**Com direcionais**

#include <Joystick.h>

// Inicializa o joystick com 12 botões e 2 eixos (X e Y)

Joystick\_ Joystick(JOYSTICK\_DEFAULT\_REPORT\_ID,

  JOYSTICK\_TYPE\_GAMEPAD, 12, 0, // 12 botões, 0 eixos "viram" 2 eixos abaixo

  true, true, true, true, true, true, true, true, true, true);

// LEDs

const byte LED1 = A5;

const byte LED2 = A4;

const byte LED3 = A3;

const byte LED4 = A2;

const byte LED5 = A1;

const byte LED6 = A0;

const byte LED7 = 13;

const byte LED8 = 12;

// Pinos dos botões

const byte BTN\_DIREITA  = 0;

const byte BTN\_ESQUERDA = 1;

const byte BTN\_BAIXO    = 2;

const byte BTN\_CIMA     = 3;

const byte SELECT       = 4;

const byte START        = 5;

const byte BOTAO\_F      = 6;

const byte BOTAO\_E      = 7;

const byte BOTAO\_D      = 8;

const byte BOTAO\_C      = 9;

const byte BOTAO\_B      = 10;

const byte BOTAO\_A      = 11;

void setup() {

  // Configura LEDs

  pinMode(LED1 , OUTPUT);

  pinMode(LED2 , OUTPUT);

  pinMode(LED3 , OUTPUT);

  pinMode(LED4 , OUTPUT);

  pinMode(LED5 , OUTPUT);

  pinMode(LED6 , OUTPUT);

  pinMode(LED7 , OUTPUT);

  pinMode(LED8 , OUTPUT);

  // Configura botões como entrada com pull-up

  for (int i = 0; i <= 11; i++) {

    pinMode(i, INPUT\_PULLUP);

  }

Joystick.begin();

}

void loop() {

  // Simula eixo X (esquerda/direita)

  int eixoX = 0;

  if (digitalRead(BTN\_ESQUERDA) == LOW) eixoX = -127;

  if (digitalRead(BTN\_DIREITA) == LOW)  eixoX = 127;

  Joystick.setXAxis(eixoX);

  // Simula eixo Y (cima/baixo)

  int eixoY = 0;

  if (digitalRead(BTN\_CIMA) == LOW)  eixoY = -127;

  if (digitalRead(BTN\_BAIXO) == LOW) eixoY = 127;

  Joystick.setYAxis(eixoY);

  // Botões digitais (4 a 11)

  for (int i = 4; i <= 11; i++) {

    bool estado = digitalRead(i) == LOW;

    Joystick.setButton(i, estado);

    // LEDs

    switch (i) {

      case 4: digitalWrite(LED7, estado); break; // Select

      case 5: digitalWrite(LED8, estado); break; // Start

      case 6: digitalWrite(LED1, estado); break; // F

      case 7: digitalWrite(LED2, estado); break; // E

      case 8: digitalWrite(LED3, estado); break; // D

      case 9: digitalWrite(LED4, estado); break; // C

      case 10: digitalWrite(LED5, estado); break; // B

      case 11: digitalWrite(LED6, estado); break; // A

    }

  }

  delay(10); // Estabilização

}

**Sem direcionais**

#include

// Inicializa o joystick com 12 botões e sem eixos

Joystick\_ Joystick(JOYSTICK\_DEFAULT\_REPORT\_ID,

  JOYSTICK\_TYPE\_GAMEPAD, 12, 0, // 12 botões, 0 eixos

  false, false, false, false, false, false, false, false, false, false);

// LEDs

const byte LED1 = A5;

const byte LED2 = A4;

const byte LED3 = A3;

const byte LED4 = A2;

const byte LED5 = A1;

const byte LED6 = A0;

const byte LED7 = 13;

const byte LED8 = 12;

// Botões de ação (A-F)

const byte BOTAO11 = 11; // A

const byte BOTAO10 = 10; // B

const byte BOTAO9  = 9;  // C

const byte BOTAO8  = 8;  // D

const byte BOTAO7  = 7;  // E

const byte BOTAO6  = 6;  // F

// Select e Start

const byte BOTAO5 = 5;   // Start

const byte BOTAO4 = 4;   // Select

// Direcionais

const byte BOTAO3 = 3;   // Cima

const byte BOTAO2 = 2;   // Baixo

const byte BOTAO1 = 1;   // Esquerda

const byte BOTAO0 = 0;   // Direita

void setup() {

  // Configura LEDs

  pinMode(LED1 , OUTPUT);

  pinMode(LED2 , OUTPUT);

  pinMode(LED3 , OUTPUT);

  pinMode(LED4 , OUTPUT);

  pinMode(LED5 , OUTPUT);

  pinMode(LED6 , OUTPUT);

  pinMode(LED7 , OUTPUT);

  pinMode(LED8 , OUTPUT);

  // Configura botões

  for (int i = 0; i <= 11; i++) {

    pinMode(i, INPUT\_PULLUP);

  }

  Joystick.begin();

}

void loop() {

  // Lê e envia estado dos botões

  for (int i = 0; i <= 11; i++) {

    bool estado = digitalRead(i) == LOW;

    Joystick.setButton(i, estado);

    // Acende LEDs correspondentes (opcional)

    switch (i) {

      case 4: digitalWrite(LED7, estado); break; // Select

      case 5: digitalWrite(LED8, estado); break; // Start

      case 6: digitalWrite(LED1, estado); break; // F

      case 7: digitalWrite(LED2, estado); break; // E

      case 8: digitalWrite(LED3, estado); break; // D

      case 9: digitalWrite(LED4, estado); break; // C

      case 10: digitalWrite(LED5, estado); break; // B

      case 11: digitalWrite(LED6, estado); break; // A

    }

  }

  delay(10); // Estabilização

}