, LOW );
  delay ( 1000 );
}
```

Specifies the LED on your board labeled D2

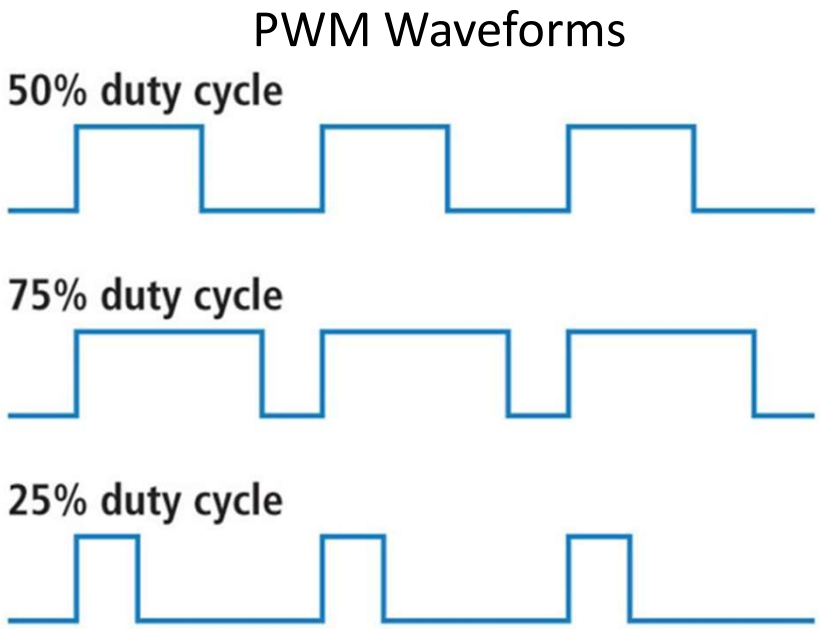
Makes pin 2 an output

Turn LED2 off (HIGH)

Turn LED2 on (LOW)

PWM & Random Functions

- PWM = Pulse Width Modulation
  - Varying duty cycle, can be used to vary LED brightness
  - Written as analogWrite(x) to a digital output pin
  - X is a value between 0 & 255, percentage duty cycle square wave



- Random (min,max) - returns a random number between min and max values

Fire Effect

```
/* LED Fire Effect
  Uses two amber and one red LED to simulate fire
  From the instructables.com web site, slightly modified

  This program uses the PWM function of the digital outputs
  to vary effective LED intensity. Analog write puts out a
  square wave of the specified duty cycle. Note that more LEDs
  can be added, but they must be attached to PWM pins,
  which are 3, 5, 6, 9, 10 & 11.
*/
int ledPin1 =  9;
int ledPin2 = 10;
int ledPin3 = 11;

void setup()
{
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
}

void loop()
{
analogWrite(ledPin1, random(135, 255));
analogWrite(ledPin2, random(135, 255));
analogWrite(ledPin3, random(135, 255));
delay(random(1, 100));
}
```

Note analogWrite: The number is x/256 so it's % of full brightness

Delay in Milliseconds

Servo Sweep

```
/*
Servo Sweep

This program slowly sweeps a servo back and forth from 0 to a presettable maximum number of degrees.
*/

#include <Servo.h>
Servo Servo1;

int pos;
int MaxPos = 60;
int DelayTime = 90;

void setup()
{
  //tell the Arduino & the servo driver that the servo is connected to pin 7
  Servo1.attach(7);
}

void loop()
{
  //sweep the servo back and forth
  for( pos = 1; pos <=MaxPos; pos++ ) //sweep from 0 to MaxPos in 1 degree steps
  {
    Servo1.write(pos);
    delay(DelayTime);
  }
  delay( 1000 );

  //now sweep it back
  for( pos = MaxPos; pos >=1; pos--)  //sweep from MaxPos to 0 in 1 degree steps
  {
    Servo1.write(pos);
    delay(DelayTime);
  }
  delay( 1000 );
}
```

Important Programming note: Use of the servo library disables analogWrite() (PWM) functionality on pins 9 and 10, whether or not there is a Servo on those pins.

Summary

- Shown how to type in, compile and download Arduino code
- Introduced some of the built-in Arduino functions
- Provided sample programs
- Next Clinic, Arduino 102:
  - Working with inputs
  - Higher level software functions
  - Tying it together

Arduino 102

Presented to PCR X2026 North

Seth Neumann

May 1, 2026

Arduino 102 Builds on 101

- More complicated programs
- More realistic effects, with random start and run times
- Interfacing with the real world: testing inputs and manipulating outputs (i/o)
- Introduce more Arduino built-in functions

Higher Level Software

- while loop

```
while(  X comparison Y  )
        {
        do something
        }
```

- Comparison operators include:

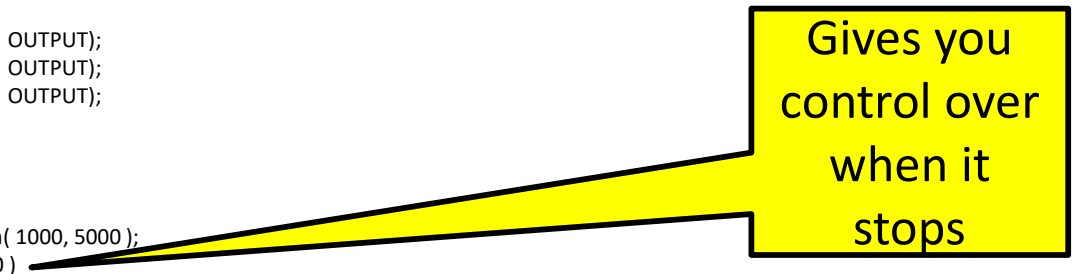
  > < >= <= == !=

  – *Note that '=' and '==' are different*

- if / else statement

WHILE Loop: Turning the fire off and on

```
/* LED Fire Effect
  Uses two amber and one red LED to simulate fire
  From the instructables.com web site

  Included a WHILE loop, so that the effect turns on and off.
*/

int ledPin1 =  9;
int ledPin2 = 10;
int ledPin3 = 11;
int OnTime;
int OffTime;
int i;

void setup()
{
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
}

void loop()
{
OnTime = random( 1000, 5000 );
while( OnTime > 0 )
 {
 analogWrite(ledPin1, random(120)+135);
 analogWrite(ledPin2, random(120)+135);
 analogWrite(ledPin3, random(120)+135);
 i = random(1, 100);
 delay( i );
 OnTime = OnTime - i;
 }

 // off time, ensure the LEDs are turned off
 analogWrite(ledPin1, 255 );
 analogWrite(ledPin2, 255 );
 analogWrite(ledPin3, 255 );
 OffTime = random ( 1000, 3000 );
 delay ( OffTime );
}
```



Debugging Your Program
or, Errors *You're* Going to Make

- Capitalization
- Spelling
- Inconsistent variable names
- Check the syntax: == is different from =
- Did you terminate the last item with ";" or is there one that doesn't belong?
- Are the {} balanced?

Tying it Together
Simplified Grade Crossing Control

- Combines the BLINK and SERVO SWEEP routines
- Need the while loop to run the two routines simultaneously
- Uses the pushbutton to actuate the sequence

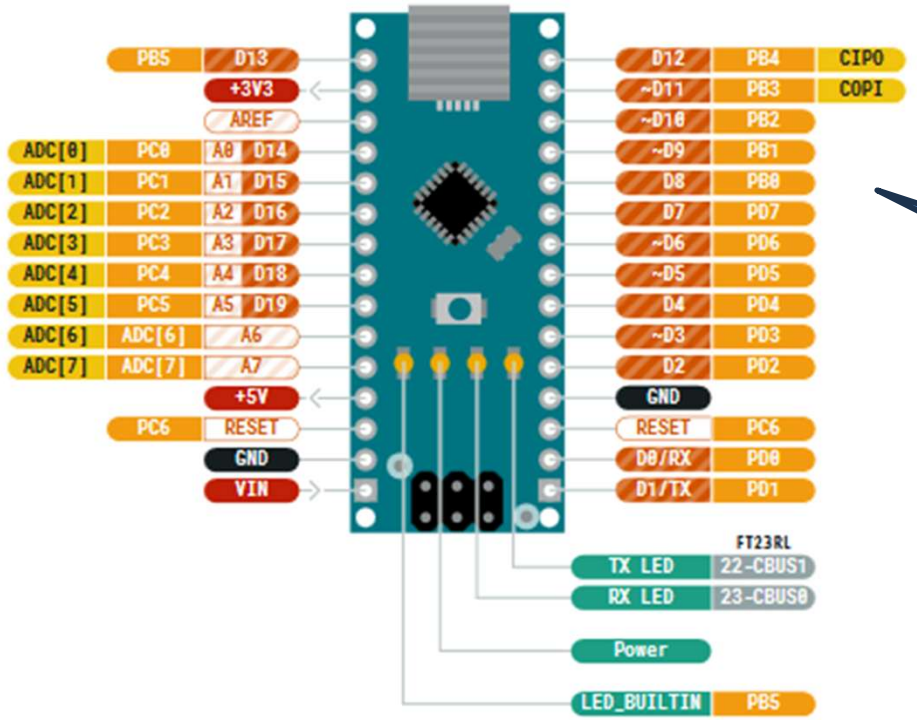
Program to Read Inputs & Set Outputs

- Look at the pots on A0, A1 as a percentage of brightness or servo deflection
  - Read value
  - Adjust as required (re-scale or map to log value)
  - Write to LED, Servo

Arduino Nano Pinouts & “Superpowers”

I2C – SDA/SCK

~ have PWM



Ground
Power
LED
Internal Pin
SWD Pin

Digital Pin
Analog Pin
Other Pin
Microcontroller's Port
Default

MAXIMUM current per I/O pin is 20mA

MAXIMUM current per +3.3V pin is 50mA

NOTE: CIPO/COPI have previously been referred to as MISO/MOSI

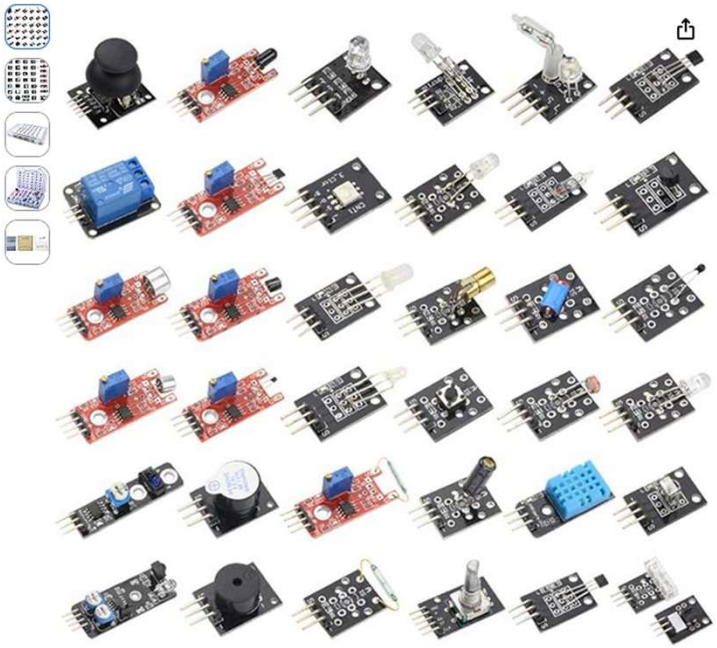
VIN 7-12 V input to the board.

Hardware options for your projects

- Additional copies of project board
- Use a nano with connectors
- Uno
- Mega
- Many new options, some with onboard WiFi or Bluetooth. There are often processor definitions for the Arduino IDE

Many Kinds of Sensors, Actuators, Indicators that work with Arduinos

Electronics › Computers & Accessories › Computer Components › Single Board Computers



Click to see full view

HiLetgo 37 Sensor Assortment Kit for Arduino & Raspberry Pi - 37 in 1 Robot Project Starter Kit

Visit the HiLetgo Store

4.2 ★★★★½ (296) | Search this page

Amazon's Choice

50+ bought in past month

$20.99

Price history

Get $60 off instantly: Pay $0.00 ~~$20.99~~ upon approval for the Prime Store Card. No annual fee.

prime Tomorrow

FREE Returns

Brand	HiLetgo
Connectivity Technology	USB
Included Components	Small passive buzzer module, 2-color LED module, Hit sensor module, Vibration switch module, Photo resistor module, Key switch module, Tilt switch module, 3-color f…
	See more
Operating System	Linux
CPU Manufacturer	Atmel

About this item

Meet the “Family” – Expanders and i/o

- **Arduino relay, 1,2,4,8,16**
- **8 position DPDT relay**
- **SMC-08 (8 position stall motor controller)**
- **i/o expander (IOX, dedicated Arduino Relay board coming) if you need more than 16 i/o lines, you can control up to 128 additional devices (16 base + 128 = 144)**

MRCS Nano Breakouts

- Dual Servo Semaphore/Grade Xing
- Approach Indicator
- Spring Switch Indicators

Sensors:

- Optical – various types
- Current sensing occupancy detectors
- Push buttons and toggles

https://www.modelrailroadcontrolsystems.com/arduino-nano-break-out-boards/

Additional Resources

- Arduino Development Kits with breadboard
- Sample code on Arduini Yahoo Group, join!
- Do a search or better yet, AI search! Lots of examples.
- Animation projects/videos on YouTube:
  - Geoff Bunza
  - Laurie McLean MMR

Something to Consider

- Multi-function programs can get complicated quickly, and are time-intensive to write and debug.
- Arduinos are inexpensive
- Dedicate an Arduino to each function
- Hardest part then is wiring it all up, not writing and debugging code

Arduino 102 Summary

- Introduced more software input and output functions and provided sample programs
- Introduced more complex software commands
- Shown the most common programming errors and what to watch out for
- Hopefully, inspired you to use Arduinos somewhere on your layout