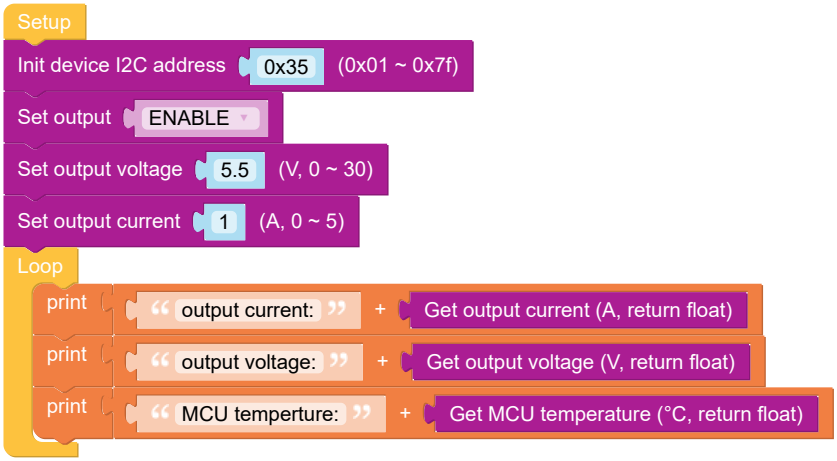
# Module13.2 PPS

## Example

Continuously read and print the output current, output voltage and MCU temperature (via serial port)




```
from m5stack import *
from m5stack_ui import *
from uiflow import *
import module

screen = M5Screen()
screen.clean_screen()
screen.set_screen_bg_color(0xFFFFFF)

pps = module.get(module.PPS)

pps.init_i2c_address(0x35)
pps.setOutput(True)
pps.setOutputVoltage(5.5)
pps.setOutputCurrent(1)
while True:
  print((str('output current:') + str((pps.readOutputCurrent()))))
  print((str('output voltage:') + str((pps.readOutputVoltage()))))
  print((str('MCU temperature:') + str((pps.readMcuTemperature()))))
  wait_ms(2)
```

## API

Get current I2C address (return int)

```
pps.getI2CAddress()
```

- Get the current device's I2C address


```
pps.init_i2c_address(0x35)
```

- Initialize the device's I2C address


```
pps.readDataUpdateFlag()
```

- Get the data update flag, returns an integer. This flag is usually used to indicate whether the data has been updated


```
pps.readInputVoltage()
```

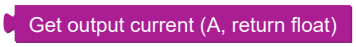
- Get the input voltage value in volts (V), returns a float. This value represents the currently measured input voltage


```
pps.readMcuTemperature()
```

- Get the MCU temperature value in degrees Celsius (°C), returns a float. This value represents the currently measured microcontroller temperature


```
pps.readModuleId()
```

- Get the module ID, returns an integer. This ID is used to uniquely identify the device or module

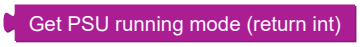



```
pps.readOutputCurrent()
```

- Get the output current value in amperes (A), returns a float. This value represents the currently measured output current


```
pps.readOutputVoltage()
```

- Get the output voltage value in volts (V), returns a float. This value represents the currently measured output voltage




```
pps.readPsuRunningMode()
```

- Get the power module's running mode, returns an integer. This is usually used to indicate the current operating status of the power module


```
pps.readUID()
```

- Get the unique identifier (UID), returns a byte array. The UID is used to uniquely identify the device or module


```
pps.setI2CAddress()
```

- Set the device's I2C address


```
pps.setOutput(True)
```

- Set the output state. This function is usually used to control the device's output, such as turning a feature or module on or off


```
pps.setOutputCurrent(1)
```

- Set the output current, range is 0 to 5 amperes. This is used to control the device's output current


```
pps.setOutputVoltage(5.5)
```

- Set the device's output voltage. The range is 0 to 30 volts, used to adjust the device's output voltage value, controlling power supply or voltage level