

UIFlow Reference V1.4.0-Beta.

Written By
Adam Bryant

Events	2
Set-up	3
Loop	4
Button Loop	5
Obtain Button Value,	6
Button A+B Press Loop	7
Button Value	7
Timer Callback Loop.	7
Set timer Period	7
Start Timer	7
Stop Timer	7
Virtual User interface Designer and Graphic Functions.	8
The Show/Hide Blocks.	9
The Show Text Blocks	9
Set Screen Rotate Mode	9
Set Color	10
Set Screen Brightness	10
Set Colour R G B.	11
Background colour	11
Set Width and Height	12
Set X and Y	12
Set Circle Radius.	12
Internal Hardware	13
Speaker,	14
Speaker.Beep,	14
Speaker Volume,	14
Play Tone.	14
RGB,	15
Set RGB Bar Colour,	15
Set RGB Bar Colour R,G,B,	15
Set (Side) RGB Bar Colour,	15
Set (Side) RGB Bar Colour R,G,B,	15
Set (individual) RGB Bar Colour,	15
Set (individual) RGB Bar Colour R,G,B,	15
Set RGB Brightness,	16
IMU,	17

Get X,	17
Get Y,	17
Get X ACC,	17
Get Y ACC,	17
Get Z ACC,	17
Get X Gyro,	17
Get Y Gyr,	17
Get Z Gyr,	18
Power	19
Is Charging,	19
Is ChargedFull,	19
Set Charging,	19
Get Battery Level,	19
Real Time Clock	20
Time and Date configuration block.	20
Get Year,	20
Get Month,	20
Get Day,	20
Get Hour,	20
Get Minute,	21
Get Second,	21
Get Local Time.	21
LED	22
LED On	22
LED Off	22
Hats	24
ENV Hat	25
Get Pressure,	25
Get Temperature,	25
Get Humidity,	25
P.I.R Hat,	26
Get hat_pir Status,	26
Units,	31
Environmental,	32
TOF,	52
Get Distance,	52

EXT I/O,	53
RFID,	54
PAHub	55
Set position state	55
Set position	55
Set port value	55
Modules,	56
I2C,	100
Set I2C Port,	100
Set I2c SDA, SCL,	100
I2C Scan,	100
I2C Available Address in List,	100
I2C Read Reg One Byte,	100
I2C Read Reg One Short,	100
I2C Read Reg One Byte	101
I2C Write Reg One Byte,	101
I2C Read Byte,	101
I2C Write Reg One Short,	101
Network,	103
ESP NOW	104
Get MAC Address	104
Add Peer	104
Set PMK	104
Broadcast Data	104
Receive MAC Address Data	104
After Send Message Flag	104
Send Message ID Data	104
Custom Blocks.	108

Introduction

UIFlow is the Integrated Development Environment (**IDE**) created by M5Stack and is the third incarnation of the programming environment.

In various documents references are made to M5Cloud and Moments, these are the two previous versions and are no longer available.

UIFlow contains two environments. Blocky, the default environment is a basic programming aimed at the young and beginners to programming. Micropython, the second environment is a text based environment that is at the core of all the firmware and functions of blocky. Once a programmer has grown beyond blocky, they can use the Micropython environment to dig deeper in to the functions and develop new blocks for Blocky or program the M5Stacks and M5Sticks at a lower level.

Blocks in UIFlow can be separated into three groups, actions, loops and values.

Actions are the blocks that contain command to preform tasks.

Loops are used for code that needs to be repeated.

Values provide numbers that can vary or be defined but the programmer.

I hope that this section will explain what the various blocks available in the Uiflow environment. In another section I will show examples on how to use the various blocks to make things work.

Events

Set-up

Setup

The set-up block is required by all programs as it works in the background to import the library required for our programs to run. This block is the equivalent of Arduino's Void Setup function and contains functions that you only want to run once at the beginning of the program.

In the Blockly windows we can't see much but if we switch to the </> python window we see the following micro python code.

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

setScreenColor(0x222222)
```

We can see in this code snippet that this block imports the three core libraries required by all programs ad set the initial screen colour to an almost black colour. By changing the value highlighted in red, we can change the initial screen colour. For more information on setting colours, check out the colour blocks.

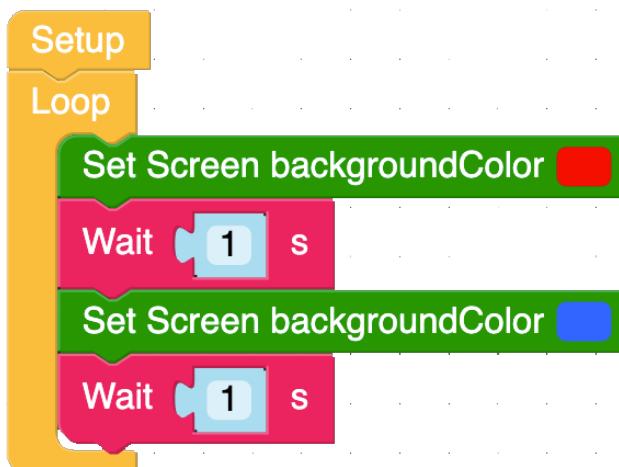
Loop



After this we have the main program loop. Inside of this block we place the main program that we want to continuously repeat.

Example

In the following example I am using a loop to continuously step through the SetscreenColour blocks. If this is run without the wait blocks then the screen will look purple because the screen is changing faster than the human eye can see. To be able to see the colour change I have added a wait block to make the program wait one second before moving between the blocks.



```

from m5stack import *
from m5ui import *
from uiflow import *

setScreenColor(0x222222)

while True:
    setScreenColor(0xff0000)
    wait(1)
    setScreenColor(0x3366ff)
    wait(1)
    wait_ms(2)

```

For more information on the wait block, check out the timer section of the book.

Button Loop



As well as the main loop, we have the button loop that continuously watches the three main buttons on the front of the M5Stacks and then runs the code inside it when it detects a triggering even. The four events the button loop waits for are as follows.

Was Pressed - This is the normal mode of operation and waits until a single short press of the button is detected before running the code inside it.

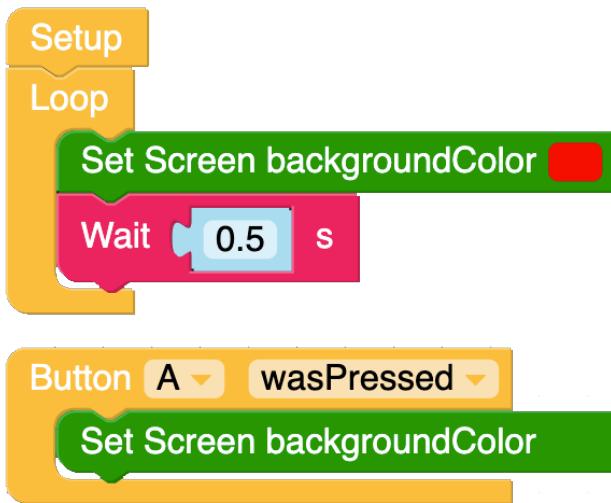
Was Released - This waits until the button is pressed and released before running the code placed inside it.

Long Press - Like the "Was Pressed" block, it waits until the button is pressed however, this block on activates if the button is pressed and held down for a few seconds.

Was Double Pressed - This function checks to see if the put-on was pressed and released twice within a few milliseconds before running the code inside it.

Example

In this example I have two loops. The main loop set the screen to red and the button loop changes the screen to green when the button is pressed. The main loop is required to reset the screen colour after the button has been released.



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

setScreenColor(0x222222)

def buttonA_wasPressed():
    # global params
    setScreenColor(0x009900)
    pass
    btnA.wasPressed(buttonA_wasPressed)

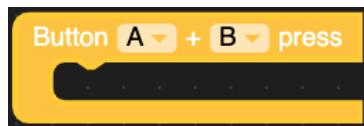
while True:
    setScreenColor(0xff0000)
    wait(0.5)
    wait_ms(2)
```

Obtain Button Value,



The obtain button block works in a similar mode to the Button loop but instead of running code placed inside it, it is a value block that tells other code if a button event has been triggered.

Button A+B Press Loop



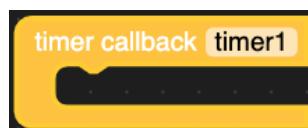
Button A+B Press expands on the function of the button loop by waiting until it detects that two buttons have been pressed at the same time.

Button Value



Similar to the Obtain Button function, this block is simplified to only having options for Pressed and Released.

Timer Callback Loop.



This loop runs a timer in the background and runs the code placed inside it when certain timer related events occur.

Set timer Period



Defines a timer function to run for a set period of milliseconds as "One Shot" which only runs once or "Periodic" which restarts when the period of milliseconds runs out.

Start Timer



Starts the timer for a defined period of time as One Shot" which only runs once or "Periodic" which restarts when the period of milliseconds runs out.

Stop Timer



Stops the timer selected in the drop down box.

Virtual User interface Designer and Graphic Functions.

The M5Stack and M5Stick core models have a selection of text and graphic functions defined for us. Some of these functions share the same code blocks. In this chapter I will explore the blocks available and teach you how to use them.

Functions are not the only things to share code. some of the hardware like the WS2812bs and RGB LEDs also share some of the code blocks.

Code Block	Title	Label	Rectangle	Circle	Image
Show Text	Yes	Yes	Yes	Yes	Yes
Show	Yes	Yes	Yes	Yes	Yes
Hide	Yes	Yes	Yes	Yes	Yes
Set Colour	Yes	Yes	Yes	Yes	Yes
Set Colour RGB	Yes	Yes	Yes	Yes	Yes
Set Background Colour	Yes	Yes	Yes	Yes	Yes
Set Background Colour RGB	Yes	Yes	Yes	Yes	Yes
Set Width and Height	No	No	Yes	No	Yes
Set Width	No	No	Yes	No	Yes
Set Height	No	No	Yes	No	Yes
Set X and Y Position	No	No	Yes	Yes	Yes
Set X Position	No	No	Yes	Yes	Yes
Set Y Position	No	No	Yes	Yes	Yes
Set Radius	No	No	No	Yes	No
Available to M5 Stack	Yes	Yes	Yes	Yes	Yes
Available to M5 Stick	No	Yes	Yes	No	No

[Table 01 UI Block compatibility.](#)

The table above shows some of the User Interface (UI) functions and which code blocks they share.

When used in conjunction with other functions we can create functions that can be as simple as displaying the value of a sensor or as complicated as you can imagine.

There are two Show blocks, the first will display the UI element defined on UI builder while the second takes a value from one off the sensors and changes the text to show that value.
Hide, hides the onscreen text.

The Show/Hide Blocks.



As you can see in the image above the Show/Hide block is only available to the Label, Rectangle and Circle UI elements. If you have more than one element on the screen you can use the block to control the individual elements by clicking on the arrow beside the element number. The second block is where you choose Hide or Show for the elements visibility. You can use this block to hide elements until certain event or conditions to happen and then set the individual elements to true in the condition or event function.



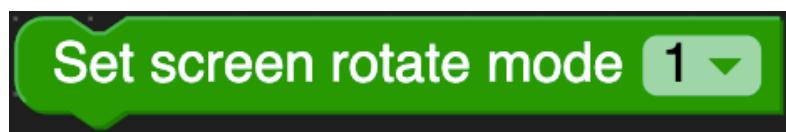
The Title UI element is slightly different in that its show and hide blocks are separate.

The Show Text Blocks



These two blocks are used to show or print a string of text onto the LCD screen. You can see that it has a puzzle-shaped space in it that allows you to use the title and label blocks for showing values returned from sensors.

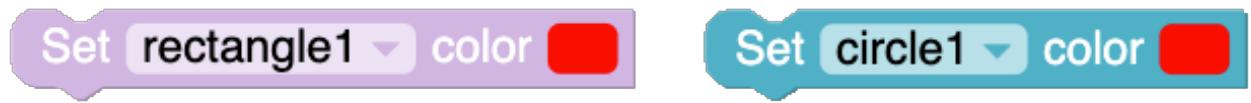
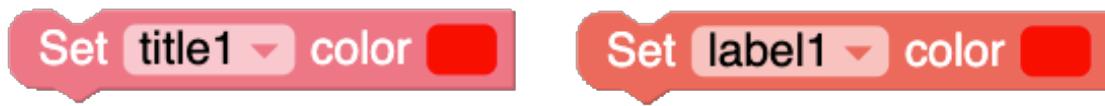
Set Screen Rotate Mode



Allows you to rotate the screen using one of the four preset modes shown below.
 Mode 0 turns the screen 90 degrees clockwise,
 Mode 1 is normal direction,
 Mode 2 is 90 degrees anticlockwise,
 Mode 3 is 180 degrees clockwise.

Set Color

The Set Colour block allows you to set each elements colour using the colour picker. Set Screen backgroundColor set the screen or "canvas" colour independent of the other U.I Elements.



Set Screen Brightness

Sets the screens brightness using a value block.

Set Colour R G B.

Set title1 color by R 0 G 0 B 0

Set label1 color by R 0 G 0 B 0

Set rectangle1 color by R 0 G 0 B 0

Set circle1 color by R 0 G 0 B 0

This Set Colour block allows you to set each elements colour by specifying individual value for each of the separate colour channels. Again for more information on colour functions available in UIFlow read chapter 4.3.3.

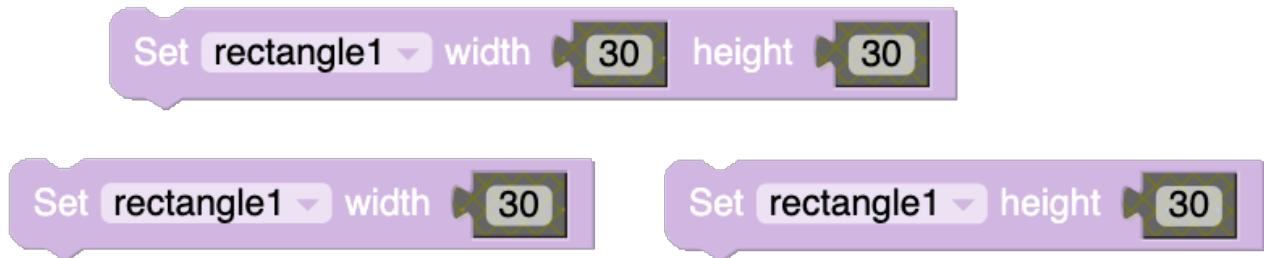
Background colour

Set title1 backgroundColor red

Set title1 backgroundColor by R 0 G 0 B 0

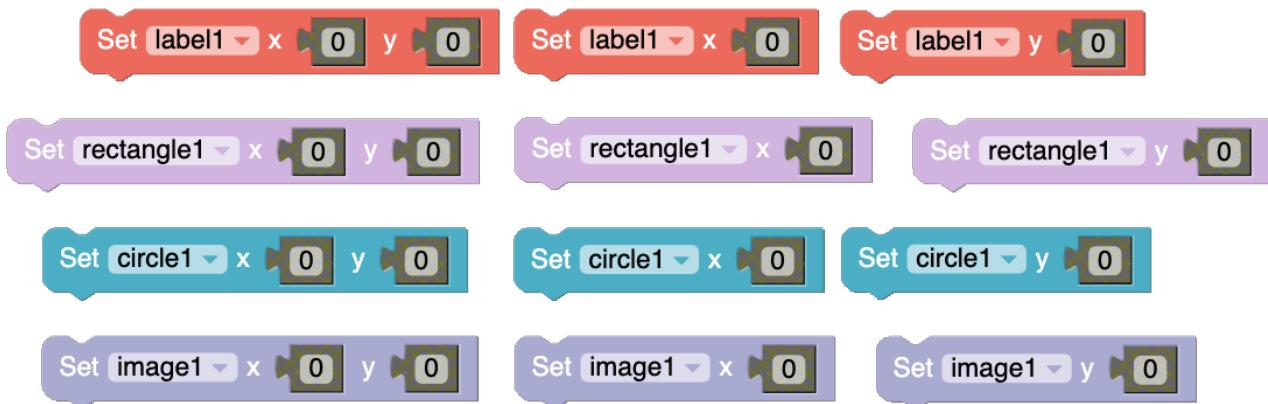
Only available to the Title UI element, the **Set backgroundColor** and **Set backgroundColor RGB** block allows us to change the colour off the title bar that appears at the top of the screen when the title element is visible. For more information on colour functions available in UIFlow read chapter 4.3.3.

Set Width and Height



Available only to the Rectangle UI element, the set width and set height blocks are used to control the size.

Set X and Y



The set X and Y blocks are used to set the position of the UI element on the M5Stacks canvas.

Set Circle Radius.



The Set Circle Radius block is only available to the circle UI element.

Internal Hardware

The M5Stack and M5Stick family have some differences on the inside due to configuration options available. The blocks in this section allow you to access those functions available in the various hardware versions.

Speaker,

Speaker.Beep,



The speaker Beep block allows us to make sounds by setting the frequency of the sound and the duration of the sound.

Speaker Volume,



Speaker volume allows us to set the volume of the speaker sounds.

Play Tone.



The Play Tone block allows users to play musical notes and set the duration in beats or fractions of Beats.

RGB, Set RGB Bar Colour,



Sets all the WS2812b LED colour in the M5Go base to the same colour using the colour selector.

Set RGB Bar Colour R,G,B,



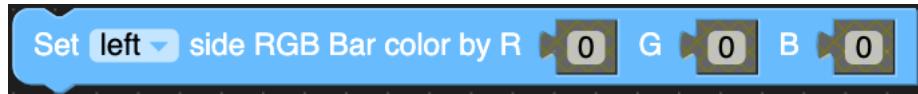
Sets all the WS2812b LED colour in the M5Go base to the same colour using individual Red, Green and Blue colour values.

Set (Side) RGB Bar Colour,



Sets all the WS2812b LED colour on the left or the right side of the M5Go base to the same colour using the colour selector.

Set (Side) RGB Bar Colour R,G,B,



Sets all the WS2812b LED colour on the left or the right side of the M5Go base to the same colour using individual Red, Green and Blue colour values.

Set (individual) RGB Bar Colour,



Sets one of the ten individual WS2812B LEDs colour on the M5Go base using the colour Picker.

Set (individual) RGB Bar Colour R,G,B,



Sets one of the ten individual WS2812B LEDs colour on the M5Go base using individual Red, Green and Blue colour values.

Set RGB Brightness,



Sets the brightness level of all the WS2812B LEDs in the M5Go baseplate.

IMU,
Get X,



Gets the X Value from the IMU.

Get Y,



Gets the Y Value from the IMU

Get X ACC,



Gets the X acceleration value from the IMU.

Get Y ACC,



Gets the Y acceleration value from the IMU.

Get Z ACC,



Gets the Z acceleration value from the IMU

Get X Gyro,



Gets the X value from the IMU's Gyro

Get Y Gyr,



Gets the Y value from the IMU's Gyro

Get Z Gyr,



Gets the Z value from the IMU's Gyro

Power

The Power blocks only work on M5Stacks and M5Sticks that are fitted with the customised I2C controlled IP5306 charge control chip.

Is Charging,



Checks if the battery is charging.

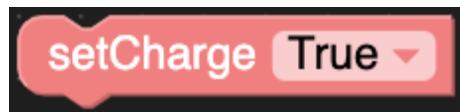
Is ChargedFull,

Checks to see if the battery is charged.



Set Charging,

Controls whether to charge the battery or not.



Get Battery Level,

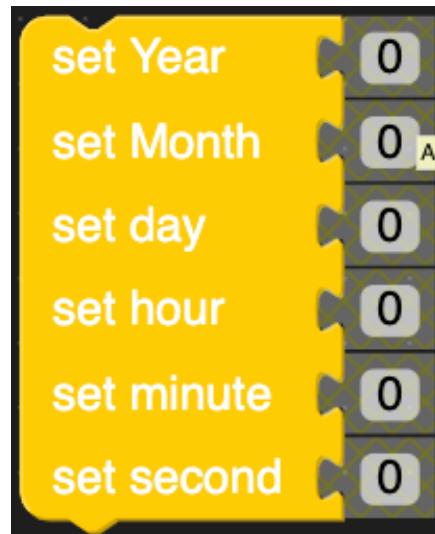
Checks the battery level.



Real Time Clock

Only available to M5Stacks and Sticks that have the built in Real Time Clock.

Time and Date configuration block.



Used to configure the initial time and date settings for the internal clock.

Get Year,



Gets the year from the RTC.

Get Month,



Gets the month from the RTC.

Get Day,



Gets the day from the RTC.

Get Hour,



Gets the hour from the RTC.

Get Minute,

Gets the minute from the RTC.

Get Second,

Gets the seconds from the RTC.

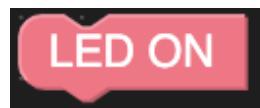
Get Local Time.

Gets the local from the RTC.

LED

On the top left hand corner of the M5Stick C is a red led. To control this led you use the following blocks.

LED On



The Micropython code for this is,

M5LED.ON()

Turns the red led on the top of the M5Stick C on.

LED Off



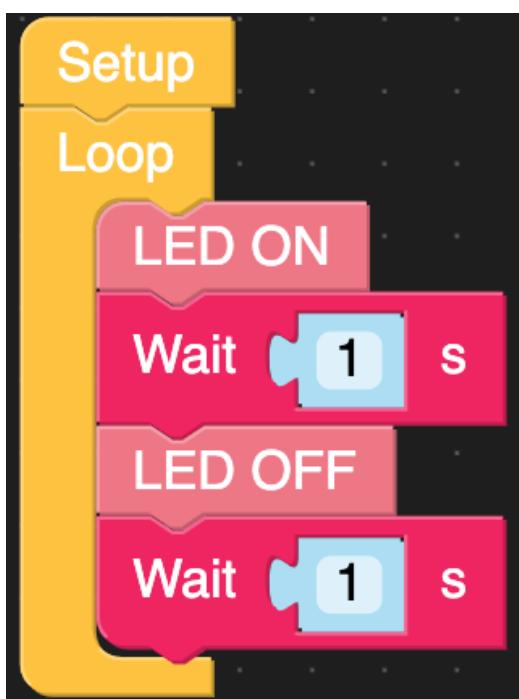
The Micropython code for this is,

M5LED.OFF()

Turns the red led on the top of the M5Stick C off.

Example.

The following example turns the LED on for one second and then off for one second.



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

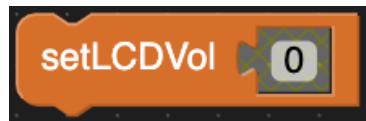
setScreenColor(0x111111)

while True:
    M5Led.on()
    wait(1)
    M5Led.off()
    wait(1)
    wait_ms(2)
```

23

AXP

SetLEDVol,



Sets the brightness of the M5Stick S's LCD screen.

Is Charge,



Checks to see if the battery of the M5Stick C is charged. Returns true or false.

GetBatVolt,



Gets the current battery voltage of the M5Stick C.

GetChargeL



Gets the charge rate/level

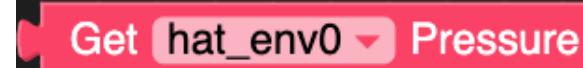
GetDischargeL



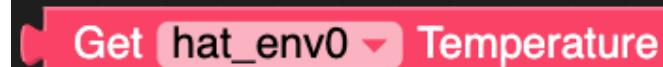
Gets the Discharge rate/level.

Hats

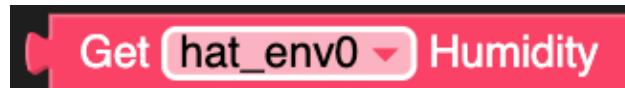
Hats are add-on modules specific to the M5Stick C. H.A.T stands for Hardware Attached on Top.

ENV Hat**Get Pressure,**A Scratch script consisting of a black hat icon followed by a pink "Get" hat block. Inside the pink block, the text "hat_env0" is followed by a dropdown menu icon, and the word "Pressure" is written.

Gets the current air pressure.

Get Temperature,A Scratch script consisting of a black hat icon followed by a pink "Get" hat block. Inside the pink block, the text "hat_env0" is followed by a dropdown menu icon, and the word "Temperature" is written.

Gets the current temperature.

Get Humidity,A Scratch script consisting of a black hat icon followed by a pink "Get" hat block. Inside the pink block, the text "hat_env0" is followed by a dropdown menu icon, and the word "Humidity" is written.

Gets the current Humidity.

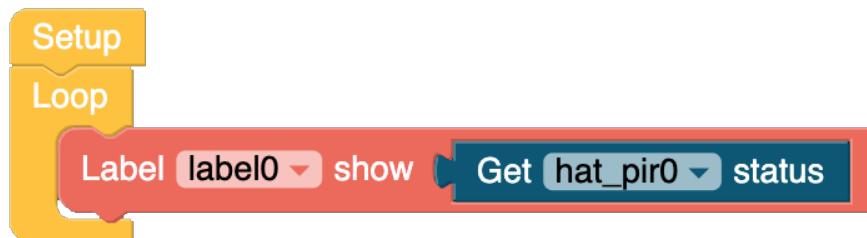
P.I.R Hat, Get hat_pir Status,



Returns True or False if the PIR detects anything.

Example

In the following example I am using a label to show the status of the P.I.R on the Stick C's screen.



```
from m5stack import *
from m5ui import *
from uiflow import *
import hat

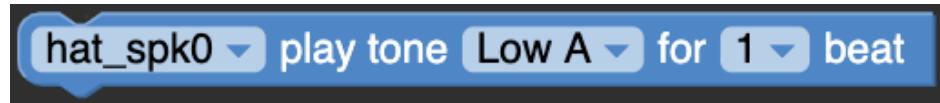
setScreenColor(0x111111)

hat_pir0 = hat.get(hat.PIR)

label0 = M5TextBox(28, 85, "Text", lcd.FONT_Default,0xFFFF, rotate=0)

while True:
    label0.setText(str(hat_pir0.state))
    wait_ms(2)
```

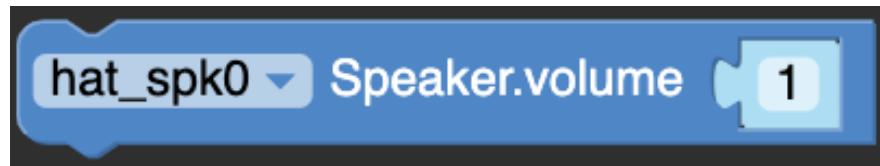
Speaker
Hat_spk0 Play Tone



The Play Tone block allows users to play musical notes and set the duration in beats or fractions of Beats.
Hat_spk0 Speaker.beep



The hat_spk0 speaker Beep block allows us to make sounds by setting the frequency of the sound and the duration of the sound.
Hat_spk0 Speaker Volume



The hat_spk0 Speaker volume allows us to set the volume of the speaker sounds.

NCIR



Returns a value from the N.C.I.R Hat Sensor.

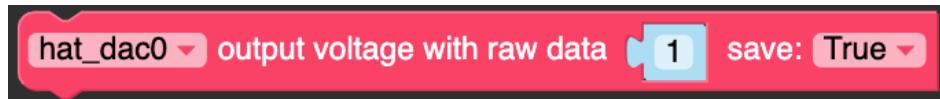
29

DAC



hat_dac0 Output Voltage

Gets a voltage reading from 0 to 3.3 volts If save it set to true then the data will be written to the M5Sticks E.E.P.R.O.M.



hat_dac0 Output Voltage with Raw Data.

Gets a raw data reading of 0 to 4096. If save it set to true then the data will be written to the M5Sticks E.E.P.R.O.M.

ADC



Returns an analogue reading from the Analogue to Digital Converter Hat.

Units,

Environmental,

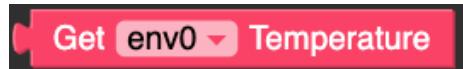
Get Pressure,



Gets the current air pressure.

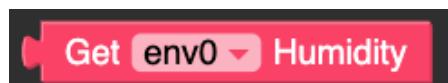
Get Temperature,

Gets the current temperature.



Get Humidity,

Gets the current Humidity.



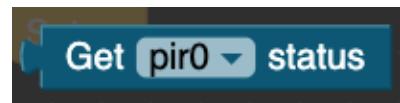
33

Angle,
Get Angle,



Returns the value generated by the angle sensor. Values are from 0 to 4096.

PIR,
Get PIR Status,



Returns the status of the PIR sensor unit.

WS2812b,

Set individual WS2812b to colour,



Sets an individual WS2812b LED colour using the Colour picker

Set WS2812b Range to colour,



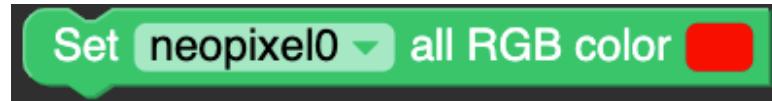
Sets a range of WS2812b LED colour using the Colour picker

Set WS2812b Range to RGB colour,



Sets a range WS2812b LED colour using individual Red, Green and Blue values.

Set all WS2812b Colour,



Sets all WS2812b LEDs to the same colour using the colour picker.

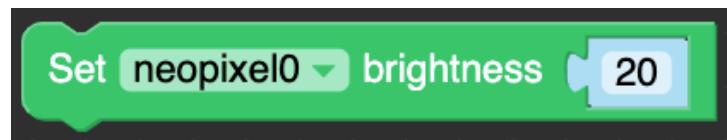


```
from m5stack import *
from m5ui import *
from uiflow import *
import unit

setScreenColor(0x222222)
neopixel0 = unit.get(unit.NEOPIXEL,
unit.PORTB, 10)

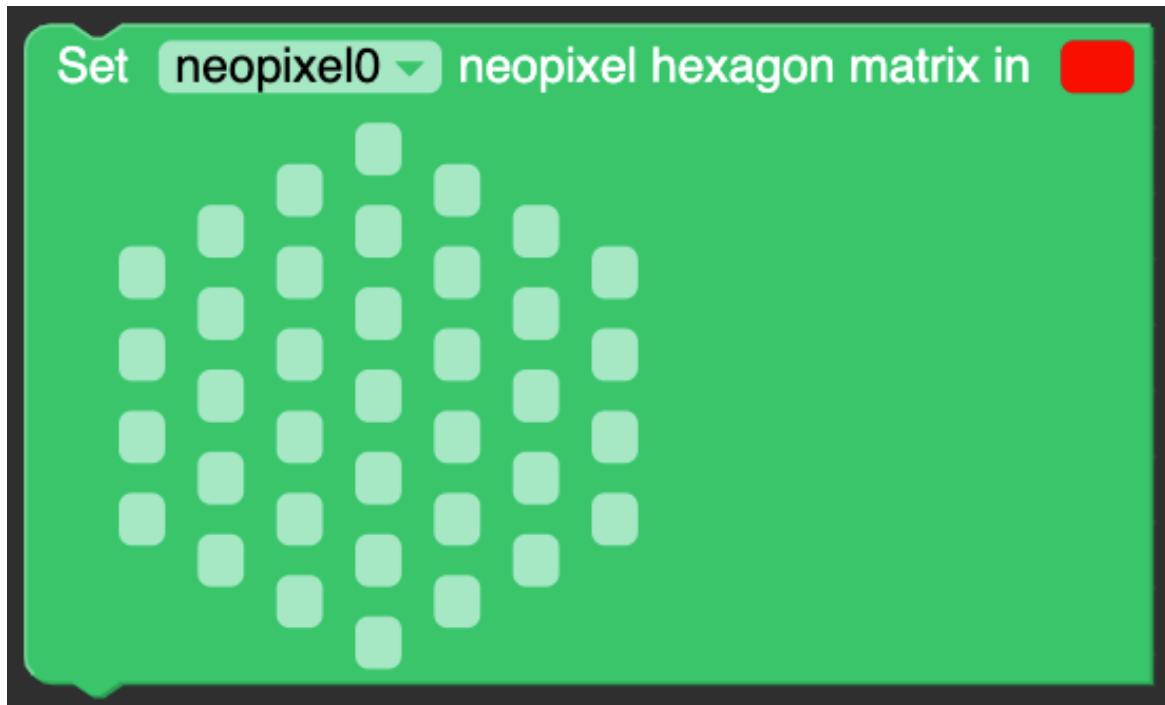
while True:
    neopixel0.setColorAll(0xff0000)
    wait(1)
    neopixel0.setColorAll(0xff0000)
    wait(1)
    wait_ms(2)
```

Set WS2812b Brightness,



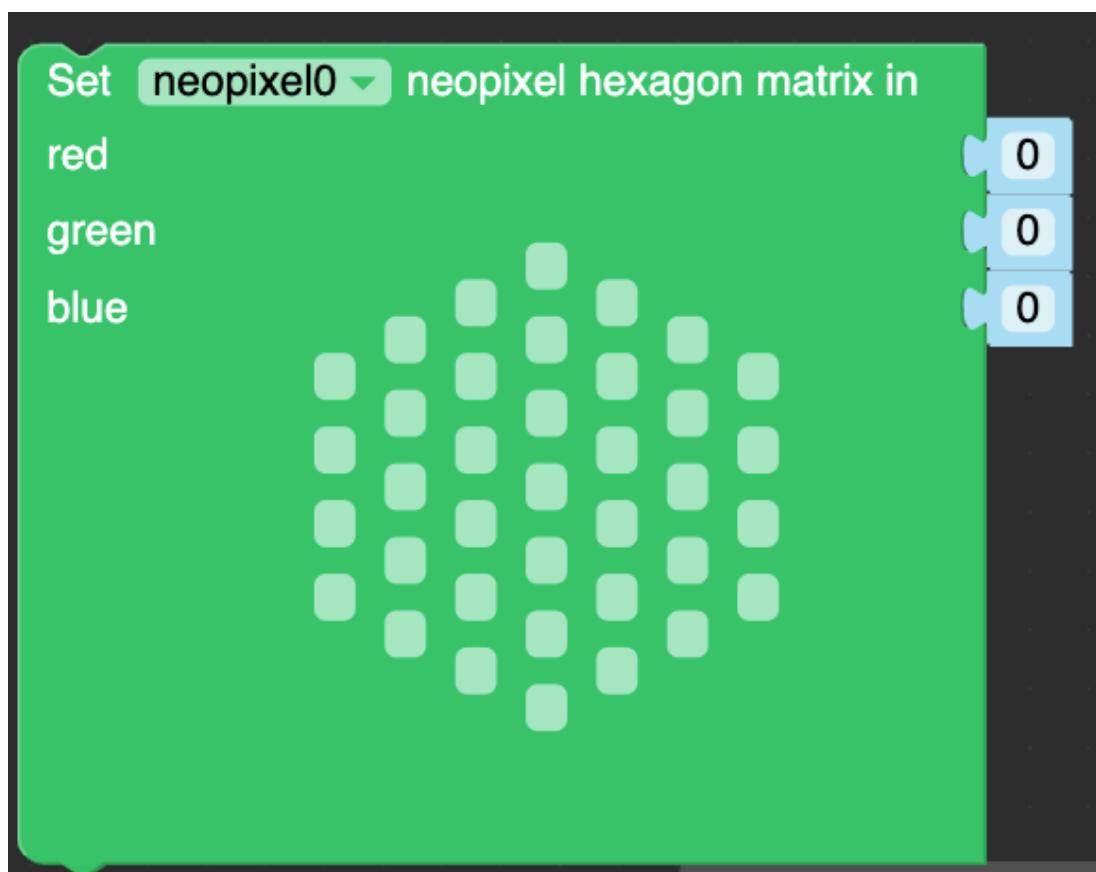
Set the brightness levels of all WS2812b LEDs

Set WS2812b Hex Colour,



Used to set the colour of the individual WS2812b LEDs on the Neohex using the colour picker.

Set WS2812b Hex (Variable) Colour,



Used to set the colour of the individual WS2812b LEDs on the Neohex using individual Red, Green and Blue values.

39

Joystick,
Get X,



Returns the joysticks X value.
Get Y,



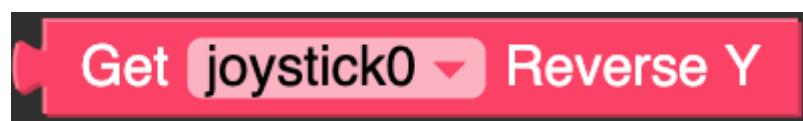
Returns the joysticks Y Value.
Is Pressed,



Returns the state of the joysticks button when pressed.
Reverse X



Returns an inverted value of the X position.
Reverse Y



Returns an inverted value of the Y position.

Light,
Get Light Analogue Value,



Returns the analogue value from the light sensor.
Get Light Digital Value,



Returns the digital value from the light sensor.

41

Earth,
Get Earth Analogue Value,



Returns the analogue value from the earth sensor.
Get Earth Digital Value,

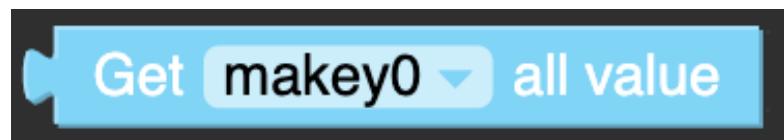


Returns the digital value from the earth sensor.

Makey,
Get Value,



Get all Value,



43

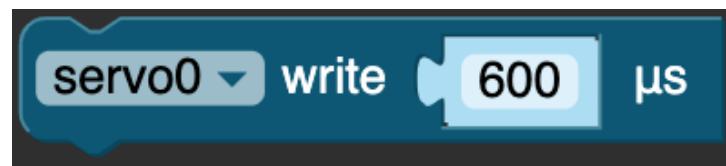
Servo,

Servo Rotate to Degree,



Tells the servo to move to a specified position.

Servo Speed,



Sets the move speed of the servo.

Weight,

Zero Weight,



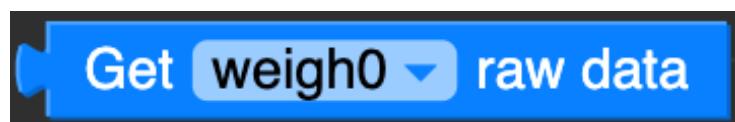
Sets the values coming from the sensor to zero.

Get Weight,



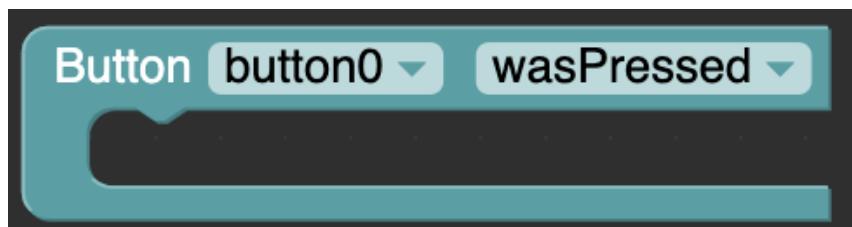
Get the weight from the sensor.

Get Raw Data



Gets the raw 24 bit value data from the sensor.

Button,
Button Loop,



The button loop continuously watches the button unit and then runs the code inside it when it detects a triggering even.

The four events the button loop waits for are as follows.

Was Pressed - This is the normal mode of operation and waits until a single short press of the button is detected before running the code inside it.

Was Released - This waits until the button is pressed and released before running the code placed inside it.

Long Press - Like the "Was Pressed" block, it waits until the button is pressed however, this block on activates if the button is pressed and held down for a few seconds.

Was Double Pressed - This function checks to see if the put-on was pressed and released twice within a few milliseconds before running the code inside it.

Button Action,



The obtain button block works in a similar mode to the Button loop but instead of running code placed inside it, it is a value block that tells other code if a button event has been triggered.

Duel Button,
Duel_Button Loop,



The duel button loop continuously watches the two buttons on the duel button unit and then runs the code inside it when it detects a triggering event on either of the buttons.

The four events the button loop waits for are as follows.

Was Pressed - This is the normal mode of operation and waits until a single short press of the button is detected before running the code inside it.

Was Released - This waits until the button is pressed and released before running the code placed inside it.

Long Press - Like the "Was Pressed" block, it waits until the button is pressed however, this block only activates if the button is pressed and held down for a few seconds.

Was Double Pressed - This function checks to see if the put-on was pressed and released twice within a few milliseconds before running the code inside it.

Button Action,

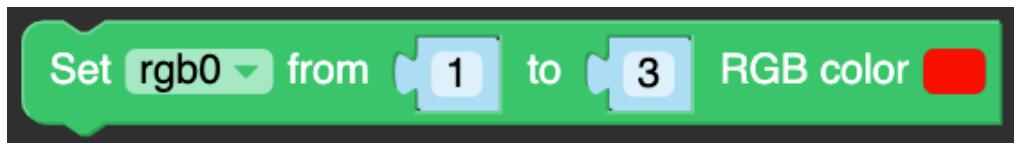


The obtain duel button block works in a similar mode to the Button loop but instead of running code placed inside it, it is a value block that tells other code if a one of the two buttons have been triggered.

RGB,
Set RGB Bar Colour,

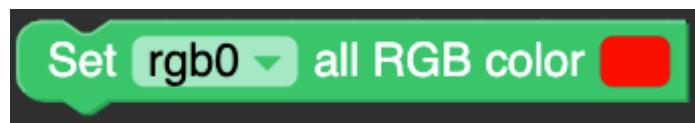


Sets one of the three R.G.B LED colours using the colour picker.
Set RGB Bar Colour R,G,B,



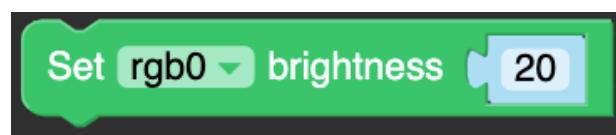
Sets more than one R.G.B LED colours using the colour picker.

Set RGB all Colour,



Sets all the R.G.B LED colours using the colour picker.

Set RGB Brightness,



Sets the Brightness of the R.G.B LEDs.

Relay,
Set Relay On,



Sets the relay unit in to the on position.
Set Relay Off,



Sets the relay unit in to the off position.

**Heart Unavailable,
Not available in UIFlow at present.**

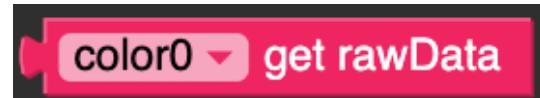
ADC,
ADC Read,



Returns an analogue reading from the Analogue to Digital Converter Unit.

49

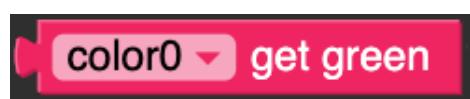
Colour,
Get Raw data.



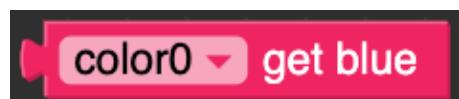
Returns the three raw values for Red, Green and blue from the sensor.
Get Red



Returns the red value from the sensor.
Get Green



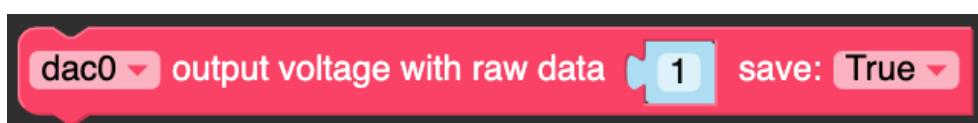
Returns therein value from the sensor.
Get Blue



Returns the blue value from the sensor.
DAC,
dac0 Output Voltage



Gets a voltage reading from 0 to 3.3 volts If save it set to true then the data will be written to the M5Sticks E.E.P.R.O.M.



dac0 Output Voltage with Raw Data.
Gets a raw data reading of 0 to 4096. If save it set to true then the data will be written to the M5Sticks E.E.P.R.O.M.

IR,
Get IR State,



Returns the state of the IR Unit.
Set IR On,



Turns the IR unit on.
Set IR Off,



Turns the IR unit off.
NCIR,
NCIR Read,



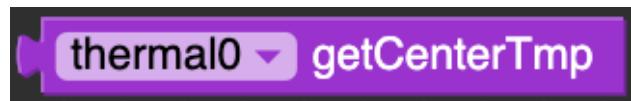
Returns values from the NCIR Unit.

51

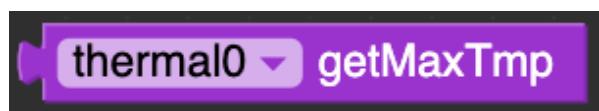
Thermal,
Get Temp X, Y,



Returns the temperature detected at the specified coordinates.
Get Centre Temp



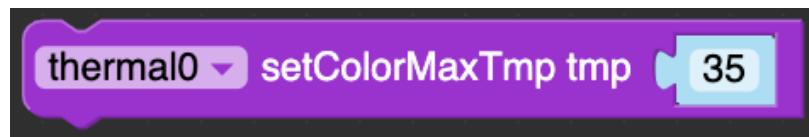
Returns the temperature detected in the middle of the sensor.
Get Max Temp,



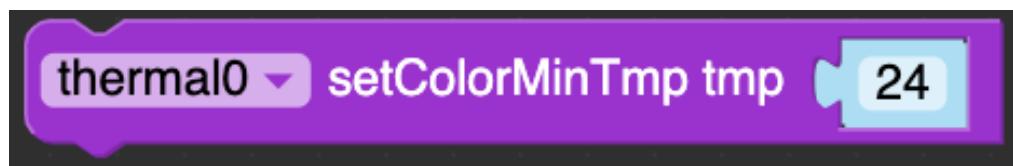
Returns the maximum temperature detected by the unit.
Get Min Temp,



Returns the minimum temperature detected by the unit.
Set Colour Max Temp,



Sets the upper level that the sensor will use.
Set Colour Min Temp,

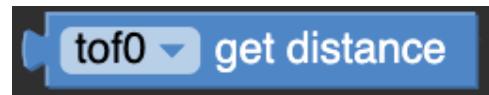


Sets the lower level that the sensor will use.
Update X, Y, show centre,



Changes the X, Y temperature read position while displaying the centre cross hair.

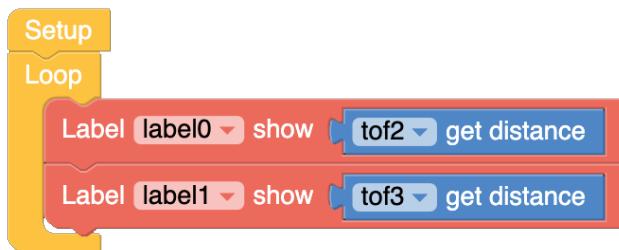
TOF, Get Distance,



Returns the distance value from the T.O.F Unit.

Example

In the following example I am using two labels to show readings from two separate but identical T.O.F units.



```

from m5stack import *
from m5ui import *
from uiflow import *
import unit

setScreenColor(0x222222)
pahub0 = unit.get(unit.PAHUB, unit.PORTA)
tof2 = unit.get(unit.TOF, unit.PAHUB0)
tof3 = unit.get(unit.TOF, unit.PAHUB1)

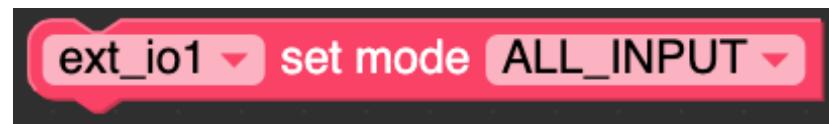
label0 = M5TextBox(29, 90, "Text",
lcd.FONT_Default,0xFFFF, rotate=0)
label1 = M5TextBox(177, 99, "Text",
lcd.FONT_Default,0xFFFF, rotate=0)

while True:
    label0.setText(str(tof2.distance))
    label1.setText(str(tof3.distance))
    wait_ms(2)

```

EXT I/O,

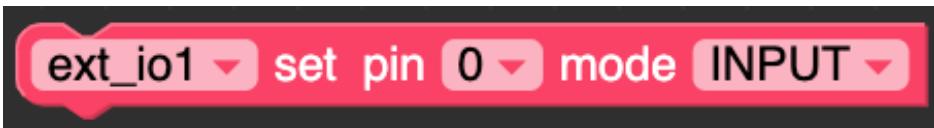
Set I/O Port Mode,



Set the mode of all the pins to Input or output.

Set I/O Pin Mode,

Sets the individual pins to input or output.



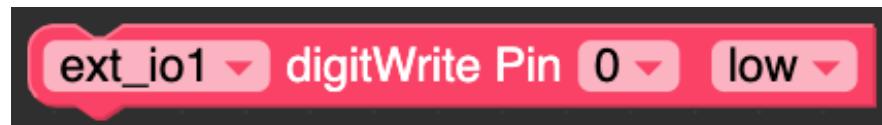
Digital WritePort,

Writes a hex value to the I/O unit.



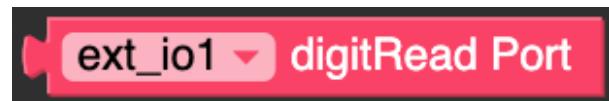
Digital Write Pin,

Set each individual pin as high or low.



Digital Read Port,

Reads the digital state of the I/O Unit.



RFID,

Read String,



Reads a string of information stored at the specified address on the RFID card or tag.

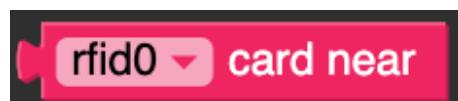
Write String,

Writes a string of date to the specified address.



Card Near,

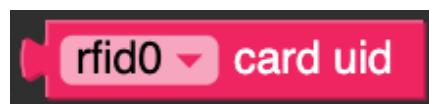
Returns true or false if an RFID tag or card is in



proximity of the unit.

Card UID,

Returns the UID stored on the RFID tag or card.

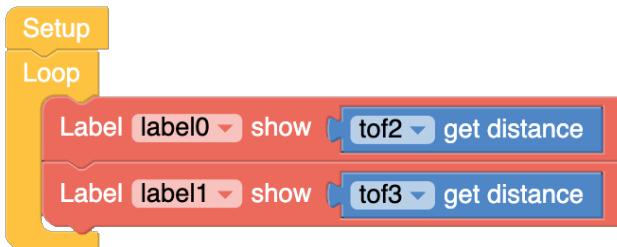


PAHub

While the PAHub has some blocks to use, the hub is designed to allow the connection of multiple identical units that have the same I2C address. The blocks provided are as follows.

Set position state

Set position



Set port value

Example

In the following example I am using two labels to show readings from two separate but identical T.O.F units.

```

from m5stack import *
from m5ui import *
from uiflow import *
import unit

setScreenColor(0x222222)
pahub0 = unit.get(unit.PAHUB, unit.PORTA)
tof2 = unit.get(unit.TOF, unit.PAHUB0)
tof3 = unit.get(unit.TOF, unit.PAHUB1)

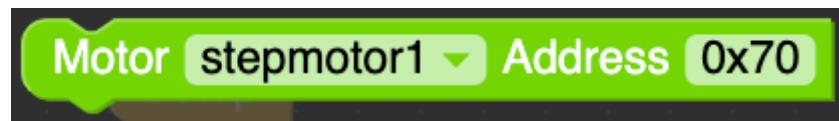
label0 = M5TextBox(29, 90, "Text",
lcd.FONT_Default,0xFFFF, rotate=0)
label1 = M5TextBox(177, 99, "Text",
lcd.FONT_Default,0xFFFF, rotate=0)

while True:
    label0.setText(str(tof2.distance))
    label1.setText(str(tof3.distance))
    wait_ms(2)
  
```

To view an introduction video on the PAHub, you can check out Lukes video on the M5Stack youtube channel <https://www.youtube.com/watch?v=nMsCwqCE5c8>

Modules,

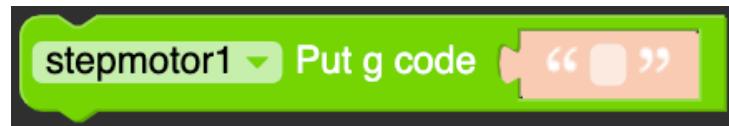
Stepper Motor,
Stepper Motor Module Address,



Sets the Hex address to use for the stepper motor driver.
Stepper Motor Position,



Sets the position and speed that the stepper motor is to move to.
Run "G" Code,



For running G-Code in the modules g-code interpreter.
Stepper Motor Movement Mode,



Sets how the stepper motor is to move. Distance is used to make it move a defined distance, Absolute is use to make the motor move to a set of coordinates.

Lock Stepper Motor,

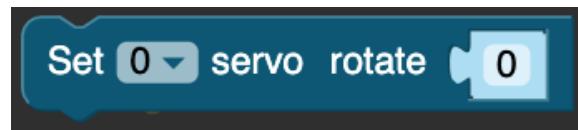


Locks the stepper motor by powering the cores preventing the motor from being mechanically turned.
Unlock Stepper Motor,



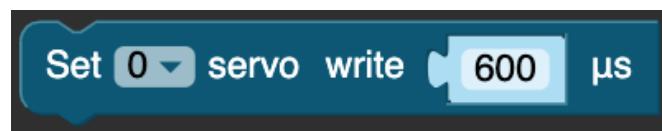
Unlocks the stepper motor by removing the power to the coils allowing them to be mechanically turned.

Servo,
Set Servo rotate,



Tells the servo on the defined channel to rotate to a specified position.

Set Servo Speed,



Sets the move speed of the servo on the defined channel.

59

Bala,
Bala Move distance,



Moves the Bala motors forward or backward by a user defined value.
Turn Wheel



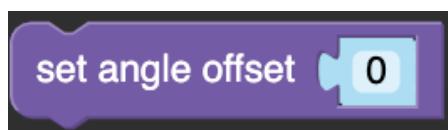
Turns the left or the right wheel wheel by a user defined value.
Rotate



Tells the bala to rotate.
Get Angle.

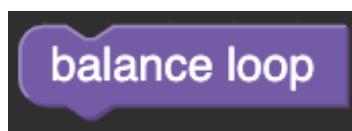


Returns the angle that the Bala unit has turned.
See Angle Offset.



Used to set an angle offset to handle inaccuracies between the motors.

Balance Loop



Tells the M5Stack to update its internal position in memory.

Bala Motor,
Set Motor direction and speed.



Sets the motors rotation direction and speed.
Run forward distance and speed.



Commands the bala to move in a direction a user defined distance at a user defined speed.
Go to Position



Commands the motor to move to a user defined position at a user defined speed.
Stop Motor

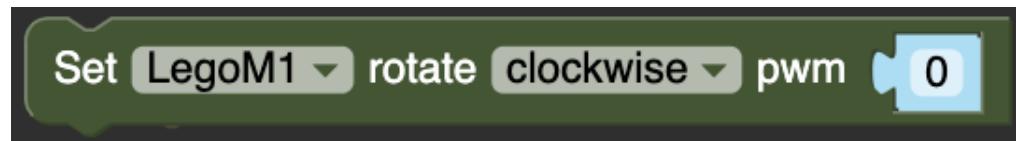


Commands the motor on the selected channel to stop.
Read encoder

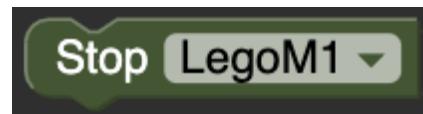


Returns the values from the Bala motors built in encoder.

Lego Motor,
Set Lego Motor Rotate PWM,



Sets the Lego motor on channel 1 to rotate clockwise with a value.
Stop Lego Motor,



Stops the motor o channel 1
Clear encoder,

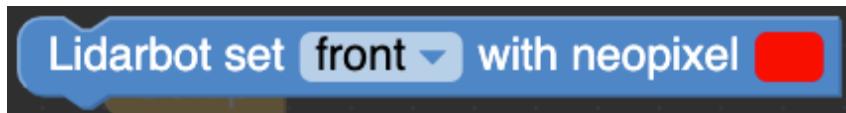


Clears decoder reading from memory.
Read encoder,



Returns the values from the lego motors built in encoder.

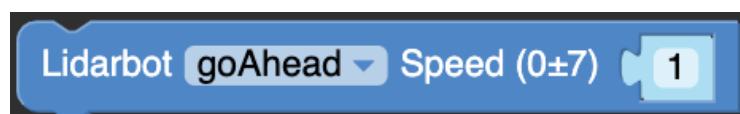
Lidarbot,
Lidarbot Set All LED Colour,



Set the colour of all the WS2812b LED's using the colour picker.
Lidarbot Set Individual LED Colour,



Sets the colour of the individual WS2812b LED using the colour picker.
Lidarbot Move,



Moves the Lidarbot in the direction at a user defined speed.
Lidarbot Set Speed,



Sets the speed of each of the individual four wheels.
Lidarbot Set Servo,



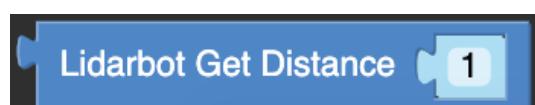
Set the angle of a servo connected to the Lidarbot.
Lidarbot Axis Speed,



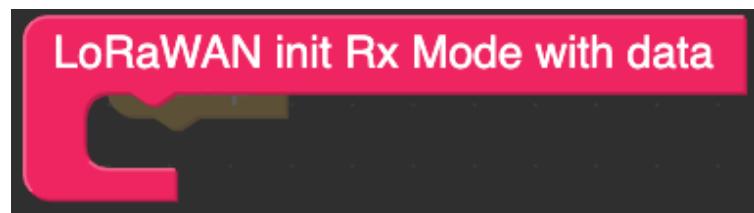
Set the X and Y axis speed.
Lidarbot Draw Map,



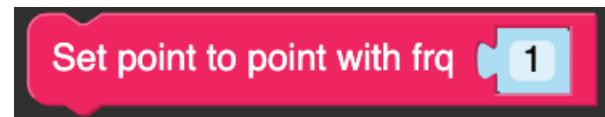
Lidarbot Get Distance,



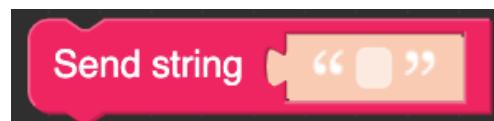
LoRaWan,
Init LoRaWan Rx Mode



Set Point to Point Frequency.



Send String.



Get Data,



65

PM2.5

Get PM2.5 Value,



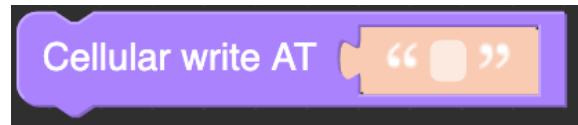
Returns particle count of selected size particles in SPM or APM value.

Get Particle Count,



Returns the amount of particles detected above the selected size.

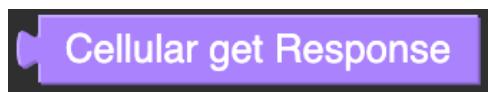
Cellular,
Cellular Write AT,



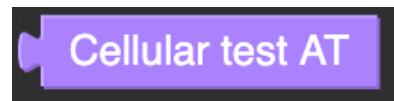
Used to send AT commands to the Cellular module.
Cellular Wait,



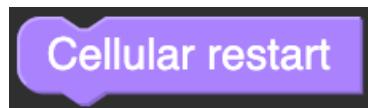
Sets the time that the cellular module waits before responding to AT commands.
Cellular Get Response,



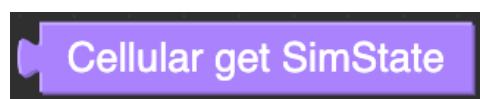
Immediately returns a response from the cellular module after an AT command is sent.
Cellular Test AT



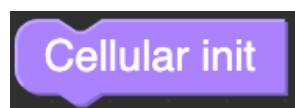
Cellular Restart



Forces the cellular module to restart independent of the M5Stack.
Cellular Get SimState



Returns the status of a Sim card plugged into the module.
Cellular Init,



Initialises the Cellular module.
Cellular GPRS Connect,



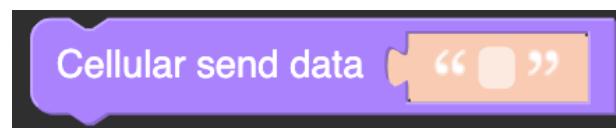
Returns the module status after ordering the module to connect to a GPRS network with the user defined credentials.

Cellular Connect host,



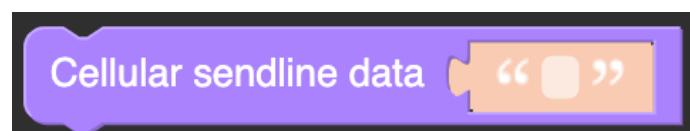
Returns the status of the module after trying to connect to a host with the following user defined credentials.

Cellular Send Data,

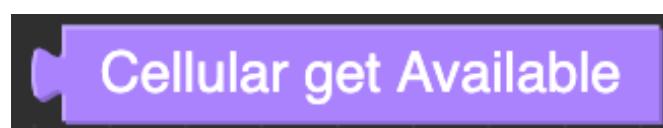


Sends data over the connection to the host device.

Cellular Send Line Data,



Cellular Get Available



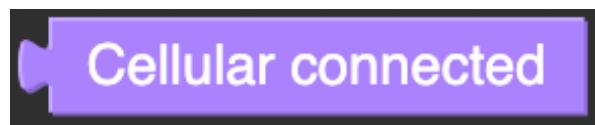
Reports if Cellular networks are available.

Cellular Read



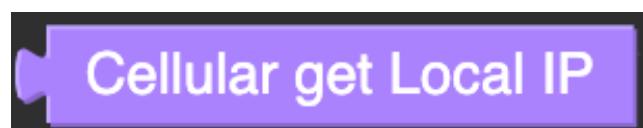
Returns available Cellular networks.

Cellular Connected



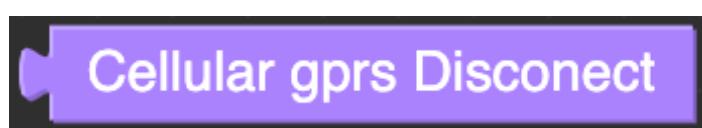
Returns the status if the cellular module has connected to a network or not.

Cellular Get Local IP



Get a local IP from a network that the module has connected to.

Cellular GPRS Disconnect,



Returns the module status after being commanded to disconnect from a network.
Cellular Get Network State,



Reports the status of the Cellular network.

69

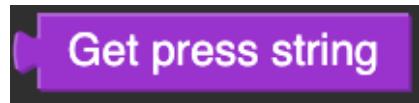
Faces Modules
Gameboy Face,
Get Direction Is Pressed,



Get Direction Was Pressed/Released,



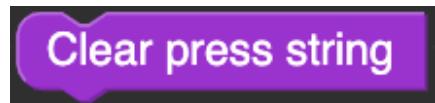
Calculator Face,
Get Press String,



Get Press Button Int Value,



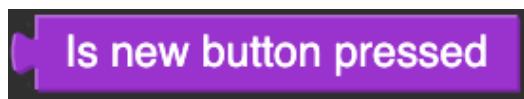
Clear Press String,



Delete Press String,



Is New Button Pressed,



71

Keyboard Face,
Get Press String,



Clear Press String,



Delete Press String,



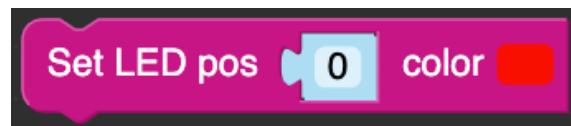
Get Press Button Int Value,



Is New Button Pressed,



Encoder Face,
Set LED Pos Colour,



Clear Encode Value to Zero,



Get Encode Value,



Get Encode Direction,



Is Encode Pressed,



73

Joystick Face,
Get X Value,



Get Y Value,



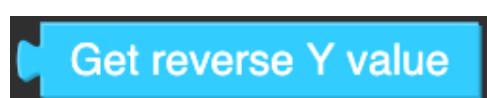
Is Pressed,



Get Reverse X Value,



Get Reverse Y Value,



Set LED Pos Colour,



Finger Face,
Get State,



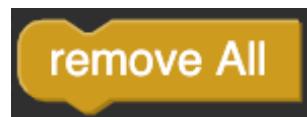
Get access,



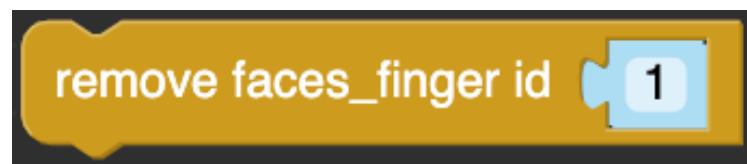
Get ID,



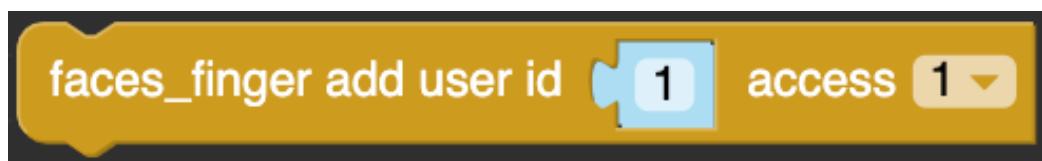
Remove all,



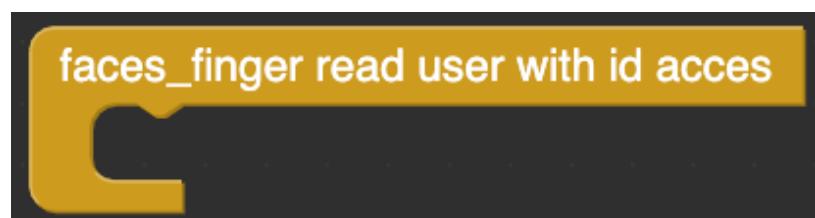
Remover Faces Finger ID (),



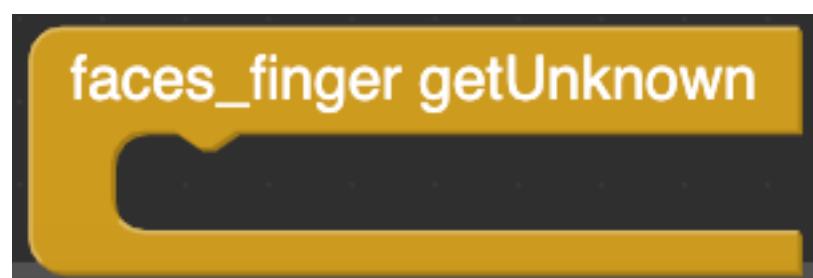
Add User Faces Finger ID (),



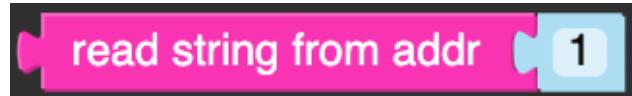
Faces Finger read User,



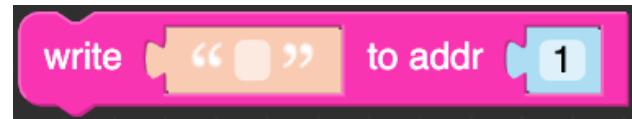
Faces Finger Get Unknown.



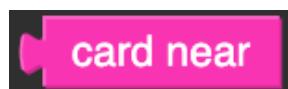
RFID Face,
Read String,



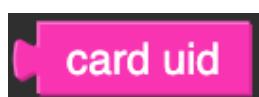
Reads a string from an RFID card or Tag.
Write String,



Writes a string to the RFID card or tags storage space.
Card Near,



Returns true if an RFID card or tag is detected.
Card UID,



Gets the Card or Tag UID.

Variables,
Create Variable,



Creates a Variable set consisting off the following blocks.
Set Variable,



This function defines a value for the variable.
Change Variable,



This function changes the value of the variable.
Variable,



The variable block that other functions can query.

Maths,
Value,



Used to hold a set value
Calculation,



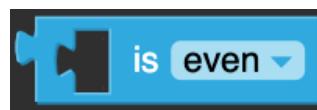
Returns the sum of the equation. Options are Plus, Minus, Multiply, Divide, To the power of.
Pi,



Allows calculations to use Pi (3.14), Eulars Number (2.718), Golden Ration (1.618), Square root 2 (1.414)
Square Root 1/2 (0.707), and infinity.
Remainder of,



Returns the remainder of one value divided by another value.
Condition is even,



Returns true or false if value is equal to even, odd, prime, whole, positive, negative or divisible by.
Sum of list,

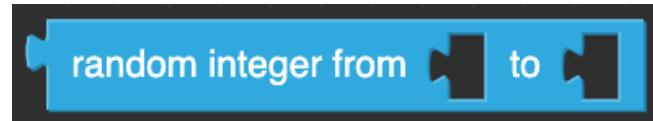


Returns the sum of a list of values.
Random Fraction,



Returns a random fraction.

Random Integer,



Returns a random integer from a range of user defined numbers.

Round,



Rounds the following value up or down.

Square Root,



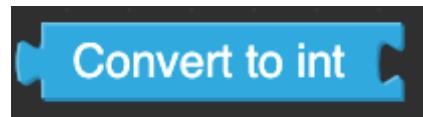
Returns the square root value, absolute value, the negation of a value, the natural logarithm of a number, a base10 logarithm of a number, e to the power of a number, or 10 to the power of a number or value connected to it.

Sin,



Returns the Sine, Cosine, Tangent, Arcsine, Arccosine, or Arctangent of a number.

Convert to int,



Converts a value to an integer.

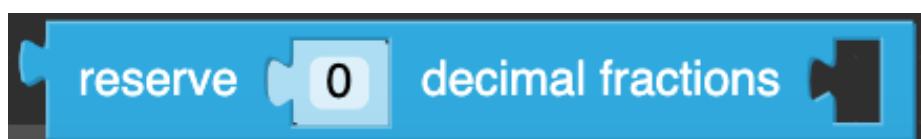
Convert to Float,

Loops,



Converts a value to a floating point number.

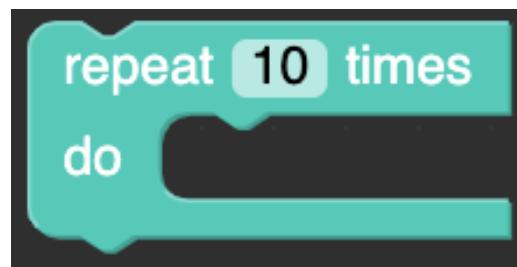
Reserve Decimal Fraction.



Reserves the user defined number as a decimal fraction.

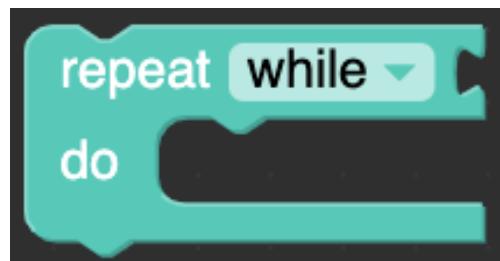
79

Loops
Repeat,



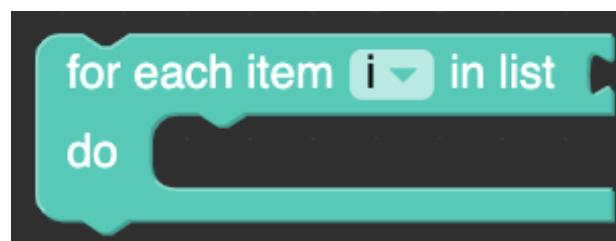
Repeats the code in side it ten times.

Repeat while Condition,



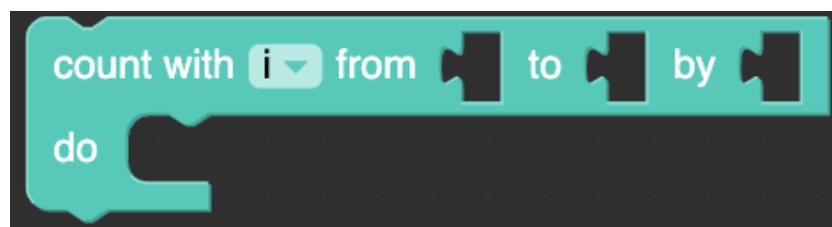
Repeats the code inside while the user defined condition is set.

For Each,



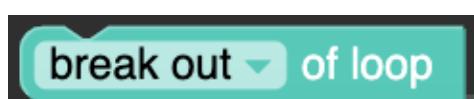
Repeats the code inside for each item in a list.

Count with,



Runs the code inside on selected items in a range using a user defined variable.

Break out,



Used to exit from a loop of code.

Logic,
If - Do,



If a defined condition is met the code inside is run.

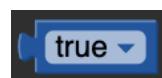
If - Else - Do,



If a defined condition is met the code inside is run otherwise another set of code is run..

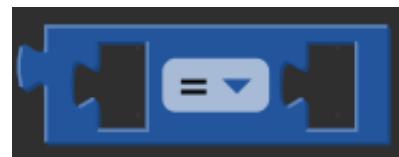


Condition True,



Returns true or false.

Condition Equals Condition,



Returns true or false if a sum two conditions are met.

Not,



81

Inverts or reverses the command.

Null,

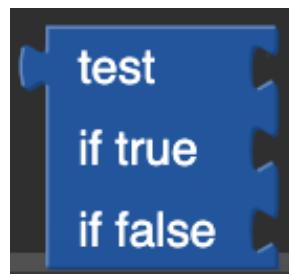


Condition and Condition,



Used to define if two conditions are required.

Test - True/False



Graphic,

Lcd.clear



Lcd.clear

Lcd.clear turns all the pixels on the screen to black.

Lcd.fill



Lcd.fill

Lcd.fill will change all of the pixels on the screen to the same colour. The colour is set in this block by using the colour picker.

lcd.print



Lcd.print



Font: FONT_Default ▾

Font specifies a font to be used for the text that will be shown on screen.

Lcd.pixel



Lcd.pixel x:

0

y:

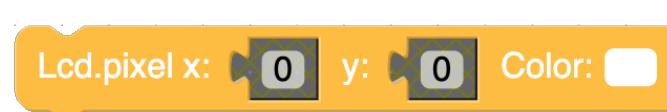
0

Color:



Lcd.pixel will colour specific pixel at the specified coordinate and in the specified colour using the colour picker.

Lcd.line,



Lcd.line x:

0

y:

0

Color:



Lcd.line draws a line starting at the specified coordinates and ending in X1 and Y1 in the specified colour using the colour picker. By specifying X1 and Y1 we can have angled lines.

Lcd.rectangle,



Lcd.rectangle x:

0

y:

0

width:

0

height:

0

color ▾



Lcd.rectangle draws a rectangle starting at the specified coordinate with a user defined height and width in the specified colour using the colour picker.

Lcd.triangle,



Lcd.triangle draws a triangle by specifying the coordinates of the three individual points of a triangle in the specified colour using the colour picker.

Lcd.circle,



Lcd.circle draws a circle with the centre point the specified coordinate with a radius defined by the user and in the specified colour using the colour picker.

Lcd.ellipse,



Lcd.ellipse draws a circle with the centre point the specified coordinate with an X radius and separate Y radius defined by the user and in the specified colour using the colour picker.

Lcd.arc,



Lcd.arc draws an arc with the centre at the specified coordinates with a user defined radius, thickness, start point and end point and in the specified colour using the colour picker.

Please Note that there is a bug in the code that keeps deleting the Radius value.

Lcd.polygon,



Lcd.polygon draws a polygon with the centre at the specified coordinates with a user definable radius, number of sides, thickness, rotation and in the specified colour using the colour picker.

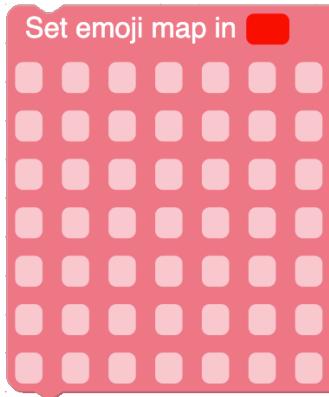
Please note that when Radius and thickness are the same, we can get a snowflake like appearance.

Emoji,

The Emoji menu contains blocks that allow us to draw Emojis and control the background shown behind them.

There are three blocks available here with the biggest being the emoji map

Emoji Map



The emoji map has a 7 X 7 grid which you click on each cell to make it light up in the specified colour using the colour picker on the right of the text. To make a cell active, all you have to do is to click on each of the lighter blocks and a tick will appear to show that you have made that cell active.

Set line 1 row 1 in colour.



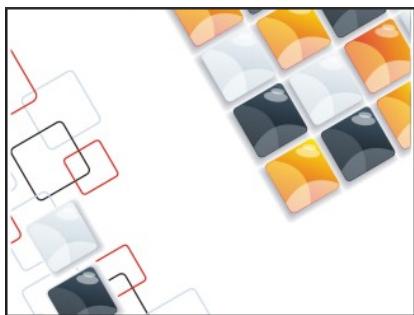
This block allows you to set the colour of each individual cell.

Change Background Image.

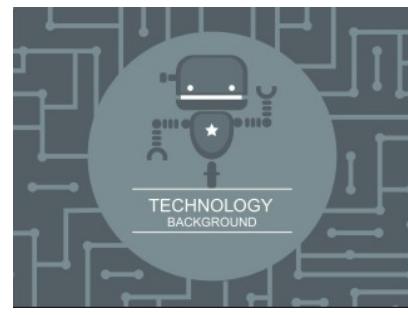
Change backgroundImage 0 ▾

Allows you to chose one of the six built in backgrounds that show behind the emoji. The six background images are shown below.

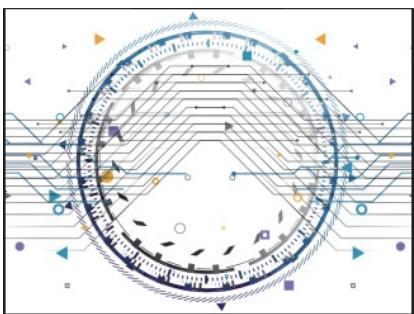
Background 0



Background 3



Background 1



Background 4



Background 2



Background 5



Timer,
Wait Seconds,



Delays a program from moving on to the next block by a user defined pause in seconds.
Wait Milliseconds,



Delays a program from moving on to the next block by a user defined pause in seconds.
Get Ticks ms



Returns ticks in milliseconds.

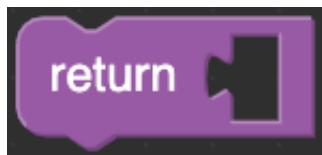
Function,
Do Something,



Creates a function with no output.
Do Something and Return Condition,



Creates a function with an output.
If Condition Return Condition,



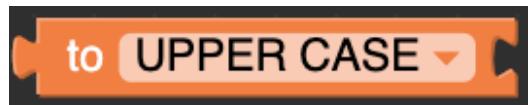
If true, returns the condition defined in the space.

Text,
Text block



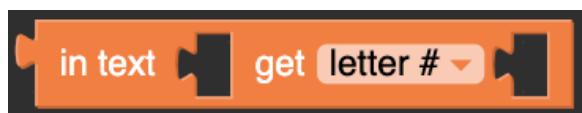
Used to display a letter, word or sting of text on the screen.

To UPPER Case



Converts the following sting to upper case.

In Text Get Letter



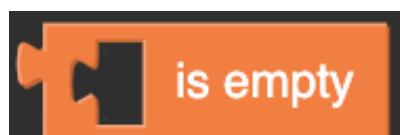
Returns one letter from the string.

Count In.



Returns how many time a string appears in another string.

Is Empty.



Returns true if string is empty.

Length Of,



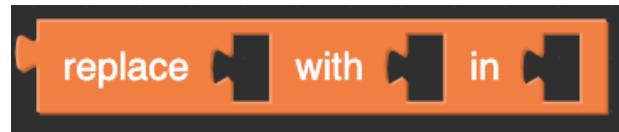
Returns the length (including spaces) of the string that is connected.

Print

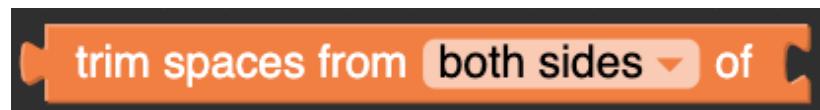


Prints the string or number connected to it on screen.

Replace With In



Finds and replaces all occurrences of a string in a string with another string.
Trim Spaces,



Returns the connected sting of txt with some or all of the spaces removed.
Prompt For Text With Message



Prints a message on screen prompting a user to insert some text.
Prompt For Text With Message ()



Prints a message on screen prompting a user to insert some text or value connected to it.
Convert To String,



Converts the following to a sting.
Message Plus



Used to add a value to the text for example batch file naming (ABCD + 001/002/003)
Decode

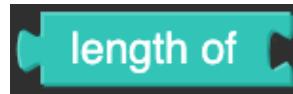


Used to decode a sting of text.
Encode,

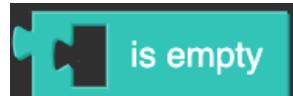


Used to encode a string of text.

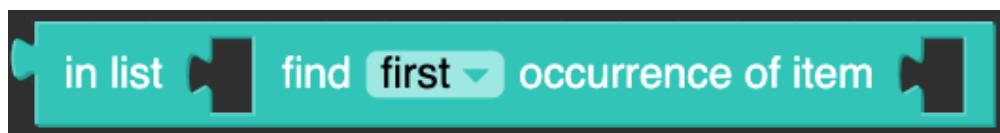
Lists,
Length Of,



Returns the length of a list.
Is Empty,



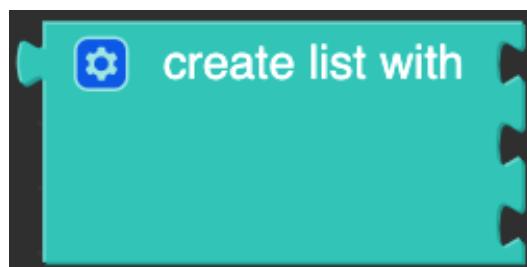
Returns true if list is empty.
In List Find Occurrence



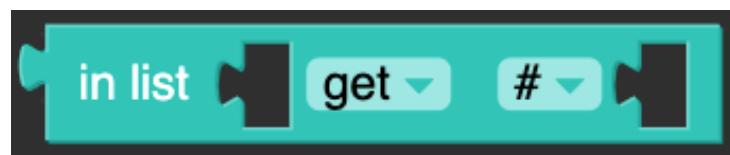
Returns index of first or last occurrence of a specified item or returns zero if an item is not found.
Create Empty List,



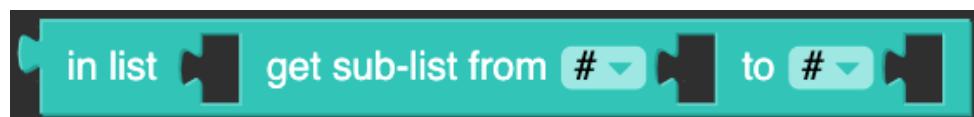
Creates an empty list.
Create List With,



Creates a list with specified items attached to it. Clicking the gear in the top left corner allows users to add additional items.
In List Get



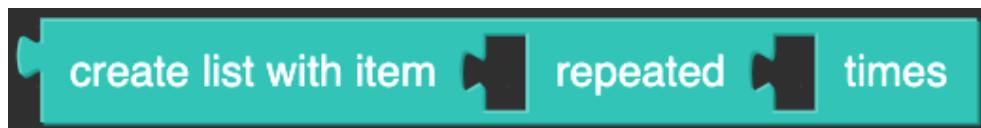
Returns the item in the specified position of a list.
In List Get Sub List,



91

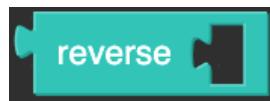
Creates a list from a specified section of another list.

Create List With Item Repeated.



Creates a list with a used defined item repeated a user defined amount of times.

Reverse,



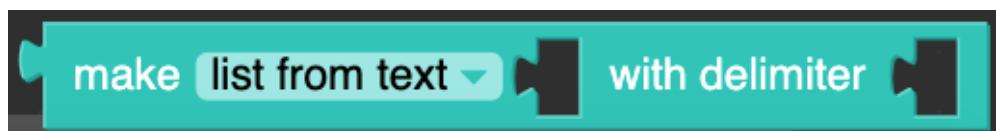
Reverses a copy of a list.

In List Set,



Allows users to define an item at a specific position in a list.

Make List From List With Delimiter.



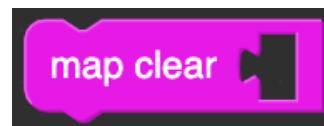
Allows users to split a list into smaller list using delimiters.

Map

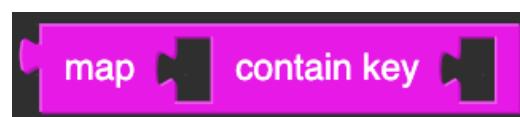
Create Map,



Map Clear,



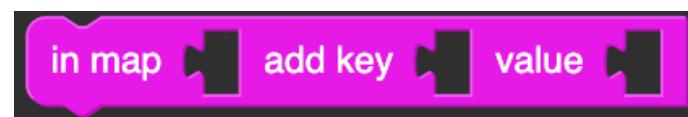
Map Contain Key,



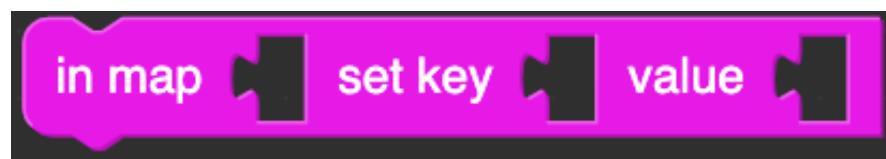
Get Key In Map,



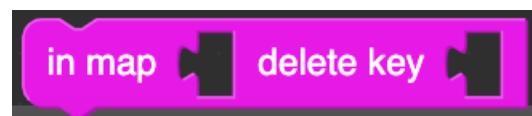
In Map Add Key



In Map Set Key,



In Map Delete Key,

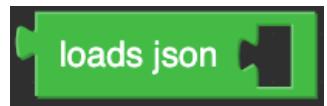


93

JSON,
Dump to JSON



Dumps a sting or value to json.
Load JSON



Used to load a JSON value or string.

Advanced,
Easy I/O,
Analog Read Pin,



Analogue Read Pin,



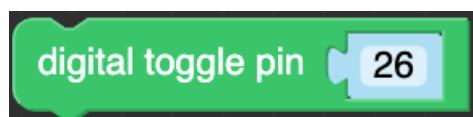
Digital Read,



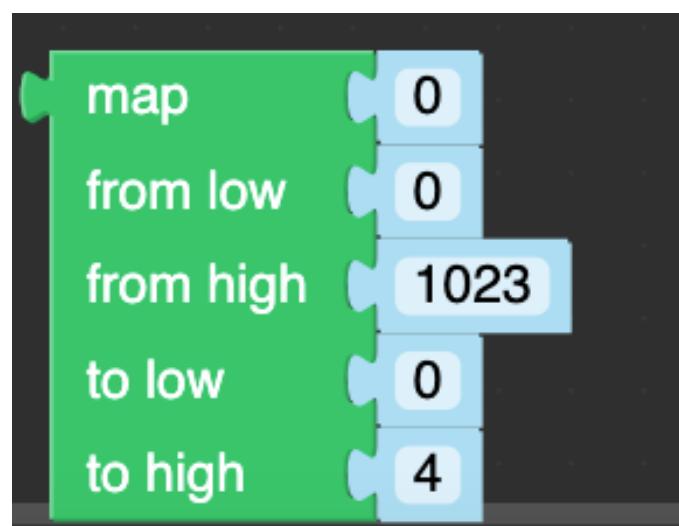
Digital Write Pin,



Digital Toggle Pin,

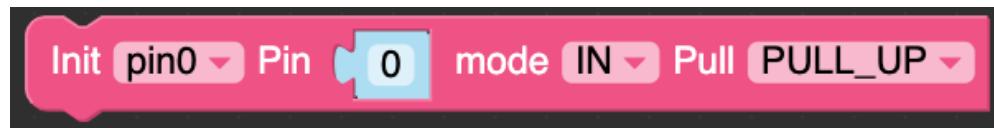


Map,



95

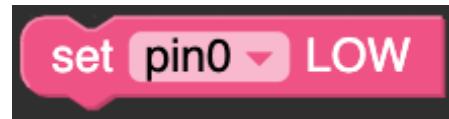
GPIO,
Init Pin Mode,



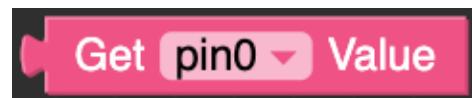
Set Pin High,



Set Pin Low,



Get Pin Value,



Set Pin Value,



PWM,
Init PWM of Pin,



Set PWM Frequency,



Set PWM Duty Cycle,



Pause PWM

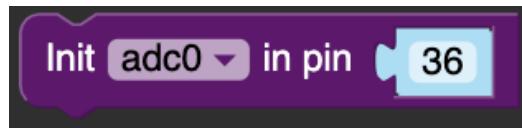


Resume PWM



97

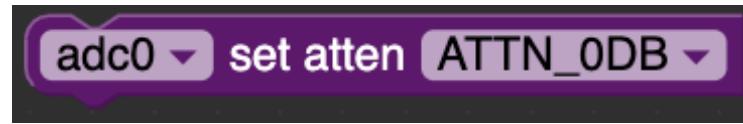
ADC,
Init ADC Pin,



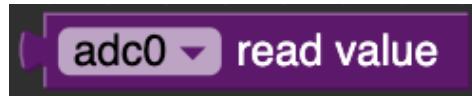
Set ADC Pin Bit Width,



Set ADC Atten,



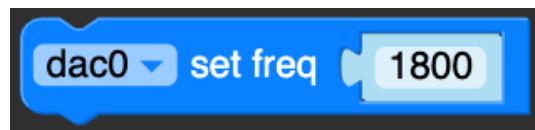
Read ADC,



DAC,
Init DAC



Set DAC Frequency,



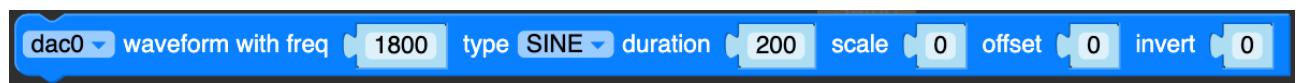
Write Value to DAC,



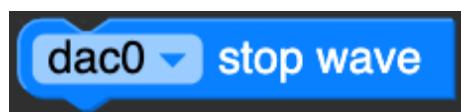
DAC Beep With Frequency,



DAC Waveform,



DAC Stop Wave



99

UART,
Set Uart,



Read Uart,



Read Uart (0) Characters,



Read a Line of Uart,



Get Remain Cache,



Write Value to UART



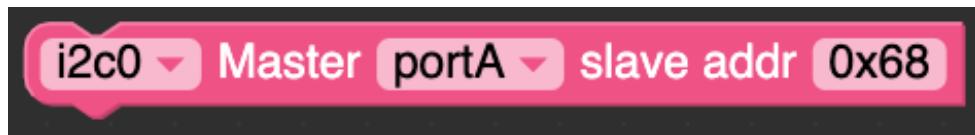
Write a Line to UART



Write String to UART,



I2C, Set I2C Port,



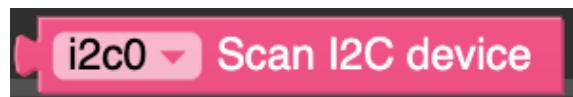
Starts communication with an I2C device with the address.

Set I2c SDA, SCL,



Sets the pins that the I2C port will use for SDA and SCL.

I2C Scan,



Returns a list of I2C addresses detected on the I2C bus. This block returns the addresses as binary instead of hexadecimal.

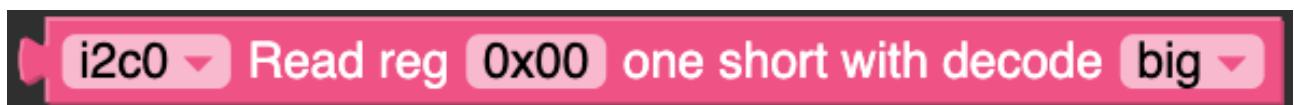
I2C Available Address in List,



I2C Read Reg One Byte,



I2C Read Reg One Short,



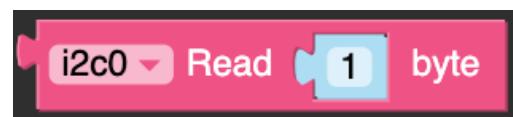
I2C Read Reg One Byte



I2C Write Reg One Byte,



I2C Read Byte,



I2C Write Reg One Short,



Execute,
Execute Code,



103

Network, WIFI Connect,



Wifi Reconnect,
Wifi is Connected,



Connect to Wifi SSID, Password,



P2P Send API Key,



P2P Read,



ESP NOW

Get MAC Address

Returns the M5Stack or sticks M.A.C address.

Add Peer

Adds a peer to the peer list using the M.A.C address as the individual identifier.

Set PMK

Set the Primary Master key that the M5Stack will use while communicating.

Broadcast Data

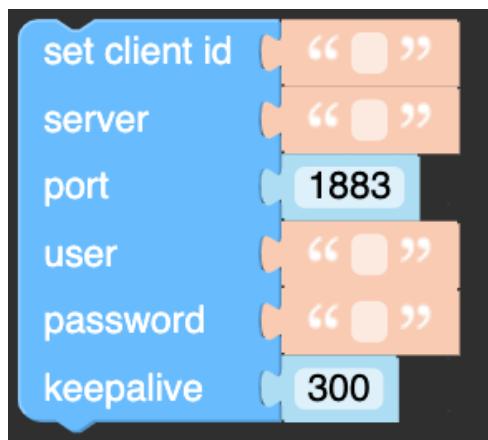
Transmits the specified data

Receive MAC Address Data

After Send Message Flag

Send Message ID Data

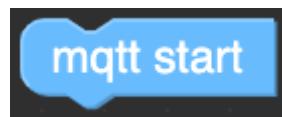
MQTT,
MQTT Setup,



This MQTT Setup block contains the credentials required to connect to a MQTT service.
MQTT Subscribe,



This loop subscribes to an MQTT topic and reruns the code inside keeping the M5Stack or Stick subscribed to the topic.
MQTT Start,



Starts the MQTT tasks.
Get Topic Data,



Returns information from the subscribed MQTT topic.
Publish Topic,



Publishes a message to the selected topic.

Remote,

Remote QRcode block display a QR code on the M5Stacks screen pointing to the remote control hosted page

Remote qrcode show in x 72 y 32 size 176

on flow.m5stack.com.



Add Remote Switch,

⚙ Add Remote Switch Button SwitchName

Add Remote Button,

Add Remote Button ButtonName

Creates an On/Off button

Add Remote Slider,

⚙ Add Remote Slider SliderName

Creates a slider which alters the value of a variable "X". This can be used as a brightness control or for controlling separate colour channels.

Add Remote Other,



Custom Blocks.

The custom block menu does not contain blocks by default but can be used to load any blocks created in the custom block editor.