

## Keyestudio Joystick Breakout Board for micro:bit



### Introduction

The BBC micro:bit is a powerful handheld, fully programmable, computer designed by the BBC. It was designed to encourage children to get actively involved in technical activities, like coding and electronics.

It features a 5x5 LED Matrix, two integrated push buttons, a compass, Accelerometer, and Bluetooth.

It supports the PXT graphical programming interface developed by Microsoft and can be used under Windows, MacOS, IOS, Android and many other operating systems without downloading the compiler.

Looking to do more with your BBC micro:bit? Unlock its potential with this Joystick breakout board for the BBC micro:bit!

Keyestudio joystick breakout board for micro:bit comes with AMS1117 chip. You can connect the external DC4.75-12V to power for micro:bit

<https://eckstein-shop.de/Keyestudio>

development board.

It can be used to simulate the mouse or keyboard. Connect the on-board pins 3V、G、P0、P1 to micro:bit main board to get analog serial port, connecting the Bluetooth devices.

This breakout board also consists of 6 buttons and a joystick. Connect the button and joystick to micro:bit pins.

Connect SELECT button to P7; MODE button to P6; UP button to P10; LEFT button to P11; RIGHT button to P9; DOWN button to P8.

Connect joystick X axis to P4; Y axis to P3; Z axis to P5. You can control the external devices by reading the interface states.

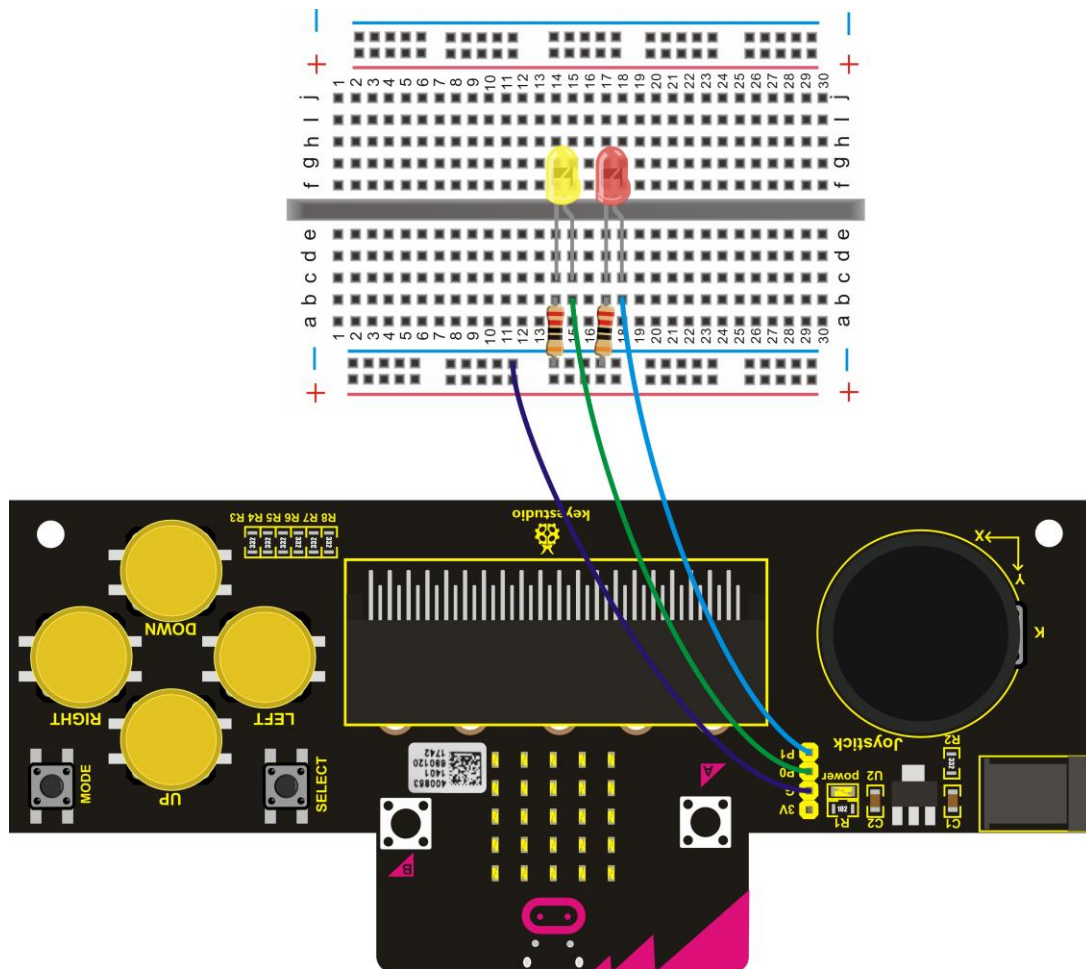
## Parameters

- Input Voltage: DC 4.75-12V



# Eckstein Keyestudio

## Wiring Diagram



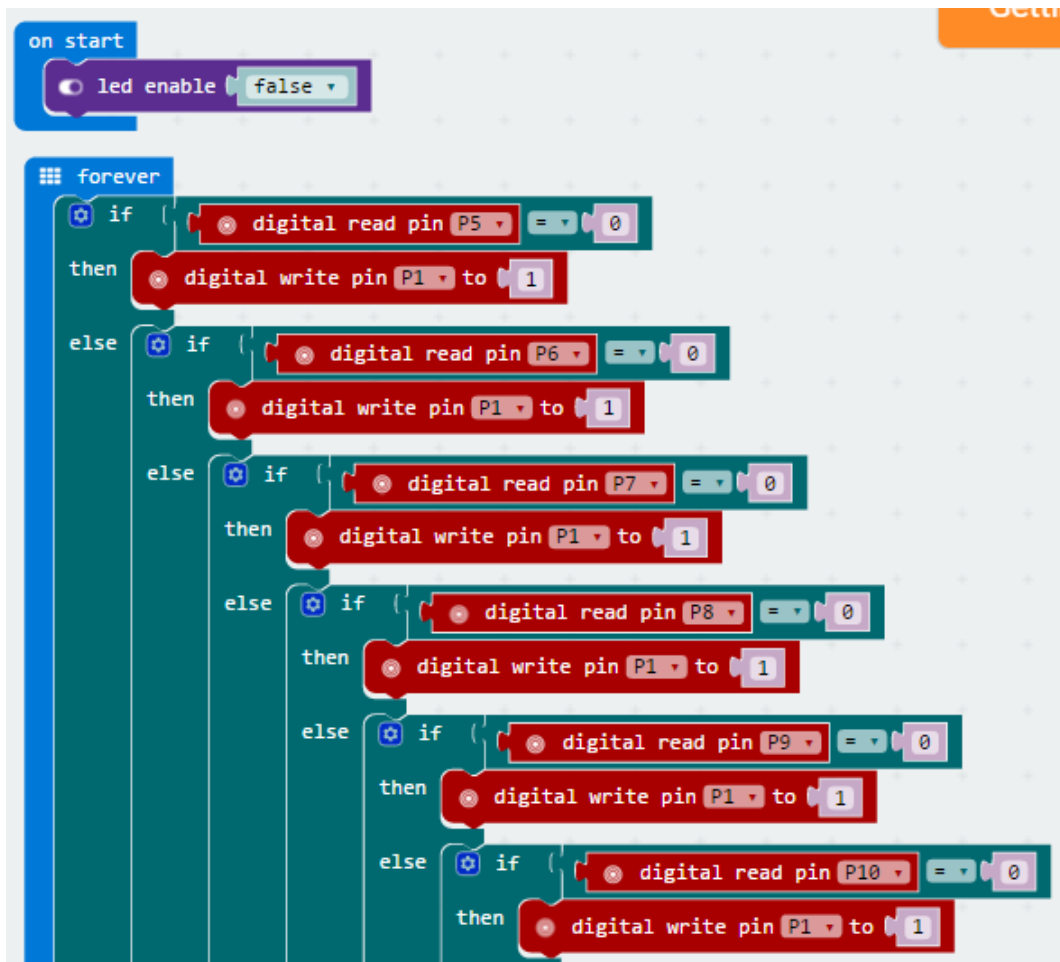
## Test Program

You can write the code below on micro:bit MakeCode editor

<https://makecode.microbit.org/#editor>

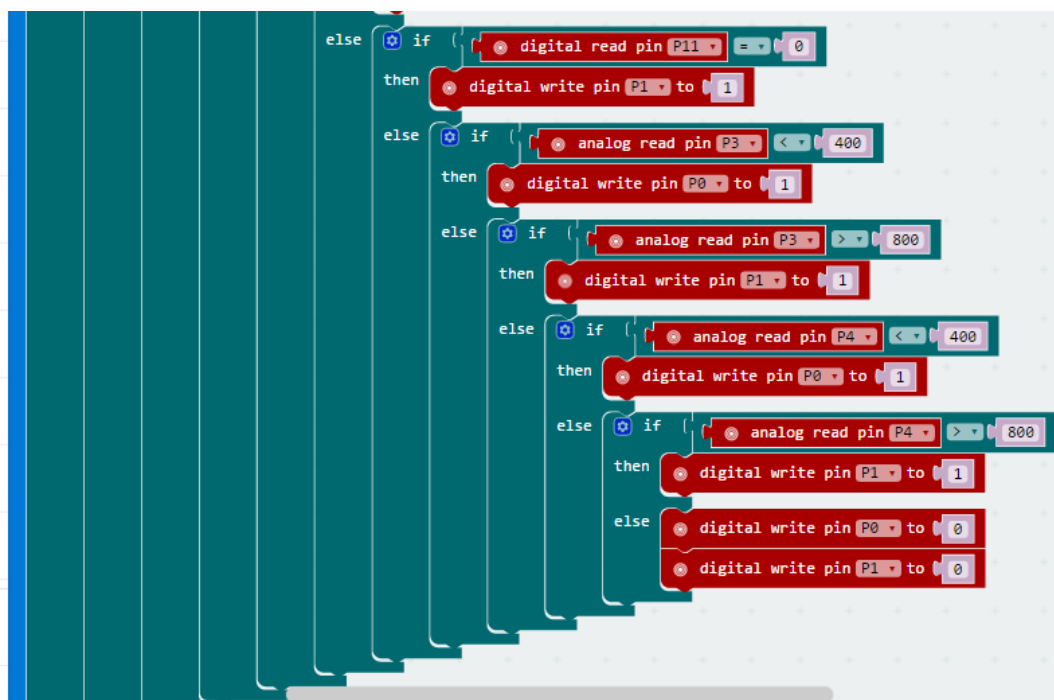
Downloaded the code you wrote well, send it to your micro:bit main board.

<https://eckstein-shop.de/Keyestudio>



```
on start
  led enable false

forever
  if (digital read pin P5 == 0)
  then
    digital write pin P1 to 1
  else
    if (digital read pin P6 == 0)
    then
      digital write pin P1 to 1
    else
      if (digital read pin P7 == 0)
      then
        digital write pin P1 to 1
      else
        if (digital read pin P8 == 0)
        then
          digital write pin P1 to 1
        else
          if (digital read pin P9 == 0)
          then
            digital write pin P1 to 1
          else
            if (digital read pin P10 == 0)
            then
              digital write pin P1 to 1
```

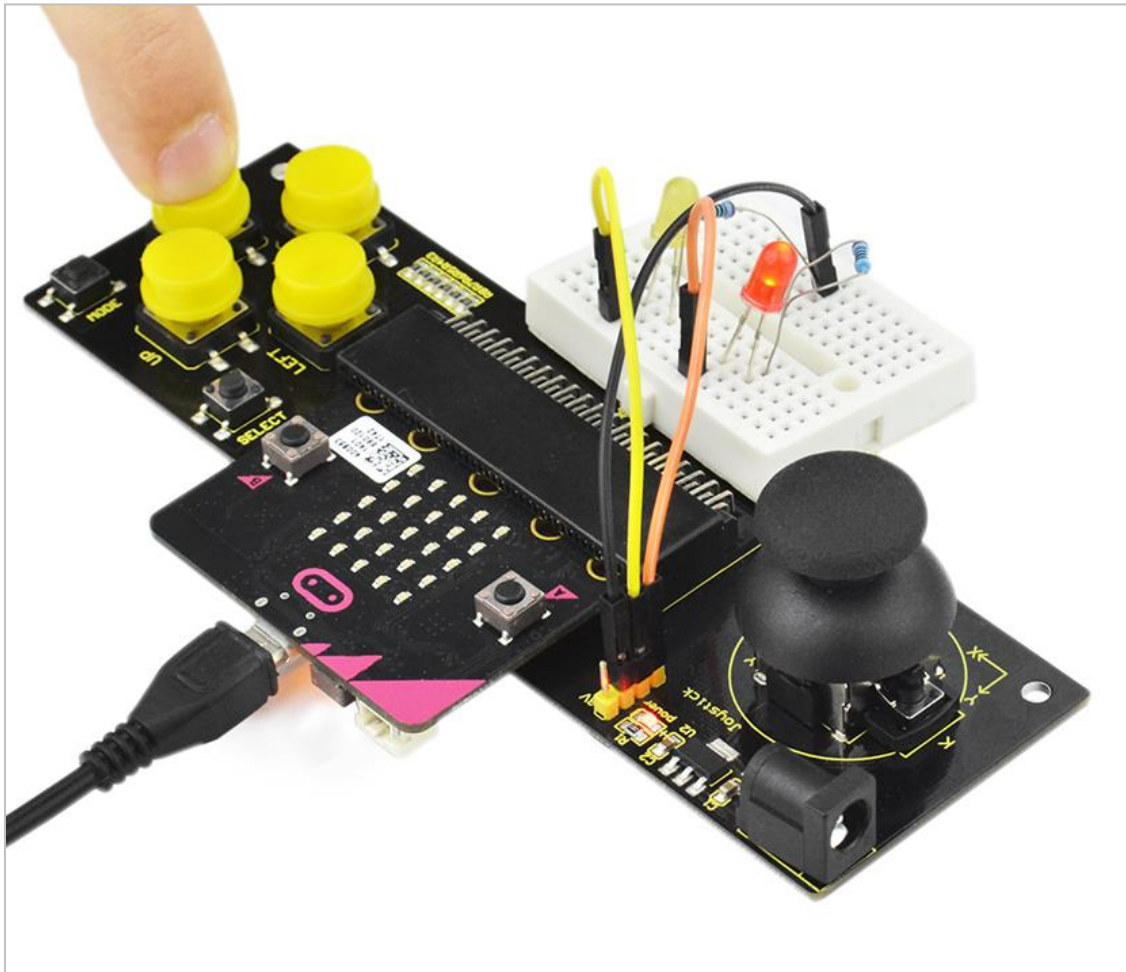


```
else
  if (digital read pin P11 == 0)
  then
    digital write pin P1 to 1
  else
    if (analog read pin P3 < 400)
    then
      digital write pin P0 to 1
    else
      if (analog read pin P3 > 800)
      then
        digital write pin P1 to 1
      else
        if (analog read pin P4 < 400)
        then
          digital write pin P0 to 1
        else
          if (analog read pin P4 > 800)
          then
            digital write pin P1 to 1
          else
            digital write pin P0 to 0
            digital write pin P1 to 0
```

## Test Result

Insert the micro:bit development board into keystudio joystick breakout board, powering up, 2 LEDs will display as below:

- 1) Do not operate both joystick and buttons, 2 LEDs are off.
- 2) Press down any buttons ( including joystick Z axis button), red LED is on; release the button, LED will be off.
- 3) Turn Joystick upward, red LED on; joystick downward, yellow LED on; joystick leftward, yellow LED on; joystick rightward, red LED on.



## Shield Experiment

In the experiment, we can insert a micro:bit control board on the shield, upload the corresponding test code, and set it as the Bluetooth host. We take a micro:bit control board, connect to the external device, upload the corresponding test code, set it as a Bluetooth slave. When both micro:bit control boards are powered up, the Bluetooth of two micro:bit control boards connect to the communication. We can control the external devices (such as LED buzzers, etc.) connected to the Bluetooth slave via external devices connected to the Bluetooth host (buttons on shield, joystick).

Here, we use this shield connected to a micro:bit control board to control the movement of a micro:bit smart car (KS0426). When using it, we only use the UP, DOWN, LEFT, RIGHT and SELECT buttons on the shield to control the smart car forward moving, backward moving, left turning, right turning and stopping. The dot matrix of the control board on the smart car displays the current motion status pattern.



## **KS0426 Smart Car Resources:**

[https://wiki.keystudio.com/Ks0426\\_keyestudio\\_Micro:bit\\_Mini\\_Smart\\_Robot\\_Car\\_Kit\\_V2](https://wiki.keystudio.com/Ks0426_keyestudio_Micro:bit_Mini_Smart_Robot_Car_Kit_V2)

## **Test Code Of Micro:bit Control Board Connected The Shield:**

```
on start
  radio set group 99
  radio set transmit power 5
  led enable false

forever
  radio send value "X" = analog read pin P4
  radio send value "Y" = analog read pin P3
  radio send value "Z" = digital read pin P5
  radio send value "U" = digital read pin P10
  radio send value "D" = digital read pin P8
  radio send value "L" = digital read pin P11
  radio send value "R" = digital read pin P9
  radio send value "S" = digital read pin P7
  radio send value "M" = digital read pin P6
```

**Test Code Of Micro: bit Control Board On KS0426 Smart Car:**



The image displays a collection of code blocks for a microcontroller project, organized into two columns. The left column contains the main event-driven logic, while the right column contains sub-routines for different directions.

**Main Logic (Left Column):**

- on radio received name value:** This block initiates the program based on received radio signals.
- if name == 'U' then:** Triggers the 'front' function when the letter 'U' is received.
- if U1 == 0 then:** Checks if the 'U1' variable is zero before calling the 'front' function.
- call front:** A function call that triggers the 'show leds' block.
- show leds:** A block that displays a 4x4 grid of LEDs in a specific pattern.
- else if name == 'D' then:** Triggers the 'back' function when the letter 'D' is received.
- if D1 == 0 then:** Checks if the 'D1' variable is zero before calling the 'back' function.
- call back:** A function call that triggers the 'show leds' block.
- else if name == 'L' then:** Triggers the 'left' function when the letter 'L' is received.
- if L1 == 0 then:** Checks if the 'L1' variable is zero before calling the 'left' function.
- call left:** A function call that triggers the 'show leds' block.
- else if name == 'R' then:** Triggers the 'right' function when the letter 'R' is received.
- if R1 == 0 then:** Checks if the 'R1' variable is zero before calling the 'right' function.
- call right:** A function call that triggers the 'show leds' block.
- else if name == 'S' then:** Triggers the 'stop' function when the letter 'S' is received.
- if S1 == 0 then:** Checks if the 'S1' variable is zero before calling the 'stop' function.
- call stop:** A function call that triggers the 'show leds' block.

**Sub-routines (Right Column):**

- on start:** Initial setup including 'radio set group 99', 'led enable true', 'clear screen', and 'init 67 0'.
- function front:** Sets LED duty cycles: LED1 (8), LED2 (100), LED3 (8), LED4 (100).
- function back:** Sets LED duty cycles: LED1 (100), LED2 (100), LED3 (100), LED4 (100).
- function stop:** Sets LED duty cycles: LED1 (8), LED2 (8), LED3 (8), LED4 (8).
- function left:** Sets LED duty cycles: LED1 (100), LED2 (50), LED3 (8), LED4 (50).
- function right:** Sets LED duty cycles: LED1 (8), LED2 (50), LED3 (100), LED4 (50).