



## SmartElex Multiplexer Breakout – 8 Channel -74HC4051

### ❖ Introduction –

The SmartElex Multiplexer Breakout provides access to all pins and features of the 74HC4051, an 8-channel analog multiplexer/demultiplexer. The 74HC4051 allows you to turn four I/O pins into eight multifunctional, individually selectable signals, which can be used to do everything from driving eight LEDs to monitoring eight potentiometers.

The 74HC4051 can function as either a multiplexer or a demultiplexer, and it features eight channels of selectable inputs/outputs. The routing of common signal to independent I/O is set by digitally controlling three select lines, which can be set either high or low into one of eight binary combinations.

One half of the board breaks out the control signals (E, S0-S2) and common input/output (Z). The other side provides access to all eight independent I/O's (Y0-Y7). Both sides include supply and ground connections (VCC, VEE, GND).

### ❖ Features –

- 8-channel analog/digital multiplexer
- Uses only 3 select pins (S0, S1, S2)
- Enable pin to activate or deactivate switching
- Supports single-supply and bipolar operation
- Wide operating voltage range (2V–10V)
- Suitable for sensor multiplexing and signal routing

### ❖ Specifications -

- IC Used: 74HC4051
- Number of Channels: 8 (Single-pole, 8-throw)
- Signal Type: Analog / Digital
- Supply Voltage (VCC): 2.0V to 10.0V
- Enable Input: Active LOW
- Select Inputs: S0, S1, S2
- On Resistance: ~60–80 ohms (typical)
- Operating Temperature: -40°C to +85°C

## ❖ Required Components -

- Arduino Uno
- SmartElex 74HC4051 Multiplexer Breakout
- Breadboard
- Jumper wires
- 8 × Potentiometers or voltage sources (optional for testing)
- USB cable for Arduino

## ❖ Pin Connections -

Multiplexer Pin	Arduino Uno Pin
VCC	5V
GND	GND
VEE	GND
S0	Digital Pin 2
S1	Digital Pin 3
S2	Digital Pin 4
Z (Common)	Analog Pin A0
Y0–Y7	Analog sensors / Potentiometers

## ❖ Test Procedure -

- Connect the multiplexer to the Arduino Uno as per the pin connection table.
- Ensure VEE is connected to GND and the Enable pin (E) is LOW (enabled).
- Upload the provided Arduino sketch to the Arduino Uno.
- Open the Serial Monitor and set the baud rate to 9600.
- Rotate potentiometers or change input voltages on Y0–Y7.
- Observe the corresponding analog values printed in the Serial Monitor.

## ❖ Expected Result -

The Serial Monitor should display eight analog values corresponding to channels Y0 to Y7. Changing the voltage on any channel should update only its respective value.

## ❖ Sample Arduino Code -

```
const int selectPins[3] = {2, 3, 4}; // S0~2, S1~3, S2~4
const int zOutput = 5;
const int zInput = A0; // Connect common (Z) to A0 (analog input)
```

```
void setup()
{
  Serial.begin(9600); // Initialize the serial port
  // Set up the select pins as outputs:
  for (int i=0; i<3; i++)
  {
    pinMode(selectPins[i], OUTPUT);
    digitalWrite(selectPins[i], HIGH);
  }
  pinMode(zInput, INPUT); // Set up Z as an input

  // Print the header:
  Serial.println("Y0\tY1\tY2\tY3\tY4\tY5\tY6\tY7");
  Serial.println("---\t---\t---\t---\t---\t---\t---\t---");
}
```

```
void loop()
{
  // Loop through all eight pins.
  for (byte pin=0; pin<=7; pin++)
  {
    selectMuxPin(pin); // Select one at a time
    int inputValue = analogRead(zInput); // and read Z
    Serial.print(String(inputValue) + "\t");
  }
  Serial.println();
  delay(1000);
}
```

```
// The selectMuxPin function sets the S0, S1, and S2 pins
// accordingly, given a pin from 0-7.
```

```
void selectMuxPin(byte pin)
{
  for (int i=0; i<3; i++)
  {
    if (pin & (1<<i))
      digitalWrite(selectPins[i], HIGH);
    else
      digitalWrite(selectPins[i], LOW);
  }
}
```