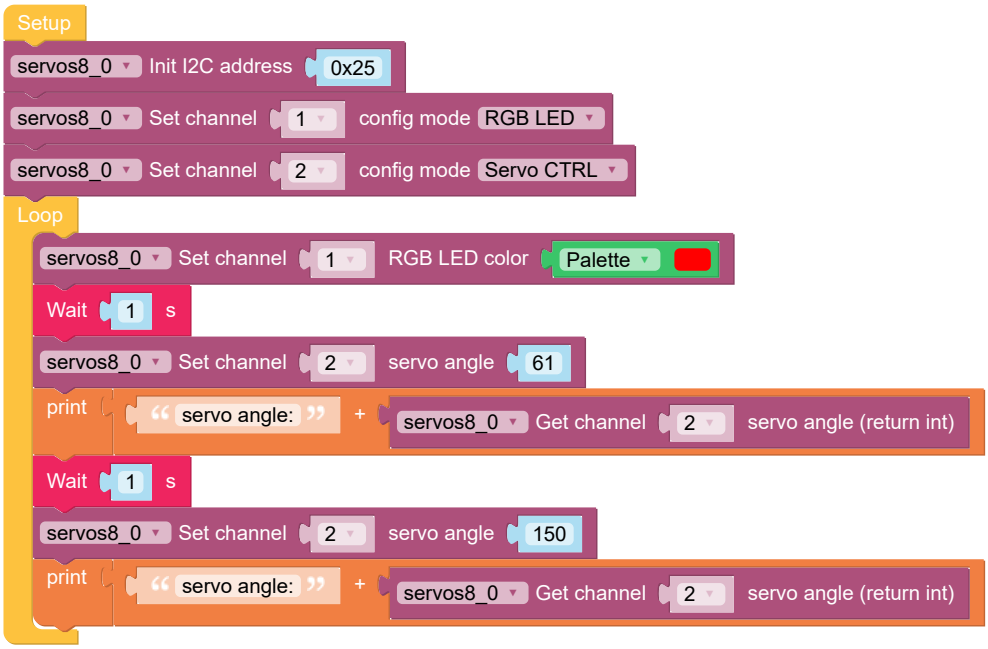
Unit 8Servos

Example

> Set the channel 1 Set the connected RGB LED to red and set the connected Servo to adjust the rotation Angle every second

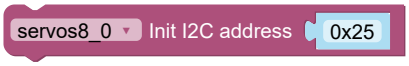



```
from m5stack import *
from m5ui import *
from uiflow import *
import time
import unit

setScreenColor(0x222222)
servos8_0 = unit.get(unit.SERVOS8, unit.PORTA)

servos8_0.init_i2c_address(0x25)
servos8_0.set_config_mode(1, 4)
servos8_0.set_config_mode(2, 3)
while True:
  servos8_0.write_rgb_led(1, 0xff0000)
  wait(1)
  servos8_0.write_servo_angle(2, 61)
  print((str('servo angle:') + str((servos8_0.read_servo_angle(2)))))
  wait(1)
  servos8_0.write_servo_angle(2, 150)
  print((str('servo angle:') + str((servos8_0.read_servo_angle(2)))))
  wait_ms(2)
```

API




```
servos8_0.init_i2c_address(0x25)
```

- Initialize I2C Address


```
print((str('config mode:') + str((servos8_0.get_config_mode(0)))))
```

- Get Configuration Mode of Specified Channel


```
print((str('ADC 12bit:') + str((servos8_0.read_adc12_pin(0)))))
```

- Get 12-bit ADC (Analog-to-Digital Conversion) Value of Specified Channel


```
print((str('ADC 8bit:') + str((servos8_0.read_adc8_pin(0)))))
```

- Get 8-bit ADC (Analog-to-Digital Conversion) Value of Specified Channel


```
print((str('rgb led color:') + str((servos8_0.read_input_pin(0)))))
```

- Get Digital Input Status of Specified Channel (Returns true or false)


```
print((str('rgb led color:') + str((servos8_0.read_rgb_led(0)))))
```

- Get RGB LED Color of Specified Channel (Returns color list)


```
print((str('servo angle:') + str((servos8_0.read_servo_angle(0)))))
```

- Get Rotation Angle of Specified Channel


```
print((str('servo current:') + str((servos8_0.read_servo_current()))))
```

- Get Servo Current (Returns float)


```
print((str('servo pulse:') + str((servos8_0.read_servo_pulse(0)))))
```

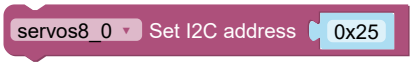
- Get Servo Pulse Width of Specified Channel (Returns integer)


```
print((str('firmware version:') + str((servos8_0.read_status(0XFE)))))
```

- Get Firmware Version

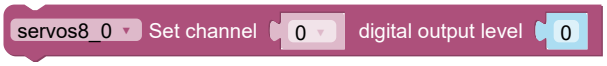

```
servos8_0.set_config_mode(0, 0)
```

- Set Configuration Mode of Specified Channel
  - Digital Input: Digital Input Mode
  - Digital Output: Digital Output Mode
  - ADC Input: Analog Input Mode
  - Servo CTRL: Servo Control Mode
  - RGB LED: RGB LED Control Mode
  - PWM Duty: PWM Duty Cycle Control Mode



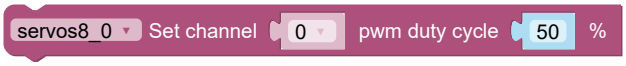

```
servos8_0.set_i2c_address(0x0x25)
```

- Set I2C Address




```
servos8_0.write_output_pin(0, 0)
```

- Set Digital Output Level of Specified Channel




```
servos8_0.write_pwm_duty(0, 50)
```

- Set PWM Duty Cycle of Specified Channel


```
servos8_0.write_rgb_led(0, 0xff0000)
```

- Set RGB LED Color of Specified Channel


```
servos8_0.write_servo_angle(0, 0)
```

- Set Rotation Angle of Specified Channel


```
servos8_0.write_servo_pulse(0, 500)
```

- Set Servo Pulse Width of Specified Channel