

Joystick

```
1 import time
2 from adafruit_crickit import crickit
3
4 # Setup seesaw
5 ss = crickit.seesaw
6 x_axis = crickit.SIGNAL1 # Joystick X-axis
7 y_axis = crickit.SIGNAL2 # Joystick Y-axis
8 sw = crickit.SIGNAL3     # Joystick button (switch)
9
10 ss.pin_mode(x_axis, ss.INPUT)
11 ss.pin_mode(y_axis, ss.INPUT)
12 ss.pin_mode(sw, ss.INPUT_PULLUP)
13
14 print("Running...")
15
16 # Motors
17 motor1 = crickit.dc_motor_1
18 motor2 = crickit.dc_motor_2
19
20 # Threshold values
21 threshold1 = 100
22 threshold2 = 900
23
24 # ---- Functions for movement ----
25 def move_forward():
26     motor1.throttle = 0.5
27     motor2.throttle = 0.5
28     print("Forward")
29
```

```
def move_backward():
    motor1.throttle = -0.5
    motor2.throttle = -0.5
    print("Backward")

def turn_right():
    motor1.throttle = 0.5
    motor2.throttle = 0.3
    print("Right")

def turn_left():
    motor1.throttle = 0.3
    motor2.throttle = 0.5
    print("Left")

def stop():
    motor1.throttle = 0
    motor2.throttle = 0
    print("Stopped")

# ---- Main loop ----
while True:
    x = ss.analog_read(x_axis)
    y = ss.analog_read(y_axis)
    z = ss.digital_read(sw)

    print("x:", x, " y:", y, " z:", z)
    time.sleep(0.1)
```

```

print("x:", x, " y:", y, " z:", z)
time.sleep(0.1)

if y < threshold1:
    move_forward()
elif y > threshold2:
    move_backward()
elif x < threshold1:
    turn_right()
elif x > threshold2:
    turn_left()
elif not z: # Button pressed
    stop()

```

Code Explanation:-

import time

from adafruit_crickit import crickit

- time → used for delays.
- crickit → to control motors and read inputs from sensors/joystick.

ss = crickit.seesaw

x_axis = crickit.SIGNAL1

y_axis = crickit.SIGNAL2

sw = crickit.SIGNAL3

- Joystick has 3 outputs:
- x_axis → left/right movement.
- y_axis → forward/backward movement.
- sw → joystick button (pressed = stop).

ss.pin_mode(x_axis, ss.INPUT)

ss.pin_mode(y_axis, ss.INPUT)

ss.pin_mode(sw, ss.INPUT_PULLUP)

Defines each pin's mode.

- For button (sw) → we use INPUT_PULLUP (default HIGH, goes LOW when pressed).

Defines each pin's mode.

- For button (sw) → we use INPUT_PULLUP (default HIGH, goes LOW when pressed).

print("Running...")

- Just confirms program started.

motor1 = crickit.dc_motor_1

motor2 = crickit.dc_motor_2

- Defines two DC motors connected to Crickit motor ports

```
threshold1 = 100
```

```
threshold2 = 900
```

- Joystick analog values range from 0 to 1023.
- Around 500 = center (neutral).
- threshold1 (100) → detects when stick is pushed strongly in one direction.
- threshold2 (900) → detects when stick is pushed strongly in the opposite direction.

```
def move_forward():
```

```
    motor1.throttle = 0.5
```

```
    motor2.throttle = 0.5
```

```
    print("Forward")
```

- Both motors forward → rover moves forward.

```
def move_backward():
```

```
    motor1.throttle = -0.5
```

```
    motor2.throttle = -0.5
```

```
    print("Backward")
```

- Both motors backward → rover moves backward.

```
def turn_right():
```

```
    motor1.throttle = 0.5
```

```
    motor2.throttle = 0.3
```

```
    print("Right")
```

- Left motor runs faster than right motor → rover turns right.

```
def turn_left():
```

```
    motor1.throttle = 0.3
```

```
    motor2.throttle = 0.5
```

```
    print("Left")
```

- Right motor runs faster than left motor → rover turns left.

```
def stop():
```

```
    motor1.throttle = 0
```

```
    motor2.throttle = 0
```

```
    print("Stopped")
```

- Stops both motors → rover halts.

```
while True:
```

```
    x = ss.analog_read(x_axis)
```

```
    y = ss.analog_read(y_axis)
```

```
    z = ss.digital_read(sw)
```

- Continuously read joystick x, y values and button state.

```
print("x:", x, " y:", y, " z:", z)
```

```
time.sleep(0.1)
```

- Displays joystick values for debugging.
- Small delay (0.1 sec) to avoid spamming.

```
if y < threshold1:
```

```
    move_forward()
```

```
elif y > threshold2:
```

```
    move_backward()
```

```
elif x < threshold1:
```

```
    turn_right()
```

```
elif x > threshold2:
```

```
    turn_left()
```

```
elif not z: # Button pressed
```

```
    stop()
```

- y small (<100) → move forward.
- y big (>900) → move backward.
- x small (<100) → turn right.
- x big (>900) → turn left.
- button pressed (z=0) → stop.