

Reboot Camp 2023

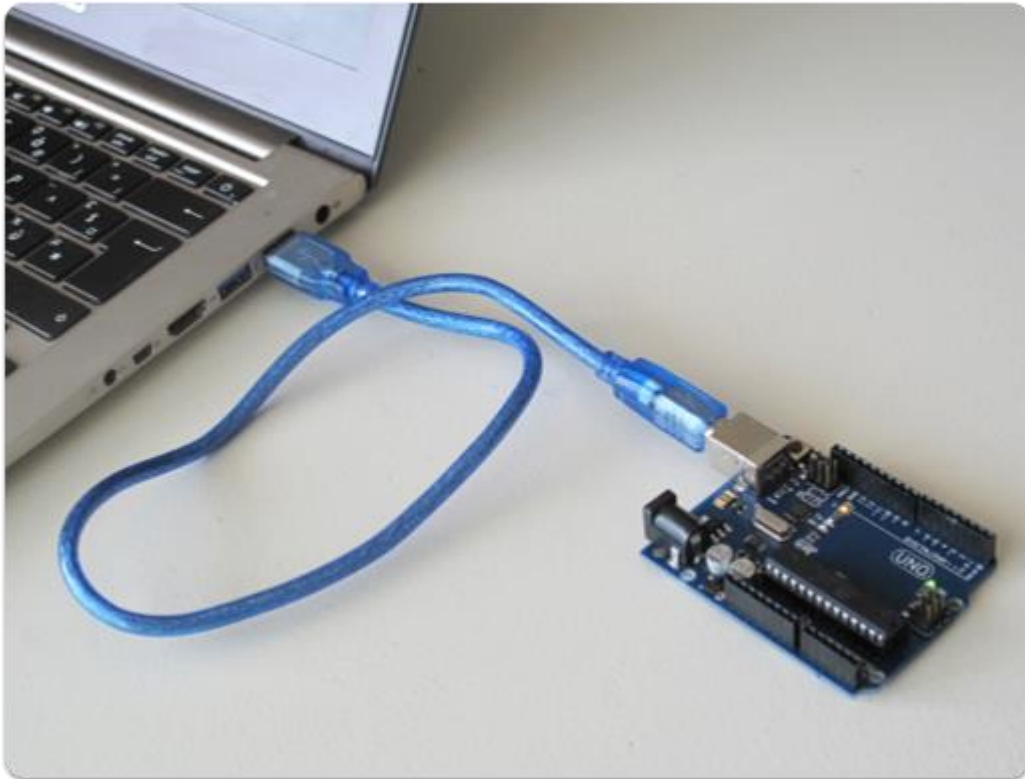
Arduino Workshop

Contents



Reboot Camp 2023 Arduino Workshop	1
How to Connect your Arduino to the Computer and Upload a Program	2
Arduino Print to Computer via Serial	5
Arduino LED Toggle	7
Arduino LED Blink Program	8
Arduino Traffic Controller.....	9
Arduino Blinking Smiley Face	11
Arduino Digitshield and Matrix Graphics	13
Arduino Joystick Calibration.....	14
Arduino Arcade – Pong	15
Arduino Arcade – Space Invaders Part 1	19
Arduino Arcade – Space Invaders Part 2	22

How to Connect your Arduino to the Computer and Upload a Program

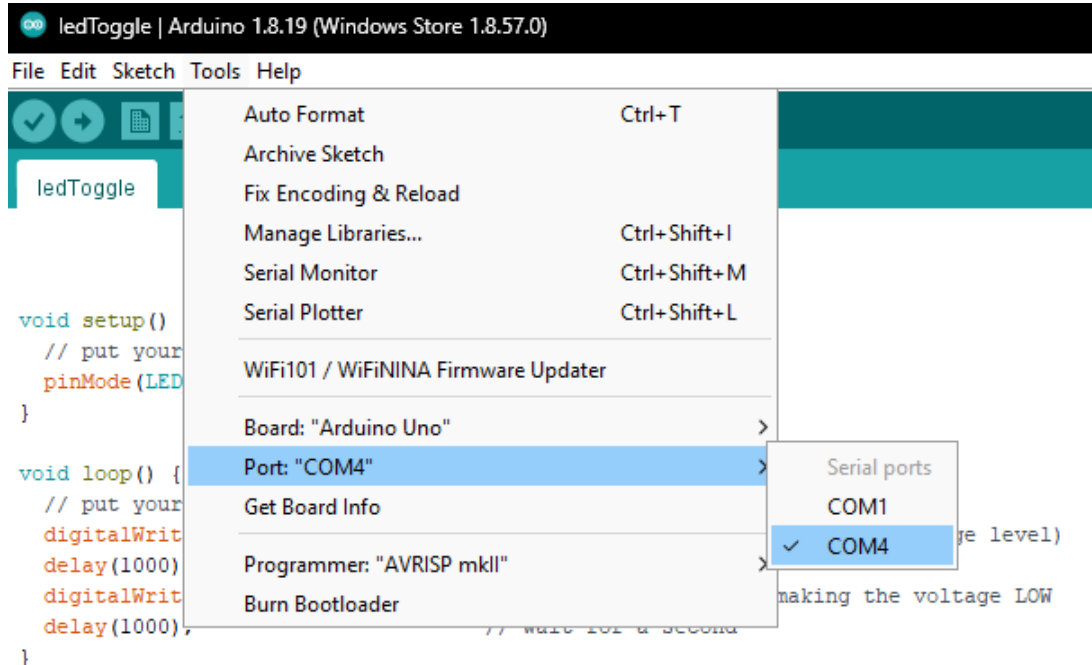
1. Connect the USB cable to the PC and the Arduino



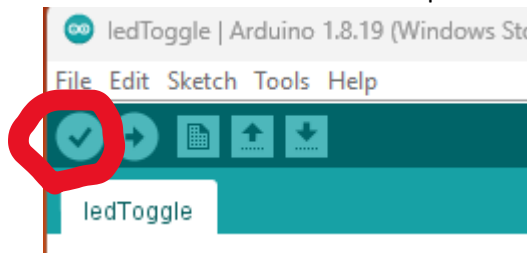
2. Open the Arduino IDE

 Arduino IDE.exe	9/09/2023 3:45 PM	Application	158,263 KB
 chrome_100_percent.pak	9/09/2023 3:45 PM	PAK File	127 KB

- Click on to “Tools” >> “Port:”,
- Select one of the options available labelled with “COM#” – For example “COM1”, “COM4”, or similar ones.
If this does not work, try another option.



- To check to see if the code will upload to the Arduino, first click on the “Verify” button.



6. If an error occurs, you will need to fix your Arduino code.

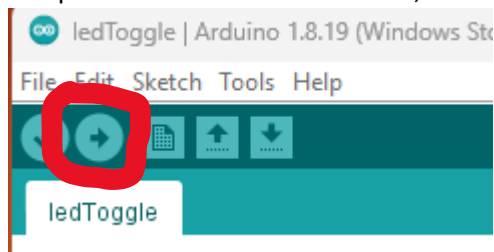


The screenshot shows the Arduino IDE interface. The code editor contains the following code:

```
}  
bad code  
  
ledToggle:l6:l: error: 'bad' does not name a type  
bad code  
^~  
exit status 1  
'bad' does not name a type
```

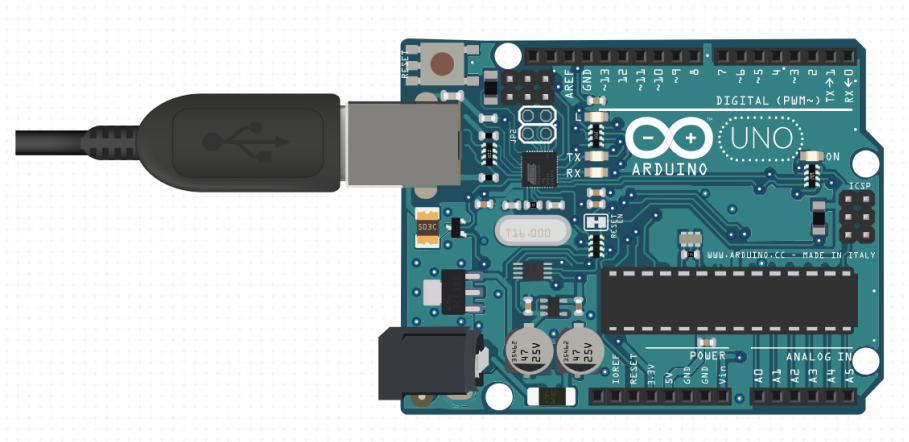
The error message "'bad' does not name a type" is displayed in the console. The line number 16 is visible at the bottom left of the console area.

7. If the “Verify” works with no error, you then upload your code to the Arduino. To upload the code to the Arduino, click on the “Upload” button.

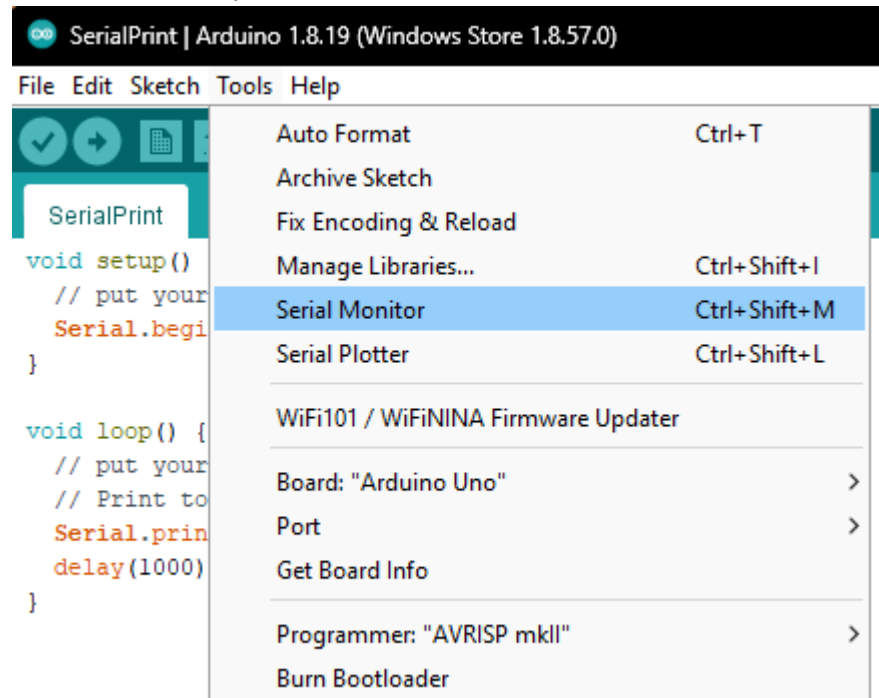


8. The Arduino will then start working with your code!

Arduino Print to Computer via Serial

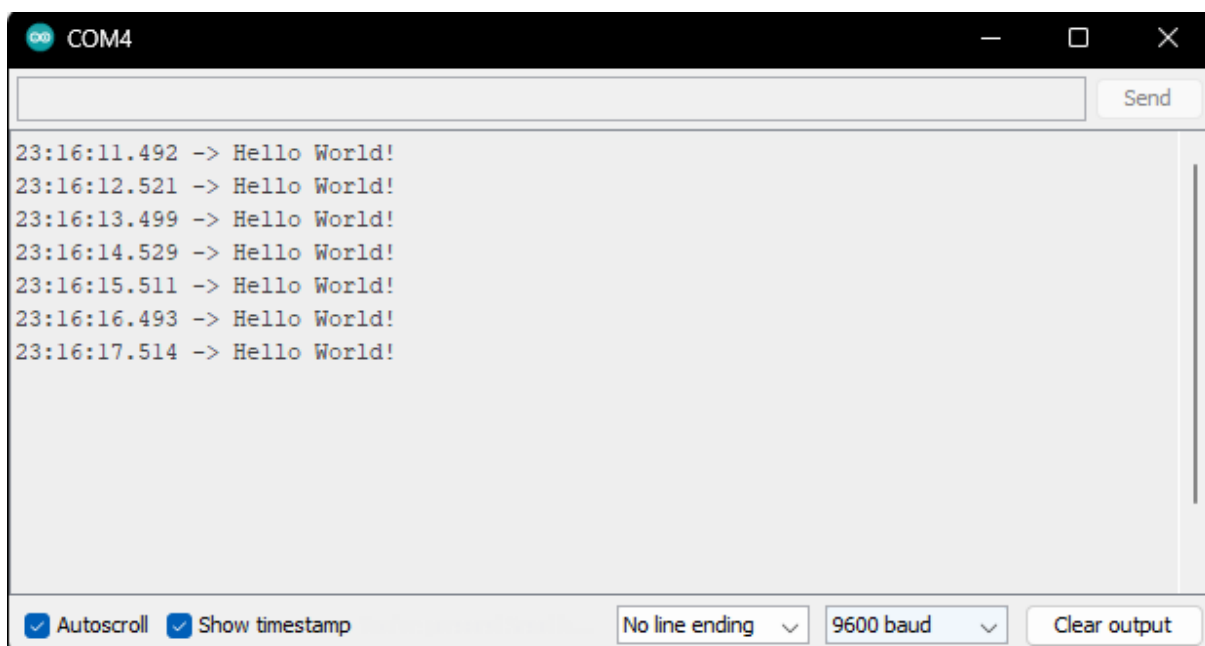


To start this one, open Serial Monitor via: “Tools” >> “Serial Monitor”



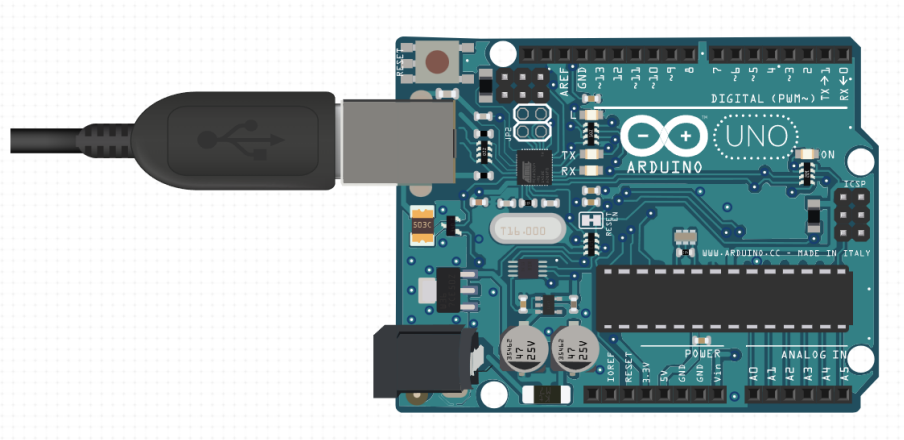
```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  // Print to serial every second  
  Serial.println("Hello World!");  
  delay(1000);  
}
```

When you run the code, the Serial Monitor will print out what you want it to write!



Arduino LED Toggle

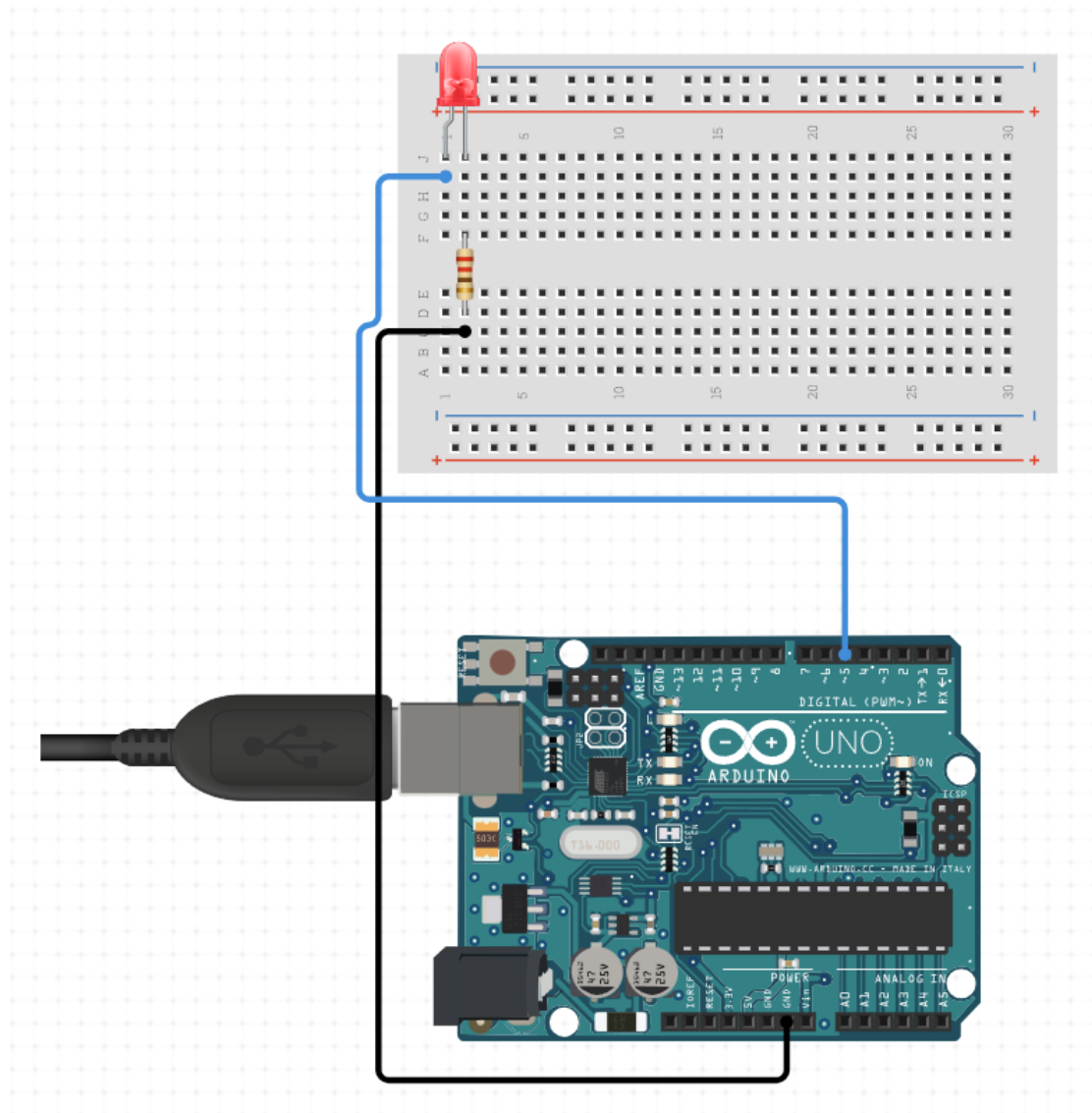
This code toggles an LED on the Arduino in 1 second intervals.



```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH)  
  delay(1000);                     // wait for a second  
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off (LOW)  
  delay(1000);                     // wait for a second  
}
```

Arduino LED Blink Program

This code will blink an LED attached to pin 5 every second.



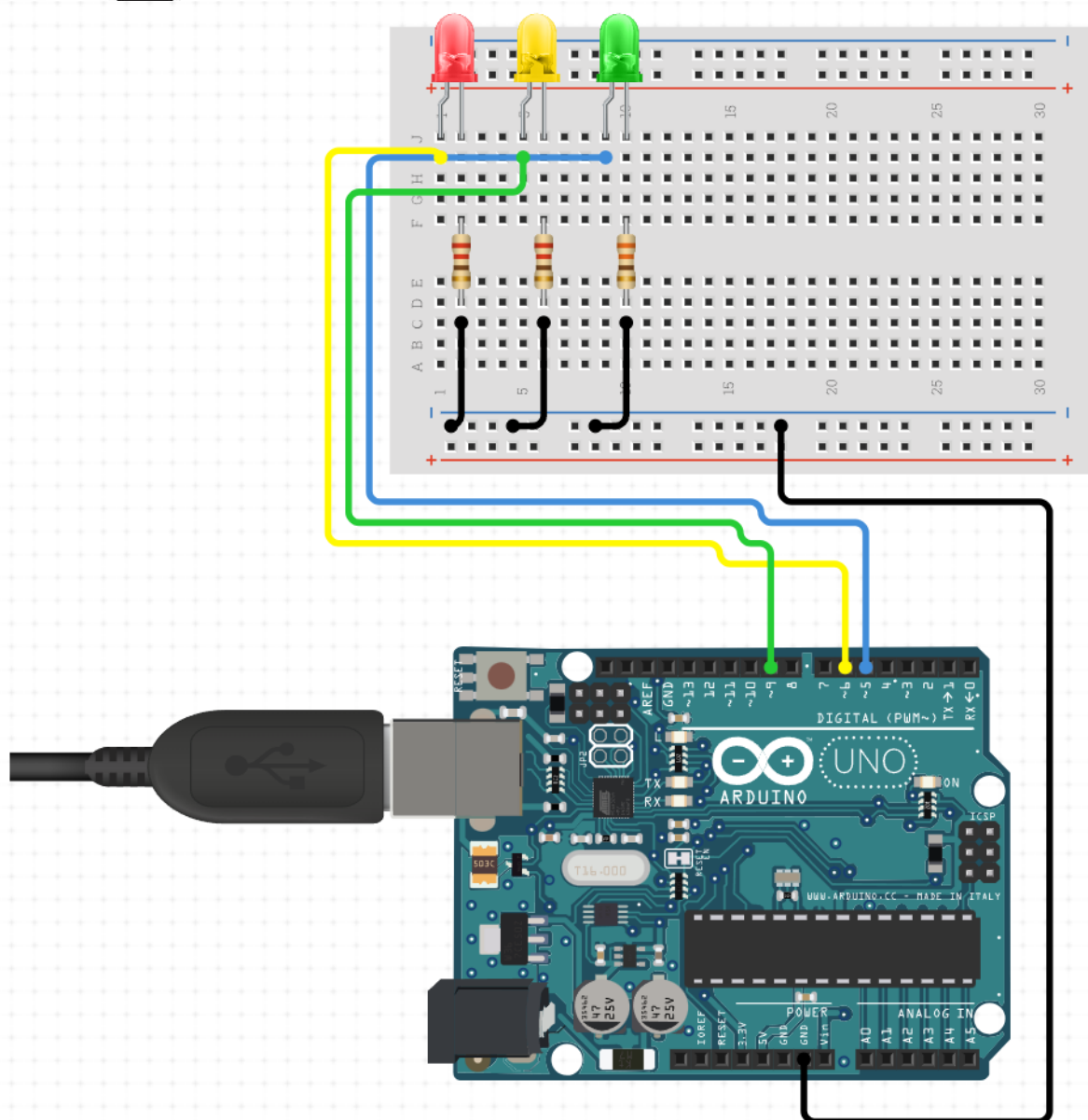
```
int ledpin = 5; // This is the pin the LED connects to.

void setup() {
  // put your setup code here, to run once:
  pinMode(ledpin, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(ledpin, HIGH); // turn the LED on (HIGH)
  delay(100);                // wait for a second
  digitalWrite(ledpin, LOW);  // turn the LED off (LOW)
  delay(1000);                // wait for a second
}
```


Arduino Traffic Controller

This program will act like traffic lights!



```
int redlight = 6;
int yellowlight = 9;
int greenlight = 5;

void setup() {
  // put your setup code here, to run once:
  pinMode(redlight, OUTPUT);
  pinMode(yellowlight, OUTPUT);
  pinMode(greenlight, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:

  // Start with green light for 10 seconds.
  digitalWrite(redlight, LOW);
  digitalWrite(yellowlight, LOW);
  digitalWrite(greenlight, HIGH);
  Serial.println("Green Light. You may pass!");
  delay(10000); // 10 seconds = 10000 milliseconds

  // Go to yellow light for 2 seconds.
  digitalWrite(redlight, LOW);
  digitalWrite(yellowlight, HIGH);
  digitalWrite(greenlight, LOW);
  Serial.println("Yellow Light. Please slow to a stop!");

  delay(2000); // 2 seconds = 2000 milliseconds

  // Go red light for 5 seconds.
  digitalWrite(redlight, HIGH);
  digitalWrite(yellowlight, LOW);
  digitalWrite(greenlight, LOW);
  Serial.println("Red Light. Stop now!");
  delay(5000); // 5 seconds = 5000 milliseconds
}
```

Arduino Blinking Smiley Face

This will create a blinking smiley face on the Arduino Arcade machine!



```

#include <DigitShield.h>
#include <Adafruit_NeoMatrix.h>

Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, 11,
  NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
  NEO_MATRIX_COLUMNS + NEO_TILE_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

// Colours
int WHITE = matrix.Color(255,255,255);
int BLACK = matrix.Color(0,0,0);
int RED = matrix.Color(255,0,0);
int BLUE = matrix.Color(0,0,255);
int GREEN = matrix.Color(0,255,0);
int YELLOW = matrix.Color(255,255,0);
int AQUA = matrix.Color(0,255,255);
int PURPLE = matrix.Color(255,0,255);

void setup() {
  // put your setup code here, to run once:
  matrix.begin();
  matrix.setBrightness(20);
}

void loop() {
  // put your main code here, to run repeatedly:

  // Normal happy face
  matrix.fillScreen(0);          // Turn off all the LEDs
  matrix.fillRect(1,5,2,2,RED);  // Left Eye
  matrix.fillRect(5,5,2,2,RED);  // Right Eye
  // Mouth
  matrix.drawLine(2,1,5,1,RED);  // Bottom of mouth
  matrix.drawPixel(0,3,RED);
  matrix.drawPixel(1,2,RED);
  matrix.drawPixel(6,2,RED);
  matrix.drawPixel(7,3,RED);
  matrix.show();
  delay(800);

  // Winking happy face
  matrix.fillScreen(0);          // Turn off all the LEDs
  matrix.fillRect(1,5,2,2,RED);  // Left Eye
  matrix.drawLine(5,5,6,5,GREEN); // Right Winking Eye
  // Mouth
  matrix.drawLine(2,1,5,1,RED);  // Bottom of mouth
  matrix.drawPixel(0,3,RED);
  matrix.drawPixel(1,2,RED);
  matrix.drawPixel(6,2,RED);
  matrix.drawPixel(7,3,RED);
  matrix.show();
  delay(400);
}

```

Arduino Digitshield and Matrix Graphics

This will display different graphics using the Digitshield and the 8x8 LED matrix.

```
#include <DigitShield.h>
#include <Adafruit_NeoMatrix.h>

Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, 11,
  NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
  NEO_MATRIX_COLUMNS + NEO_TILE_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

// Colours
int WHITE = matrix.Color(255,255,255);
int BLACK = matrix.Color(0,0,0);
int RED = matrix.Color(255,0,0);
int BLUE = matrix.Color(0,0,255);
int GREEN = matrix.Color(0,255,0);
int YELLOW = matrix.Color(255,255,0);
int AQUA = matrix.Color(0,255,255);
int PURPLE = matrix.Color(255,0,255);

void setup() {
  // put your setup code here, to run once:
  DigitShield.begin();
  matrix.begin();
  matrix.setBrightness(40);
  DigitShield.setValue(7777);
}

void loop() {

  // Explosion graphics
  for (int i = 0; i < 5; i++) {
    matrix.fillScreen(0);
    matrix.fillCircle(4, 4, i, RED);
    matrix.show();
    delay(500);
  }

  // Blink score counter
  for (int j = 0; j < 3; j++) {
    DigitShield.setBlank(true);
    delay(250);
    DigitShield.setBlank(false);
    delay(250);
  }
  delay(1000);
}
```

Arduino Joystick Calibration

This code can be used to calibrate the joystick, by printing the maximum and minimum values to the Serial Monitor.

```
#include <Adafruit_NeoMatrix.h>

Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, 11,
  NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
  NEO_MATRIX_COLUMNS + NEO_TILE_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

// Colours
int WHITE = matrix.Color(255,255,255);

const byte PIN_ANALOG_X = 3;
const byte PIN_ANALOG_Y = 2;
int analogX;
int analogY;
int myx;
int myy;
int dt = 0;

// Edit these
int scaleX = 900;
int scaleY = 900;

void setup() {
  // put your setup code here, to run once:
  matrix.begin();
  matrix.setBrightness(20);
  Serial.begin(9600);
  dt = 0;
}

void loop() {
  // put your main code here, to run repeatedly:

  // Get Joystick X and Y values
  analogX = analogRead(PIN_ANALOG_X);
  analogY = analogRead(PIN_ANALOG_Y);

  // Print X and Y values to Serial Monitor
  // But only once every second
  dt = dt + 1;
  if (dt > 20)
  {
    Serial.print("Joystick X = ");
    Serial.println(analogX);
    Serial.print("Joystick Y = ");
    Serial.println(analogY);
    dt = 0;
  }

  // Change X and Y values to match scale 0 to 7
  myx = map(analogX, 0, scaleX, 7, 0);
  myy = map(analogY, 0, scaleY, 0, 7);

  // Display the dot of where the joystick is pointing.
  matrix.fillScreen(0);
  matrix.drawPixel(myx,myy, WHITE);
  matrix.show();

  // Wait a short time before doing it again
  delay(50);
}
```

Arduino Arcade – Pong

This is the classic Arcade game called Pong, one of the first every made video games.

A ball will bounce on the screen, and you must use the joystick to move the bat so that the ball does not hit the ground. You get more points the more you hit the ball, and at 15 hits in a row you win!



This code is long and will go over the next 3 pages.

```

#include <DigitShield.h>
#include <Adafruit_NeoMatrix.h>

Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, 11,
  NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
  NEO_MATRIX_COLUMNS + NEO_TILE_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

// Colours
int WHITE = matrix.Color(255,255,255);
int BLACK = matrix.Color(0,0,0);
int RED = matrix.Color(255,0,0);
int BLUE = matrix.Color(0,0,255);
int GREEN = matrix.Color(0,255,0);
int YELLOW = matrix.Color(255,255,0);
int AQUA = matrix.Color(0,255,255);
int PURPLE = matrix.Color(255,0,255);

const byte PIN_ANALOG_X = 3;
const byte PIN_ANALOG_Y = 2;
int analogX = 0;
int myx = 0;
int myxleft = 0;
int myxright = 0;
int myy = 0;
int score = 0;
int mscount = 0;
int gamespeed = 160;
int gamespeedchange = 10;
int ballx = 0;
int bally = 1;
int ballvx = 1;
int ballvy = 1;

// Change bat size to 1, 2, or 3 for difficulty setting.
int mysize = 3;

void setup() {
  // put your setup code here, to run once:
  DigitShield.begin();
  matrix.begin();
  matrix.setBrightness(20);
  randomSeed(analogRead(0));
  ballx = random(0, 7);
  ballvx = random(-1, 1);
  bally = 1;
  ballvy = 1;
}

```



```

void loop() {
  // put your main code here, to run repeatedly:
  DigitShield.setValue(score);
  analogX = analogRead(PIN_ANALOG_X);
  myx = map(analogX, 0, 850, 6, 0);
  if (mysize == 3) { // Bat size 3
    myxleft = myx - 1;
    myxright = myx + 1;
  }
  else if (mysize == 2) { // Bat size 2
    myxleft = myx;
    myxright = myx + 1;
  }
  else if (mysize == 1) { // Bat size 1
    myxleft = myx;
    myxright = myx;
  }
  else { // Only allow bat sizes between 1 and 3
    mysize = 3;
    return;
  }

  // mscount states how many 'loops' will it take to create a new frame
  mscount++;

  if (mscount > gamespeed) { // Create new frame
    mscount = 0;

    // Bounce ball off the walls
    if (ballvx > 0 && (ballx + 1) > 7) {
      ballvx = -ballvx;
    }
    else if (ballvx < 0 && (ballx - 1) < 0) {
      ballvx = -ballvx;
    }
    ballx = ballx + ballvx;

    // Bounce the ball off the roof
    if (ballvy > 0 && (bally + 1) > 7) {
      ballvy = -ballvy;
    }
    bally = bally + ballvy;
  }
}

```

```

if (ballvy < 0 && bally == 1) {
    // Check if ball is on top of the bat
    if (ballx <= myxright && ballx >= myxleft) {
        // If the bat is only 1 pixel size, bounce the ball normally
        if (mysize == 1) {

        }
        // If on the right of the bat, bounce the ball right
        if (ballx == myxright) {
            ballvx = 1;
        }
        // If on the left of the bat, bounce the ball left
        else if (ballx == myxleft) {
            ballvx = -1;
        }
        else { // If in the middle of the bat, bounce normally

        }
        ballvy = -ballvy;
        score++;
        gamespeed = gamespeed - gamespeedchange;
    }
}
else if (bally < 1) { // Ball missed the bat. Game over!
    score = 0;
    gamespeed = 160;
    ballx = random(0, 7);
    ballvx = random(-1, 1);
    bally = 1;
    ballvy = 1;
}

if (score == 15) { // WIN!
    matrix.fillScreen(0);
    gamespeed = 160;
    score = 0;
    ballx = random(0, 7);
    ballvx = random(-1, 1);
    bally = 1;
    ballvy = 1;
}

matrix.fillScreen(0);
matrix.drawPixel(myx,myy, BLUE);
matrix.drawPixel(myxright,myy, BLUE);
matrix.drawPixel(myxleft,myy, BLUE);
matrix.drawPixel(ballx,bally, RED);
matrix.show();
}

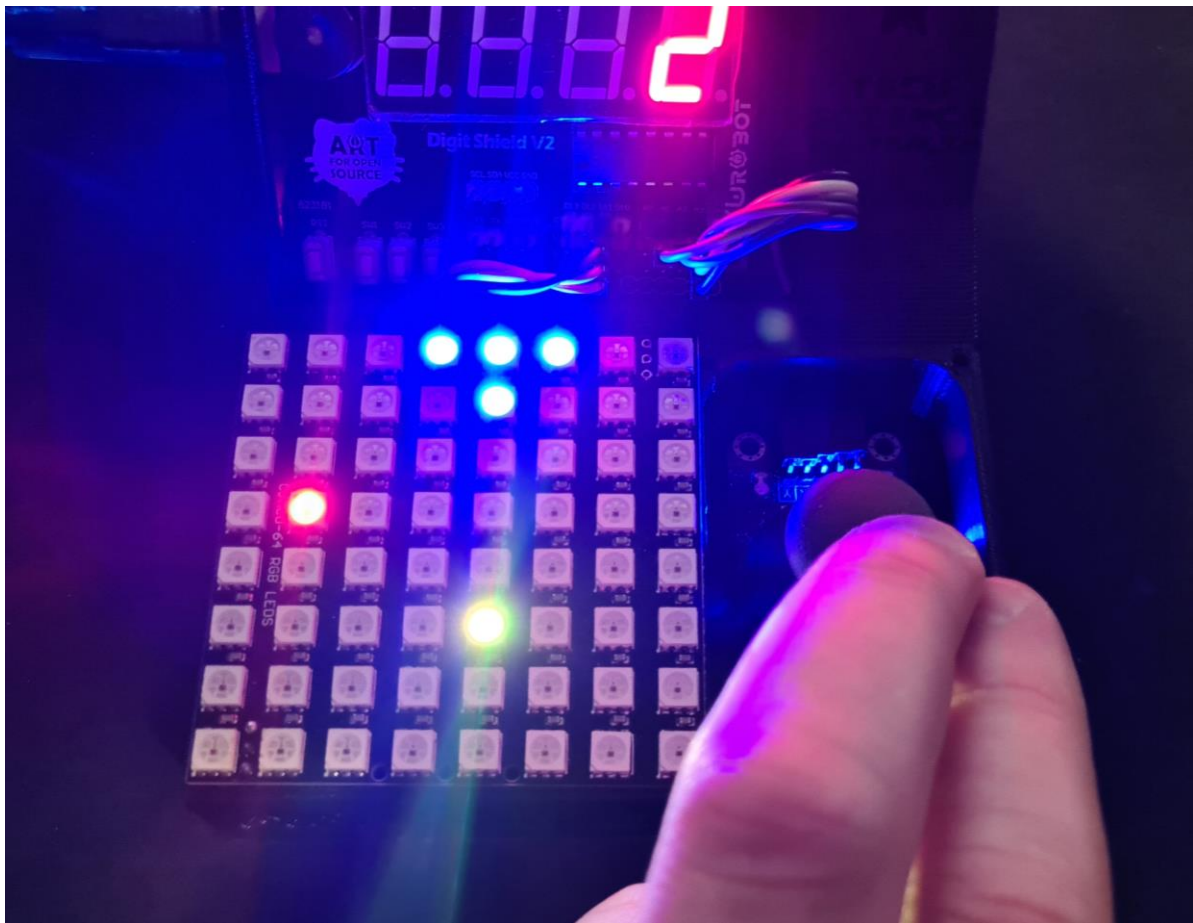
```

Arduino Arcade – Space Invaders Part 1

This is a classic Arcade game called Space Invaders. You are a spaceship with aliens flying to you.

Using your joystick, you move your spaceship left and right. You can press down on the joystick to shoot to fire a bullet that will destroy the alien on hit.

You win if you shoot enough aliens before they get to your spaceship, however they move faster every time you hit one!



This code is long and will go over the next 3 pages.

```

#include <DigitShield.h>
#include <Adafruit_NeoMatrix.h>

Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, 11,
  NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
  NEO_MATRIX_COLUMNS + NEO_TILE_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

// Colours
int WHITE = matrix.Color(255,255,255);
int BLACK = matrix.Color(0,0,0);
int RED = matrix.Color(255,0,0);
int BLUE = matrix.Color(0,0,255);
int GREEN = matrix.Color(0,255,0);
int YELLOW = matrix.Color(255,255,0);
int AQUA = matrix.Color(0,255,255);
int PURPLE = matrix.Color(255,0,255);

const byte PIN_ANALOG_X = 3;
const byte PIN_ANALOG_Y = 2;
int myx = 0;
int myy = 7;
int score = 0;
int enemyx = 0;
int enemyy = 0;
int mscount = 0;
int bulletx = 10;
int bulley = 10;
int firebutton = 0;
int gamespeed = 80;

void setup() {
  // pinMode(buzzerPin, OUTPUT);
  DigitShield.begin();
  matrix.begin();
  matrix.setBrightness(20);
}

void loop() {
  DigitShield.setValue(score);
  myx = analogRead(PIN_ANALOG_X);
  myx = map(myx, 0, 900, 7, 0);
  firebutton = digitalRead(A1);

  // When Shooting
  if (firebutton == 1) {
    bulletx = myx;
    bulley = myy-2;
  }

  // loop counter
  mscount++;
}

```

```

if (mscount > gamespeed) { // Create new frame
    mscount = 0;
    enemyx++;
    bullety--;
}

// When hitting enemy
if (bulletx == enemyx && bullety == enemyy) {
    score++;
    enemyy = 0;
    enemyx = 0;
    bullety = -1; // Make bullet dissapear
    gamespeed = gamespeed - 10; // Make it faster
}

// When enemy gets to the edge of the screen
if (enemyx > 7){
    enemyx = 0;
    enemyy++;
}

if (enemyy >= 6){ // Lose a point
    enemyy = 0;
    score--;
    gamespeed = gamespeed + 10;
}

if (score < 0){ // Game over
    score = 0;
    gamespeed = 80;
}

if (score == 5){ // WIN!
    matrix.fillScreen(0);
    gamespeed = 80;
    score = 0;
}

matrix.fillScreen(0);
matrix.drawPixel(myx,myy, BLUE);
matrix.drawPixel(myx+1,myy, BLUE);
matrix.drawPixel(myx-1,myy, BLUE);
matrix.drawPixel(myx,myy-1, BLUE);
matrix.drawPixel(enemyx,enemyy, RED);
matrix.drawPixel(bulletx,bulety, YELLOW);
matrix.show();
}

```

Code Page 2

Arduino Arcade – Space Invaders Part 2

This is the same as the last, but with some improvements, bugfixes, and extra graphics.

```
#include <DigitShield.h>
#include <Adafruit_NeoMatrix.h>

Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, 11,
  NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
  NEO_MATRIX_COLUMNS + NEO_TILE_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

// Colours
int WHITE = matrix.Color(255,255,255);
int BLACK = matrix.Color(0,0,0);
int RED = matrix.Color(255,0,0);
int BLUE = matrix.Color(0,0,255);
int GREEN = matrix.Color(0,255,0);
int YELLOW = matrix.Color(255,255,0);
int AQUA = matrix.Color(0,255,255);
int PURPLE = matrix.Color(255,0,255);

const byte PIN_ANALOG_X = 3;
const byte PIN_ANALOG_Y = 2;
int myx = 0;
int myy = 7;
int score = 0;
int enemyx = 0;
int enemyy = 0;
int mscount = 0;
int bulletx = 10;
int bullety = -1;
int firebutton = 0;
int gamespeed = 80;

void setup() {
  // pinMode(buzzerPin, OUTPUT);
  DigitShield.begin();
  matrix.begin();
  matrix.setBrightness(20);
}

void loop() {
  DigitShield.setValue(score);
  myx = analogRead(PIN_ANALOG_X);
  myx = map(myx, 0, 900, 7, 0);
  firebutton = digitalRead(A1);

  // When Shooting, only allow shoot if:
  // - bullet has hit enemy or,
  // - bullet has hit the bottom of the screen
  if (firebutton == 1 && bullety < 0) {
    bulletx = myx;
    bullety = myy-2;
  }
}
```

Code Page 1

```

// Loop counter
mscount++;

if (mscount > gamespeed) { // Create new frame
    mscount = 0;
    enemyx++;
    bullety--;
}

// When hitting enemy
if (bulletx == enemyx && bullety == enemyy) {
    score++;
    DigitShield.setValue(score);
    // Explode enemy
    matrix.fillScreen(0);
    matrix.drawPixel(myx,myy, BLUE);
    matrix.drawPixel(myx+1,myy, BLUE);
    matrix.drawPixel(myx-1,myy, BLUE);
    matrix.drawPixel(myx,myy-1, BLUE);
    matrix.fillCircle(enemyx, enemyy, 1, RED);
    matrix.show();

    // WIN if enemy hit and then Score is 5
    if (score == 5) {
        // Blink score counter
        DigitShield.setBlank(true);
        delay(250);
        DigitShield.setBlank(false);
        delay(250);
        DigitShield.setBlank(true);
        delay(250);
        DigitShield.setBlank(false);
        delay(250);
        DigitShield.setBlank(true);
        delay(250);
        DigitShield.setBlank(false);
        delay(250);
        gamespeed = 80;
        score = 0;
    }

    delay(250);
    bullety = -1; // Make bullet disappear
    enemyy = 0;
    enemyx = 0;
    gamespeed = gamespeed - 10; // Make it faster
}

if (bulley < 0) {
    bullety = -1;
}

```

Code Page 2

```
// When enemy gets to the edge of the screen
if (enemyx > 7){
  enemyx = 0;
  enemyy++;
}

if (enemyy >= 6){ // Lose a point
  enemyy = 0;
  score--;
  gamespeed = gamespeed + 10;
}

if (score < 0){ // Game over
  score = 0;
  gamespeed = 80;
  // Explode Spaceship of game over
  for (int i = 2; i < 7; i++) {
    matrix.fillScreen(0);
    matrix.fillCircle(myx, myy, i, RED);
    matrix.show();
    delay(500);
  }
}

matrix.fillScreen(0);
matrix.drawPixel(myx,myy, BLUE);
matrix.drawPixel(myx+1,myy, BLUE);
matrix.drawPixel(myx-1,myy, BLUE);
matrix.drawPixel(myx,myy-1, BLUE);
matrix.drawPixel(bulletx,bulety, WHITE);
matrix.drawPixel(enemyx,enemyy, RED);
matrix.show();
}
```

Code Page 3