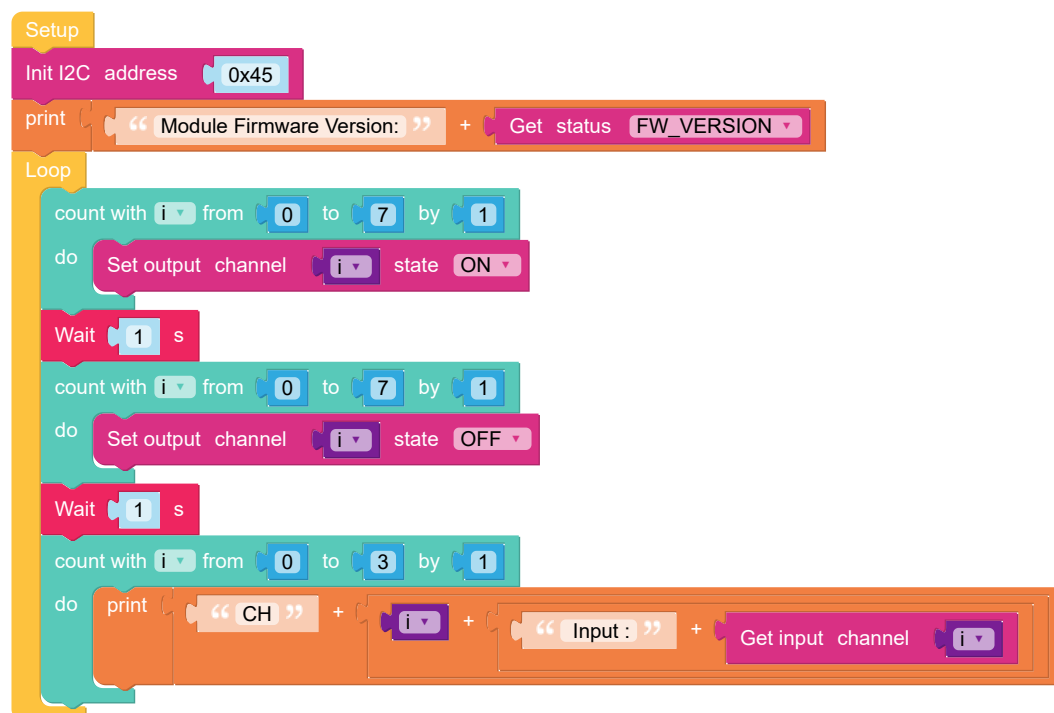


Module13.2 4In8Out

| Example

Print out the firmware version, control the output channel on and off, and read the input channel status



```
from m5stack import *
from m5ui import *
from uiflow import *
import module

import time

setScreenColor(0x222222)
i = None
module_4in8out = module.get(module.MODULE_4IN8OUT)

module_4in8out.init_i2c_address(0x45)
print((str('Module Firmware Version:') + str((module_4in8out.read_status(0xFE)))))
while True:
    for i in range(8):
        module_4in8out.write_output_pin(i, 1)
    wait(1)
    for i in range(8):
        module_4in8out.write_output_pin(i, 0)
    wait(1)
    for i in range(4):
        print((str('CH') + str(((str(i) + str(((str(' Input: ') + str((module_4in8out.read_input_pin(i)))))
    wait_ms(2)
```

API

Init I2C address  69

```
module_4in8out.init_i2c_address(0x45)
```

- Initializes the I2C address of the device.

Get input channel  0

```
module_4in8out.read_input_pin(0)
```

- Retrieves the status or value of the input channel.

Get status **FW_VERSION** ▼

```
module_4in8out.read_status(0xFE)
```

- Retrieves the status information of the device, selecting to get the firmware version (FW_VERSION).

Set I2C address  69

```
module_4in8out.set_i2c_address(0x45)
```

- Sets the I2C address of the device.

Set output channel  0 state **ON** ▼

```
module_4in8out.write_output_pin(0, 1)
```

- Controls the status of the output channel. This example sets the output status of channel 0, with options to set it to "ON" or "OFF." This is used to control devices or circuits connected to the channel, such as turning a relay or LED on or off.