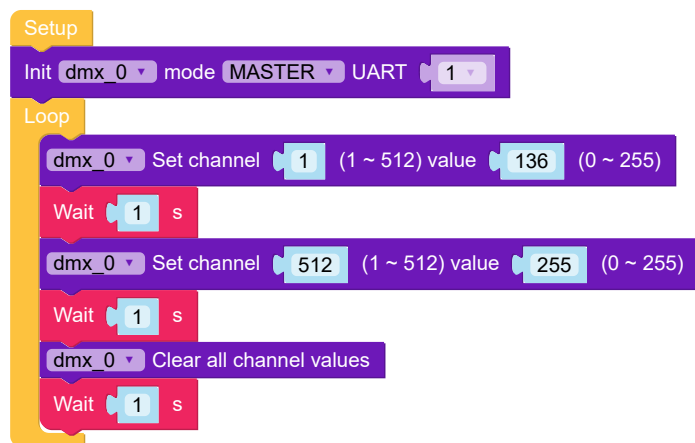


Unit DMX

Example

Control multiple channels (up to 512 channels) of a device through the DMX protocol, sending specified values (e.g., channel 0 to 255 represent RGB values of light colors) to adjust the device's state. This is commonly used for stage lighting control or other digital equipment adjustment.



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

setScreenColor(0x222222)
dmx_0 = unit.get(unit.DMX, unit.PORTA)

dmx_0.init_dmx(1, 1)
while True:
    dmx_0.write_dmx_value(1, 136)
    wait(1)
    dmx_0.write_dmx_value(512, 255)
    wait(1)
    dmx_0.clear_dmx_buffer()
    wait(1)
    wait_ms(2)
```

API

Init **dmx_0** mode **MASTER** UART **1**

```
dmx_0.init_dmx(1, 1)
```

- Initialize the DMX module **dmx_0**, set it to MASTER mode, and specify the use of UART 1 for communication.

dmx_0 Deinit

```
dmx_0.deinit()
```

- Deinitialize the DMX module **dmx_0**, stopping the module's operation and releasing the associated resources.

dmx_0 Set channel **1** (1 ~ 512) value **0** (0 ~ 255)

```
dmx_0.write_dmx_value(1, 0)
```

- Set the output value of DMX channel 1 to 0, with a value range of 0 to 255.

dmx_0 Clear all channel values

```
dmx_0.clear_dmx_buffer()
```

- Clear all DMX channel values and reset them to default values (typically 0).

dmx_0 Get channel **1** (1 ~ 512) value (return int)

```
dmx_0.read_dmx_value(1)
```

- Retrieve the current value of DMX channel 1 and display the value through **print** (returns the value as an integer).