

# **EXPERIENCIA BICI-VILIDAD**

**Por:**  
**Vicente Jaramillo**  
**Franco Andahur**

# CONTENIDO

**01**

OBEJETIVOS DE APRENDIZAJE

**02**

COMPONENTES

**03**

DESAFÍO

**04**

ARMADO

**05**

RESULTADOS

**06**

CONCLUSIÓN

# OBJETIVOS DE APRENDIZAJE

## Objetivo n° 1

Aprender a conectar una matriz de 8x8 y un joystick controlador.

## Objetivo n° 2

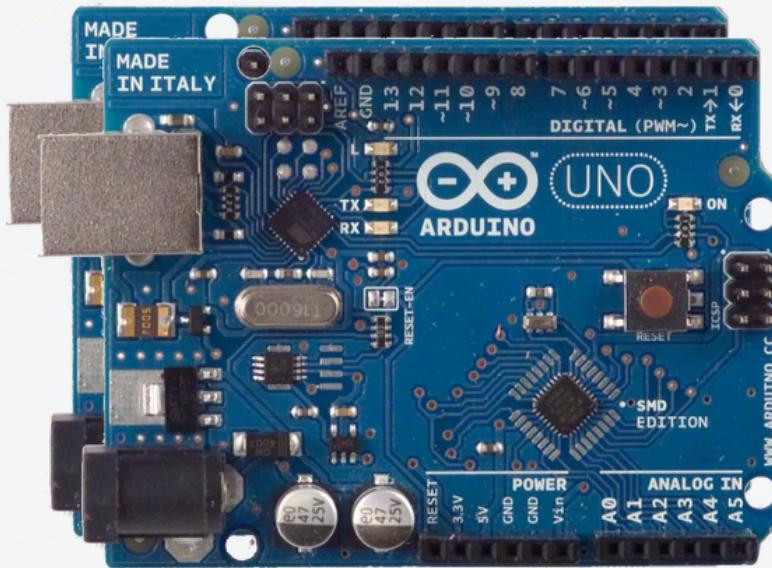
Aprender a programar los dispositivos utilizando arduino UNO.

## Objetivo n° 3

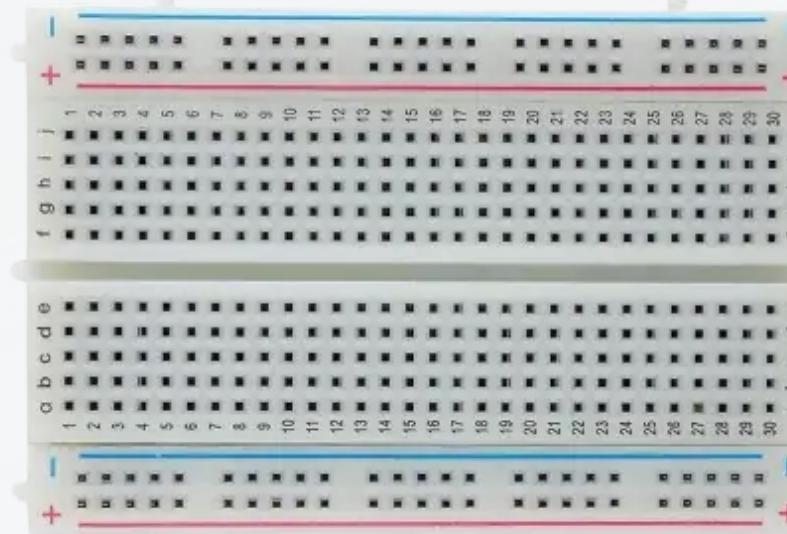
Emsamblar y orientar todos los componentes para lograr realizar las tareas descritas por el desafio



# COMPONENTES



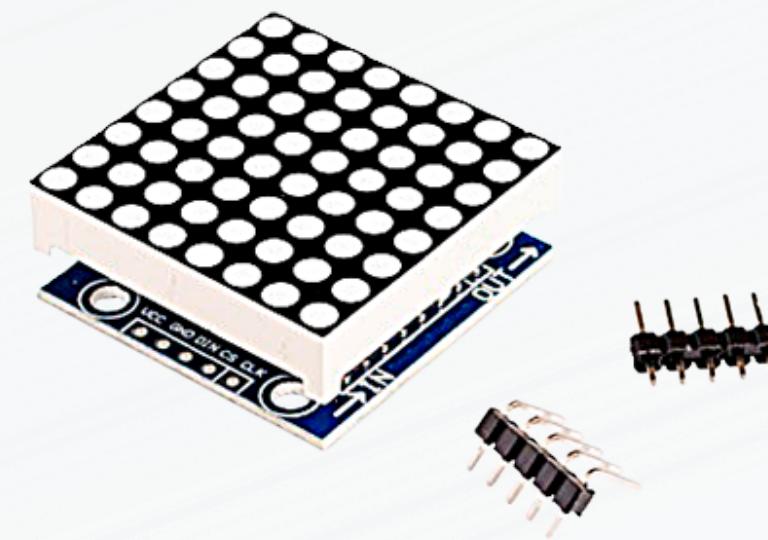
- x1 Placa de Arduino UNO



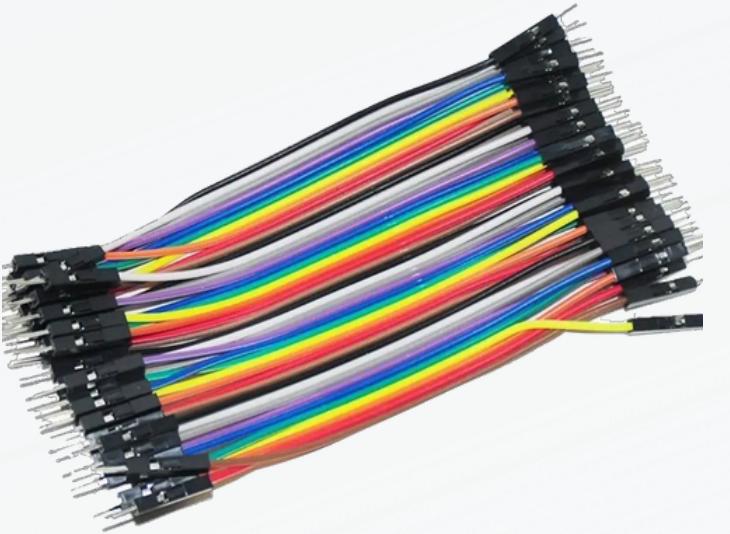
- x2 Protoboards



- x1 Joystick



- x1 Matriz 8x8



- Cables conectores

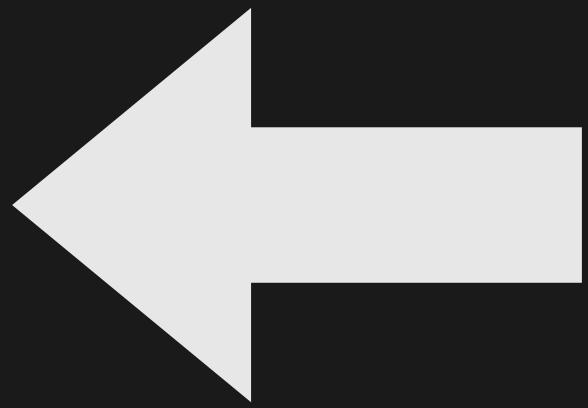
# DESAFIO

Lograr realizar un prototipo utilizando arduino, que muestre los estados de la posición del joystick, a través de la matriz 8x8.

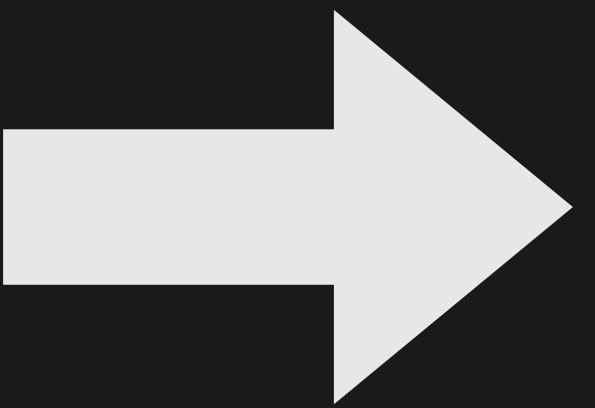
Los estados estarán representados por los siguientes simbolos:



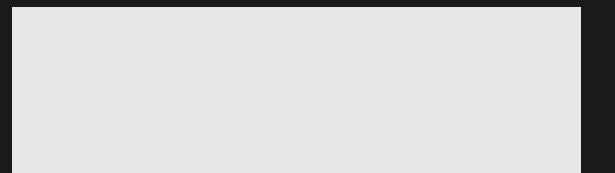
Forward



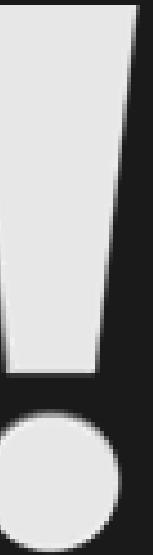
Left



Right



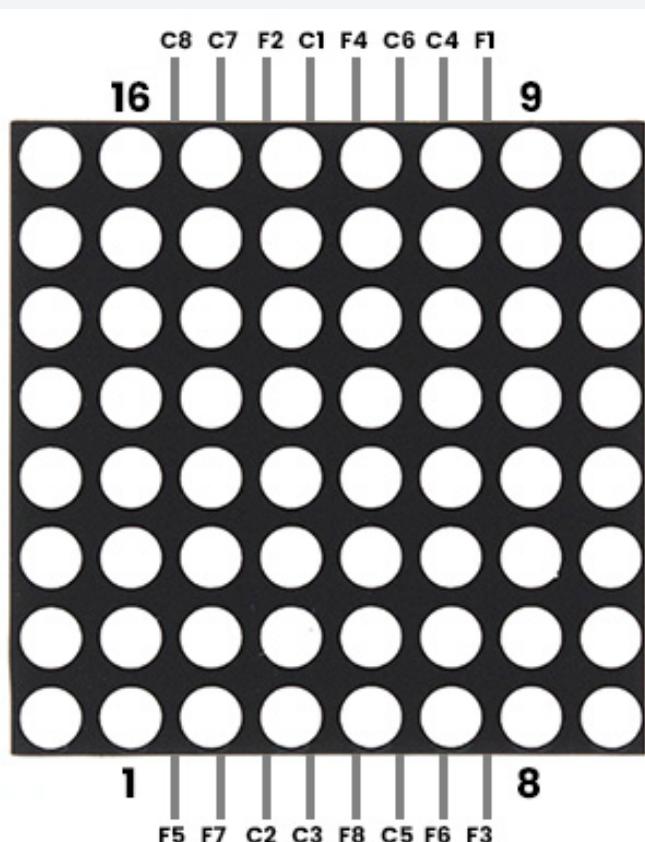
Stand-by



Stop

# ARMADO

- El ensamblaje de la matriz LED de 8x8 se llevó a cabo siguiendo cuidadosamente una serie de diagramas de conexión. Estas imágenes proporcionaron una guía clara para realizar el cableado necesario entre la matriz y el Arduino Uno, asegurando una correcta conexión de cada LED a los pines correspondientes de la placa. Gracias a este paso, se logró una integración adecuada que permite el control individual de cada punto en la matriz a través del Arduino.



PinOut Matriz Led 8x8

Pines	C1	C2	C3	C4	C5	C6	C7	C8
Columnas Matriz	13	3	4	10	6	11	15	16
Pines Matriz	2	3	4	5	6	7	8	9
Arduino Uno	F1	F2	F3	F4	F5	F6	F7	F8
Pines:	9	14	8	12	1	7	2	5
Filas Matriz	10	11	12	13	A0	A1	A2	A3
Pines Matriz	Arduino Uno							

Tabla de conexión

# ARMADO

- Luego investigamos el funcionamiento de la matriz 8x8 para controlar el encendido y apagado de cada uno de los 64 LEDs que la componen.
- Este mecanismo se basa en dos funciones esenciales, que permiten formar las figuras requeridas.
- Gracias a este enfoque, logramos crear las figuras que se mostrarán en la matriz LED.

```
// Encender un solo LED en la posición especificada (columna, fila)
void encenderUnLed(int columna, int fila) {
    // Apagar todos los LEDs antes de encender uno nuevo
    apagarMatriz();

    // Activar la columna y fila deseadas para encender el LED específico
    digitalWrite(columnas[columna], HIGH); // Activa la columna
    digitalWrite(filas[fila], LOW);      // Activa la fila
}

// Apagar todos los LEDs en la matriz
void apagarMatriz() {
    // Coloca todas las columnas en LOW y todas las filas en HIGH para apagar todos los LEDs
    for (int i = 0; i < 8; i++) {
        digitalWrite(columnas[i], LOW); // Desactivar todas las columnas
        digitalWrite(filas[i], HIGH);  // Desactivar todas las filas
    }
}
```

# ARMADO

- Posteriormente, investigamos el funcionamiento del joystick, el cual resultó ser mucho más sencillo en comparación con la matriz 8x8.
- El joystick opera mediante cuatro cables que proporcionan alimentación y transmiten los datos de salida.

```
int xPin = A4;
int yPin = A5;
int xVal; // variable para almacenar los valores del eje x del joystick
int yVal; // variable para almacenar los valores del eje y del joystick

void setup() {
    pinMode(xPin, INPUT);
    pinMode(yPin, INPUT);
    Serial.begin(9600); // inicializa el monitor serial
}

void loop() {
    // lee los valores de x, y y el estado del interruptor del joystick
    xVal = analogRead(xPin);
    yVal = analogRead(yPin);

    // imprime las lecturas en el monitor serial
    Serial.print("X: ");
    Serial.print(xVal);
    Serial.print(" | Y: ");
    Serial.println(yVal);
    |

    delay(100); // pausa para que las lecturas sean visibles
}
```

# ARMADO

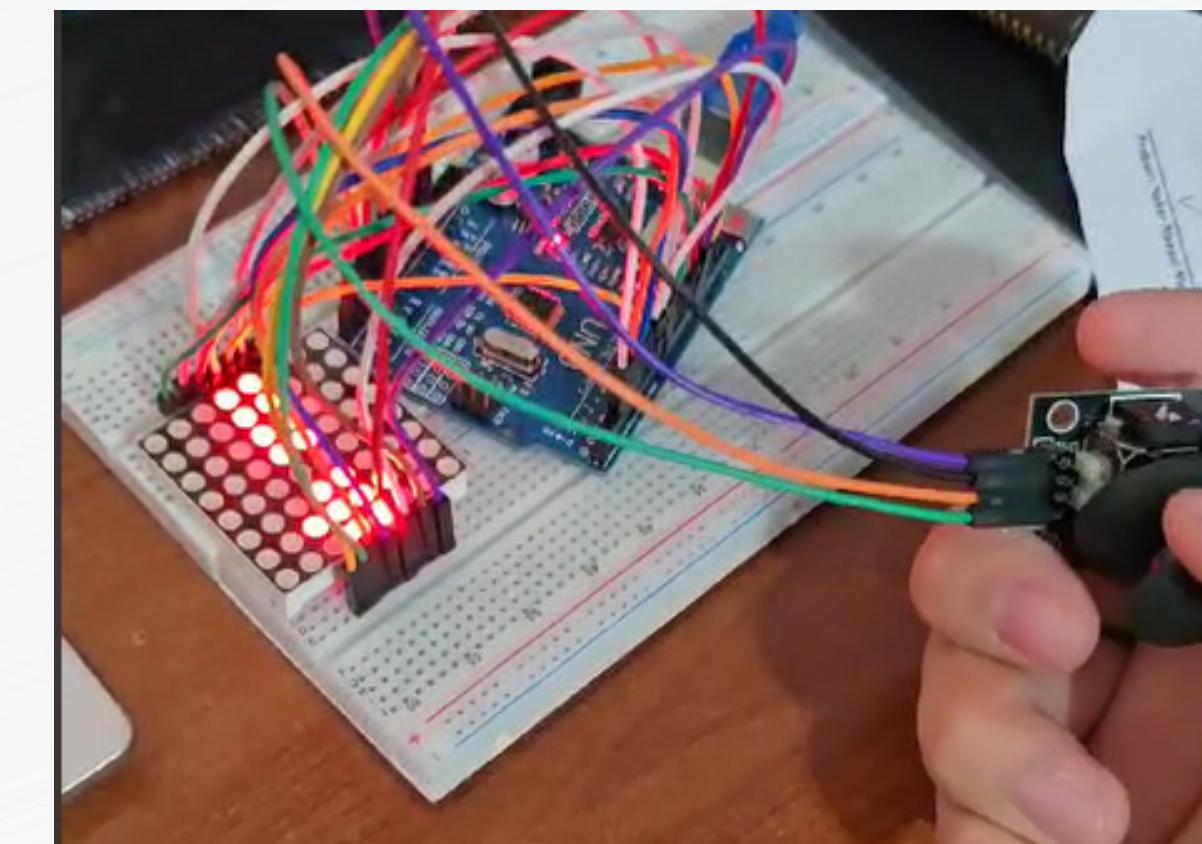
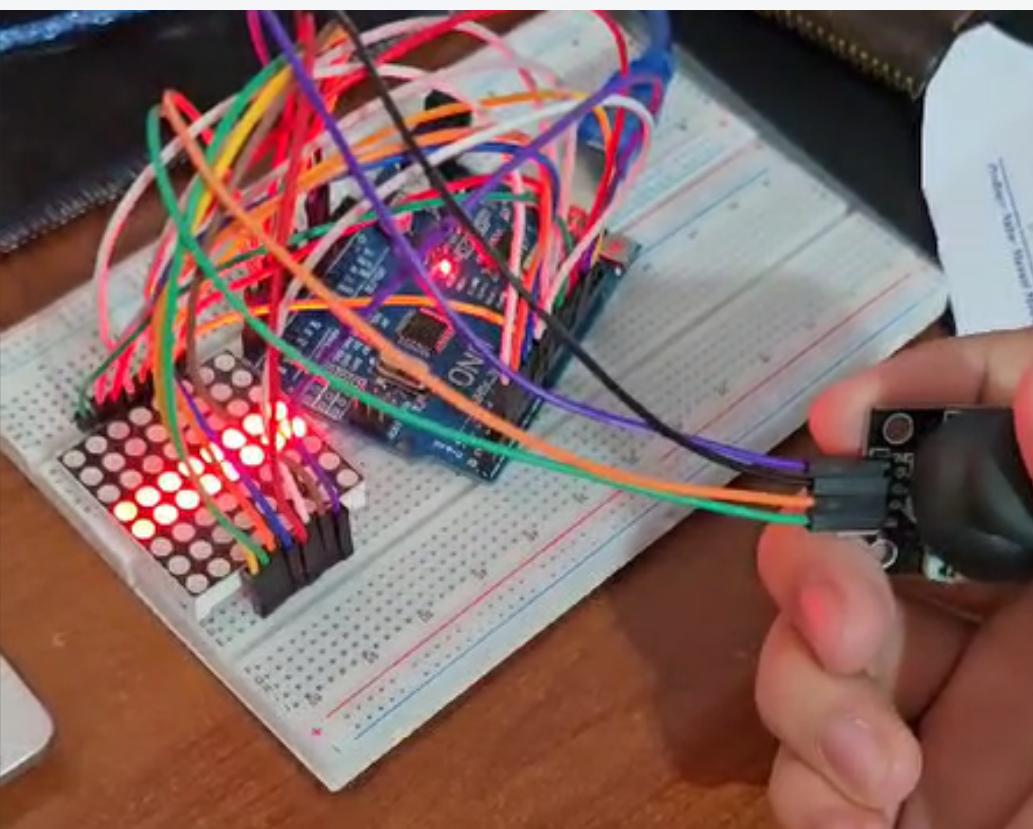
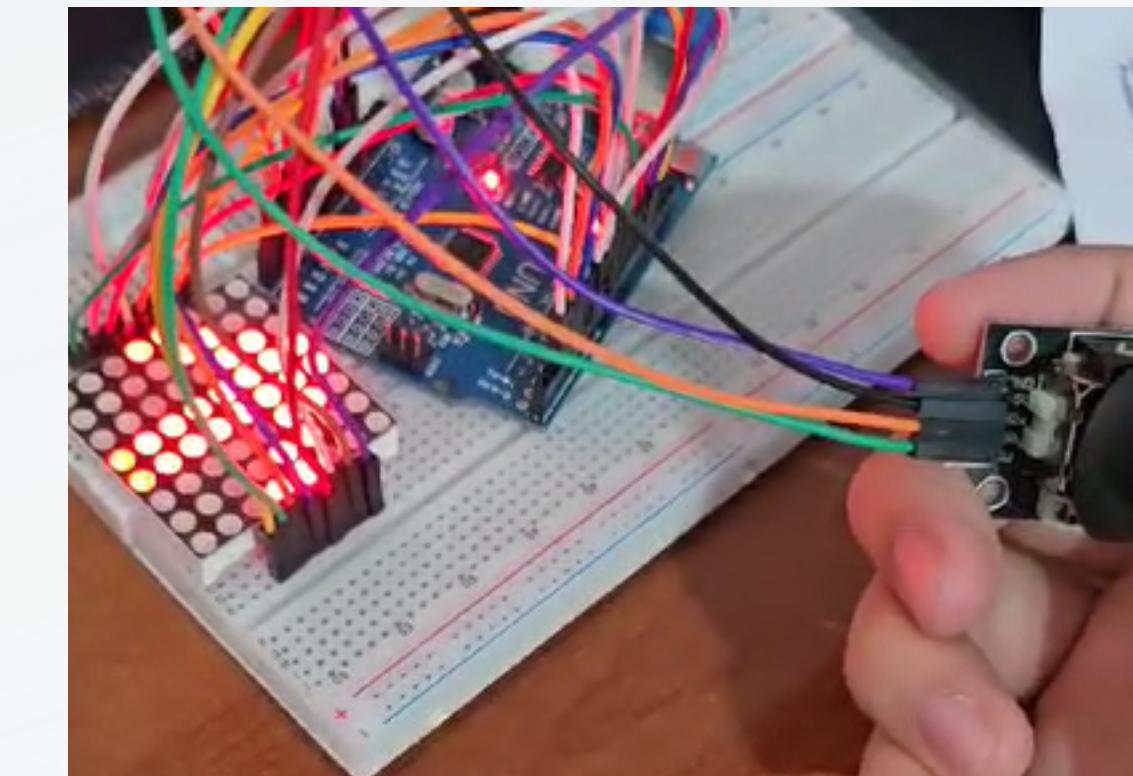
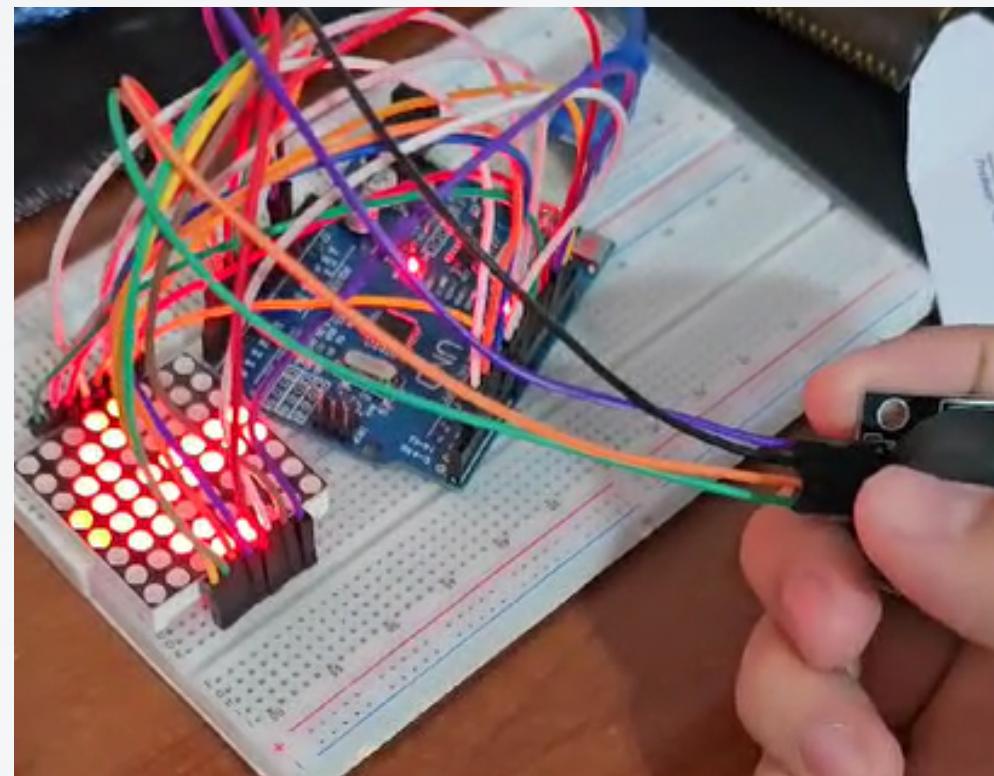
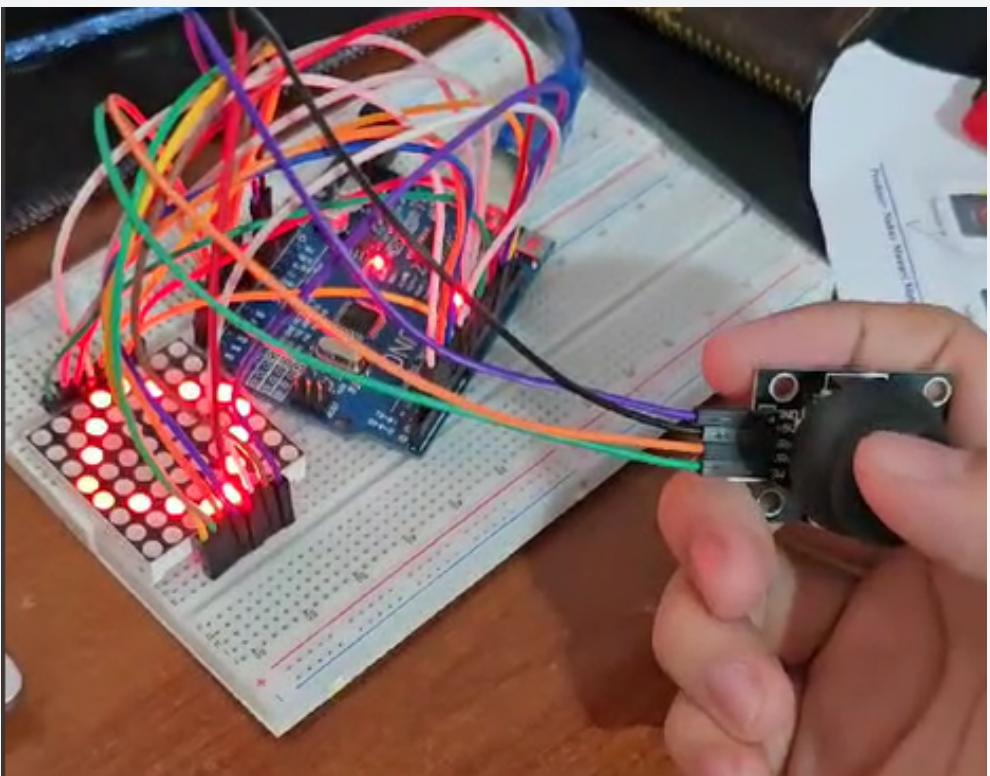
- Aprovechando las salidas del joystick, que varían de 0 a 1023 en los ejes X e Y, se crearon funciones para determinar la dirección del movimiento: izquierda, derecha, arriba o abajo. Estas funciones analizan los valores de salida y proporcionan información sobre la dirección en que apunta el joystick, considerando también una zona de reposo que se activa cuando el joystick no está en uso.

```
// Detectar el movimiento del joystick en las direcciones principales
if (xVal > 600) {
    moverDerecha(); // Mover derecha si el valor de X es alto
} else if (xVal < 400) {
    moverIzquierda(); // Mover izquierda si el valor de X es bajo
} else if (yVal > 600) {
    moverAbajo(); // Mover abajo si el valor de Y es alto
} else if (yVal < 400) {
    moverArriba(); // Mover arriba si el valor de Y es bajo
} else if (xVal > 512 - deadZone && xVal < 512 + deadZone && yVal > 512 - deadZone && yVal < 512 + deadZone) {
    enReposo(); // El joystick está en reposo dentro de la zona muerta
}
```

# ARMADO

- Finalmente, se integraron todos los programas en uno solo, combinando el control del joystick con la visualización en la matriz 8x8. Este programa unificado permite que, dependiendo del movimiento detectado en el joystick, la matriz LED muestre distintas figuras previamente creadas. Las funciones diseñadas interpretan la dirección en la que se mueve el joystick (izquierda, derecha, arriba o abajo) y, en respuesta, activan una figura específica en la matriz. Además, en la zona de reposo, se muestra una figura adicional cuando el joystick no está en uso. Así, el sistema es capaz de responder en tiempo real a los movimientos del usuario, proporcionando patrones visuales dinámicos y sincronizados con el joystick.
- Este sistema podría simplificarse aún más mediante el uso de un multiplexor, lo que permitiría reducir la cantidad de cables necesarios para controlar la matriz de LEDs. Al integrar un multiplexor, se podría minimizar el cableado sin perder precisión en la visualización, optimizando así la estructura del circuito.

# RESULTADOS



# CONCLUSION

En conclusión, este proyecto integra una matriz LED de 8x8, un joystick y una placa Arduino Uno para crear un sistema interactivo de visualización de figuras. A través de la programación, logramos que la matriz muestre diferentes patrones en respuesta al movimiento del joystick, incluyendo una figura específica en la zona de reposo. La implementación podría optimizarse aún más mediante el uso de un multiplexor, reduciendo así el número de conexiones necesarias y simplificando el cableado del circuito. Este prototipo demuestra cómo se pueden combinar componentes básicos de electrónica y programación para desarrollar sistemas dinámicos y personalizables, abriendo la puerta a aplicaciones más avanzadas en el futuro.

# **GRACIAS POR SU ATENCIÓN**

**LINK DEL VIDEO:**

**[HTTPS://YOUTUBE.COM/SHORTS/CPJGMF1ZJBY?  
FEATURE=SHARE](https://youtube.com/shorts/CPJGMF1zjBy?feature=share)**

