



LEGO Arduino  
Bluetooth Arcade Joystick  
Home Project  
Douglas Gammill  
April 20, 2023



# Introduction

This home DIY project shows how I built two Bluetooth arcade joysticks using LEGO bricks, an ESP32, and a few other components. Included in this document is a schematic, bill of materials, assembly instructions, and programming instructions.



If any reader of this document is unfamiliar with Arduino IDE or programming ESP32s, a tutorial can be found here:

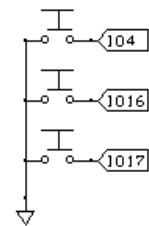
<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>.

This project is tested to work on Windows 10 and RetroPie running on a Raspberry Pi 4. It does not appear to be compatible with Recalbox x64 x86 9.0.2. No other systems have been tested at this time.

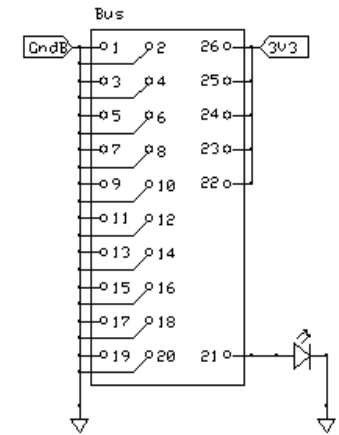
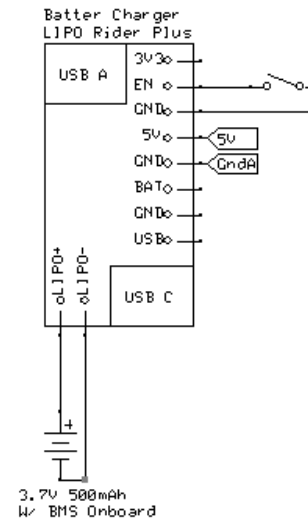
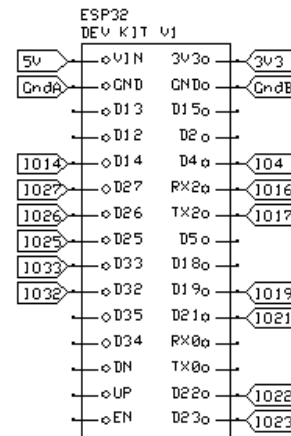
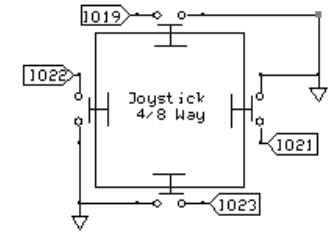
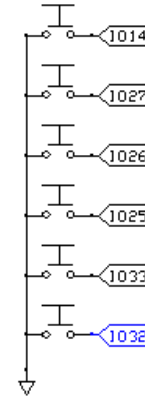


# Schematic

Small Menu Buttons



Large Arcade Buttons



LEGUINO		
BLE Arcade Joystick		
DCamill	Rev 1.0	Page 1 of 1
	4/21/2023	

# Bill of Materials

- A bunch of LEGO Bricks
- A couple of LEGO baseplates
- LEGO Parts and Pieces: Black 2x2 Tile x50
- 1 box: EG STARTS 2 Player Arcade Game DIY Kits (Arcade joysticks and buttons) (amazon.com)
- 2 each: ESP32 Dev Kit V1 (amazon.com)
- 2 each: 3.7V 500mAh LIPO batteries (with built in BMS and JST2.0 connector) (amazon.com)
- 2 each: LIPO Rider Plus (Charger/Booster) - 5V/2.4A USB Type C (digikey.com or mouser.com)
- 1 pack: TWIDEC/12Pcs 1A 250V AC 2 Pins SPST 6 Colors Normal Open Mini Momentary Push Button (amazon.com)
- 28 each: M2.5 x 6mm button head screws (amazon.com or mcmaster.com)
- 28 each: M2.5 standoffs (any length) (amazon.com or mcmaster.com)
- 28 each: M2.5 x 4mm counter sunk screws (amazon.com or mcmaster.com)
- 1 pack: Dupont Female to Female Breadboard Jumper Wires 7.9 inch 2.54mm Connector Ribbon Cables (amazon.com)
- 2 each: 1k $\Omega$  resistors (amazon.com)
- 2 each: SPST toggle switches (amazon.com)
- 2 each: AAOTOKK USB Type C 3.1Panel Mount Adapter Cable (amazon.com)
- 1 pack: MCIGICM 10pcs Male Header Pin, 40 Pin Header Strip (2.54 mm) for Arduino Connector (amazon.com)
- 2 each: 5mm LED (amazon.com)
- 2 each: 5mm LED panel mount holder (amazon.com)
- 1 each: DOLLATEK 36Pcs Double Sided PCB Board Prototype Kit 5 Sizes Universal Printed Circuit Protoboard) (amazon.com)
- ABS Plastic Sheets, 1/8 Inch thick
- Double sided tape (M3 Scotch Adhesive Transfer Hi-Strength Acrylic 5 mil) 3m part number: 40657

# Programming the ESP32:

This code can also be found at my GitHub:

<https://github.com/douggmiller/ESP32-BLE-Arcade-Joystick-LEGO-Arduino/blob/master/ARCADEv3/ARCADEv3.ino>.

If you need to set up your ESP32 for the first time, you will find a link of the tutorial in the introduction. This program uses the BleGamepad.h library and can be installed from the Library manager in Arduino IDE. This code is a modified example from the BleGamepad.h library. It uses the Bounce2.h library which might also need to be installed through the library manager if it wasn't installed as a dependency from the previous library. Where the code says:  
BleGamepad bleGamepad("Name", "Name", 100);  
The variables for "Name" is meant to name your device and to add company information, you can also customize the naming how you see fit. Once programmed, it is suggested to try connecting a few buttons to test before you continue building.

```
#define BOUNCE_WITH_PROMPT_DETECTION // Make button state changes available immediately

#include <Arduino.h>
#include <Bounce2.h> // https://github.com/thomasfredericks/Bounce2
#include <BleGamepad.h> // https://github.com/lemmingDev/ESP32-BLE-Gamepad
#define numButtons 13

Bounce debouncers[numButtons];

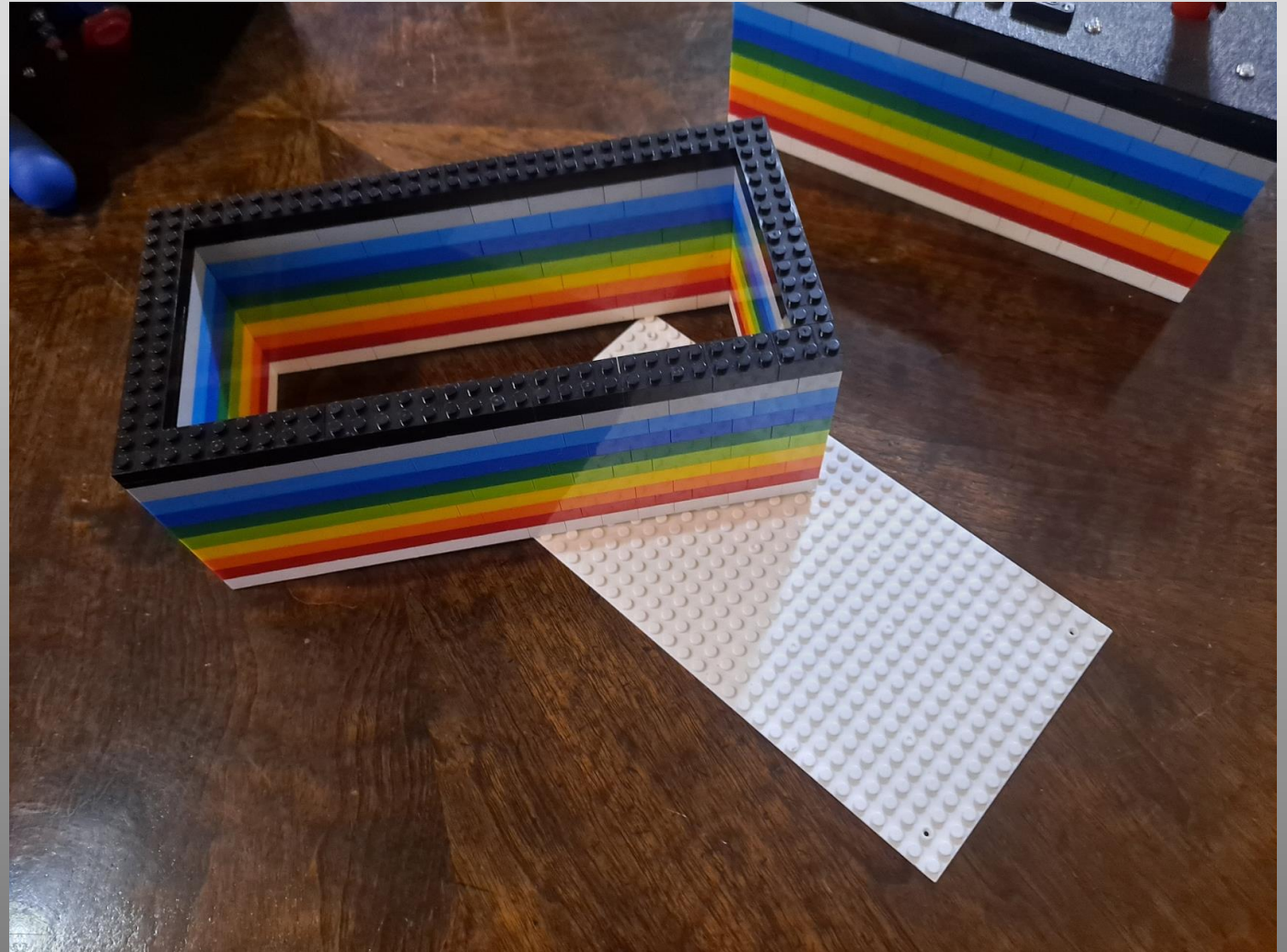
//BleGamepad bleGamepad("Name1", "Name1", 100);
BleGamepad bleGamepad("Name2", "Name2", 100);

byte buttonPins[numButtons] = { 4, 14, 16, 17, 19, 21, 22, 23, 25, 26, 27, 32, 33 };
byte physicalButtons[numButtons] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 };

void setup() {
  for (byte currentPinIndex = 0; currentPinIndex < numButtons; currentPinIndex++) {
    pinMode(buttonPins[currentPinIndex], INPUT_PULLUP);
    debouncers[currentPinIndex] = Bounce();
    debouncers[currentPinIndex].attach(buttonPins[currentPinIndex]);
    debouncers[currentPinIndex].interval(5);
  }
  BleGamepadConfiguration bleGamepadConfig;
  bleGamepadConfig.setButtonCount(numButtons);
  bleGamepadConfig.setAutoReport(false);
  bleGamepad.begin(&bleGamepadConfig);
}

void loop() {
  if (bleGamepad.isConnected()) {
    bool sendReport = false;
    for (byte currentIndex = 0; currentIndex < numButtons; currentIndex++) {
      debouncers[currentIndex].update();
      if (debouncers[currentIndex].fell()) {
        bleGamepad.press(physicalButtons[currentIndex]);
        sendReport = true;
      } else if (debouncers[currentIndex].rose()) {
        bleGamepad.release(physicalButtons[currentIndex]);
        sendReport = true;
      }
    }
    if (sendReport) {
      bleGamepad.sendReport();
    }
  }
}
```

Build the LEGO box: Build the walls around the outside edges of a 5-inch by 10-inch LEGO base plate. Stack the walls 9 LEGO bricks high in order to have space to fit the electrical components inside of it. The trick to building most things with LEGO bricks is to lay them like actual bricks, where they should overlap each other. This will create a stable structure for your components.

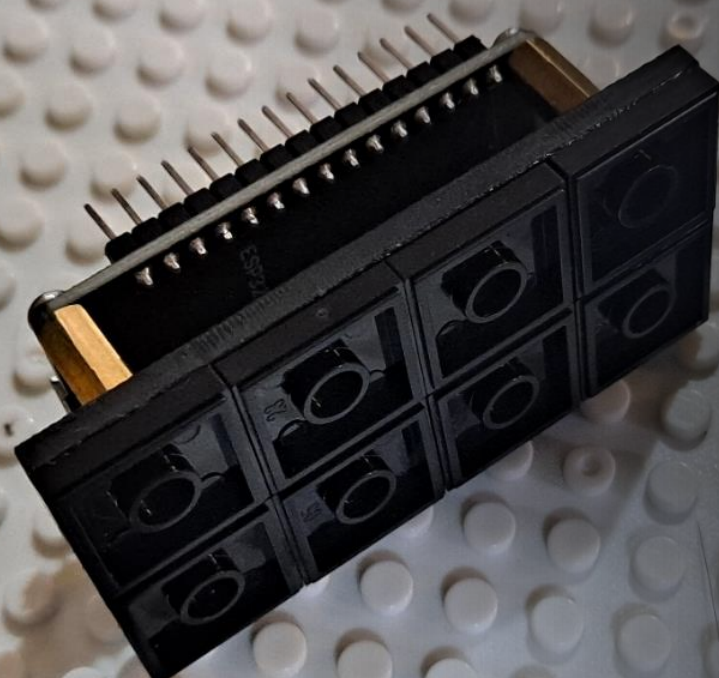




To turn the Arduino ESP32 and dev boards into LEGO compatible devices you will need to do a few things.

First, cut some ABS plastic into rectangles slightly larger than the dev boards. Make sure the dimensions will fit perfectly onto the standard dimensions of the LEGO brick tiles. Second, drill mounting holes into the ABS plastic using the same dimensions as the mounting holes on the device. On the bottom side counter sink the holes. Third, mount the boards onto the ABS plastic using M2.5 screws and standoffs. On the bottom side use the counter sunk screws. The batteries were taped directly onto the ABS plastic without the need to drill any holes. Finally, tape the bottom of the ABS plastic on to the top of the LEGO brick tiles.

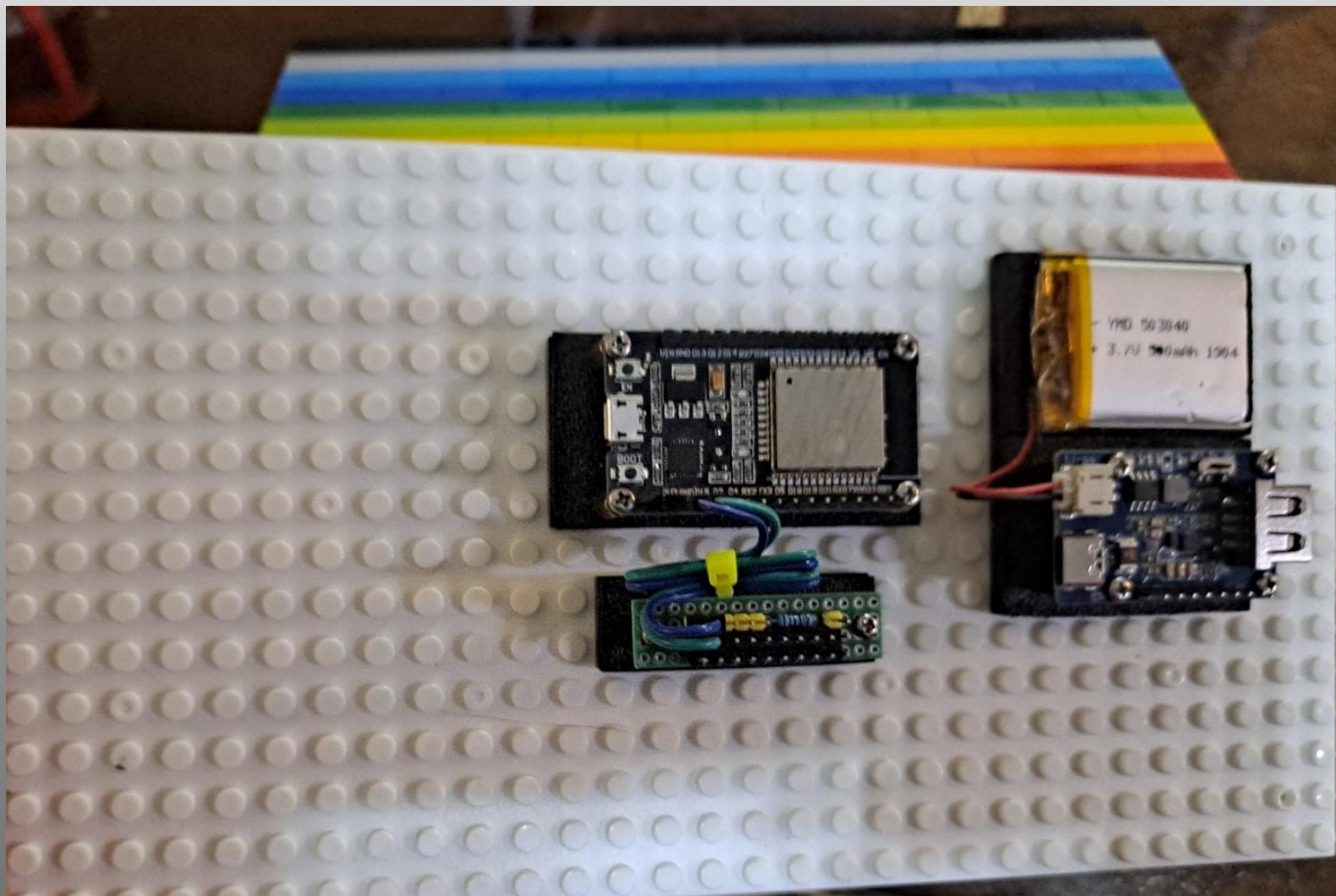


A black ESP32 microcontroller module is mounted on a white LEGO Technic plate. The module is oriented vertically, showing its pins and the integrated circuit. The plate is placed on a dark wooden surface. The text "And we have an ESP32 Arduino LEGO." is overlaid on the right side of the image, with a white horizontal line underneath it.

And we have  
an ESP32  
Arduino LEGO.

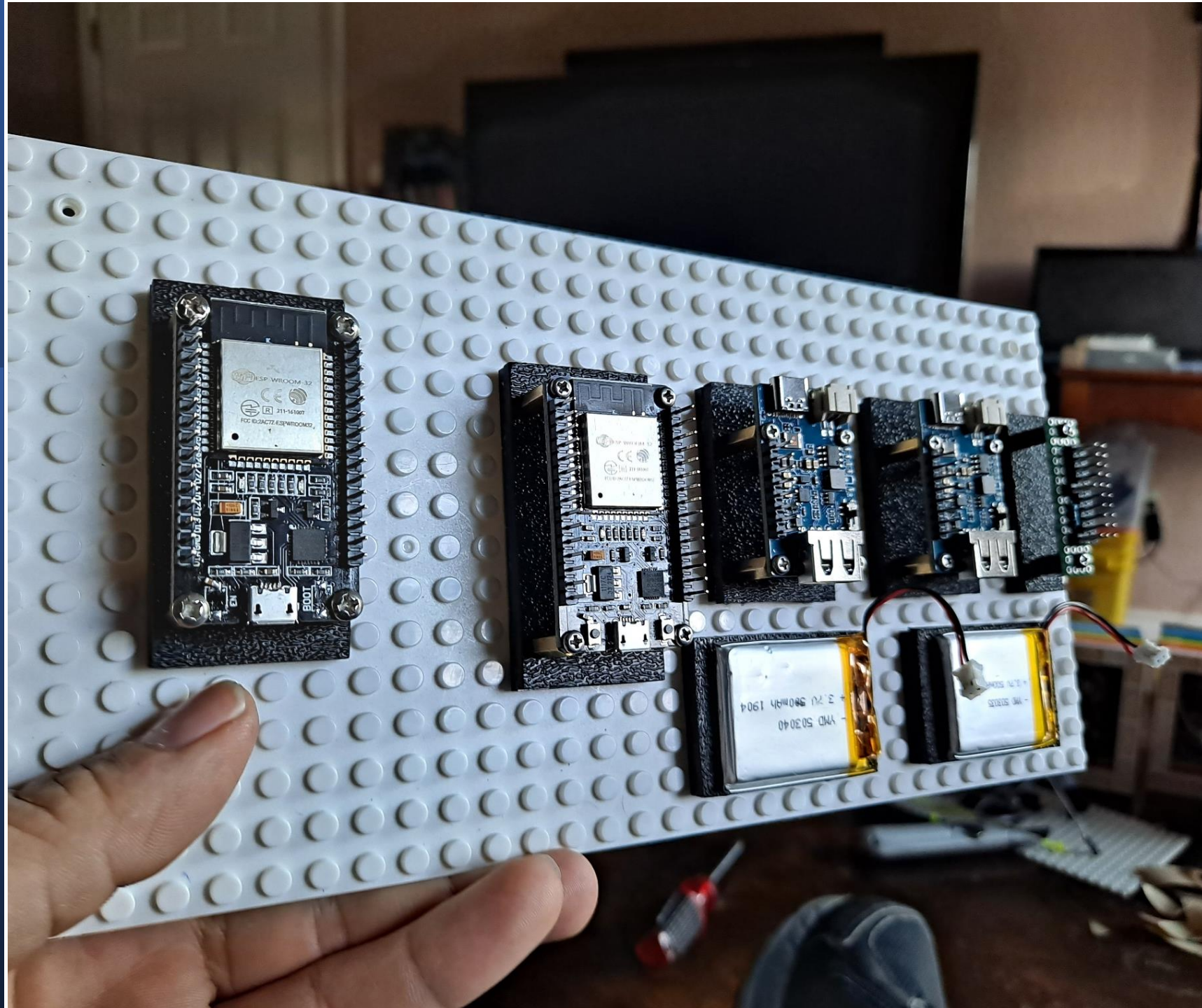


# Example Photo

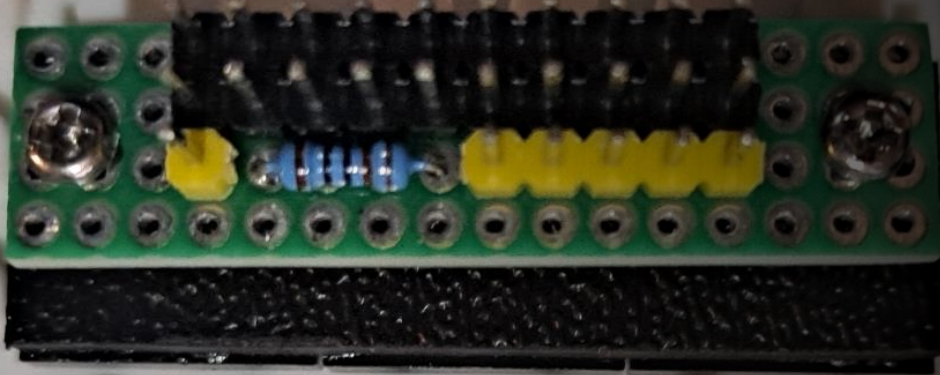




You can make  
enough to do two  
joysticks.







You may have noticed that one of the boards doesn't look like the others. This is just a simple DIY bus bar. This may be optional, but it is highly recommended to build this. Having a place to connect all your ground and 3.3V wires will help with cable management in future steps. The black terminals are meant for ground, and the yellow terminals are meant for 3.3V. Add a 1kΩ resistor attached to 3.3V to have a place to connect your power indicator LED as well.

---

I experimented with straight button placement to save space, but having a more ergonomic design for hand comfort is well worth the extra design time.

<https://www.slagcoin.com/joystick/layout.html>

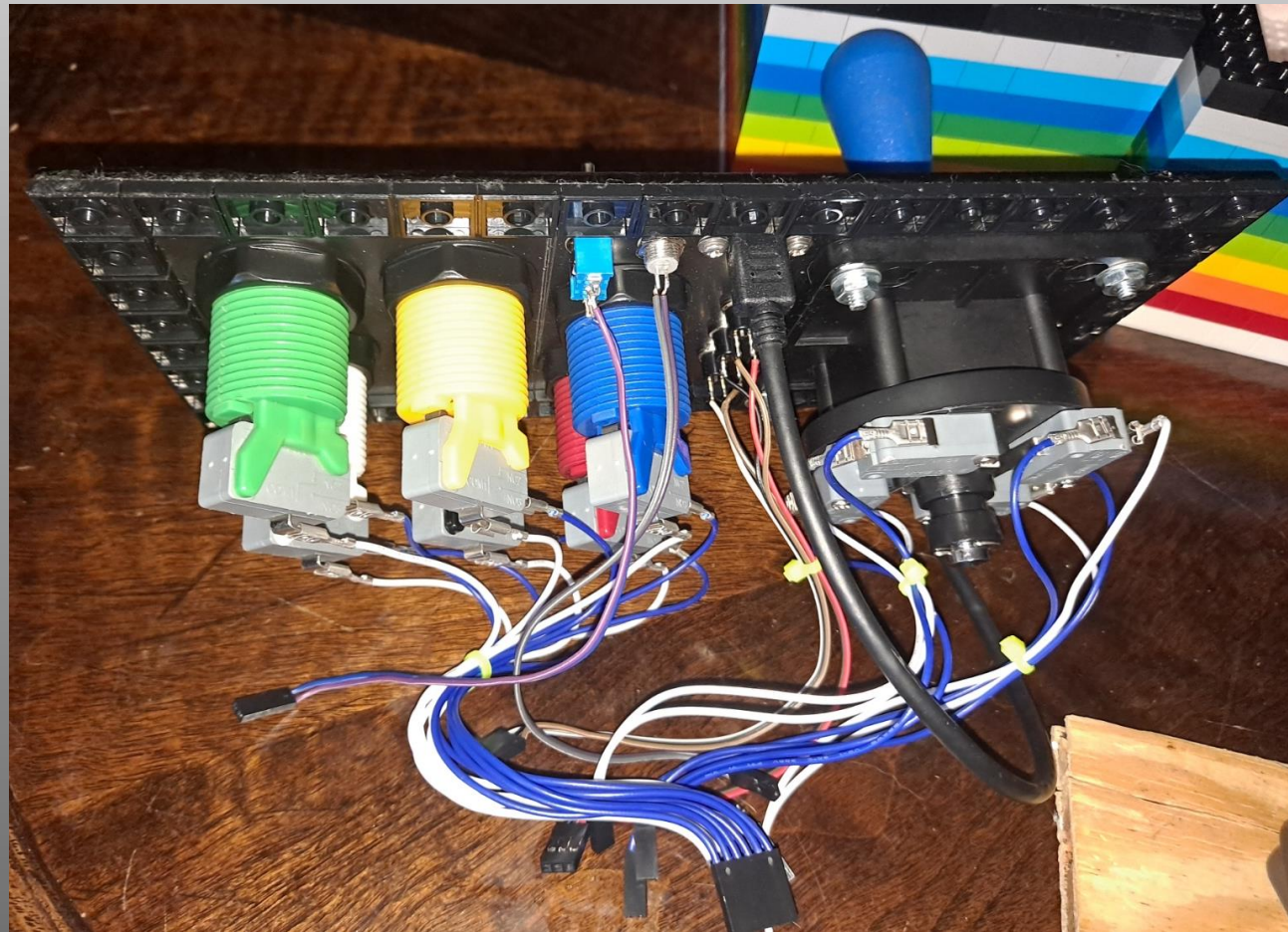
Image: Generic shift layout ([96 PPI](#) [100 PPI](#) [300 PPI](#)) (36.5mm spacing: [72 PPI](#) [96 PPI](#) [100 PPI](#) [300 PPI](#))



Cutting joystick and push-button holes can be done with a hole saw, drill, or a laser cutter. The joystick and large buttons use 1 ¼-inch holes. Three smaller push buttons were added to do non-gameplay functions such as start, adding coins, and accessing menu options. A USB-type C connector is installed for battery charging, a toggle switch is used to power the device on and off, and a standard blue LED is used for power indication. The LED will shut off as soon as the ESP32 3.3V regulator shuts off, letting you know when the battery needs charging. The library that is used in the ESP32 code does have battery level indication. Currently this function is not utilized, but this feature could be added.



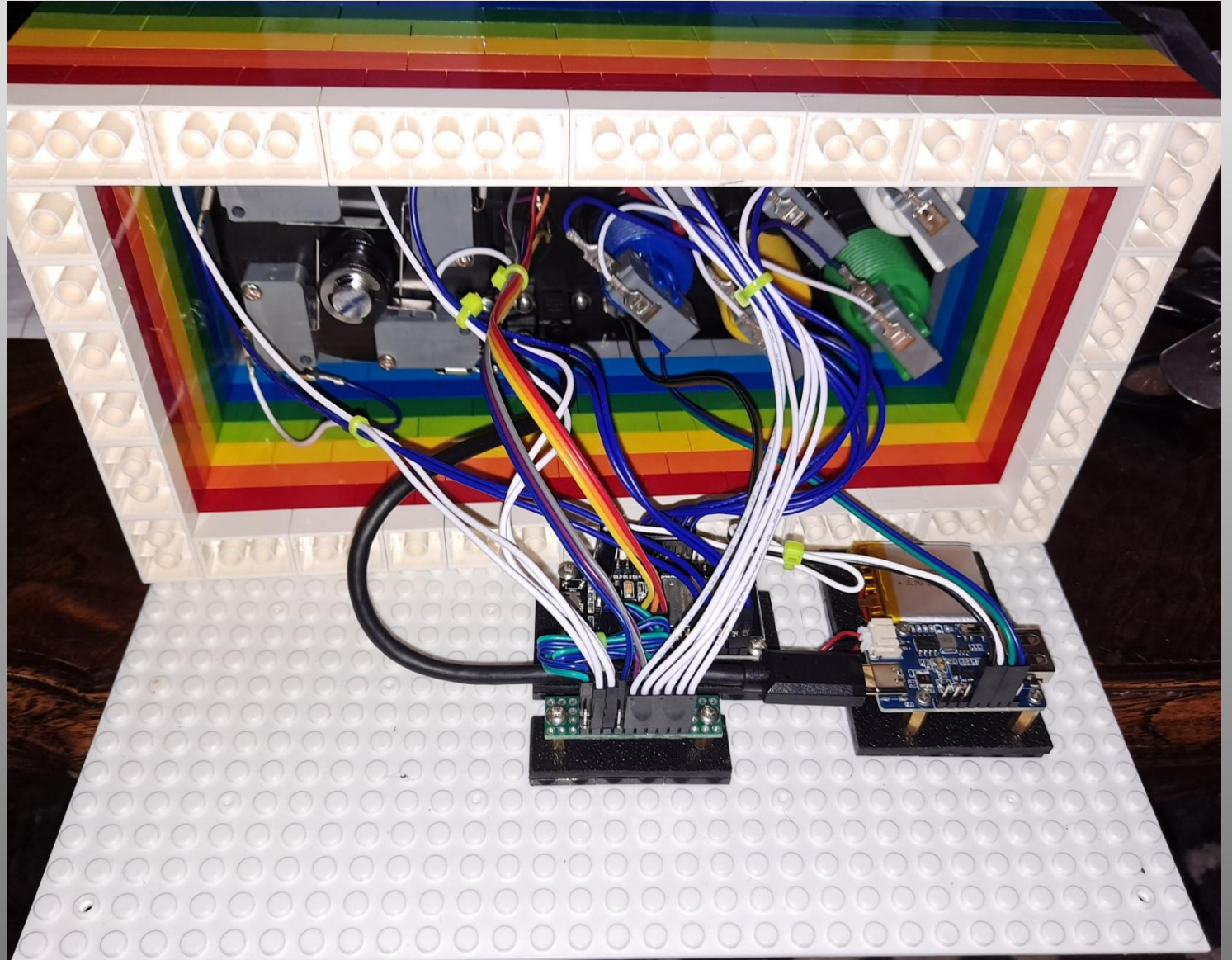
The wires that come with the arcade buttons and joystick have a spade connector on one end and a JST-XH 2.54mm on the other end. These wires will need to be modified. Cut the JST-XH connectors off and crimp on DuPont style connectors that are compatible with the ESP32 pin headers. Another way you could do this is to cut one end of the ribbon cable that is listed in the BOM, then crimp on spade connectors where you cut. There are other ways that cabling could be added, just be creative. For all other switches and the LED, cut one end of a ribbon cable, strip the wire back, and solder directly on to the components. Finally, tape LEGO brick tiles around the outer edge of the ABS plastic with double sided tape.





Install the ABS plastic to the LEGO box, then connect the wires. Once wired up, install the base plate to the bottom of the LEGO box.

---







Now enjoy your favorite classic arcade games.

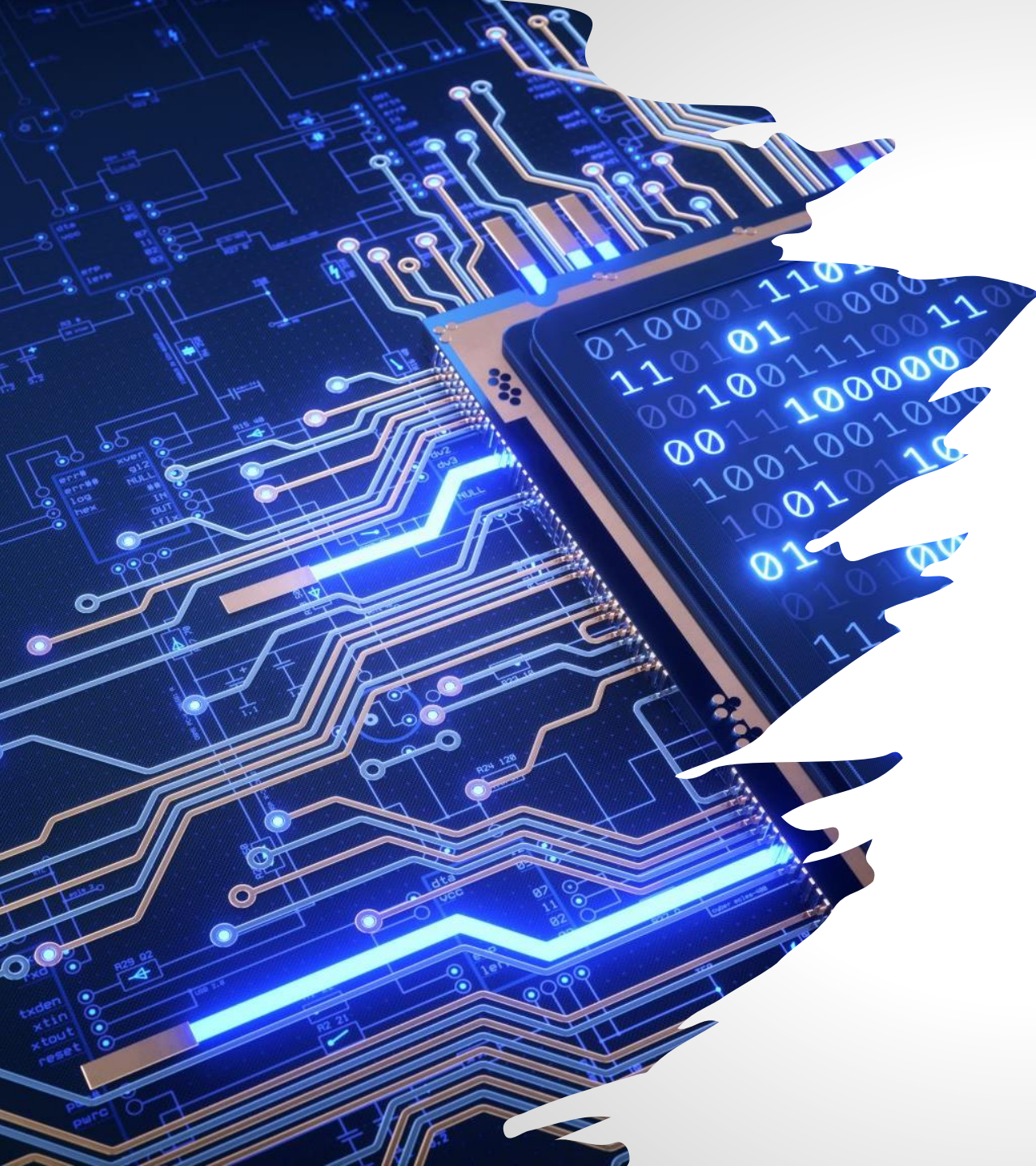




# Special Thanks

- I'd like to thank lemmingDev for writing the BleGamepad.h library, their GitHub is at: <https://github.com/lemmingDev>.
- I'd like to thank thomasfredricks for writing the bounce2.h library, their GitHub can be found at: <https://github.com/thomasfredericks>.





# References

- *Thomasfredericks - Overview.* (n.d.). GitHub. <https://github.com/thomasfredericks>
- *LemmingDev - Overview.* (n.d.). GitHub. <https://github.com/lemmingDev>
- *Installing ESP32 in Arduino IDE (Windows, Mac OS X, Linux).* (2023, March 9). Random Nerd Tutorials. <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>
- *BLE-arcade-Joystick-LEGO-Arduino/ARCADEv3.ino at master · douggmill/BLE-arcade-Joystick-LEGO-Arduino.* (n.d.). GitHub. <https://github.com/douggmill/BLE-Arcade-Joystick-LEGO-Arduino/blob/master/ARCADEv3/ARCADEv3.ino>