M5Stack Documentation

Release 1.0.1

M5Stack

Oct 10, 2018

Contents

Welcome to M5Stack Documents!

Note: *The green arrows designate "more info" links leading to advanced sections about the described task.*

M5Stack is a series of stackable boards. We devote ourselves to develop this boards for lower development threshold and creating things more quickly.

Our a series of core boards are based on **ESP32** chip.

Our aim is "**Stackable Boards is Product**". Here's the Official Website

If you have any questions, you can contact us by Email or Forum. If you want to get our products, access our store please.

Product Documents	Get Started	API Reference
M5Stack Cases	M5Stack Awesome	FAQ

CHAPTER 1

Product Documents

1.1 M5Stack Core

BASIC	GRAY
FIRE	PANDA

1.2 Modules

1.2.1 M5Stack GPS Module

DESCRIPTION

The M5Stack GPS Module is a module with small GPS module. The small GPS module named UBLOX NEO-M8N. You can program it through Blockly, Arduino or MicroPython after connected to any series of M5Stack Core.

GPS module is built on the high performing u-blox M8 GNSS engine and exhibit high performance and high sensitivity.And it can supply your global positioning information even you in the wild and get lost.

FEATURES

- GPS NEO-M8N Module
- high-performance
- high-sensitivity
- Concurrent reception of up to 3 GNSS
- Industry leading –167 dBm navigation sensitivity

INCLUDES

- 1x M5Stack GPS Module
- 1x M5Stack Antenna

Applications

- Child positioning bracelet
- Logistics Tracking Management based on GPS

DOCUMENTS

- WebSite
- Example
  - Arduino Example
- GPS Info (GPS)
- GitHub
  - Arduino GitHub
- Purchase

1.2.2 M5Stack LORA Module

DESCRIPTION

The M5Stack LoRa Module is a module with small LoRa module named Ra-02. You can program it after connected to any series of M5Stack Core through Blockly, Arduino or MicroPython.

M5Stack LoRa Module can be used for ultra-long distance spread spectrum communication, and compatible FSK remote modulation and demodulation quickly, to solve the traditional wireless design can not take into account the distance, anti-interference and power consumption

FEATURES

- LoRa Module named RA-02 supply by Ai-Thinker
- Supports FSK, GFSK, MSK, GMSK, LoRa ™ and OOK modulation modes
- Receive sensitivity as low as -141 dBm
- Programmable bit rate up to 300Kbps
- Build-in Antenna

INCLUDES

- 1x M5Stack LoRa Module

Applications

- Automatic meter reading
- Home building automation
- Remote irrigation system

DOCUMENTS

- WebSite
- Example
- LoRa Info (LoRa)
- GitHub
- Purchase

NOTE

If your board LCD can't display or has some other problem, we suggest you to add the two statements code followed by `m5.begin();` as shown below

```
m5.begin();
pinMode(5,OUTPUT);
digitalWrite(5,HIGH);
```

Because GPIO5 who has connected NSS pin of LoRa module need be pull-up at the moment your board(or system) power on to prevent system's LCD can't display.

1.2.3 M5Stack LAN Module

DESCRIPTION

The M5Stack LAN Module is a ethernet module which with a I2C Grove port and 6 pins user ports.

When the speed of our wireless network is so slow that M5Stack Core can not normally connect to the wireless network. So we need a ethernet port board which is connected with M5Stack Core. This **LAN module** is that one who solves the awkward problem.

FEATURES

- W5500 - High-performance ethernet chip
- Less code and Hand-in-hand program experience

INCLUDES

- 1x M5Stack LAN Module(including USER-DEFINE 6 PIN and 1 Grove port)
- 1x DIN-RAIL
- 1x STICKER

Applications

- A mini portable Web server + TCP/UDP Server + HTTP Server
- A mini portable Web Clinet + TCP/UDP Clinet + HTTP Clinet
- SMTP/NTP Server

DOCUMENTS

- WebSite
- Example
  - Arduino Example
- LAN Info (LAN)
- GitHub
  - Arduino GitHub
- Purchase

1.2.4 M5Stack SIM800L Module

DESCRIPTION

The M5Stack SIM800L Module is a module with small SIM800L GSM/GPRS module. You can program it after connected to any series of M5Stack Core through Blockly, Arduino or MicroPython.

SIM800L is a complete Quad-band GSM/GPRS solution. SIM800L moudle could be connected with M5Stack Core via a serial port named USART2. Absolutely, you also can change the serial port number with jumper by your own.

FEATURES

- SIM800L Module
- Build-in Antenna
- 3.5 mm phone audio jack
- Microphone
- Parameter:
- GSM/GPRS
- support Quad-band 850/900/1800/1900MHz
- transmit Voice, SMS and data information with low power consumption
- Featur Bluetooth and Embedded AT

INCLUDES

- 1x M5Stack SIM800L Module

Applications

- Nitrogen dioxide alarm
- Automatic Web Spider SMS-notifier
- Remote meter reading system

DOCUMENTS

- WebSite
- Example
- SIM800L Info (SIM800L)
- GitHub
- Purchase

1.2.5 M5Stack BATTERY Module

DESCRIPTION

The M5Stack BATTERY Module is module with 850mAh High-Capacity Battery. User can create a portble device with any series of M5Stack Core and M5Stack BATTERY Module easily.

FEATURES

- 850mAh High-Capacity Battery

INCLUDES

- 1x M5Stack BATTERY Module

DOCUMENTS

- WebSite
- Purchase

1.2.6 M5Stack BTC Module

DESCRIPTION

The M5Stack BTC Module is a base including DHT12 module which can detect temperature and humidity. Your M5Stack Core board can stay as a small displayer(like a small TV or a small IOT central contronller) after adding this BTC Module. Absolutely, it is more easier to charge M5Stack Core via Type-C Cable after adding this BTC Module.

FEATURES

- DHT12 inside

INCLUDES

- Type-C USB Cable
- M3 x 16
- Tools

DOCUMENTS

- WebSite

1.2.7 M5Stack PROTO Kit

DESCRIPTION

The M5Stack PROTO Module is a flexible blank circle with 30 pins which could connect with all series of M5Stack Core. You can create any circle that could controlled by any M5Stack Core on M5Stack PROTO Module as you like.

FEATURES

- Flexible extended blank circle
- Compatible with all series of M5Stack Core
- A environment detector - DHT12 Module

Interface

LINE0	LINE1
GND	IO35(ADC1)
GND	IO36(ADC2)
GND	EN
IO23(MOSI)	IO25(DAC0)
IO19(MISO)	IO26(DAC1)
IO18(EXT_SCK)	3V3
IO3(U1_RX)	IO1(U1_TX)
IO16(U1_RX)	IO17(U2_TX)
IO21(I2C_SDA)	IO22(I2C_SCL)
IO2	IO5
IO12(I2S_SCLK)	IO13
IO15(I2S_OUT)	IO0
HPOWR	IO34
HPOWR	5V
HPOWR	BAT

INCLUDES

- 1x M5Stack PROTO Module
- 1x DHT12 Temperature & Humidity Sensor

- 1x Bus Socket
- 1x GROVE Cable
- 1x Packing Box
- User Manual

DOCUMENTS

- WebSite
- Example
- GitHub
- Purchase

1.2.8 M5Stack PROTO Module

DESCRIPTION

The M5Stack PROTO Module is a flexible blank circle with 30 pins which could connect with all series of M5Stack Core. You can create any circle that could controlled by any M5Stack Core on M5Stack PROTO Module as you like.

FEATURES

- Flexible extended blank circle
- Compatible with all series of M5Stack Core

Interface

LINE0	LINE1
GND	IO35(ADC1)
GND	IO36(ADC2)
GND	EN
IO23(MOSI)	IO25(DAC0)
IO19(MISO)	IO26(DAC1)
IO18(EXT_SCK)	3V3
IO3(U1_RX)	IO1(U1_TX)
IO16(U2_RX)	IO17(U2_TX)
IO21(I2C_SDA)	IO22(I2C_SCL)
IO2	IO5
IO12(I2S_SCLK)	IO13
IO15(I2S_OUT)	IO0
HPOWR	IO34
HPOWR	5V
HPOWR	BAT

INCLUDES

- 1x M5Stack PROTO Module
- User Manual

DOCUMENTS

- WebSite
- GitHub

1.3 BASE

1.3.1 M5Stack PLC Module

DESCRIPTION

The M5Stack PLC Module is a prototye industrial board, including RS484 adapter and electricity meter module.

With DC9~24V power input, PLC-Proto motherboard reserved 6Pin or 4Pin relay output, digital input, communication interface etc.

FEATURES

- Free DIY
- Programmable Logic Controller
- Individual package weight: 0.1kg (0.22lb.)
- Package size: 5cm * 5cm * 5cm (1.97in * 1.97in * 1.97in)

INCLUDES

- 1x PLC-Proto Broad
- 1x RS485 module
- 1x PLC Plastic Enclosure
- 1x Slide Guide
- 1x Magnet
- 1x 6 Pin 3.96 Pitch Terminal
- 1x 4 Pin 3.96 Pitch Terminal
- 3x Hex Key
- 7x Electrical Terminal
- 1x Sticker

Applications

- Programmable Logic Controller
- Programmable Motion Controller
- Digital Operation Processor
- Strong Electric Controller

DOCUMENTS

- WebSite
- GitHub
  - Arduino GitHub

1.4 ACCESSORY

1.4.1 Headers Socket

DESCRIPTION

1.5 UNIT

ENV	HUB	IR
PIR	POT	RGB
3.96PORT	EARTH	LIGHT
MAKEY	PROTO	RELAY
THERMAL		

1.6 APPLICATION

1.6.1 M5Stack BALA

DESCRIPTION

The M5Stack BALA is a balance bot based on M5Stack FIRE, including a 2 DC driver module based on Mega328p which is a core chip on Arduino UNO You can even program The M5Stack BALA through Arduino or MicroPython with few code

The 2 DC driver module communicates with M5Stack FIRE through I2C bus. It's default I2C address is **0x56**

FEATURES

- Programming Support
- Python
- Compatible LEGO
- POGO Pin
- TF Card Support

PARAMETER

Model	M5Stack FIRE
ESP32	240MHz dual core, 600 DMIPS, 4MB SRAM, Wi-Fi, dual mode Bluetooth
Flash	16M-Bytes
Input	5V @ 500mA
Interface	TypeC x 1, GROVE(I2C+I/0+UART), Pogo Pin x 1
LCD	2 inch, 320x240 Colorful TFT LCD, ILI9342
Speaker	1W-0928
Microphone	MEMS Analog BSE3729 Microphone
LED	SK6812 3535 RGB LED x 10
MEMS	MPU6050, MAG3110
Battery	550mAh @ 3.7V, inside
Op.Temp.	32°F to 104°F (0°C to 40°C)
Size	54 x 54 x 21 mm
C.A.S.E	Plastic (PC)
Weight	56g

INCLUDES

- 1x M5Stack BALA
- 1x Motor Driver
- 2x N20(Encoder included)
- Type-C USB Cable

DOCUMENTS

- Example
  - Arduino Example
  - MicroPython Example
- GitHub
- QuickStart
- Purchase

1.7 Tools

1.7.1 M5Stack USB Downloader

DESCRIPTION

1. USB to URAT Chip. CP2104 supports automatic firmware download of ESP32/ESP8266
2. TXD light, RXD light, Power light and 6pin @ 2.54mm bus sockets.

PARAMETER

PinNumber	PinName
1	GND
2	GPIO0
3	EN
4	TXD
5	RXD
6	3.3V

NOTE

There are two reserved pins(RTS, DTR) on M5Stack USB Downloader for other applications.

INCLUDES

- 1x M5Stack USB Downloader

DOCUMENTS

- Schematic
- UserGuide

CHAPTER 2

Get Started

This document is intended to help users set up the software environment for development of applications. Through a simple example we would like to illustrate how to develop M5Stack boards, firmware(*Arduino IDE*), Blockly or source files(*Micropython*) download to M5Stack boards.

2.1 Introduction

2.2 What You Need

To develop applications for M5Stack Core you need:

- **PC** loaded with either Windows, Linux or Mac operating system
- a **M5Stack Core** with Type-C cable

2.3 Your boards

Note: Make sure you have installed USB driver so that your board can establish serial connection with PC. If not, please view this article establish_serial_connection for connection.

At the first time, you need to burn the specific firmware file(.bin) to your board following this article How to burn firmware befor developing it.

If you have one of ESP32 development boards listed below, click the corresponding one to start your development.

		
M5Stack Core	M5GO	M5CAMERA
		
M5Bala	M5Stack STEPMOTOR	M5CAMERA

2.3.1 M5Stack Core Get Started(Blockly/MicroPython)

This article will guide you for getting started with Blockly(or MicroPython) through UIFlow.

Note: If your M5Stack Core was not burnt with a firmware in advance, please visit this article How to burn firmware for burnning.

First we need to know how to upload code onto the M5. After powering on Core and pressing the red button on the left hand side of the M5you will be greeted by this screen.

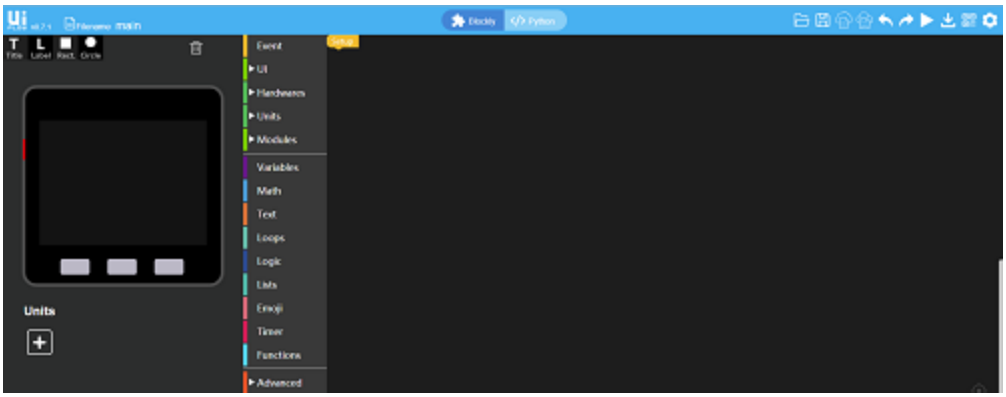
After pressing the upload button you will arrive at a screen with a QR code which you scan with your phone or tablet to start programming on your mobile device. If you want to program the M5 from your computer, enter the url shown at the top of the screen flow.m5stack.com

After a few seconds if nothing is pressed the M5 will automatically run the code that was previously uploaded. If we want to upload new code we have to make sure we press the A button which is the upload button on this menu before the M5 boots the code in it's memory.

Note: But if it's first time to use M5Stack Core or you want to change the networkable AP that means the Core can't access `flow.m5stack.com`, you need visit this article for setting wifi How to connect wifi using Core.

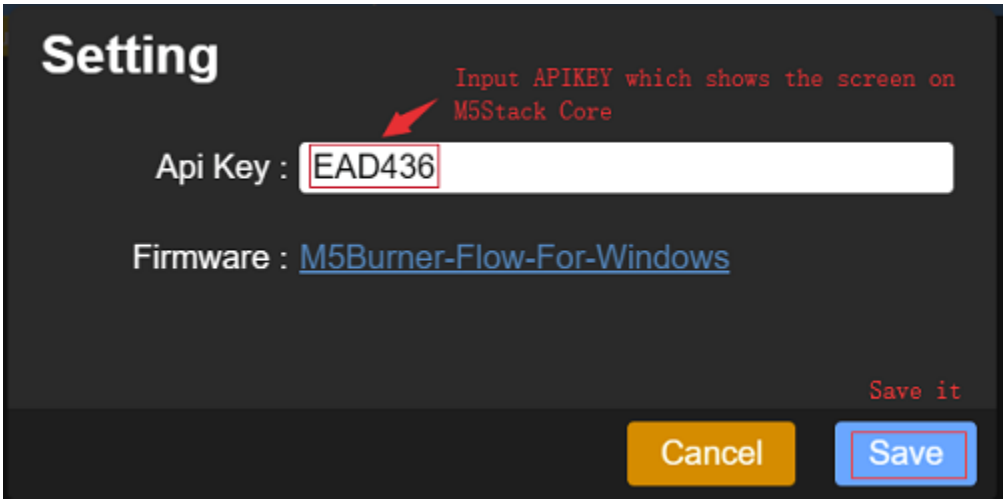
Program with Core

Visit the WebIDE, it will show as following figure.



Whenever we want to upload code to the M5 from UI flow we need to make sure the device is paired.

So press the little gear in the top right corner of the screen and enter the `APIKEY` which shows on the screen of M5(Now, my APIKEY is `9C6469`) and click `SAVE`.



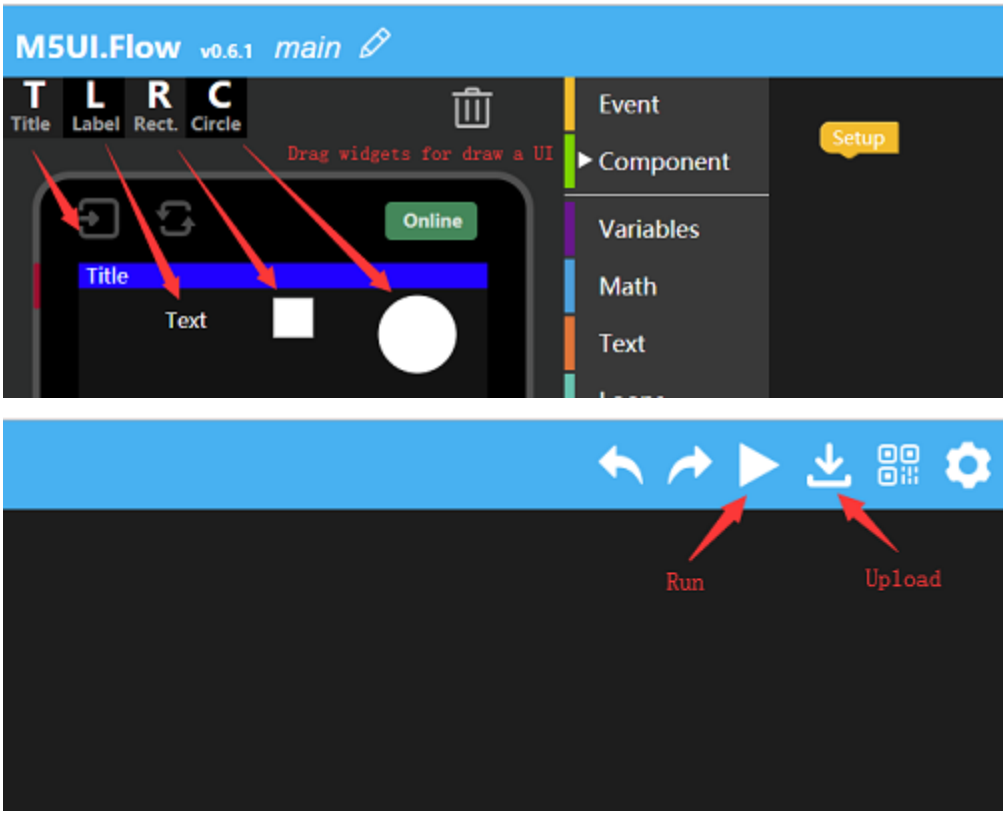
Then M5Flow will connect with this Core.

At the moment, you can draw a UI or program it through Blockly(or Python) as shown below.

Note: Once you've run another program

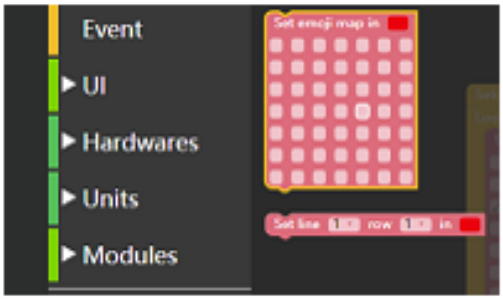
1. Draw a UI

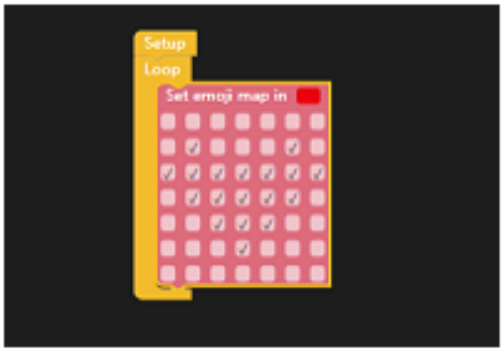
Drag 4 kinds of elements into `M5Stack Core` UI and click `Run` buttom on M5UI.Flow

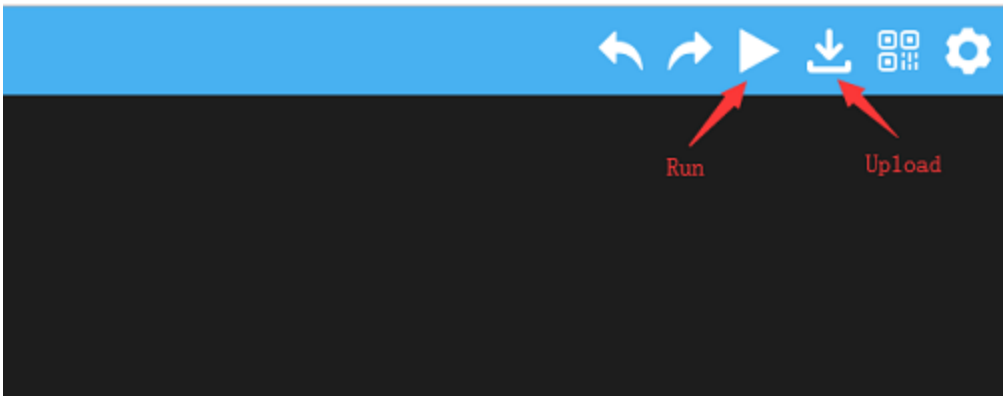


2. Program with Blockly

Drag some blocks named `Set emoji map in0` from `Emoji` class and click `Run` buttom on M5UI.Flow





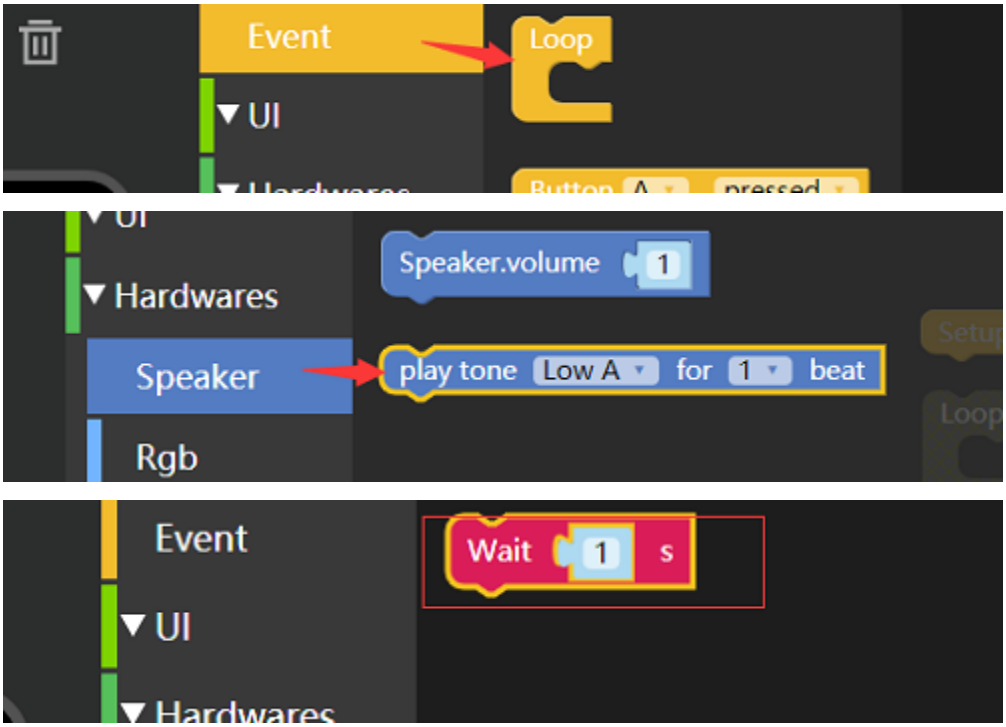


3. Program with MicroPython

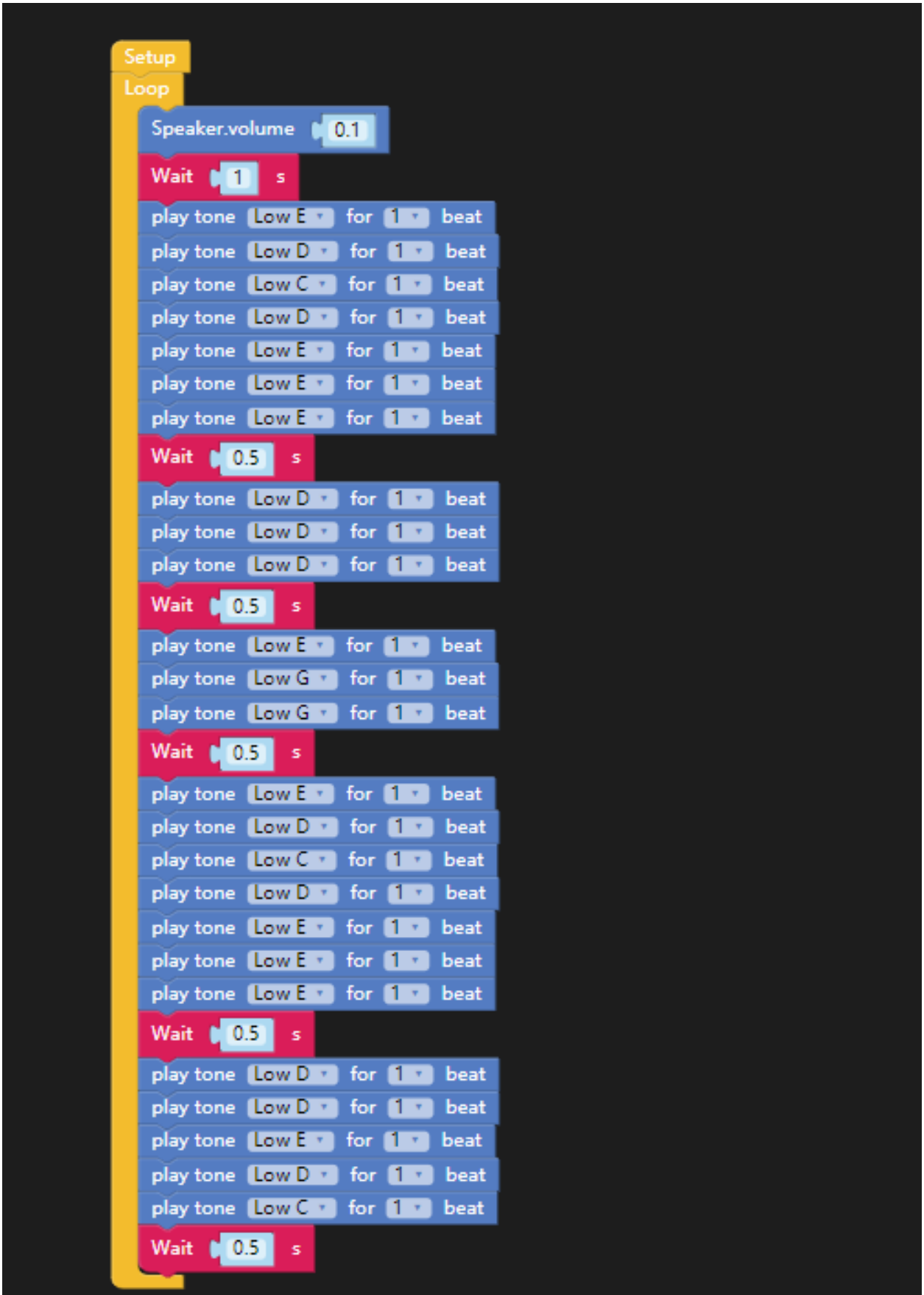
Play a song now

Now, let's make a music player and play a song in a few minutes using M5Stack Core.

Drag a loop, music and timer block into the coding area from the components section.



Then set parameters of `music block` and `timer block` as shown belown


Setup
Loop
Speaker.volume 0.1
Wait 1 s
play tone Low E for 1 beat
play tone Low D for 1 beat
play tone Low C for 1 beat
play tone Low D for 1 beat
play tone Low E for 1 beat
play tone Low E for 1 beat
play tone Low E for 1 beat
Wait 0.5 s
play tone Low D for 1 beat
play tone Low D for 1 beat
play tone Low D for 1 beat
Wait 0.5 s
play tone Low E for 1 beat
play tone Low G for 1 beat
play tone Low G for 1 beat
Wait 0.5 s
play tone Low E for 1 beat
play tone Low D for 1 beat
play tone Low C for 1 beat
play tone Low D for 1 beat
play tone Low E for 1 beat
play tone Low E for 1 beat
play tone Low E for 1 beat
Wait 0.5 s
play tone Low D for 1 beat
play tone Low D for 1 beat
play tone Low E for 1 beat
play tone Low D for 1 beat
play tone Low C for 1 beat
Wait 0.5 s

Now, run it and enjoy your musical work!

Note: Also, here some simple pratices and workshop for you being familiar with M5.

- Blockly

  https://m5stack.readthedocs.io/en/latest/get-started/practices_blockly.html

- MicroPython

  https://m5stack.readthedocs.io/en/latest/get-started/practices_micropython.html

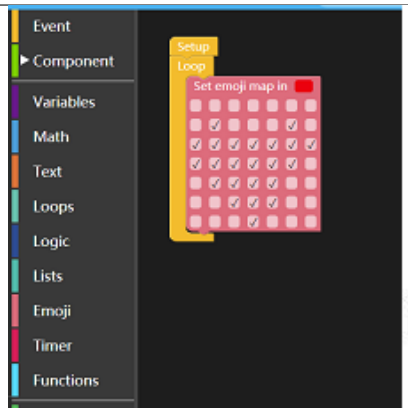
- WorkShop - if you want to participate our workshop, contact us through support email.

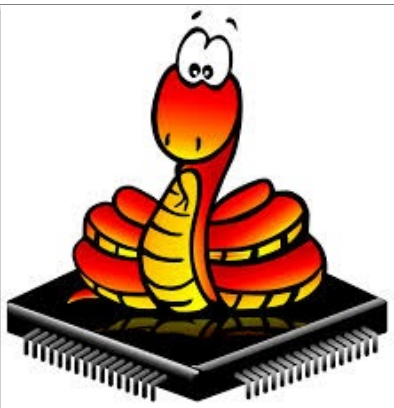
  support@m5stack.com

2.3.2 M5Stack Core Get Started

Pick up your programming mode below for getting started

Arduino	Blockly/MicroPython





2.3.3 ESP32CAM User Guide

1. Out-of-the-box Demo

It is really really out of the box. Your ESP32CAM will immediately run without any code after you power it.

1. plug usb cable into ESP32CAM and open the serial terminal on your computer.

```
][0;32mI (522) system_api: Base MAC address is not set, read default base MAC address from BLKD
][0;32mI (622) phy: phy_version: 3910, c0c45a3, May 21 2018, 18:07:06, 0, 0□[0m
][0;32mI (622) camera_demo: wifi_init_softap finished.SSID:M5Cam password:□[0m
][0;32mI (622) camera_demo: Open http://192.168.4.1/jpg for single image/jpg image□[0m
][0;32mI (632) camera_demo: Open http://192.168.4.1/jpg_stream for multipart/x-mixed-replace stre
][0;32mI (642) camera_demo: Free heap: 142128□[0m
][0;32mI (642) camera_demo: Camera demo ready□[0m
][0;32mI (64292) camera_demo: station:94:65:2d:8e:a1:08 join, AID=1□[0m
```

2. Then waitting a few seconds, you connect to a AP named "M5CAM" with your computer(or mobile phone).

3. And you open the browser on the computer(or mobile phone), enter a URL `http://192.168.4.1`. At the moment, your can see the real-time transmission of video by ESP32CAM on the browser.

Now, A WebCam you achieved successfully !

Note: ESP32CAM AP only can connect with one device at a time.

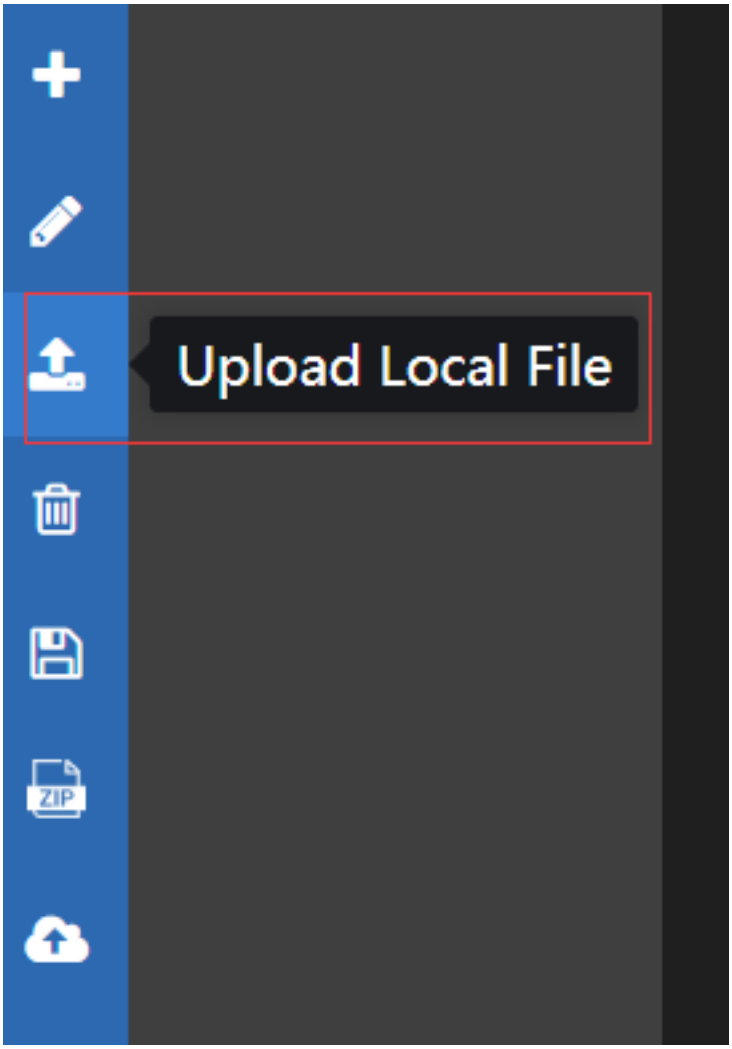
2.3.4 M5Stack StepMotor Module

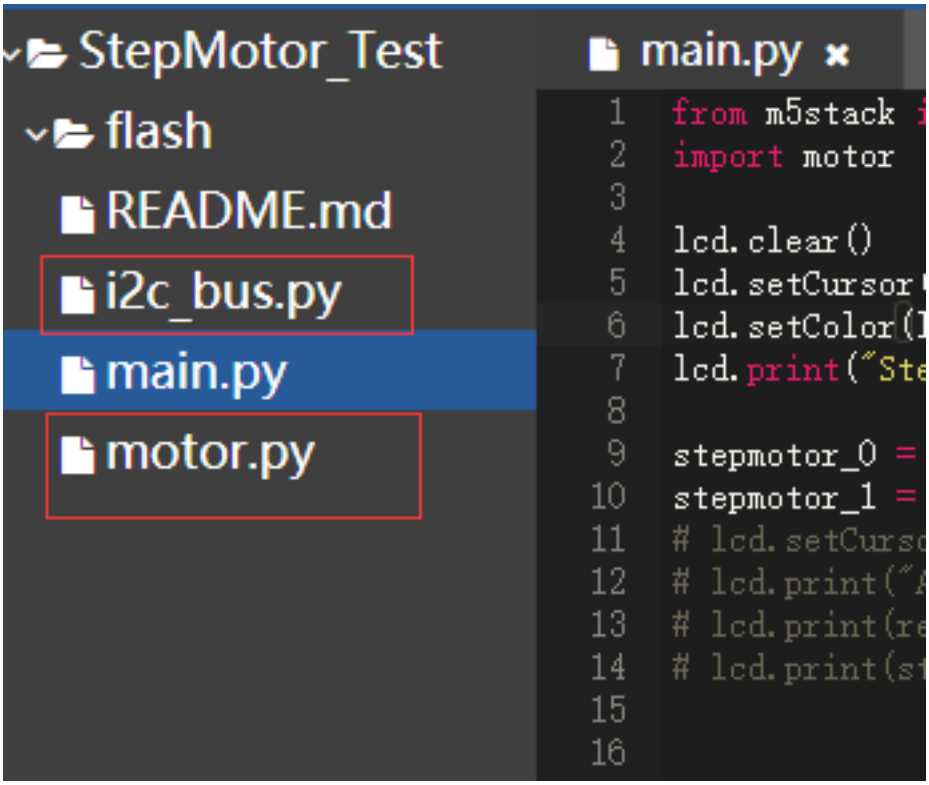
Now, we consident that you can program M5Stack Core with M5Cloud. If not, please read the article Getting Started with MicroPython

Quick Start

1. M5Stack Core connect with StepMotor Module

Click `Upload Local File` for adding two necessary files `motor.py`, `i2c_bus.py` as shown below


Upload Local File
ZIP



2. Copy the below code to `main.py`, upload all file and `run`

```
from m5stack import lcd
import motor
import utime

lcd.clear()
lcd.setCursor(0, 0)
lcd.setColor(lcd.WHITE)
lcd.print("StepMotor Test: ")

stepmotor_0 = motor.StepMotor(0x70)

stepmotor_0.StepMotor_XYZ(0, 0, 0, 500)
utime.sleep(3)
stepmotor_0.StepMotor_XYZ(10.5, 10.5, 10.5, 500)
utime.sleep(3)
stepmotor_0.StepMotor_XYZ(0, 0, 0, 500)
```

Note: When step motor is running, supply it with 12V power.*

2.3.5 M5Bala

M5Stack balance car

Quick Start

```
git clone https://github.com/m5stack/M5Bala.git
cd M5Bala
pio run
```

Installing and compiling the software

This project used Arduino framework develop, you must install the necessary tools and prepare the IDE environment. - Download (and unzip) this repository - Download and Install Visual Studio Code https://code.visualstudio.com/ - Install the PlatformIO Extension - Install M5Stack USB Driver - Install ESP32 Platform on PlatformIO - Open the M5Bala Project folder on PlatformIO - Build your project with *ctrl+alt+b* hotkey or using **Build** button on the PlatformIO Toolbar 

Dependent library

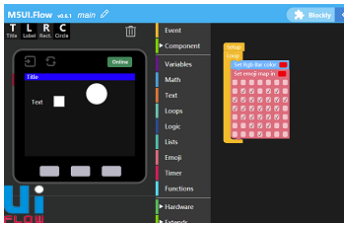
- M5Stack - https://github.com/m5stack/M5Stack
- MPU6050_tockn - https://github.com/tockn/MPU6050_tockn
- NeoPixelBus - https://github.com/Makuna/NeoPixelBus

MicroPython

- Examples

2.4 Which programming mode you like

For being familiar with the programming mode you lik, We suggest you following the corresponding option to do more practices.

		
Arduino	Blockly	MicroPython

2.5 Related Documents

2.5.1 Establish Serial Connection

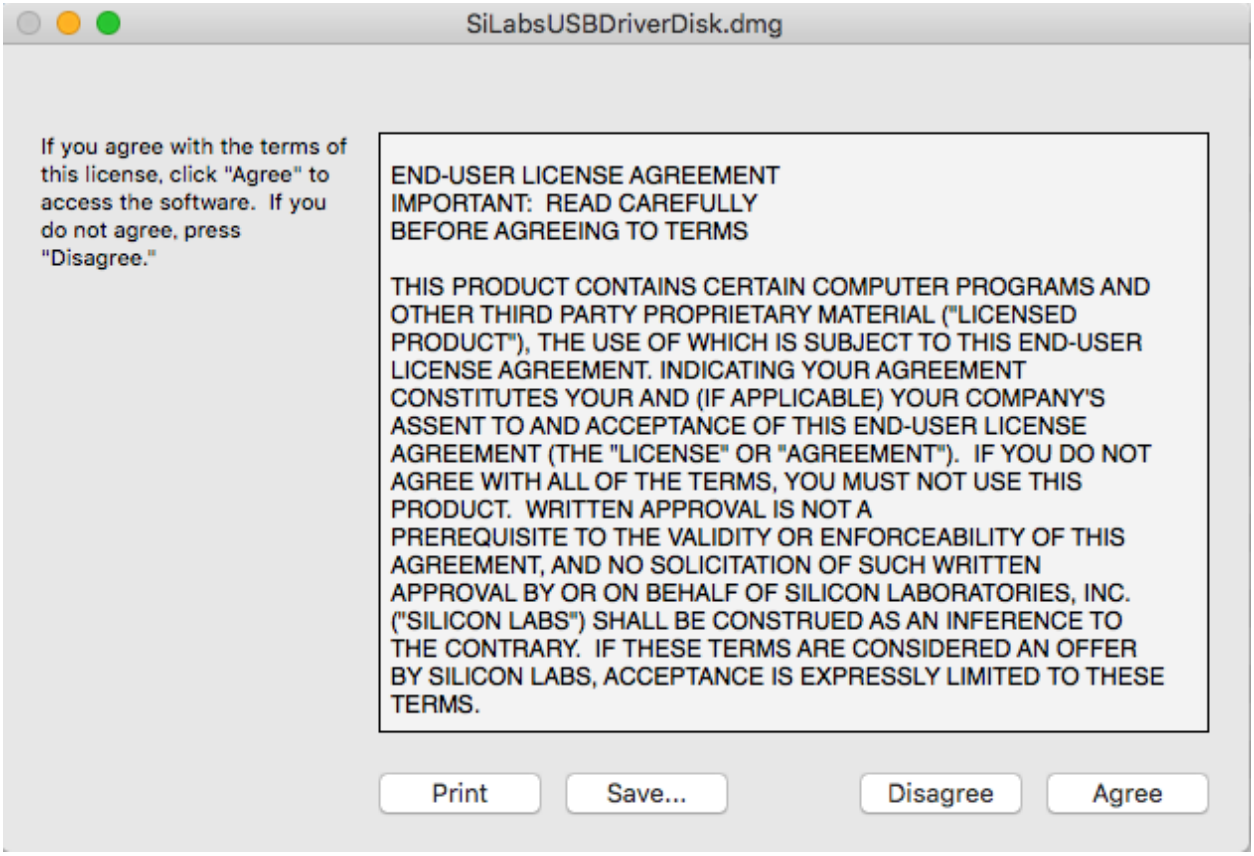
This section provides guidance how to establish serial connection between your board and PC.

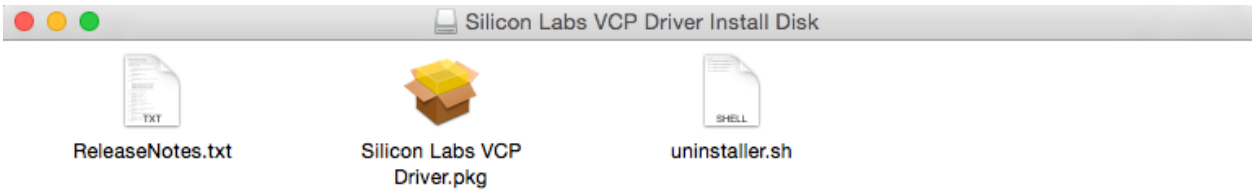
For MacOS

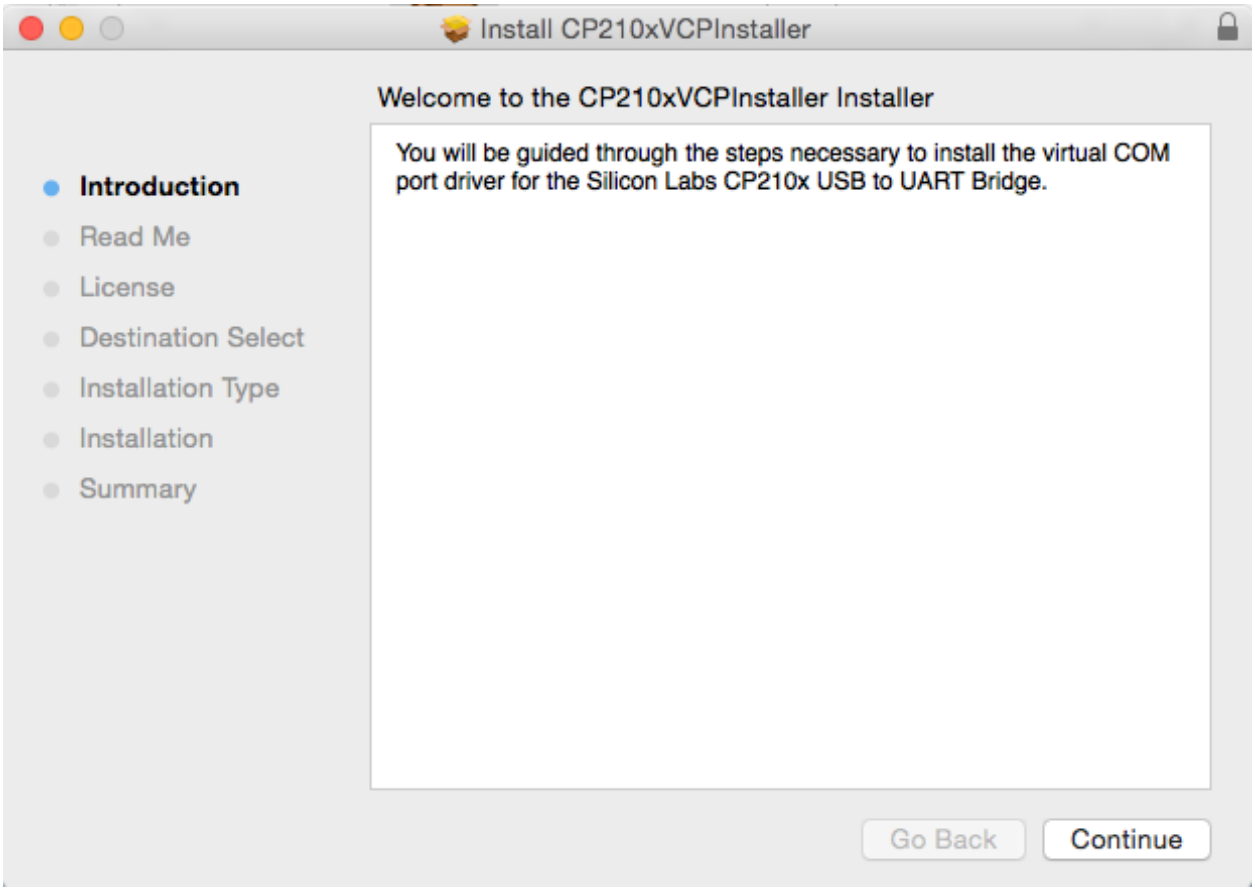
1. Install the USB driver

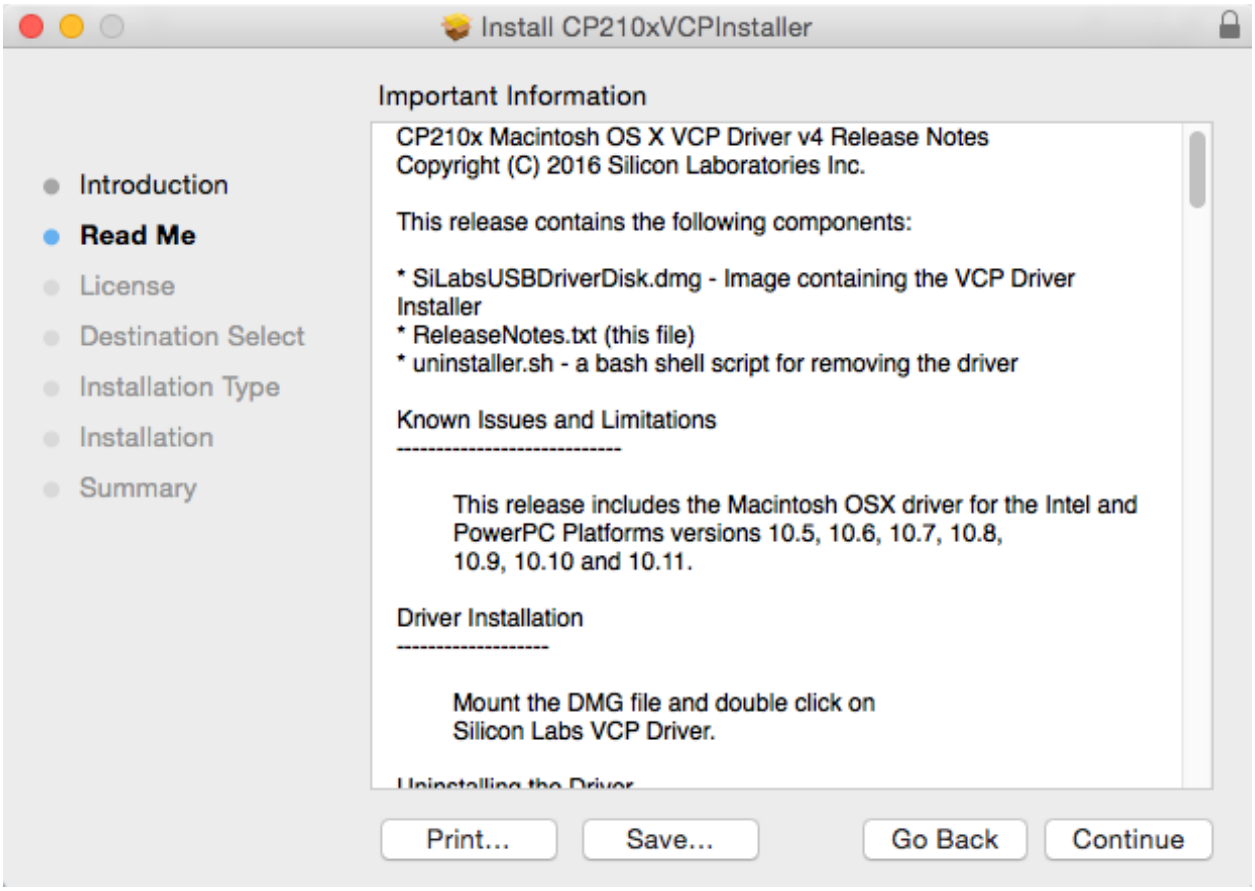
Download the SiLabs CP2104 Driver

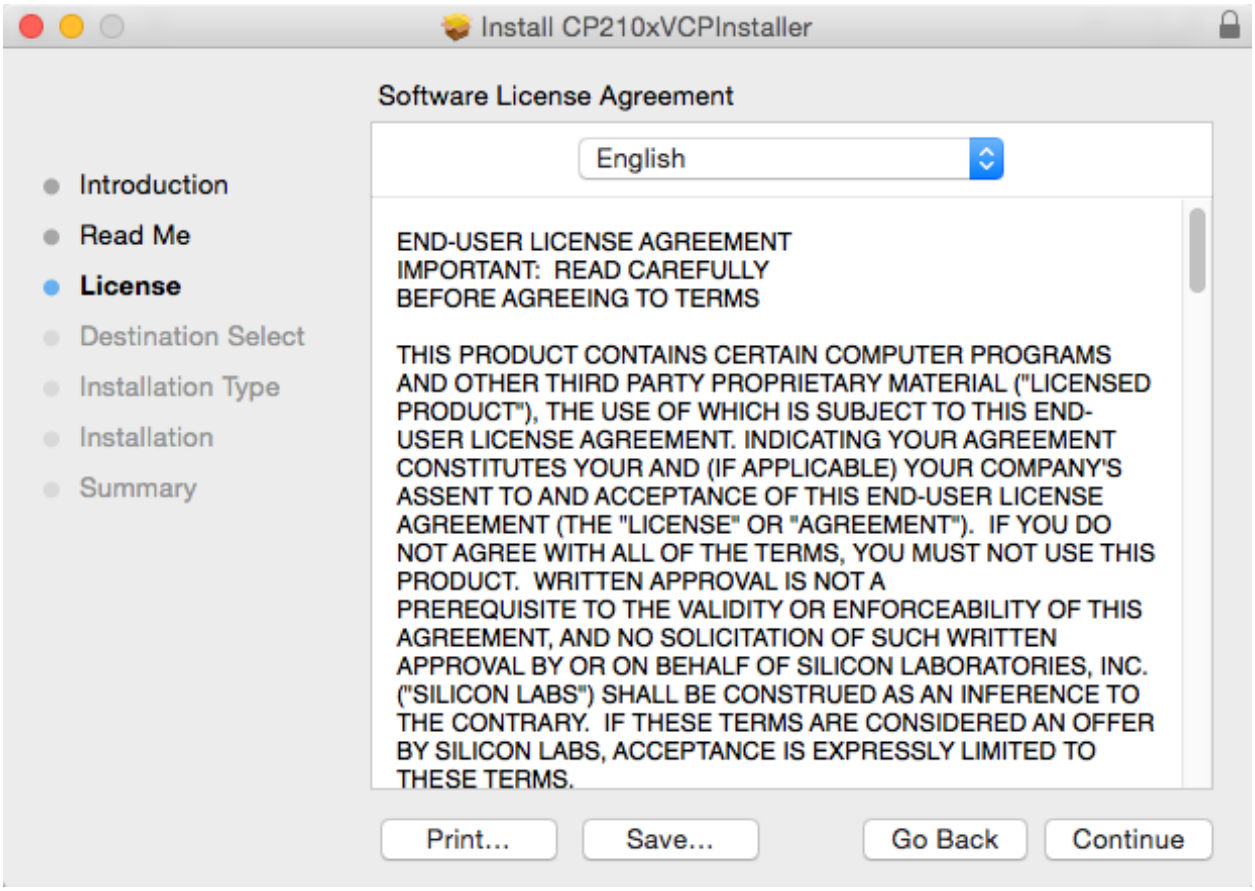
As the disk image SiLabsUSBDriverDisk.dmg is downloaded, mount it. Proceed according to the instructions OK.

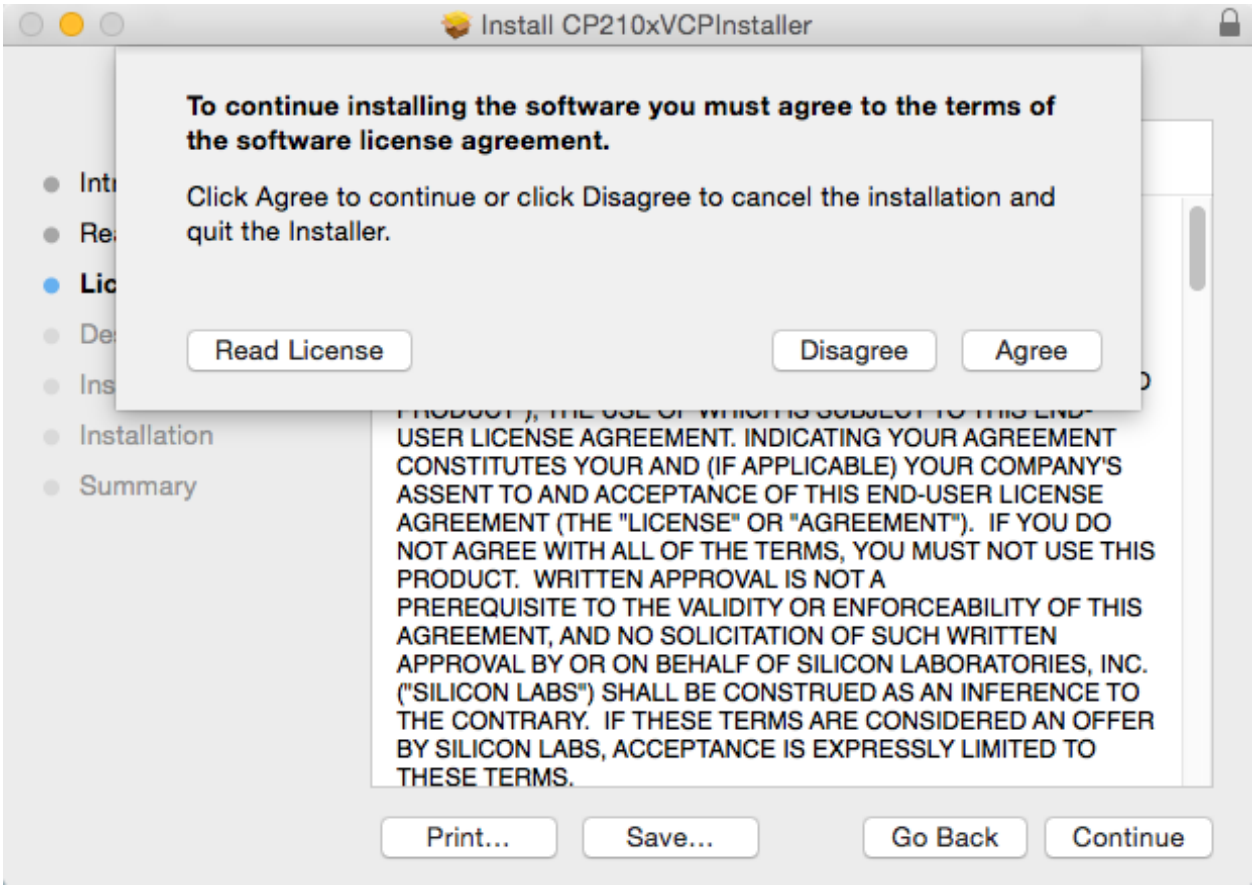


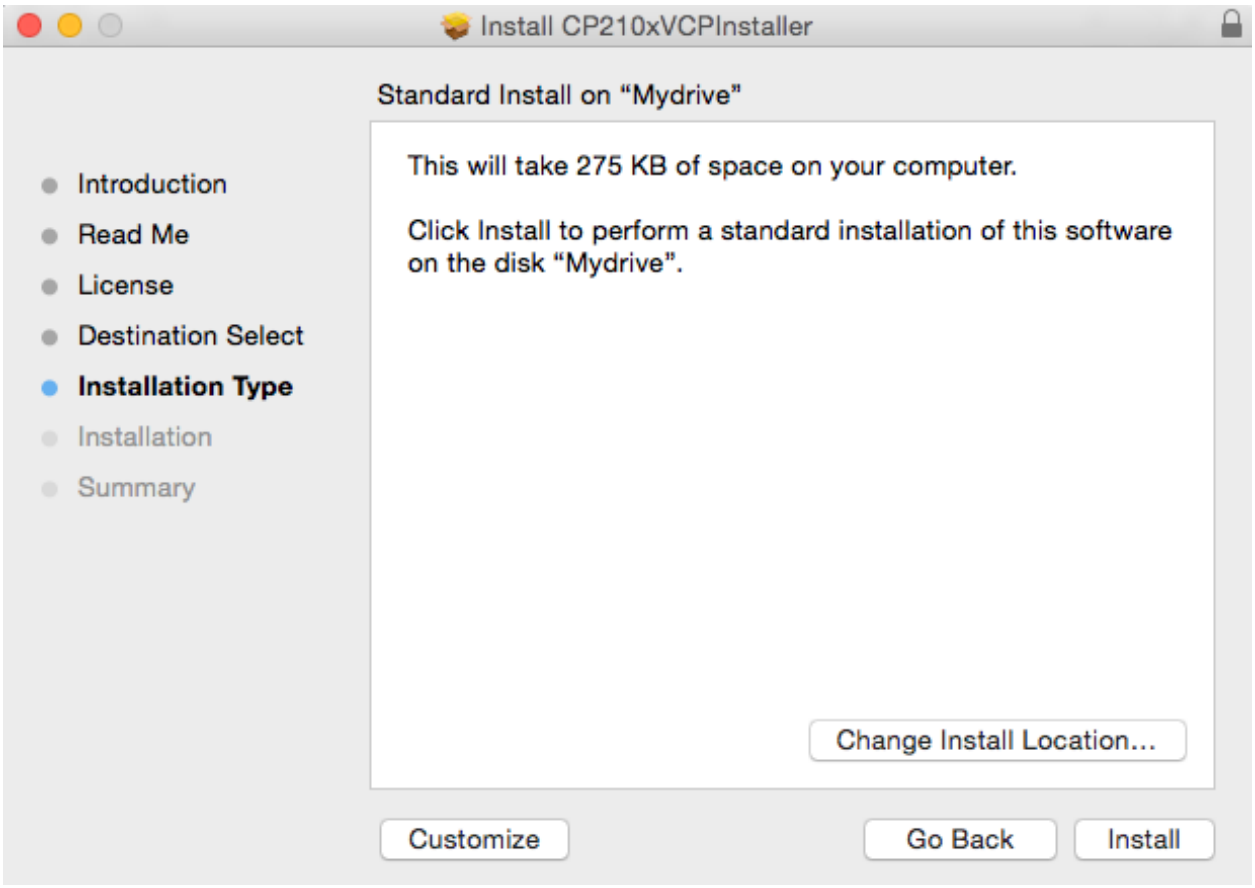

Silicon Labs VCP Driver Install Disk
TXT
ReleaseNotes.txt
Silicon Labs VCP Driver.pkg
SHELL
uninstaller.sh

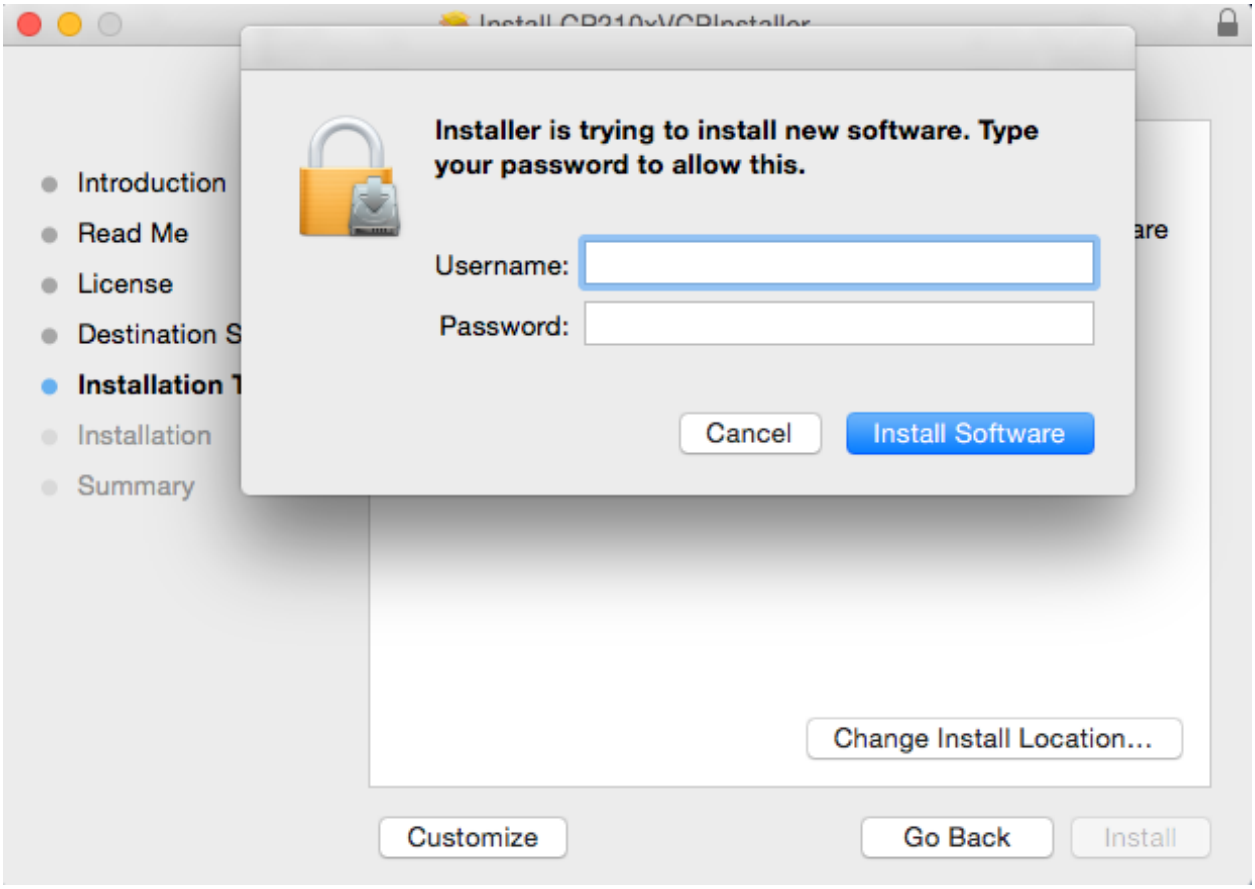

Install CP210xVCPInstaller
Welcome to the CP210xVCPInstaller Installer
Introduction
Read Me
License
Destination Select
Installation Type
Installation
Summary
You will be guided through the steps necessary to install the virtual COM port driver for the Silicon Labs CP210x USB to UART Bridge.
Go Back
Continue

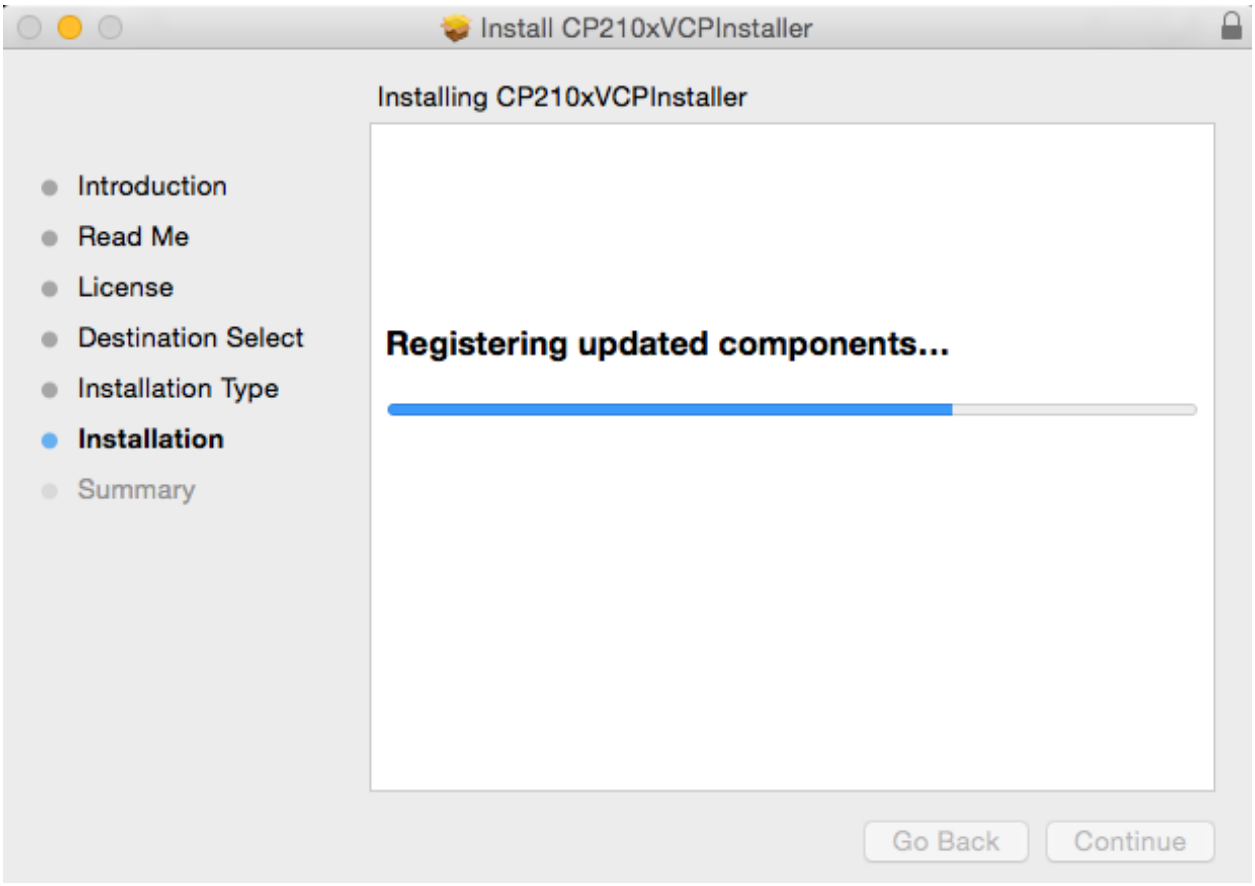

Install CP210xVCPInstaller
Introduction
Read Me
License
Destination Select
Installation Type
Installation
Summary
Important Information
CP210x Macintosh OS X VCP Driver v4 Release Notes
Copyright (C) 2016 Silicon Laboratories Inc.
This release contains the following components:
* SiLabsUSBDriverDisk.dmg - Image containing the VCP Driver Installer
* ReleaseNotes.txt (this file)
* uninstaller.sh - a bash shell script for removing the driver
Known Issues and Limitations
This release includes the Macintosh OSX driver for the Intel and PowerPC Platforms versions 10.5, 10.6, 10.7, 10.8, 10.9, 10.10 and 10.11.
Driver Installation
Mount the DMG file and double click on Silicon Labs VCP Driver.
Print...
Save...
Go Back
Continue

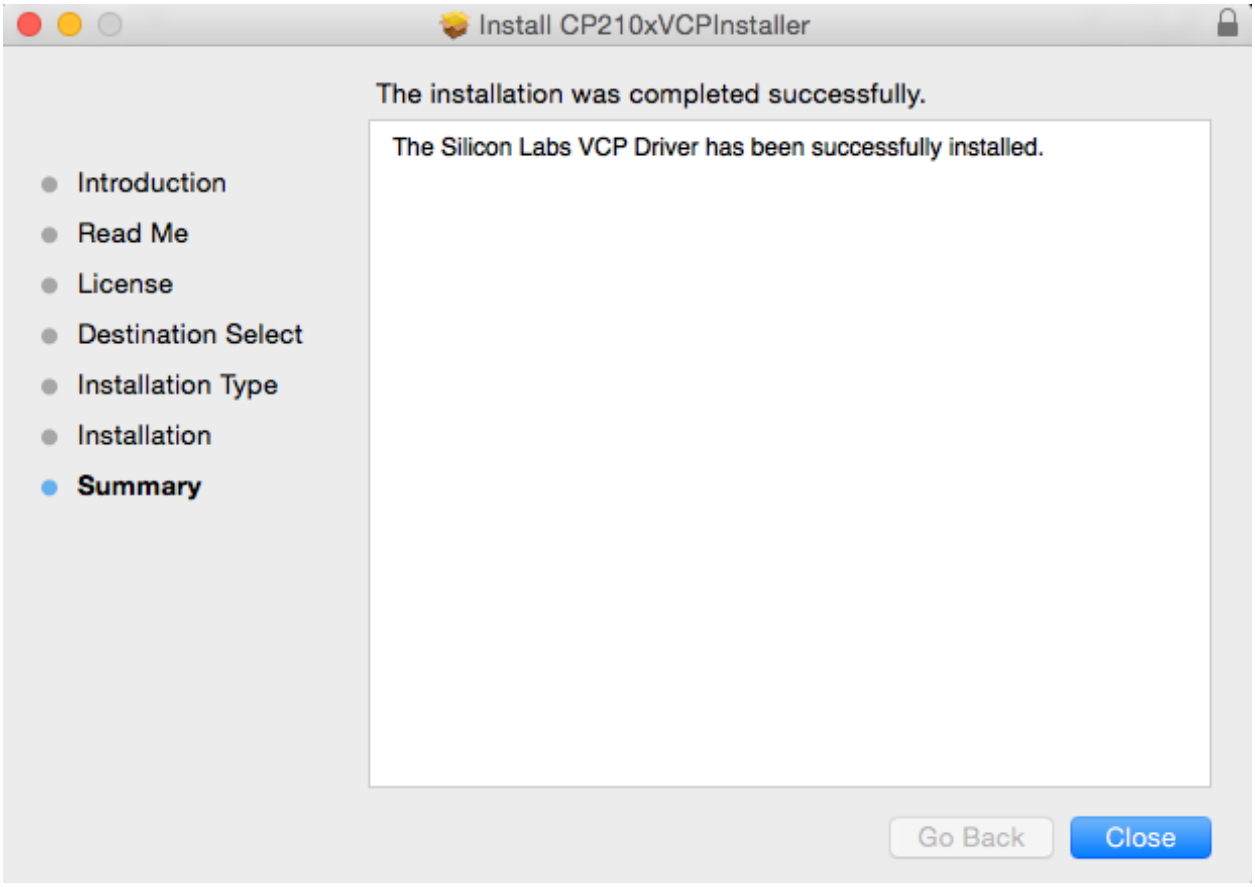

Install CP210xVCPInstaller
Software License Agreement
Introduction
Read Me
License
Destination Select
Installation Type
Installation
Summary
English
END-USER LICENSE AGREEMENT
IMPORTANT: READ CAREFULLY
BEFORE AGREEING TO TERMS
THIS PRODUCT CONTAINS CERTAIN COMPUTER PROGRAMS AND OTHER THIRD PARTY PROPRIETARY MATERIAL ("LICENSED PRODUCT"), THE USE OF WHICH IS SUBJECT TO THIS END-USER LICENSE AGREEMENT. INDICATING YOUR AGREEMENT CONSTITUTES YOUR AND (IF APPLICABLE) YOUR COMPANY'S ASSENT TO AND ACCEPTANCE OF THIS END-USER LICENSE AGREEMENT (THE "LICENSE" OR "AGREEMENT"). IF YOU DO NOT AGREE WITH ALL OF THE TERMS, YOU MUST NOT USE THIS PRODUCT. WRITTEN APPROVAL IS NOT A PREREQUISITE TO THE VALIDITY OR ENFORCEABILITY OF THIS AGREEMENT, AND NO SOLICITATION OF SUCH WRITTEN APPROVAL BY OR ON BEHALF OF SILICON LABORATORIES, INC. ("SILICON LABS") SHALL BE CONSTRUED AS AN INFERENCE TO THE CONTRARY. IF THESE TERMS ARE CONSIDERED AN OFFER BY SILICON LABS, ACCEPTANCE IS EXPRESSLY LIMITED TO THESE TERMS.
Print...
Save...
Go Back
Continue


Install CP210xVCPInstaller
To continue installing the software you must agree to the terms of the software license agreement.
Click Agree to continue or click Disagree to cancel the installation and quit the Installer.
Read License
Disagree
Agree
Installation
Summary
USER LICENSE AGREEMENT. INDICATING YOUR AGREEMENT CONSTITUTES YOUR AND (IF APPLICABLE) YOUR COMPANY'S ASSENT TO AND ACCEPTANCE OF THIS END-USER LICENSE AGREEMENT (THE "LICENSE" OR "AGREEMENT"). IF YOU DO NOT AGREE WITH ALL OF THE TERMS, YOU MUST NOT USE THIS PRODUCT. WRITTEN APPROVAL IS NOT A PREREQUISITE TO THE VALIDITY OR ENFORCEABILITY OF THIS AGREEMENT, AND NO SOLICITATION OF SUCH WRITTEN APPROVAL BY OR ON BEHALF OF SILICON LABORATORIES, INC. ("SILICON LABS") SHALL BE CONSTRUED AS AN INFERENCE TO THE CONTRARY. IF THESE TERMS ARE CONSIDERED AN OFFER BY SILICON LABS, ACCEPTANCE IS EXPRESSLY LIMITED TO THESE TERMS.
Print...
Save...
Go Back
Continue


Install CP210xVCPInstaller
Standard Install on “Mydrive”
Introduction
Read Me
License
Destination Select
Installation Type
Installation
Summary
This will take 275 KB of space on your computer.
Click Install to perform a standard installation of this software on the disk “Mydrive”.
Change Install Location...
Customize
Go Back
Install


Introduction
Read Me
License
Destination S
Installation T
Installation
Summary
Installer is trying to install new software. Type your password to allow this.
Username:
Password:
Cancel
Install Software
Change Install Location...
Customize
Go Back
Install


Install CP210xVCPInstaller
Installing CP210xVCPInstaller
Introduction
Read Me
License
Destination Select
Installation Type
Installation
Summary
Registering updated components...
Go Back
Continue



2. Check port on MacOS

To check the device name for the serial port of your your board board (or external converter dongle), open terminal and run this command two times, first with the board / dongle unplugged, then with plugged in. The port which appears the second time is the one you need:

MacOS

```
ls /dev/cu.*
```

For Windows

1.Install the USB driver

Download the SiLabs CP2104 Driver and choice the version of USB driver according to your windows version(Windows7/8/10).

Download Software

The CP210x Manufacturing DLL and Runtime DLL have been updated and must be used with v6.0 and later of the CP210x Windows VCP Driver. Application Note Software downloads affected are AN144SW.zip, AN205SW.zip and AN223SW.zip. If you are using a 5.x driver and need support you can download archived Application Note Software.

Legacy OS software and driver package download links and support information ›

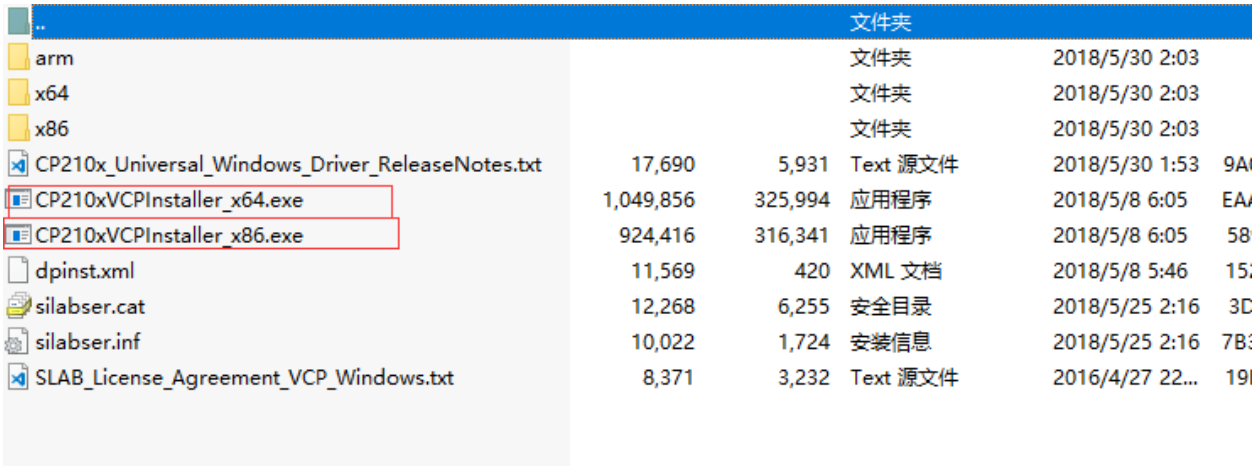
Download for Windows 10 Universal (v10.1.3)

Platform	Software	Release Notes
Windows 10 Universal	Download VCP (2.3 MB)	Download VCP Revision History

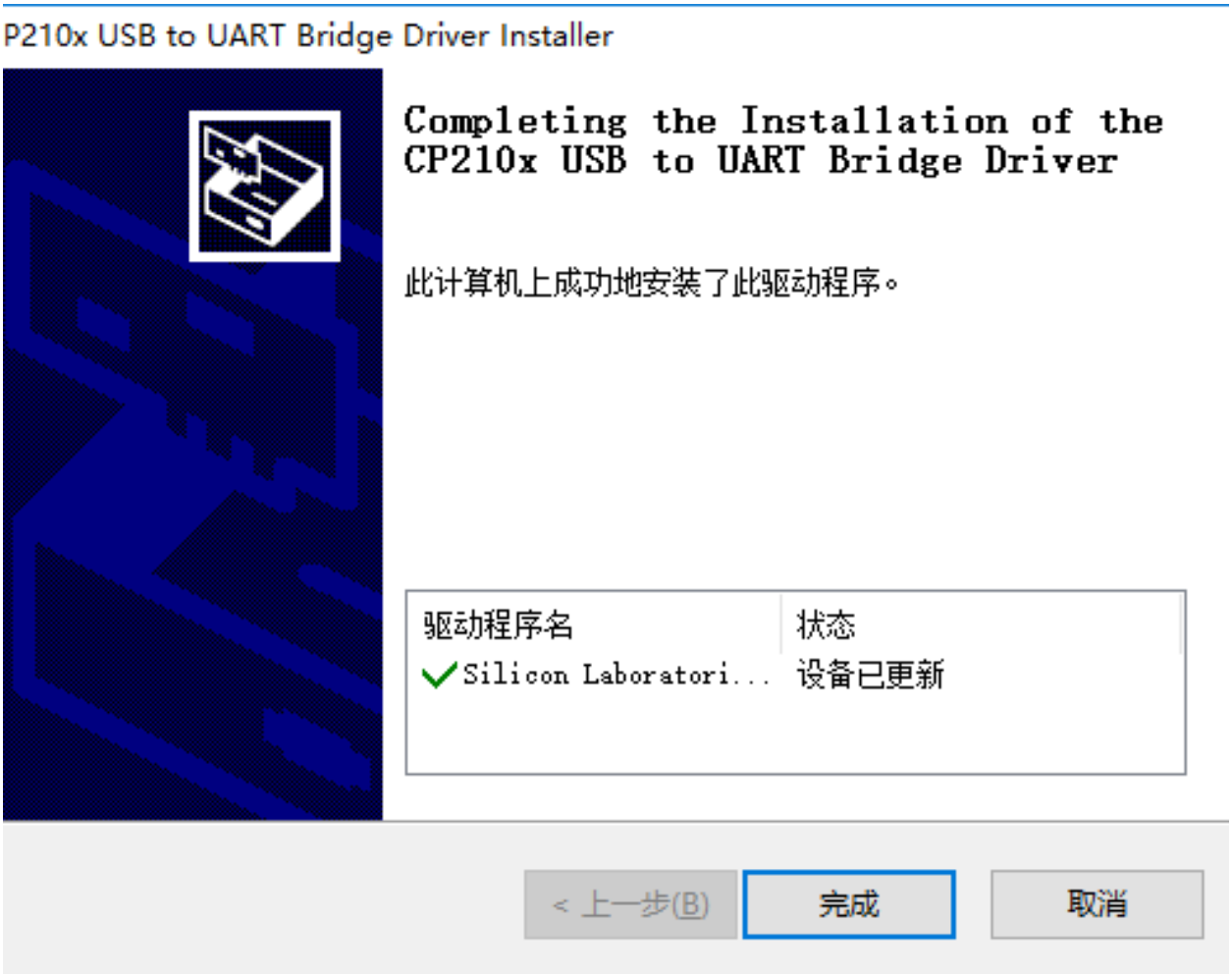
Download for Windows 7/8/8.1 (v6.7.6)

Platform	Software	Release Notes
Windows 7/8/8.1	Download VCP (5.3 MB) (Default)	Download VCP Revision History
Windows 7/8/8.1	Download VCP with Serial Enumeration (5.3 MB) Learn More »	Download VCP Revision History

Choice the right version installer(x64/x86), and install it.



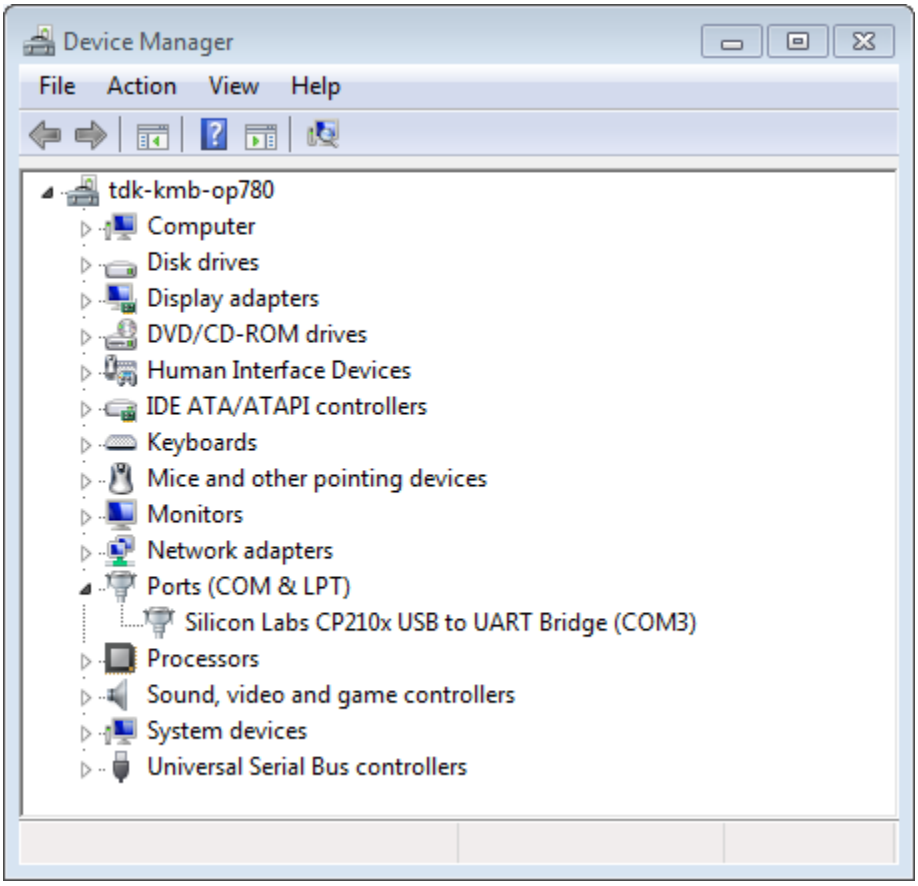

CP210x USB to UART Bridge Driver Installer
Welcome to the CP210x USB to UART Bridge Driver Installer
This wizard will help you install the drivers for your CP210x USB to UART Bridge device.
要继续，请单击“下一步”。
< 上一步(B)
下一步(N) >
取消



2. Check port on Windows

Check the list of identified COM ports in the Windows Device Manager. Disconnect your board and connect it back, to verify which port disappears from the list and then shows back again.

Figures below show serial port for M5Stack Core board



2.5.2 How to Burn Firmware

This article will guide you how to burn a right firmware to your board via M5Burner.

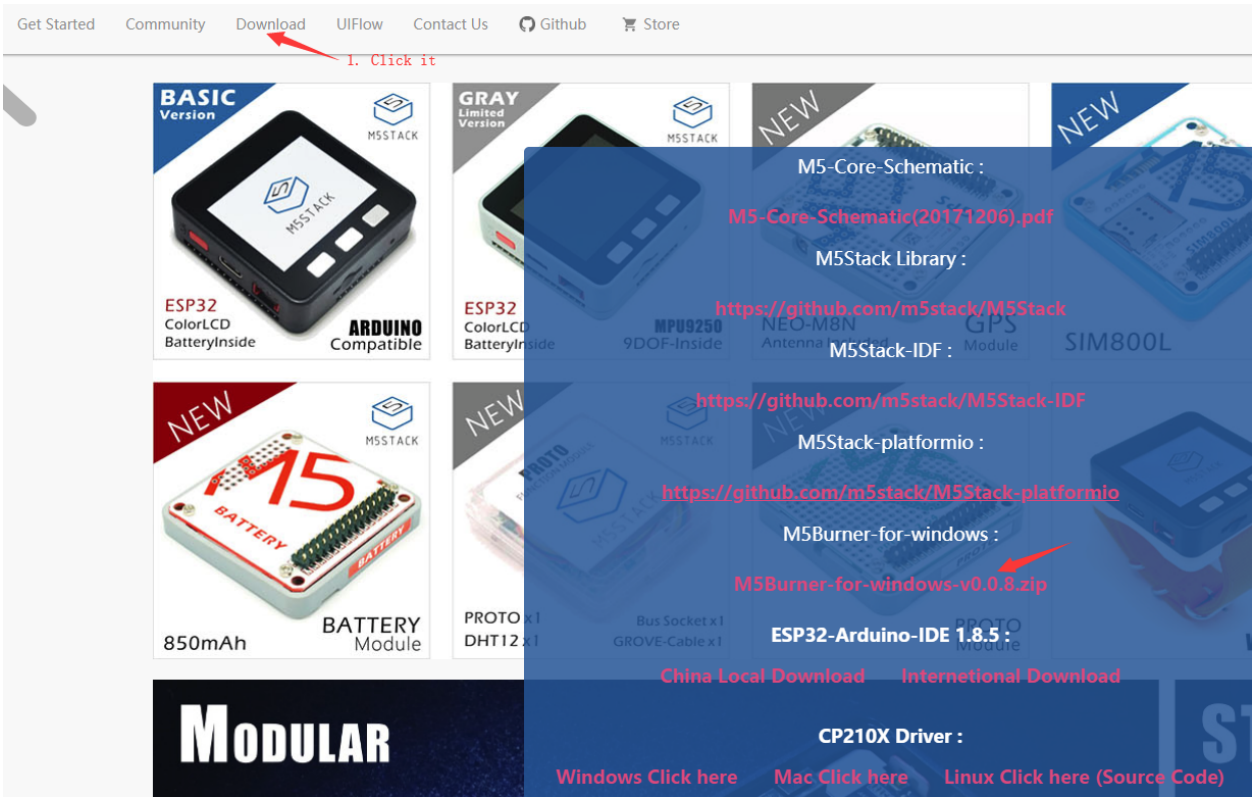
For MacOS

(Coming soon. . .)

For Windows

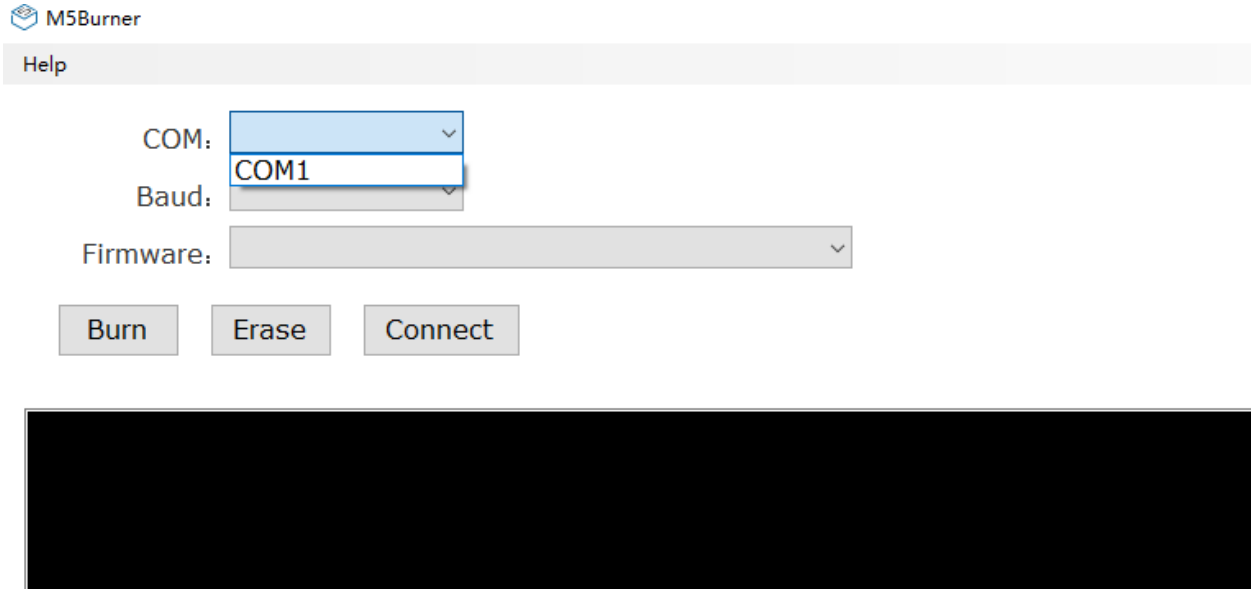
1. Download M5Burner

For downloading M5Burner, visit the offical website please.

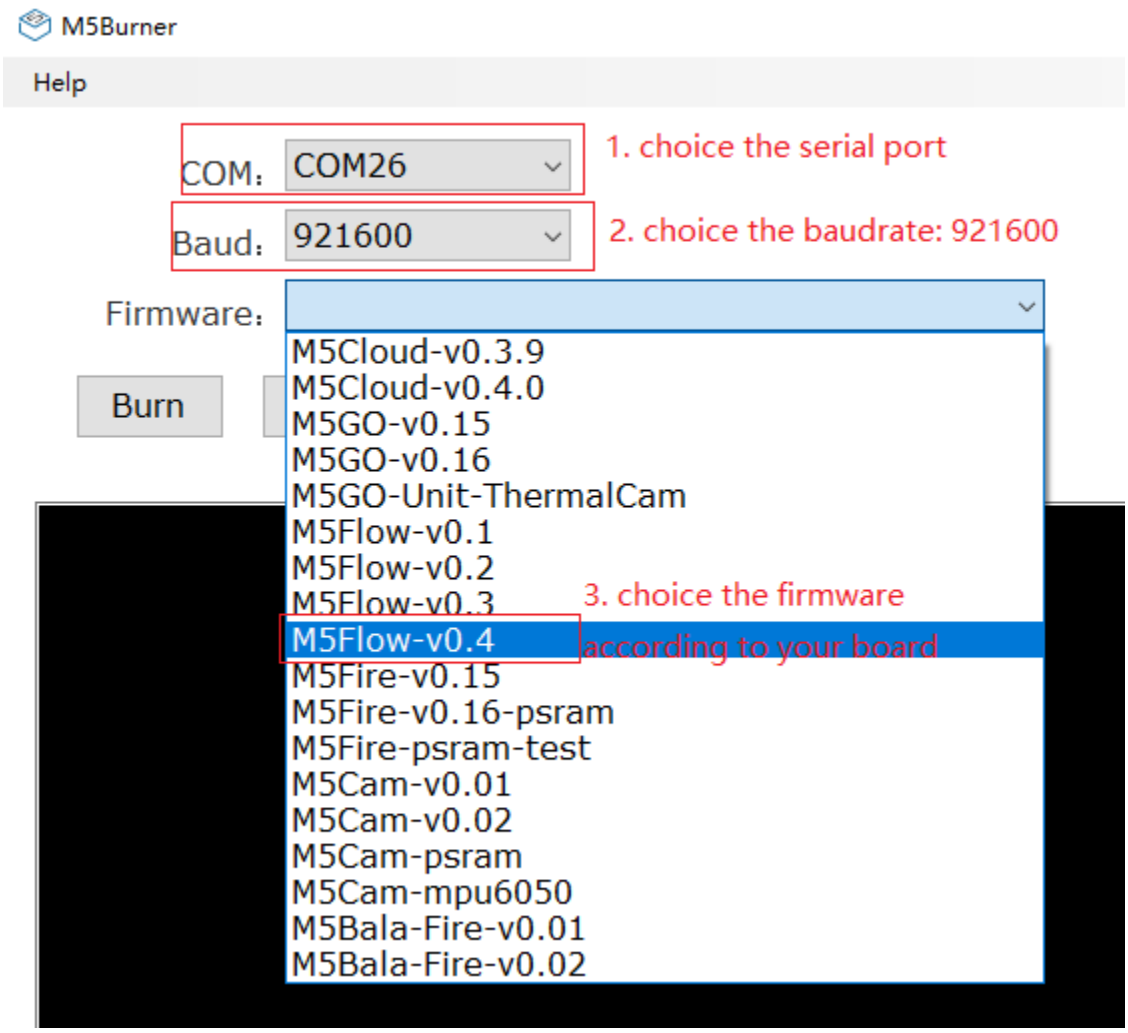


2. Burn the firmware

Unzip the M5Burner tool which you donwloaded for official website just now, then double click `M5Burner.exe`.



Then choice the `serial port` which is connected with your board and the `Baud` which is 921600 following below steps.

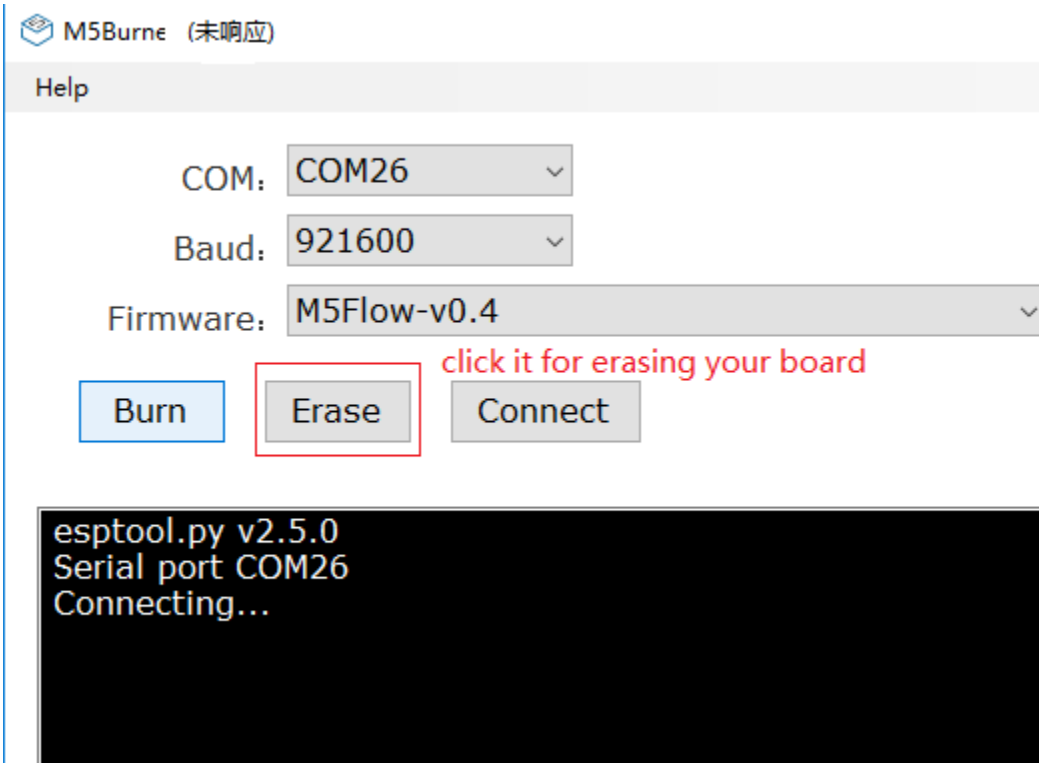


Note: If it does not display any `COMx` port or only `COM1` exists at the option, you need to visit this article establish_serial_connection and reinstall the USB driver.

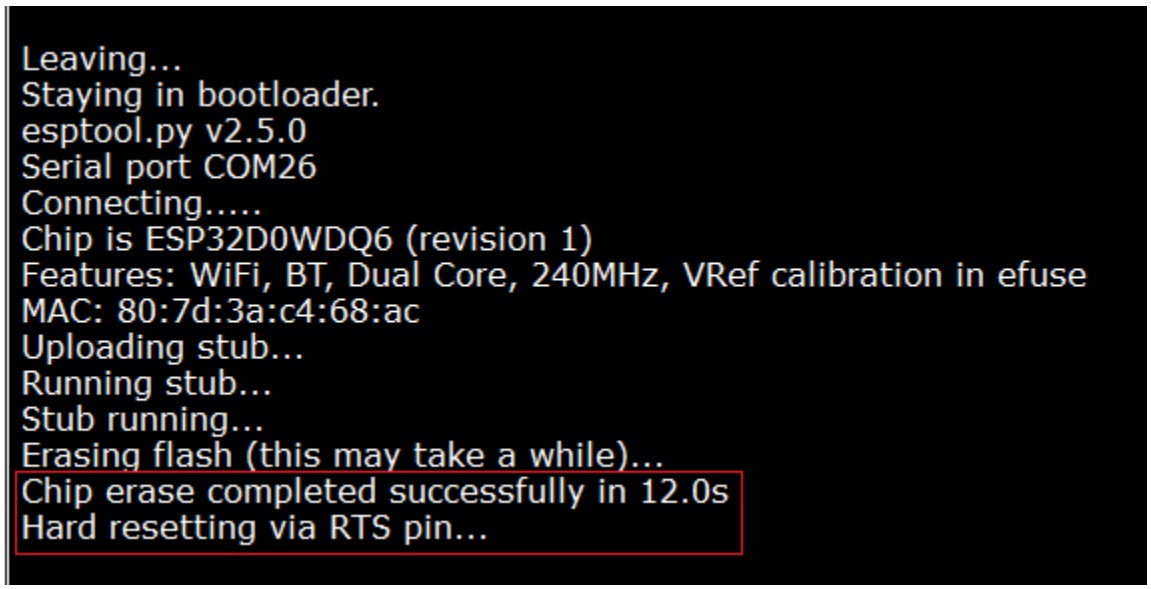
a. Choice a right firmware

1. select `M5Flow-vx.x` option(the lastest version), if you want to program with M5Flow
2. select `M5GO-vx.x` option(the lastest version), if you own a M5GO Kit
3. select `M5Cam-vx.x (/M5Cam-psram)` option, if you own a ESP32CAM (/ M5CAMERA)

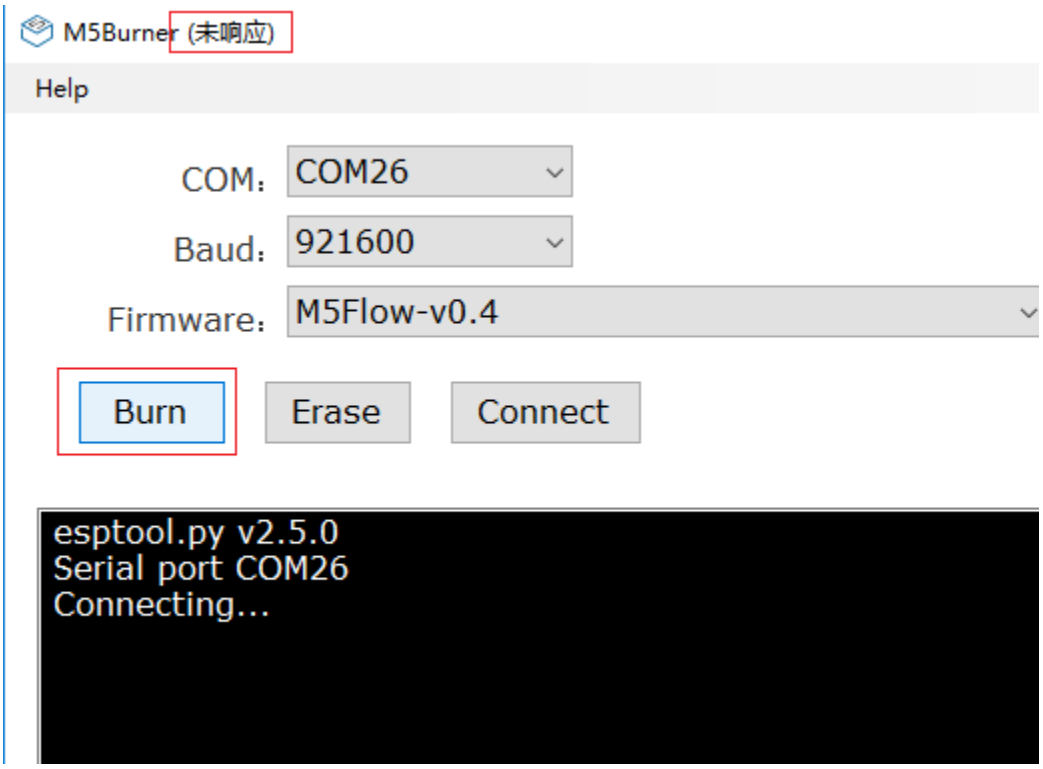
b. Click Erase



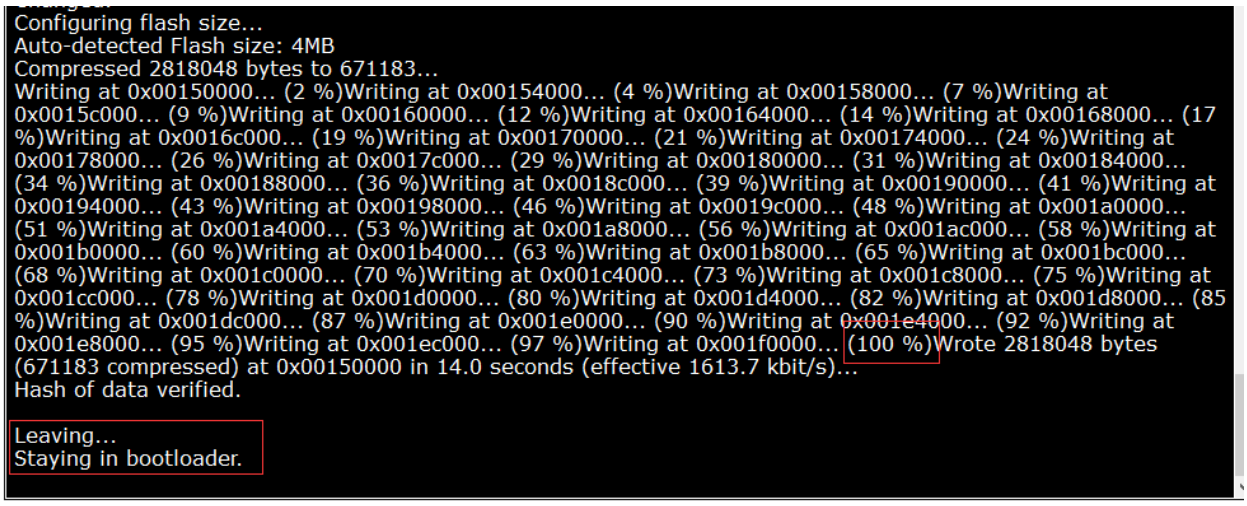
If M5Burner shows the information `Hard resetting via RTS pin...` below, it means chip has been erased successfully.



c. Click Burn



If M5Burner shows the information `Leaving... Staying in bootloader.` below, it means chip has been burnt successfully..



3. Reset your board

Note:

- If M5Burner means be busy after clicking `Burn`, please wait for a few minutes. It'll be normal after the firmware has been burnt successfully.

- If the burning procedure has been interrupted(like M5Burner has been closed suddenly...), it's better to burn your board again.

2.5.3 How to Connect WIFI Using Core

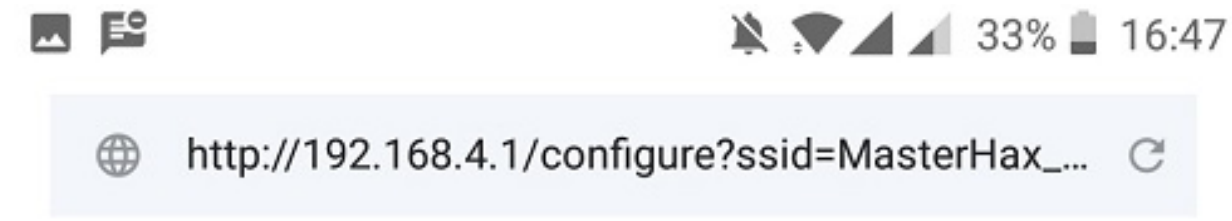
if it's first time to use M5Stack Core or you want to change the networkable AP that means the Core can't access http://flow.m5stack.com, please execute the following setps for setting wifi.

Connect Wi-Fi

Now, after you've pressed the upload buttom, the screen will show this message.

Then use Mobile Phone or PC for connectting to M5Stack AP(like `M5Stack-a67c`), and then open brower to login 192.168.4.1 for setting your networkable WIFI name and password. (Now, my networkable wifi is named `MasterHax_5G`)

After connecting wifi successfully, reset your core according to the prompt on `192.168.4.1`



Once you've reset M5Stack Core, the upload buttom, you will arrive at a screen with a QR code which you scan with your phone or tablet to start programming on your mobile device. If you want to program the M5 from your computer, enter the url shown at the top of the screen flow.m5stack.com

Note: Similarly, if you want another networkable WIFI AP to connect with Core, press the SETUP buttom while core was power-up.

2.5.4 Upgrade M5Stack Lib

1. Start up Arduino IDE, then Select `Sketch` -> `Include Library` -> `Manage Libraries...`

2. Type `M5Stack` into the search box, search it.

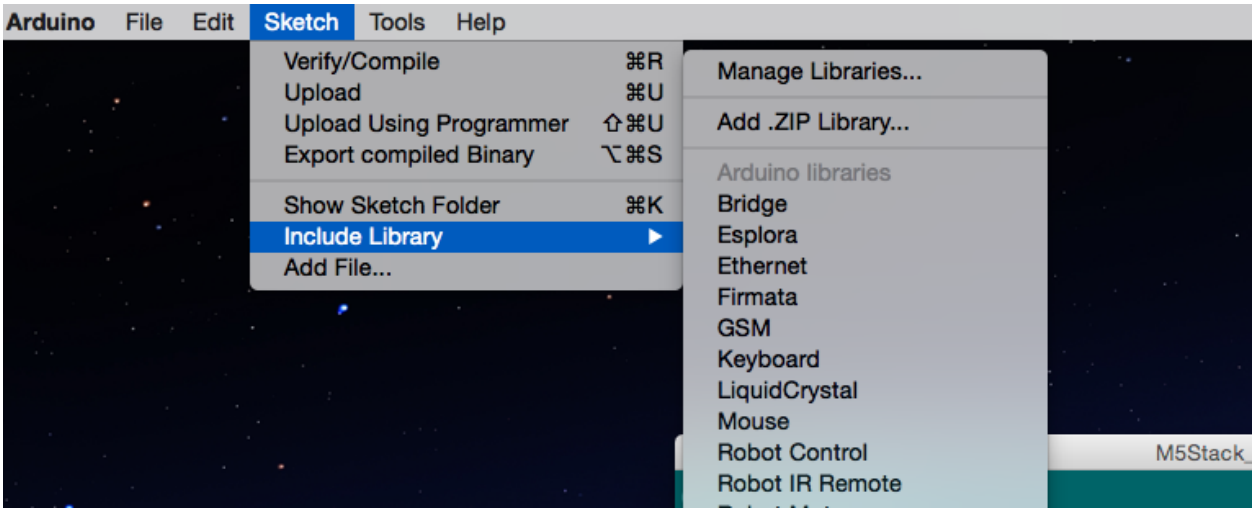


Fig. 1: image

3. If it shows as below figure, click `Update`.

But if it shows `Install`, it means you has not installed M5Stack Lib.

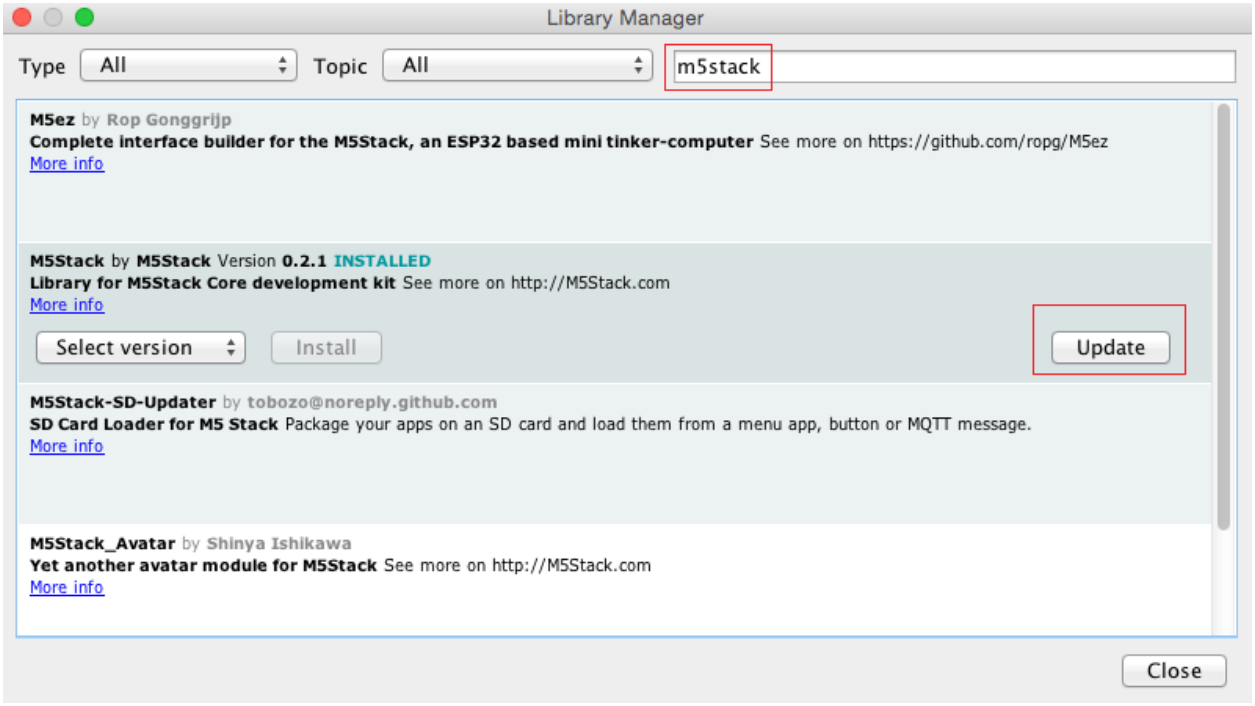

Library Manager
Type
All
Topic
All
m5stack
M5ez by Rop Gonggrijp
Complete interface builder for the M5Stack, an ESP32 based mini tinker-computer See more on https://github.com/ropg/M5ez
More info
M5Stack by M5Stack Version 0.2.1 INSTALLED
Library for M5Stack Core development kit See more on http://M5Stack.com
More info
Select version
Install
Update
M5Stack-SD-Updater by tobozo@noreply.github.com
SD Card Loader for M5 Stack Package your apps on an SD card and load them from a menu app, button or MQTT message.
More info
M5Stack_Avatar by Shinya Ishikawa
Yet another avatar module for M5Stack See more on http://M5Stack.com
More info
Close

Fig. 2: image

CHAPTER 3

API Reference

3.1 LCD API

3.1.1 Function

lcd.setbrightness(uint8_t brightness)

lcd.setRotation(degree)

Description:

set the angle of rotation of the entire screen

Parament:

degree: the angle of rotation

Example:

```
#The M5Stack Core LCD has been initialized
lcd.setRotation(90)
```

lcd.setColor(color [, background_color])

Description:

Set the default foreground/background color

Parament:

color: the color of text

background_color: the fill color of text

Example:

```
#The M5Stack Core LCD has been initialized
lcd.setColor(lcd.RED)
lcd.setColor(lcd.ORANGE, LCD.DARKCYAN)
```

lcd.setTextColor(color [,background_color])

Description:

This function is as same as *setColor(color [, background_color])*

Example:

```
#The M5Stack Core LCD has been initialized
lcd.setTextColor(lcd.PINK)
lcd.setTextColor(lcd.ORANGE, LCD.DARKCYAN)
```

lcd.fillScreen(color)

Description:

Fill the entire screen with the given color

Parament:

color: color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.fillScreen(lcd.RED)
```

lcd.drawPixel(x, y [,color])

Description:

Draw the pixel at position **(x,y)**

Note:

If color is not given, current foreground color is used

Parament:

color color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.drawPixel(22,22,lcd.RED)
```

lcd.drawLine(x, y, x1, y1 [,color])

Description:

Draw the line from point **(x,y)** to point **(x1,y1)**

Note:

If color is not given, current foreground color is used

Parament:

color color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.drawLine(0,0,12,12,lcd.WHITE)
```

lcd.drawTriangle(x, y, x1, y1, x2, y2 [,color])

Description:

Draw the triangel between points **(x,y)**, **(x1,y1)** and **(x2,y2)**

Note:

If color is not given, current foreground color is used

Parament:

color: color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.drawTriangle(22,22,69,98,51,22,lcd.RED)
```

lcd.fillTriangle(x, y, x1, y1, x2, y2 [,color])

Description:

Fill the triangel between points **(x,y)**, **(x1,y1)** and **(x2,y2)**

Note:

If color is not given, triangle will be filled in current foreground color

Parament:

color color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.fillTriangle(122, 122, 169, 198, 151, 182, lcd.RED)
```

lcd.drawCircle(x, y, r [,color])

Description:*

Draw the circle with center at **(x,y)** and radius **r**

Note:

If color is not given, current foreground color is used

Parament:

r: the radius of circle

color: color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.drawCircle(180, 180, 10, lcd.BLUE)
```

lcd.fillCircle(x, y, r [,color])

Description:

Fill the circle with center at **(x,y)** and radius **r**

Note:

If color is not given, current foregroundcolor will be used

Parament:

r: the radius of circle

color: color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.fillcircle(100, 100, 10, lcd.BLUE)
```

lcd.drawRect(x, y, w, h, [,color])

Description:

Draw the rectangle from the upper left point at **(x,y)** and **width** and **height**

Note:

If color is not given, rectangle will be drawn in current foreground color

Parament:

w: display phisical width in pixels (display's smaller dimension)
h: display phisical height in pixels (display's larger dimension)
color: optional, color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.drawRect(180, 12, 122, 10, lcd.BLUE)
```

lcd.fillRect(x, y, w, h, [,color])

Description:

Fill the rectangle from the upper left point at **(x,y)** and **width** and **height**

Note:

If fillcolor is not given, rectangle will be filled in current foreground color

Parament:

w: display phisical width in pixels (display's smaller dimension)
h: display phisical height in pixels (display's larger dimension)
color: optional, color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.fillRect(180,30,122,10,lcd.BLUE)
```

lcd.drawRoundRect(x, y, w, h, r [,color])

Description:

Draw the rectangle with rounded corners from the upper left point at **(x,y)** and **width** and **height**. Corner radius is given by **r** argument

Note:

*If *color* is not given, current foreground color will be used**

Parament:

w: display phisical width in pixels (display's smaller dimension)

h: display phisical height in pixels (display's larger dimension)

r: the radius of circle

color: optional, color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.drawRoundRect(180,50,122,10,4,lcd.BLUE)
```

lcd.fillRoundRect(x, y, w, h, r [,color])

Description:

Fill the rectangle with rounded corners from the upper left point at **(x,y)** and **width** and **height**. Corner radius is given by **r** argument

Note:

If **color** is not given, current foreground color will be used

Parament:

w: display phisical width in pixel (display's smaller dimension)

h: display phisical height in pixels (display's larger dimension)

r: the radius of circle

color: optional, color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.fillRoundRect(180,70,122,10,4,lcd.BLUE)
```

lcd.print('text', [x, y])

Description:

Print the **text** at position **(x,y)**

Parament:

text: the string need to print

Example:

```
#The M5Stack Core LCD has been initialized
lcd.print('this is a print text function', 80, 80)
```

lcd.clear([color])

Description:

Clear the screen with default background color orspecific color if given

Parament:

color: optional, color values

Example:

```
#The M5Stack Core LCD has been initialized
lcd.clear()
```

3.1.2 Usage

```
from machine import SPI, Pin
from display import LCD

spi = SPI(1, baudrate=32000000, mosi=Pin(23), miso=Pin(19), sck=Pin(18))

lcd = LCD(spi = spi) #lcd init
lcd.fillScreen(lcd.BLACK) #set the default background color

lcd.drawLine(0, 0, lcd.WHITE)
lcd.drawTriangle(22, 22, 69, 98, 51, 22, lcd.RED)
lcd.fillTriangle(122, 122, 169, 198, 151, 182, lcd.RED)
lcd.drawCircle(180, 180, 10, lcd.BLUE)
lcd.fillcircle(100, 100, 10, lcd.BLUE)
lcd.drawRect(180, 12, 122, 10, lcd.BLUE)
lcd.fillRect(180, 30, 122, 10, lcd.BLUE)
lcd.drawRoundRect(180, 50, 122, 10, 4, lcd.BLUE)
lcd.fillRoundRect(180, 70, 122, 10, 4, lcd.BLUE)
lcd.print('this is a print text function', 80, 80)
```

3.2 Button API

3.2.1 Function

isPressed()

isReleased()

pressedFor(timeout)

wasPressed(callback=None) wasReleased(callback=None) releasedFor(timeout, callback=None)

3.2.2 Usage

```
from m5stack import *
from machine import SPI, Pin
from display import LCD
import utime

spi = SPI(1, baudrate=32000000, mosi=Pin(23), miso=Pin(19), sck=Pin(18))

lcd = LCD(spi = spi) #lcd init

while True:
  if buttonA.wasPressed():
    lcd.print('Button A was Pressed\n')

  if buttonA.wasReleased():
    lcd.print('Button A was Released\n')

  if buttonA.pressedFor(1.5):
    lcd.print('Button A pressed for 1.5s\n')

  if buttonA.releasedFor(2):
    lcd.print('Button A released for 2s press hold\n')

  utime.sleep(0.1)
```

```
#Button Callback

from m5stack import *

def on_wasPressed():
  lcd.print('Button B was Pressed\n')

def on_wasReleased():
  lcd.print('Button B was Released\n')

def on_releasedFor():
  lcd.print('Button B released for 1.2s press hold\n')

buttonB.wasPressed(on_wasPressed)
```

(continues on next page)

(continued from previous page)

```
buttonB.wasReleased(on_wasReleased)
buttonB.releasedFor(1.2, on_releasedFor)
```

3.3 Peripherals API

3.3.1 GPIO API

Function

value()

Usage

```
import machine

pinout = machine.Pin(0, machine.Pin.OUT)
pinout.value(1)

pinin = machine.Pin(2, machine.Pin.IN)
val = pinin.value()
```

3.3.2 UART API

Description

An Universal Asynchronous Receiver/Transmitter (UART) is a component known to handle the timing requirements for a variety of widely-adapted protocols (RS232, RS485, RS422, ...). An UART provides a widely adopted and cheap method to realize full-duplex data exchange among different devices. There are three UART controllers available on the ESP32 chip. They are compatible with UART-enabled devices from various manufacturers. All UART controllers integrated in the ESP32 feature an identical set of registers for ease of programming and flexibility. In this documentation, these controllers are referred to as UART0, UART1, and UART2.

Function

machine.UART(uart_num, tx=pin, rx=pin [,args])

Description:

Create the uart instance object

Parament:

uart_num: The hardware SPI host. machine-SPI.HSPI (1) or machine.SPI.VSPI (2) can be used. Default: 1
tx: the gpio used for tx
rx: the gpio used for rx

Example:

```
from machine import UART

uart2 = UART(2, tx=17, rx=16)
uart2.init(115200, bits=8, parity=None, stop=1)
```

write(buf [, len [, off]])

Description:

Write bytes from buffer object buf to UART

Parament:

buf: data buffer to be write
len: writes *en* bytes
off: starts writting from *off* position in *buf*

Example:

```
from machine import UART

uart2 = UART(2, tx=17, rx=16)
uart2.init(115200, bits=8, parity=None, stop=1)
uart2.write('abc')    # write the 3 characters
```

read()

Description:

read all available characters

Example:

```
from machine import UART

uart2 = UART(2, tx=17, rx=16)
uart2.init(115200, bits=8, parity=None, stop=1)
uart2.read()          # read all available characters
```

readline([max_len])

Description:

Reads all bytes from the receive buffer up to the line end character **n**

Parament:

max_len: maximum length to be read

Example:

```
from machine import UART

uart2 = UART(2, tx=17, rx=16)
uart2.init(115200, bits=8, parity=None, stop=1)
uart2.readline()     # read a line
```

Usage

```
from machine import UART

uart2 = UART(2, tx=17, rx=16)
uart2.init(115200, bits=8, parity=None, stop=1)
uart2.read(10)       # read 10 characters, returns a bytes object
uart2.read()         # read all available characters
uart2.readline()     # read a line
uart2.readinto(buf)  # read and store into the given buffer
uart2.write('abc')   # write the 3 characters
```

3.3.3 ADC API

Description

ESP32 integrates two 12-bit SAR (Successive Approximation Register) ADCs (Analog to Digital Converters) and supports measurements on 18 channels (analog enabled pins)

ESP32 DAC output voltage range is 0-Vdd (3.3 V), the resolution is 8-bits

---

Function

machine.ADC(pin [,unit=1])

Description:

Create the adc instance object

Parament:

pin: pin argument defines the gpio which will will be used as adc input

unit: Optional unit argument select ESP32 ADC unit for this instance. Values 1 (ADC1, default) or 2 (ADC2) can be selected

Example:

```
import machine

adc = machine.ADC(35)
```

read()

Description:

Read the ADC value as voltage (in mV)

Parament:

None Parament

Example:

```
import machine

adc = machine.ADC(35)
adc.read()
```

Usage

```
import machine

adc = machine.ADC(35)
adc.read()
```

3.3.4 DAC API

Description

ESP32 has two 8-bit DAC (digital to analog converter) channels, connected to GPIO25 (Channel 1) and GPIO26 (Channel 2). The DAC driver allows these channels to be set to arbitrary voltages.<br The DAC channels can also be driven with DMA-style written sample data, via the I2S driver when using the "built-in DAC mode".

ESP32 DAC output voltage range is 0-Vdd (3.3 V), the resolution is 8-bits

---

Function

machine.DAC(pin)

Description:

Pin argument defines the gpio which will will be used as dac output
Note: Only GPIOs 25 and 26 can be used as DAC outputs

Parament:

pin: The pin argument can be given as pin number (integer) or the machine.Pin object

Example:

```
import machine

dac = machine.DAC(machine.Pin(26))
dac.write(128)
```

write(value)

Description:

Set the DAC value
*Note: The value of 255 sets the voltage on dac pin to 3.3 V

Parament:

value: DAC vaule.Valid range is: 0 - 255

Example:

```
import machine

dac = machine.DAC(machine.Pin(26))
dac.write(128)
```

Usage

```
import machine

dac = machine.DAC(machine.Pin(26))
dac.write(128)
```

3.3.5 Timer API

Description

The ESP32 chip contains two hardware timer groups. Each group has two general-purpose hardware timers. They are all 64-bit generic timers based on 16-bit prescalers and 64-bit auto-reload-capable up / down counters.

Note: Due to MicroPython callback latency, some callbacks may not be executed if the timer period is less than 15 ms. The number of events and executed callbacks can be checked using tm.events() method.

Function

machine.Timer(timer_no)

Description:
Create the Timer instance object

Parament:
timer_no: the timer number to be used for the timer

Example:

```
import machine

p1 = machine.Pin(27)
```

init(period, mode, callback, dbgpin)

Description:
initialize paraments of the timer

Parament:

period: Timer period(Default: 10 ms)

Note: Timer period in ms, only used for PERIODIC and ONE_SHOT timers

mode: Timer mode of operation(Default: PERIODIC)

callback: The Python callback function to be executed on timer event(Default: None)

dbgpin: GPIO pin to be used as debug output(Default: -1 (not used))

Note: If used, the gpio level will toggle on each timer event

Return:

None

Example:

```
from machine import SPI, Pin

spi = SPI(
    spihost=SPI.HSPI,
    baudrate=2600000
    sck=Pin(18),
    mosi=Pin(23),
    miso=Pin(19),
    cs=Pin(4)
)

spi.write(buf) #NOHEAP
```

value()

Description:

Return:

Returns the current timer counter value in **µs** if timer mode is CHRONO or in **ms** for other modes

Example:

```
import machine

p1 = machine.Pin(27)
p1.init(p1.OUT)
p1.value(1)
```

Usage

```
import machine

tcounter = 0

p1 = machine.Pin(27)
p1.init(p1.OUT)
p1.value(1)

def tcb(timer):
    global tcounter
    if tcounter & 1:
        p1.value(0)
    else:
        p1.value(1)
    tcounter += 1
    if (tcounter % 10000) == 0:
        print("[tcb] timer: {} counter: {}".format(timer.timernum(), tcounter))

t1 = machine.Timer(2)
t1.init(period=20, mode=t1.PERIODIC, callback=tcb)
```

3.3.6 Neopixel API

Description

This class includes full support for various types of Neopixel, individually-addressable RGB(W) LEDs. ESP32 RMT peripheral is used for very precise timing (+/- 50 ns). Up to 8 Neopixel strips can be used, 1~1024 pixels each.

Function

machine.Neopixel(pin, pixels, type)

Description:

Create the Neopixel instance object

Parament:

pin: the data gpio which connects to rgb

pixels: The number of rgb

type: Neopixel type: 0 (machine.Neopixel.RGB) for RGB LEDs, or 1 (machine.Neopixel.RGBQ) for RGBW LEDs

Example:

```
import machine
np = machine.Neopixel(22, 115, 0)
npw = machine.Neopixel(machine.Pin(25), 24, machine.Neopixel.RGBW)
```

setHSB(pos, hue, saturation, brightness [, white, num, update]))

Description:

Write bytes from buffer object buf to the Neopixel device

Parament:

pos: required, pixel position; 1 ~ pix_num

hue: float: any number, the floor of this number is subtracted from it
to create a fraction between 0 and 1.
This fractional number is then multiplied by 360 to produce the hue angle in the HSB color model

saturation: float; 0 ~ 1.0

brightness: float; 0 ~ 1.0

num: optional; default: 1; number of pixels to set to the same color, starting from pos

update: optional, default: True; update the Neopixel strip.
Note: If False np.show() has to be used to update the strip

Return:

None

Example:

```
import machine, time

np = machine.Neopixel(machine.Pin(22), 24)

def rainbow(loops=120, delay=1, sat=1.0, bri=0.2):
    for pos in range(0, loops):
        for i in range(0, 24):
            dHue = 360.0/24*(pos+i);
            hue = dHue % 360;
            np.setHSB(i, hue, sat, bri, 1, False)
        np.show()
        if delay > 0:
            time.sleep_ms(delay)

def blinkRainbow(loops=10, delay=250):
    for pos in range(0, loops):
        for i in range(0, 24):
            dHue = 360.0/24*(pos+i);
            hue = dHue % 360;
            np.setHSB(i, hue, 1.0, 0.1, 1, False)
        np.show()
        time.sleep_ms(delay)
        np.clear()
        time.sleep_ms(delay)
```

Usage

```
import machine, time

np = machine.Neopixel(machine.Pin(22), 24)

def rainbow(loops=120, delay=1, sat=1.0, bri=0.2):
    for pos in range(0, loops):
        for i in range(0, 24):
            dHue = 360.0/24*(pos+i);
            hue = dHue % 360;
            np.setHSB(i, hue, sat, bri, 1, False)
        np.show()
        if delay > 0:
            time.sleep_ms(delay)

def blinkRainbow(loops=10, delay=250):
    for pos in range(0, loops):
        for i in range(0, 24):
            dHue = 360.0/24*(pos+i);
            hue = dHue % 360;
            np.setHSB(i, hue, 1.0, 0.1, 1, False)
        np.show()
        time.sleep_ms(delay)
        np.clear()
        time.sleep_ms(delay)
```

3.3.7 I2C API

Description

Both master and slave modes are supported Master and slave modes can be used at the same time, on different I2C interfaces

---

Function

machine.I2C(id, mode, speed, sda, scl, slave_addr, slave_bufflen, slave_rolen, slave_busy)

Description:

Create the I2C instance object

Parament:

id: The hardware I2C peripheral ID; 0 or 1 can be used. Default: 0

mode: I2C interface mode; master or slave. Use the constants **machine.I2C.MASTER** or **machine.I2C.SLAVE** Default: master

speed: I2C clock frequency in **Hz**. Default: 100000

sda: I2C **sda** pin; can be given as integer gpio number or Pin object

scl: I2C **scl** pin; can be given as integer gpio number or Pin object

slave_addr: I2C slave address to be assigned to this i2c interface. Only used if SLAVE mode is selected
slave_bufflen: Size of slave buffer used for master<->slave comunication in bytes
slave_rolen: Size of **read-only** area at the end of the slave buffer in bytes
slave_busy: Only used if **SLAVE** mode is selected
Note: Only sda and scl are required, all the others are optional and will be set to the default values if not given

init(args)

Description:
Reinitialize an existing I2C object

Parament:
The arguments are the same as for creating a new i2c instance object

scan()

Description:
Scan for i2c devices on I2C bus. Does not scan reserved 7-bit addresses: **0x00-0x07** & **0x78-0x7F**
Note: can only be used in master mode

Parament:
None

Return:
Returns the list of detected addresses

readfrom(addr, nbytes)

Description:
Read **nbytes** bytes from i2c device with address **addr**
Note: can only be used in master mode

Parament:
None:

Return:
Return bytearray of read bytes

writeto(addr, buf [,stop=True])

Description:

Write the content of the buffer object **buf** to the i2c device with address **adr**
Note: can only be used in master mode

Parament:
Note: If optional stop argument is set to False, the stop signal is not issued

readfrom_into(addr, buf)

Description:
Read from i2c device with address **addr** into buffer object **buf**
Note: can only be used in master mode

Parament:
Note: Size of buf bytes are read

Usage

```
from machine import I2C

i2c = I2C(freq=400000, sda=21, scl=22)
                                # create I2C peripheral at frequency of 400kHz
                                # depending on the port, extra parameters may be␣
↪required
                                # to select the peripheral and/or pins to use

i2c.scan()                      # scan for slaves, returning a list of 7-bit addresses

i2c.writeto(42, b'123')         # write 3 bytes to slave with 7-bit address 42
i2c.readfrom(42, 4)             # read 4 bytes from slave with 7-bit address 42

i2c.readfrom_mem(42, 8, 3)      # read 3 bytes from memory of slave 42,
                                #   starting at memory-address 8 in the slave
i2c.writeto_mem(42, 2, b'\x10') # write 1 byte to memory of slave 42
                                #   starting at address 2 in the slave
```

3.3.8 SPI API

Description

This class includes full support for using ESP32 SPI peripheral in master mode

Only SPI master mode is supported for now.

Python exception wil be raised if the requested spihost is used by SD Card driver (sdcard in spi mode). If the requested spihost is VSPI and the psRAM is used at 80 MHz, the exception will be raised. The exception will be raised if SPI cannot be configured for given configurations.

Function

machine.spi(spihost, baudrate, polarity, phase, firstbit, sck, mosi, miso, cs, duplex, bits)

Description:

Create the spi instance object

Parament:

spihost: The hardware SPI host. machine-SPI.HSPI (1) or machine.SPI.VSPI (2) can be used. Default: 1

baudrate: SPI clock speed in Hz; Default: 1000000

Example:

```
from machine import SPI, Pin

spi = SPI(
    spihost=SPI.HSPI,
    baudrate=2600000
    sck=Pin(18),
    mosi=Pin(23),
    miso=Pin(19),
    cs=Pin(4)
)
```

write(buf)

Description:

Write bytes from buffer object buf to the SPI device

Parament:

buf: data buffer to be write

Return:

Returns *True* on success, *False* ion error

Example:

```
from machine import SPI, Pin

spi = SPI(
    spihost=SPI.HSPI,
    baudrate=2600000
    sck=Pin(18),
    mosi=Pin(23),
    miso=Pin(19),
    cs=Pin(4)
)

spi.write(buf) #NOHEAP
```

write_readinto(wr_buf, rd_buf)

Description:

Write bytes from buffer object wr_buf to the SPI device and reads from SPI device into buffer object rd_buf

Parament:

wr_buf: data buffer to be write

rd_buf: data buffer to be read

*Note: * In fullduplex mode write and read are simultaneous. In halfduplex mode the data are first written to the device, then read from it

Return:

Returns *True* on success, *False* ion error

Example:

```
import machine

adc = machine.ADC(35)
adc.read()
```

Usage

```
from machine import SPI, Pin

spi = SPI(
    spihost=SPI.HSPI,
    baudrate=2600000
    sck=Pin(18),
    mosi=Pin(23),
    miso=Pin(19),
    cs=Pin(4)
```

(continues on next page)

(continued from previous page)

```
)

spi.write(buf) #NOHEAP
spi.read(nbytes, *, write=0x00) #write is the byte to ?output on MOSI for each byte␣
↪read in
spi.readinto(buf, *, write=0x00) #NOHEAP
spi.write_readinto(write_buf, read_buf) #NOHEAP; write_buf and read_buf can be the␣
↪same
```

3.3.9 RTC API

Description

The content of the RTC memory is preserved during the deep sleep.

Up to 64 32-bit integers can be saved in RTC memory.

One string of up to 2048 characters can be saved in RTC memory. The string can be, for example, json string containing the parameters which has to be restored after deep sleep wake-up.

Integers and string saved in RTC memory are protected by 16-bit CRC

Function

machine.RTC()

Description:

Create the RTC instance object

init(date)

Description:

Set the system time and date

Parament:

date: it's a tuple containing the time and date information

Note: the tuple is equal to (year, month, day [,hour [,minute [, second]]])

now()

Description:

get the current time

Parament:

None

Return:
Return the current time as tuple: *(year, month, day, hour, minute, second)*

ntp_sync(server [,update_period] [,tz])

Description:
Write bytes from buffer object buf to the RTC device

Parament:
server : the NTP server domain name or IP, for example "pool.ntp.org"
update_period: optional, time update interval in seconds; default: 3600
tz: optional, time zone string; default: the one set in menuconfig

Return:
Returns *True* on success, *False* ion error

Example:

```
import machine
import utime

rtc = machine.RTC()
rtc.ntp_sync(server="hr.pool.ntp.org", tz="CET-1CEST")
```

synced()

Description:
Sync the system time from NTP server

Parament:
None:

Return:
Return *True* if the system time was synced from NTP server, *False* if not

Usage

```
import machine
import utime

rtc = machine.RTC()
rtc.ntp_sync(server="hr.pool.ntp.org", tz="CET-1CEST")
rtc.synced()
True
utime.gmtime()
(2018, 1, 29, 16, 3, 18, 2, 29)
utime.localtime()
(2018, 1, 29, 17, 3, 30, 2, 29)
```

3.4 Mic API

3.4.1 Function

3.4.2 Usage

```
from m5stack import *
import machine, _thread

mic_adc = 0
buffer = []
def microphone_enter():
        print('microphone_enter')
global mic_adc, buffer
        try:
                mic_adc = machine.ADC(34)
                mic_adc.atten(mic_adc.ATTN_11DB)
        dac = machine.DAC(machine.Pin(25))
        dac.write(0)
        except:
                pass
        buffer = []
        for i in range(0, 55):
                buffer.append(0)


def microphone_loop():
        global mic_adc, buffer
        val = 0
        for i in range(0, 32):
                raw = (mic_adc.readraw() - 1845) // 10
        if raw > 20:
                raw = 20
        elif raw < -20:
                raw = -20
```

(continues on next page)

(continued from previous page)

```
        val += raw
val = val // 32
        buffer.pop()
        buffer.insert(0, val)
        for i in range(1, 50):
        lcd.line(i*2+44, 120+buffer[i+1], i*2+44+2, 120+buffer[i+2], lcd.WHITE)
        lcd.line(i*2+44, 120+buffer[i],   i*2+44+2, 120+buffer[i+1], lcd.BLACK)

microphone_enter()

while 1:
        microphone_loop()
```

3.5 Speaker API

3.5.1 Function

volume(volume)

Description:

Set the volume of sound

Parament:

volume: sound volume

Example:

```
from m5stack import *

speaker.volume(2)
speaker.tone(freq=1800)
```

tone(freq [, duration])

Description:

Speak a sound with *frequency* and *duration*

Parament:

frequency: the frequency of sound
duration: the duration of sound continued

Return:

None

Example:

```
from m5stack import *

speaker.volume(2)
speaker.tone(freq=1800)
speaker.tone(freq=1800, duration=200) # Non-blocking
```

3.5.2 Usage

```
from m5stack import *

speaker.volume(2)
speaker.tone(freq=1800)
speaker.tone(freq=1800, duration=200) # Non-blocking
```

3.6 SD Card API

3.6.1 Usage

```
import uos

uos.mountsd()
uos.listdir('/sd')
```

CHAPTER 4

Basic Cases

4.1 M5Stack Core Cases

Coming soon! Please wait!

4.2 M5GO Cases

Coming soon! Please wait!

4.3 ESP32CAM Cases

Coming soon! Please wait!

Coming soon! Please wait!

CHAPTER 5

M5Stack-awesome

- M5Stack-SD-Updater - Customizable menu system for M5Stack - loads apps from the Micro SD card
- TFT_eSPI - TFT library for the ESP8266 and ESP32 that supports different driver chips
- M5Widgets - Widgets for the M5Stack
- M5StackSAM - Simple Applications Menu Arduino Library for M5Stack
- cfGUI - A simple GUI library for M5Stack (ESP32)
- GUIslice - A lightweight GUI framework suitable for embedded displays
- M5ez - The easy way to program on the M5Stack
- M5Stack MultiApp Advanced - A M5Stack firmware made on PlatformIO
- M5Stack ESP32 Oscilloscope - A fully functional oscilloscope based on ESP32 M5Stack
- M5Stack-Avatar - An M5Stack library for rendering avatar faces
- M5Stack_CrackScreen - Crack your M5Stack!!
- M5_Shuttle_Run - M5_Shuttle_Run
- nixietubeM5 - (Fake) Nixie Tube Display on a M5Stack
- M5Stack_BTCTicker - A small Bitcoin price ticker using an M5Stack (ESP32) and the Coindesk API
- M5Stack_ETHPrice - Dependence on example Wifi Setting to get ETH Price from Maicoin
- M5Stack-PacketMonitor - M5Stack ESP32 Packet Monitor
- M5-FFT - Graphic Equalizer on the M5Stack platform
- M5Stack_ESP32_radio - Playing mp3 stream out of internet using M5Stack prototype
- mp3-player-m5stack - MP3 player for M5Stack
- ArduinoWiFiPhotoBackup - M5STACK Arduino WiFi Photo Backup device
- M5StackHIDCtrlAltDel - You can send ctrl+alt+del to your PC from M5Stack
- M5Stack Markdown Web Server - Markdown & icons loaded from an Micro SD card/TF card to run a web page

- M5Stack-Tetris - Tetris for M5Stack Ported to M5Stack by macsbug - https://macsbug.wordpress.com/
- M5Stack_FlappyBird_game - M5Stack FlappyBird Playable
- M5Stack-SpaceShooter - Space Invaders knock-off for M5Stack
- M5Stack-Pacman-JoyPSP - Pacman on M5Stack/PSP Joypad, with sounds
- M5Stack-Thermal-Camera - M5Stack Thermal Camera with AMG8833 thermal sensor
- M5Stack-3DPrintFiles - Links to files for 3D printing custom case parts for the M5Stack

CHAPTER 6

M5Stack-FAQ

6.1 M5Stack-Core FAQ

Note: Thank you for being interested in our products and services, Please contact us by tech support email or forum if you need any assistance.

6.2 M5Stack-Module FAQ

Note: Thank you for being interested in our products and services, Please contact us by tech support email or forum if you need any assistance.

Note: Thank you for being interested in our products and services, Please contact us by tech support email or forum if you need any assistance.