

DIY Arcade Joystick Kit



SUPER

Wake!

UNLEASH YOUR INNER MAKER!

Whether you're 8 or 80, the Maker Shed offers an incredible array of **kits for learning about electronics, engineering, design, and coding.** So what are you waiting for? Dive in and discover a whole new universe of possibilities!

Make:

ARDUINO ELECTRONICS STARTER PACK

Learn How to Make Electronics That Interact with the Physical World

- Skill level: Beginner to Advanced
- Some soldering required
- Computer required

Books to Boost Your Experience:
Getting Started with Arduino
Learn Electronics with Arduino

Maker Shed

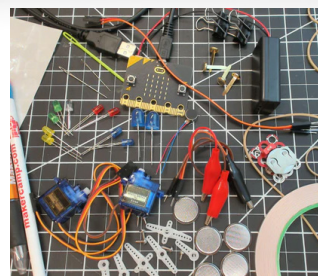
Make:
Getting Started with **ARDUINO**
4th Edition

The Open Source Electronics Prototyping Platform
by Massimo Banzai co-founder of Arduino and Michael Shloh

Make: Arduino Electronics Starter Pack & Getting Started with Arduino, 4th Edition book



Makey bit: Mobile Kit & Getting Started with the micro:bit book



Making Simple Robots Starter Pack with kit & Making Simple Robots book



Save 15% off select Kits with promo code **MAKE84** at makershed.com/kits

Maker Shed



Make:

DIY Arcade Controller

Written by Tyler Capps

TIME REQUIRED:

1-2 Hours

DIFFICULTY:

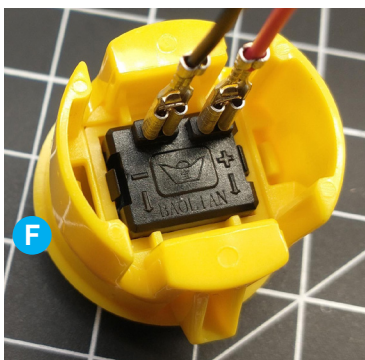
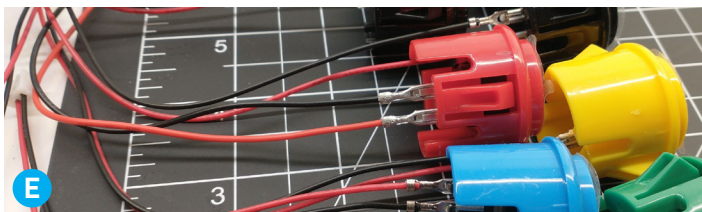
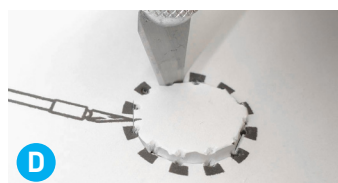
Easy

MATERIALS

- » Box x 1
- » Arcade stick x 1
- » Ball Top x 1
- » Dust Guard x 1
- » Buttons (30mm) x 8
- » Buttons (24mm) x 5
- » Button Leads x 13
- » USB Encoder x 1
- » USB Connecting wire x 1
- » Templates x 2
- » Sticker x 1

TOOLS

- » X-Acto Knife or Box Cutter
- » Masking Tape
- » Hot Glue Gun
- » Screwdriver
- » M4 or 8-32 bolts and nuts (optional)

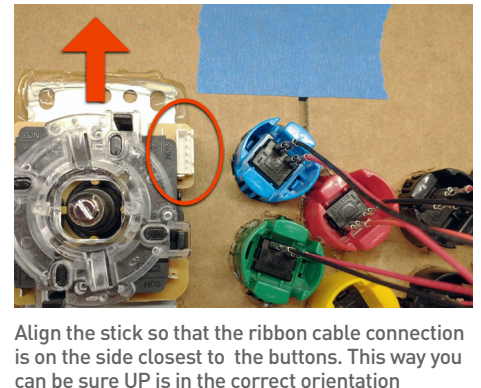
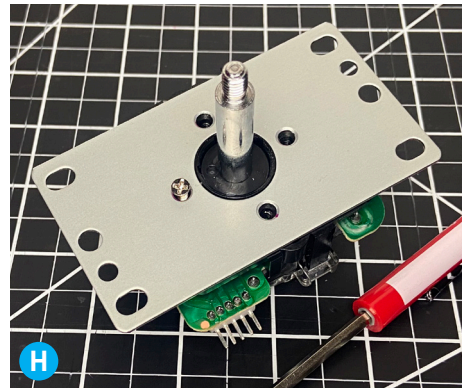


Red goes to +
Black goes to -

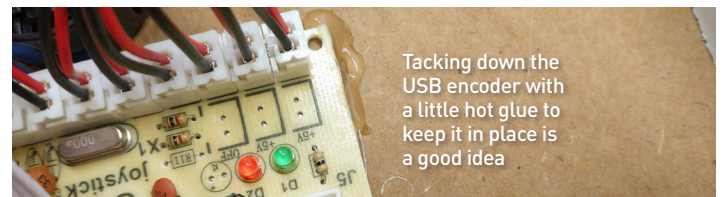
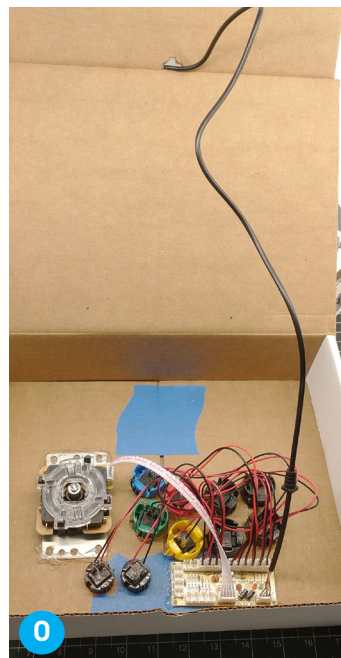
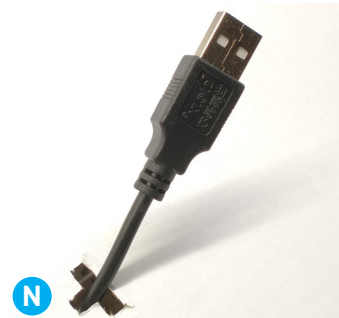
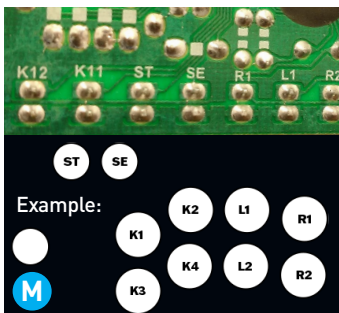
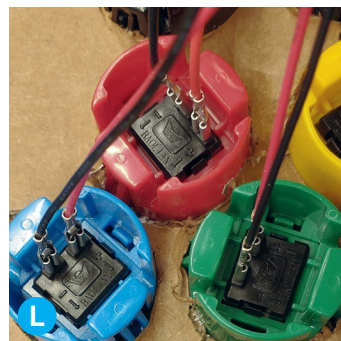
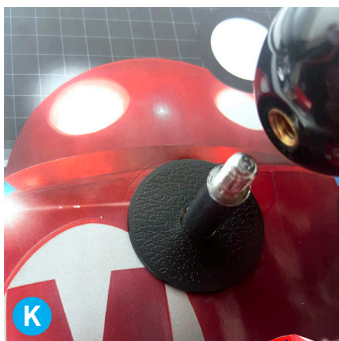
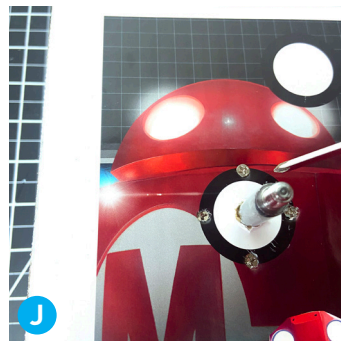
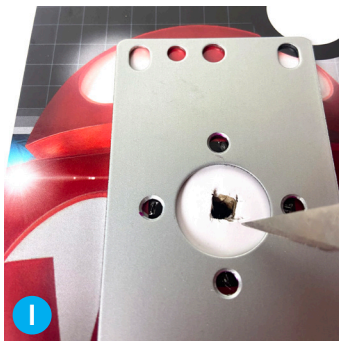
THE BUILD:

1. Open your kit box and unpack the contents, paint or decorate your box as desired.
2. Carefully pull the interior side tabs out of the box, lay it out flat and cut 4 3/4" off each flap (Figure A), this will make it easier to cut out your button and joystick holes.
3. Refold the box and close the lid, then flip over to the bottom of the box, the thumb notch should be facing away from you on the lower side of the box.
4. Use the Makey Robot sticker or the provided paper template, center on your box (Figure B).
5. For the sticker, carefully peel/crack one side of the sticker and align on your box. Peel remaining backing and smooth across the box using a credit card or other flat item to remove any air bubbles from your sticker. For the paper template use masking tape.
6. Using a sharp razor hobby knife or equivalent blade carefully cut out the large 8 button holes and the smaller 2 button holes, cutting inside the white dot area only. Do not cut out the joystick hole yet. Tip: Try pie cuts with your blade or use a screwdriver or skewer to poke holes in a radial pattern to make cutting out your circles a snap! (Figures C and D).
7. Test the fit of the buttons but don't press any buttons all the way in yet. If it's a little too tight, you can carefully remove a little more cardboard from the circumference of the hole until the button can be pressed in snugly. Connect the ends of each lead to the prongs on each button (Figures E and F).
8. The arcade stick can be attached in a couple of ways, for this build we are going to use the screws found in the joystick itself. You can also use hot glue or M4 bolts and nuts (or 8-32 for an imperial

DIY Arcade Controller



Align the stick so that the ribbon cable connection is on the side closest to the buttons. This way you can be sure UP is in the correct orientation



- equivalent) for more permanent builds or when using your own controller housing (Figure G).
9. Remove the 4 small screws holding the joystick's metal plate (Figure H).
 10. Take the metal plate and align it straight over the final uncut joystick hole on your sticker.
 11. Using a pen or screwdriver point, mark the center for the handle along with the four screw holes. You can widen the center point (about 1/2") with an exacto blade or screwdriver, just enough for the joystick handle to come through. Poke smaller holes through for the 4 plate screws (Figure I).
 12. Realign the plate over the joystick and from the inside of your box poke the handle through the hole. Use the screws to secure the box to the joystick plate. Do not overtighten (Figure J). Optionally use hot glue or other screw mounts for extra strength.
 13. Place the joystick dust cover over the handle and screw on the ball top (Figure K).

INSERT YOUR CONTROLS

14. Choose your buttons and then feed the wires through the holes and press each button in firmly (Figure L). You'll have a bit of a wirey mess on the opposite end now, but we'll get to that.
15. Next let's take a look at the USB encoder. You'll notice labels on the back for all of your button connections (Figure M). These buttons can be reassigned later however you like. So hook them up in whatever way makes sense for you.
16. All of these connectors including the joystick and USB cable can only be plugged in one way so you don't have to worry about hooking anything up backwards.
17. Our last connection is the USB cable. Before we connect that we need a way for it to exit the box. I used the USB connector end to mark and cut a small hole (Figure N).
18. The cable is then fed through the hole such that it can be connected to the board and pass through the box when it's closed (Figure O). The stopper near the end of the cable should keep it

DIY Arcade Controller



from pulling loose easily, but you can secure this by feeding some slack back into the box (so you can still open it while the cable is connected) and using a dab of hot glue or tape to secure the cable in the slot.

19. Close the box lid and you have a fully functional plug & play arcade controller (Figure P).

You might notice that you have three more buttons and cables left over (Figure Q). These buttons, or any of the buttons really, can be placed wherever your heart desires. For example I found it useful to have buttons on each side of the box for pinball games (Figure R).

The controller is completely plug and play. When you connect it to a Windows PC a generic gamepad driver will automatically be installed and you can check the functionality using Windows Gamepad Setup. If you want to use it for modern games it can be configured through Steam's Big Picture mode and is completely customizable. For retro games, MAME and most emulators will automatically recognize it.

It is also plug & play and fully configurable with Raspberry Pi, Linux and RetroPie setups. Check out our [Getting Started with Raspberry Pi](#) book for more about retro gaming.

CODE:

```
//Tyler Capps
//MAKE:
//April 2023
// Controlling an Arduino with an arcade style joystick and buttons
```

```
//selecting pins for each directional input
int dirDown =4;
int dirUp =5;
int dirRight =7;
int dirLeft =6;
```

```
//selecting pins for the output of our buttons
int buttonRed =8;
int buttonBlue =9;
int buttonGreen =10;
int buttonYellow =11;
```

```
int buttonValue;
```

```
void setup() {
  // put your setup code here, to run once:
  pinMode( dirDown , INPUT_PULLUP);
  pinMode( dirUp , INPUT_PULLUP);
  pinMode( dirLeft , INPUT_PULLUP);
  pinMode( dirRight , INPUT_PULLUP);
```

```
  pinMode( buttonRed , OUTPUT);
  pinMode( buttonBlue , OUTPUT);
  pinMode( buttonGreen , OUTPUT);
  pinMode( buttonYellow , OUTPUT);
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

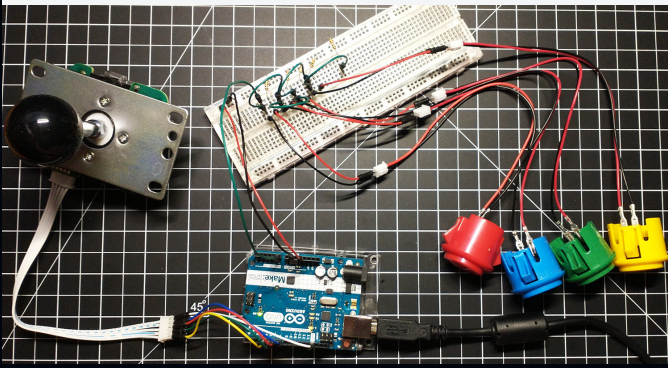
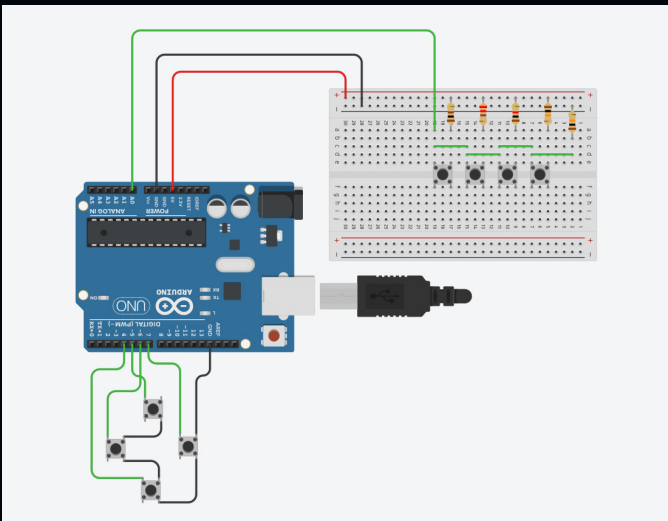
DIY Arcade Controller

NOTE: The **USB encoder board** that comes with the kit can take up to 12 buttons of input. The 4 inputs for CLR, AUTO, TURBO and MODE, are state modifiers that aren't reboundable or usable as extra button inputs.

- **MODE:** Switches between digital and analog mode. The joystick is not analog so this function is not useful for this build.
- **TURBO:** When the turbo button is held down any other button pressed will repeat rapidly.
- **AUTO:** sets all buttons to turbo mode until CLR is pressed.
- **CLR:** Clears AUTO state.

So if you connect a button to AUTO you also need a button connected to clear, otherwise turbo will be permanently turned on.

There are also 4 extra inputs for AU, AD, AL, AR connections, but they mirror the directional inputs from the joystick ribbon cable port. Meaning you can use the ribbon cable for the joystick OR the individual connections, but not both.



// put your main code here, to run repeatedly:

```
buttonValue = analogRead(A0); //Read the analog value from pin A0
```

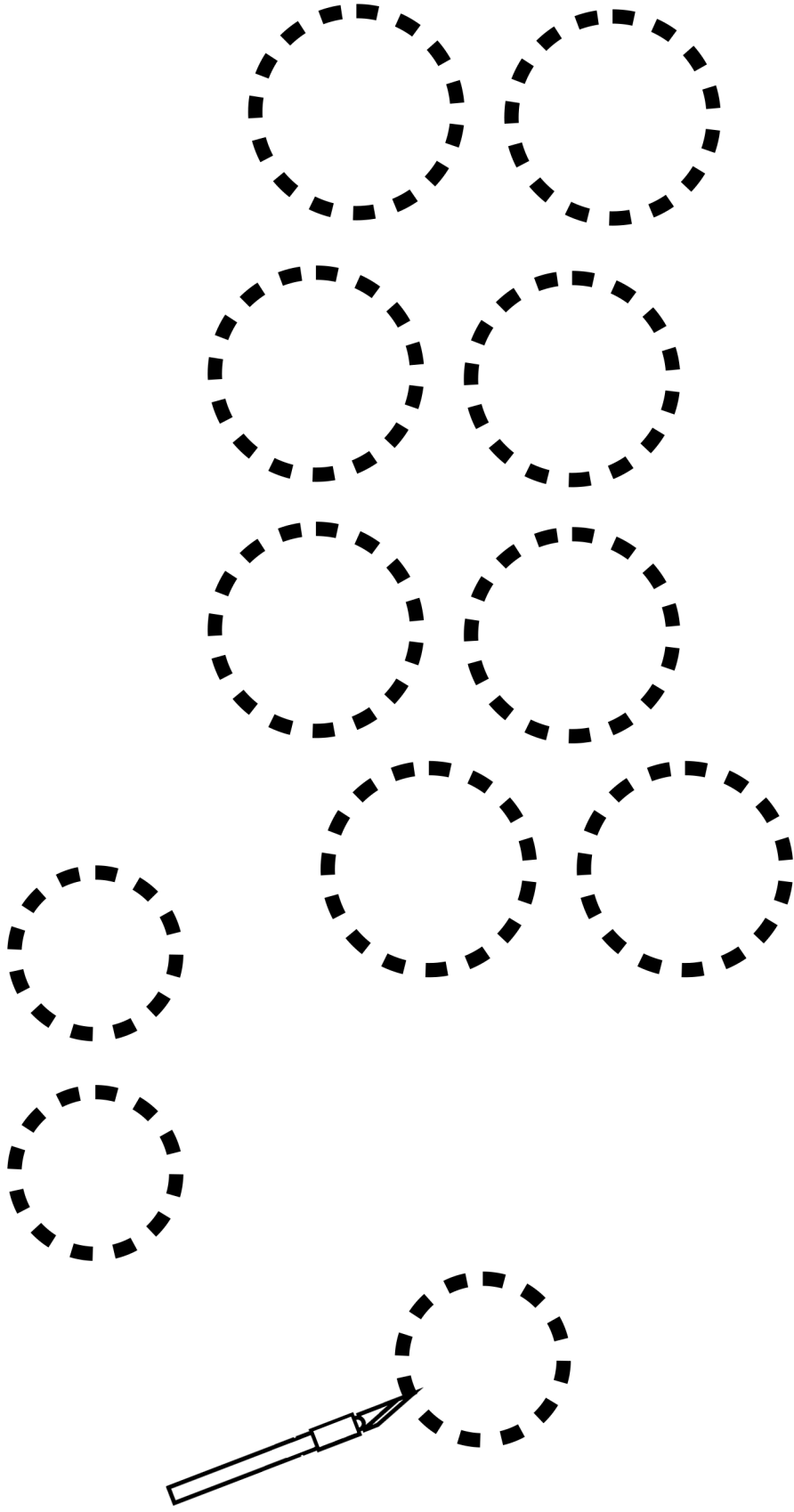
```
int Up = digitalRead(dirUp);  
int Left = digitalRead(dirLeft);  
int Right = digitalRead(dirRight);  
int Down = digitalRead(dirDown);
```

```
if(Up == LOW && Left == LOW){  
  Serial.println( "Direction: UPLEFT");  
}  
else if(Up == LOW && Right == LOW){  
  Serial.println( "Direction: UPRIGHT");  
}  
else if(Down == LOW && Right == LOW){  
  Serial.println( "Direction: DOWNRIGHT");  
}  
else if(Down == LOW && Left == LOW){  
  Serial.println( "Direction: DOWNLEFT");  
}  
else if(digitalRead( dirUp ) == LOW){  
  Serial.println( "Direction: UP");  
}  
else if(digitalRead( dirLeft ) == LOW){  
  Serial.println( "Direction: LEFT");  
}  
else if(digitalRead( dirRight ) == LOW){  
  Serial.println( "Direction: RIGHT");  
}  
else if (digitalRead( dirDown ) == LOW){  
  Serial.println( "Direction: Down");  
}  
else{ //This is where you would add an action for when no  
direction is pressed.
```

```
}  
// The resistors in the circuit will cause different resistances to be  
read for each button when pressed  
// This is how the Arduino knows which specific button is being  
pressed when all of the buttons are connect to a single pin  
// Here is where we tell the Arduino what ranges to look for for  
each button
```

```
//for the Red button:  
if (buttonValue>=1010 && buttonValue<=1015){  
  Serial.println( "RED");  
}  
//for the Blue button:  
else if (buttonValue>=1000 && buttonValue<=1008){  
  Serial.println( "BLUE");  
}  
//for the Green button:  
else if (buttonValue>=900 && buttonValue<=950){  
  Serial.println( "GREEN");  
}  
//for the Yellow button:  
else if (buttonValue>=500 && buttonValue<=600){  
  Serial.println( "YELLOW");  
}  
else{
```

```
  // This is where you would add an action for when no  
  button is pressed.  
  delay(100);  
}
```



TIP: Cut just inside the dotted lines. You can always trim more out if the holes are too small.



For all templates and resources go to make.co/joystick

Make:

WAYS TO READ **Make:**



Make: is the premier magazine for makers, DIY enthusiasts, hackers, and tinkerers

OVER 120 PAGES OF:

- Full project tutorials and skill builders, from electronics and robots to 3D fabrication and woodworking
- Maker profiles and interviews
- Detailed photos and diagrams to help your builds
- Project inspiration
- and more!

Subscribe to Make: at makezine.com/go/subscribe
Join Make: Community to access the digital magazine at make.co
Make: Community members and print subscribers have access online and in the iOS and Android apps.