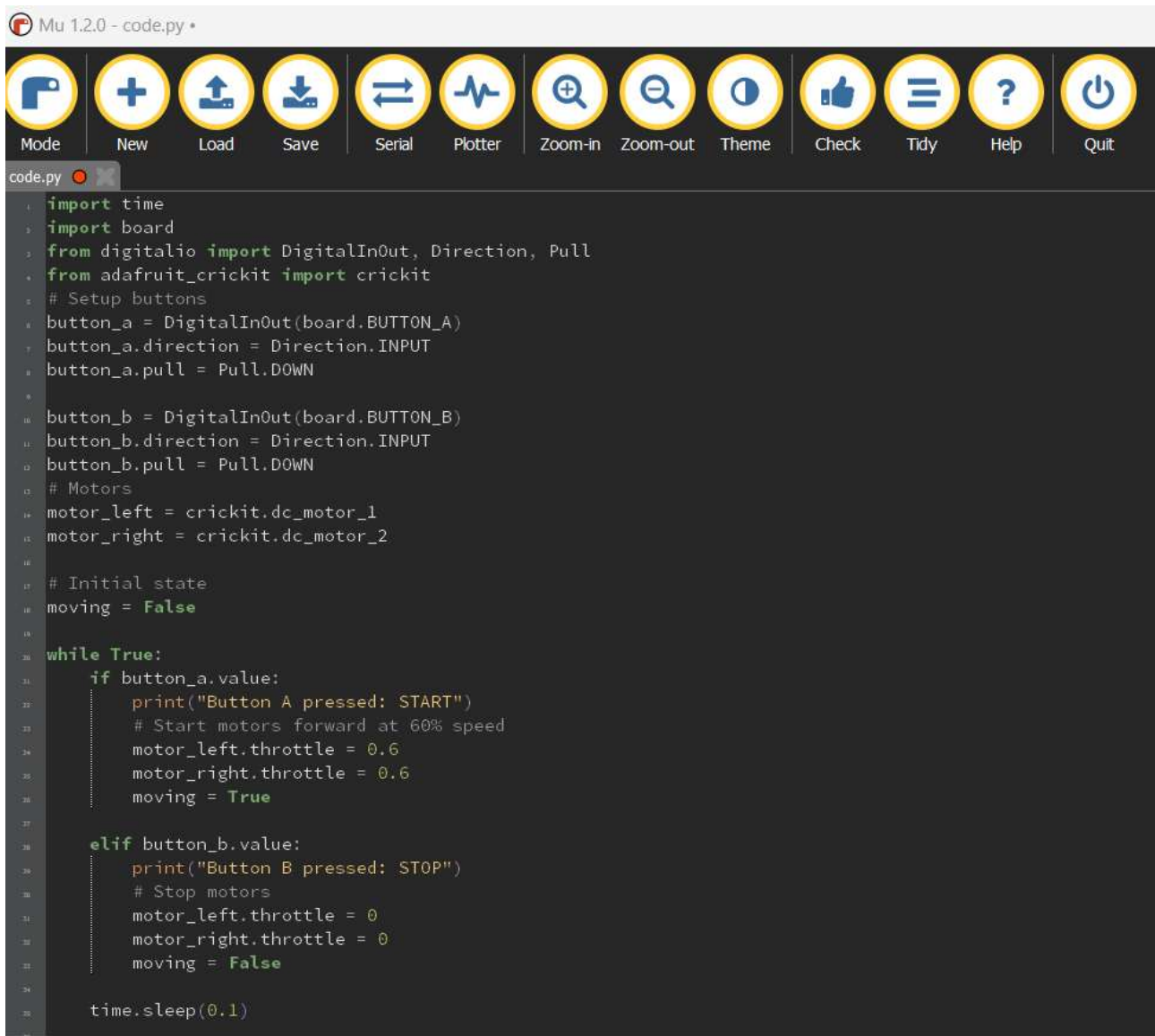


Start-Stop Rover

When you press button A the Rover will move and when you press button B the Rover will stop.



```
1 import time
2 import board
3 from digitalio import DigitalInOut, Direction, Pull
4 from adafruit_crickit import crickit
5 # Setup buttons
6 button_a = DigitalInOut(board.BUTTON_A)
7 button_a.direction = Direction.INPUT
8 button_a.pull = Pull.DOWN
9
10 button_b = DigitalInOut(board.BUTTON_B)
11 button_b.direction = Direction.INPUT
12 button_b.pull = Pull.DOWN
13 # Motors
14 motor_left = crickit.dc_motor_1
15 motor_right = crickit.dc_motor_2
16
17 # Initial state
18 moving = False
19
20 while True:
21     if button_a.value:
22         print("Button A pressed: START")
23         # Start motors forward at 60% speed
24         motor_left.throttle = 0.6
25         motor_right.throttle = 0.6
26         moving = True
27
28     elif button_b.value:
29         print("Button B pressed: STOP")
30         # Stop motors
31         motor_left.throttle = 0
32         motor_right.throttle = 0
33         moving = False
34
35     time.sleep(0.1)
```

Code explanation

- import time-gives access to time.sleep()
- import board-provides names for the board's pins and buttons (e.g. board.BUTTON_A, board.BUTTON_B).
- from digitalio import DigitalInOut, Direction, Pull
-imports classes to configure and read digital input pins (buttons):

- DigitalInOut creates a pin object,
- Direction sets input/output,
- Pull sets internal pull-up or pull-down resistors.
-
- from adafruit_crickit import crickit- imports the Crickit helper object that gives access to motors, servos, and sensors on the Crickit board.

Setup buttons-comment describing that the next lines configure the two hardware buttons.

- button_a = DigitalInOut(board.BUTTON_A)-creates a DigitalInOut object for Button A on the board.
- button_a.direction = Direction.INPUT-configures Button A pin as an **input** (we will read it).
- button_a.pull = Pull.DOWN- enables an internal **pull-down** resistor so the pin reads False when the button is not pressed and True when pressed.
- button_b = DigitalInOut(board.BUTTON_B)-creates a DigitalInOut object for Button B.
- button_b.direction = Direction.INPUT- sets Button B as an input.
- button_b.pull = Pull.DOWN-sets an internal pull-down for Button B (same behaviour as Button A).
- # Motors- comment: next lines select the DC motors attached to Crickit.
- motor_left = crickit.dc_motor_1-assigns the first Crickit DC motor (left motor) to the variable motor_left.
- motor_right = crickit.dc_motor_2-assigns the second Crickit DC motor (right motor) to motor_right.
- # Initial state- comment for the state variable.
- moving = False-boolean variable to remember whether the robot/motors are currently running.
- while True- starts an infinite loop so the program continuously checks the buttons and updates motors.
- if button_a.value - checks if **Button A** is pressed (.value is True when pressed because of Pull.DOWN setup).

- `print("Button A pressed: START")`- prints a message to the console for debugging/feedback.
- `# Start motors forward at 60% speed`- comment describing the next actions.
- `motor_left.throttle = 0.6`- sets the left motor throttle (speed) to **0.6** (60% forward). throttle ranges from -1.0 to 1.0 (negative = reverse, positive = forward).
- `motor_right.throttle = 0.6`- sets the right motor throttle to **0.6** as well.
- `moving = True`-updates the state variable to indicate motors are running.
- `elif button_b.value`-otherwise, if **Button B** is pressed, execute the stop branch.
- `print("Button B pressed: STOP")`- prints a stop message.
- `# Stop motors`- comment describing the next actions.
- `motor_left.throttle = 0`-sets left motor speed to **0** (stop).
- `motor_right.throttle = 0`- sets right motor speed to **0** (stop).
- `moving = False`- updates state to indicate motors are stopped.
- `time.sleep(0.1)`- small delay (100 ms) at the end of each loop iteration to:
 - reduce CPU usage,
 - debounce the buttons a little,
 - avoid a very tight polling loop that floods the serial output.