

HotSwap Arcade Kit

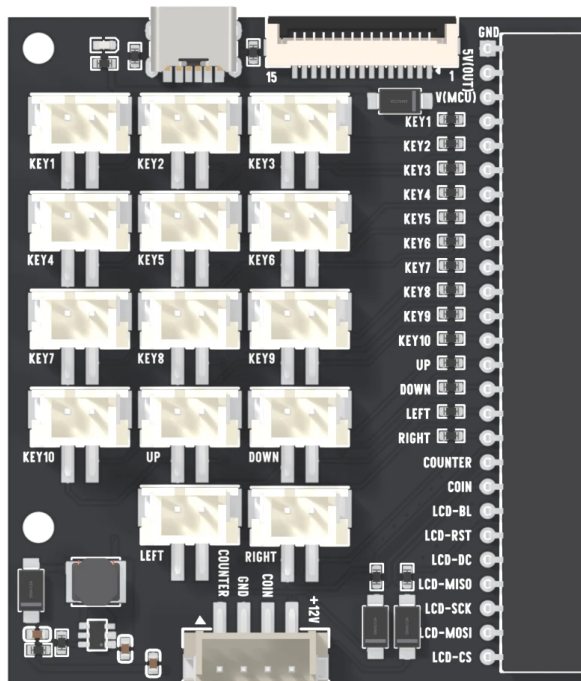
One mainboard for ten buttons, a joystick, a coin acceptor and a screen. Four adapters: ESP32-S3, Nano, Pico, Zero 2 W.



Unplug every cable before you change adapters

An adapter one position out puts the mainboard's 5 V onto your board's ground or its 3.3 V pin. **USB-C takes 5 V only**: the acceptor's 12 V is made on the mainboard, and nothing may feed 12 V in.

The mainboard



Position 1, GND, is at the USB-C end. The adapter plugs into the long socket on the right edge and lies flat beside the mainboard, printed side up.

Two USB sockets

PLUGGED IN	BUTTONS	ACCEPTOR	SERIAL
Board USB only	✓	—	✓
USB-C only	✓	✓	—
Both	✓	✓	✓

The mainboard's USB-C has no data lines: programs and messages go through the development board's own socket. A diode stops that socket feeding the mainboard, so with it alone the power LED stays dark and the acceptor has no 12 V.

Where each position lands

#	MAINBOARD	ESP32-S3	NANO	PICO	ZERO 2 W
1	GND	GND	GND	GND	GND
2	5V (OUT)	5V	5V	VBUS	5V
3	V(MCU)	3V3	5V	3V3	3V3
4	KEY1	GPI04	D2	GP16	BCM4
5	KEY2	GPI05	D3	GP17	BCM5
6	KEY3	GPI06	D4	GP18	BCM6
7	KEY4	GPI07	D5	GP19	BCM12
8	KEY5	GPI015	D6	GP20	BCM13
9	KEY6	GPI016	D7	GP21	BCM16
10	KEY7	GPI017	D8	GP15	BCM17
11	KEY8	GPI018	D9	GP14	BCM19
12	KEY9	GPI08	D10	GP13	BCM14
13	KEY10	GPI040	D11	GP12	BCM15
14	UP	GPI039	D12	GP22	BCM20
15	DOWN	GPI038	A4	GP11	BCM21
16	LEFT	GPI09	A3	GP10	BCM22
17	RIGHT	GPI047	A2	GP26	BCM23
18	COUNTER	GPI021	A1	GP27	BCM24
19	COIN	GPI014	A0	GP28	BCM26
20	LCD-BL	GPI041	—	GP9	BCM18
21	LCD-RST	GPI042	—	GP8	BCM27
22	LCD-DC	GPI02	—	GP7	BCM25
23	LCD-MISO	GPI013	—	GP6	BCM9
24	LCD-SCK	GPI012	—	GP5	BCM11
25	LCD-MOSI	GPI011	—	GP4	BCM10
26	LCD-CS	GPI010	—	GP3	BCM8

Pressed reads HIGH

Each button joins its input to V(MCU), your board's own supply: ESP32-S3, Pico, Zero 2 W at 3.3 V, the Nano at 5 V. A 10 kΩ resistor holds it LOW when released. So no pull-up in code, and no level shifter: a pin never sees more than its own board's voltage.

Input monitor, the first program

Arduino IDE · Nano: Board Arduino Nano, ATmega328P · ESP32-S3: ESP32S3 Dev Module, USB CDC On Boot Enabled · Serial Monitor 115200

```
#include <Arduino.h>

#if defined(ARDUINO_AVR_NANO)
const uint8_t INPUT_PINS[] = {2,3,4,5,6,7,8,9,10,11,12,A4,A3,A2};
const uint8_t INPUT_MODE = INPUT; // 10k pull-downs on the mainboard
#elif defined(CONFIG_IDF_TARGET_ESP32S3)
const uint8_t INPUT_PINS[] = {4,5,6,7,15,16,17,18,8,40,39,38,9,47};
const uint8_t INPUT_MODE = INPUT_PULLDOWN;
#else
#error "Select classic Arduino Nano or ESP32S3 Dev Module."
#endif

const char *NAMES[] = {"KEY1", "KEY2", "KEY3", "KEY4", "KEY5",
                       "KEY6", "KEY7", "KEY8", "KEY9", "KEY10",
                       "UP", "DOWN", "LEFT", "RIGHT"};

bool raw[14], stable[14];
unsigned long changed[14];

void setup() {
  Serial.begin(115200);
  delay(1000);
  for (uint8_t i=0; i<14; ++i) {
    pinMode(INPUT_PINS[i], INPUT_MODE);
    raw[i] = stable[i] = digitalRead(INPUT_PINS[i]) == HIGH;
    changed[i] = millis();
  }
  Serial.println("Ready. Press a button or push the stick.");
}

void loop() {
  unsigned long ms = millis();
  for (uint8_t i=0; i<14; ++i) {
    bool value = digitalRead(INPUT_PINS[i]) == HIGH;
    if (value != raw[i]) { raw[i] = value; changed[i] = ms; }
    if (value != stable[i] && ms - changed[i] >= 30) {
      stable[i] = value;
      Serial.print(NAMES[i]);
      Serial.println(value ? " pressed" : " released");
    }
  }
  delay(1);
}
```

KEY1 pressed
KEY1 released
Press each button and push the stick each way. The Pico and Zero 2 W versions print the same lines.

When it does not work

Nothing prints

The cable is in the mainboard's USB-C, which has no data lines. Use the development board's own socket, at 115200.

Joystick directions swapped

Wired by position. Push each way, read the name the monitor prints, and move that lead.

One coin counts as several

The acceptor's pulses are further apart than the 200 ms group gap. Lengthen the gap or speed up the pulses.

Specifications

Supply	USB-C, 5 V, 2 A or more	Acceptor supply	About 12 V, MT3608 boost
Switch inputs	14, 10 kΩ pull-down each	Logic level	3.3 V, or 5 V on the Nano
Coin lines	COIN and COUNTER, active LOW	Adapter socket	26 positions, 2.54 mm, right-angle
Screen socket	15-way FPC, 1.0 mm	Code	Kit examples, tag v1.0

Coins

The acceptor socket reads COUNTER, GND, COIN, +12V. COIN is header position 19 and COUNTER 18; each passes through a diode and is pulled up to V(MCU), so a pulse is the line going LOW and the acceptor's 12 V never reaches a pin.

One coin is a burst of pulses. The kit's code ignores an edge within 60 ms of the last and ends a coin after 200 ms of silence. The acceptor's pulse spacing has to fall between the two.

Joystick

Four microswitches, wired COM and NO. The switch that closes is opposite the way the handle moves, so UP takes the switch nearest you. A diagonal closes two.

Screen

A 15-way 1.0 mm ribbon socket. Contacts 3 to 9 are BL, RESET, DC, MISO, SCK, MOSI and CS, on header positions 20 to 26. The Nano adapter does not connect them. The kit's display code is for an ILI9488 panel.

A button always reads pressed

The switch is on COM and NC. Move the wire from NC to NO.

A coin is taken and nothing counts

The acceptor runs on the mainboard's 12 V: plug in the USB-C as well as the board's USB.

KEY9, KEY10 silent on a Zero 2 W

They are the serial port's pins. Turn off the serial console and serial hardware in raspi-config, then reboot.