

# M5GO Lite



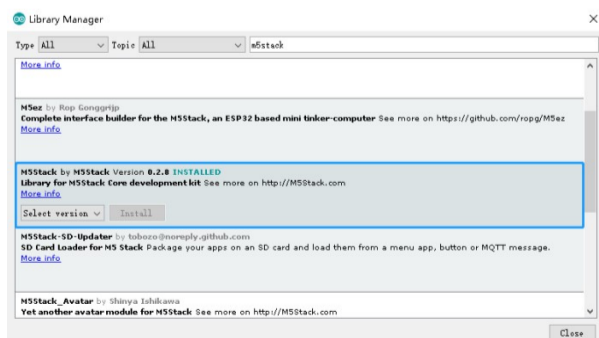
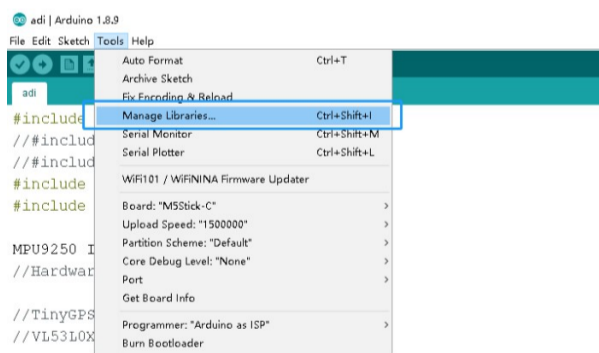
## DESCRIPTION

**M5GO Lite** is a light pack of M5GO kit, instead of having 6 M5units, M5GO-Lite provides 1 ENV + accessories and the same M5 controller as M5GO. M5GO-Lite is also designed for STEM education. M5GO controller itself is already a full-feathered, highly integrated, upgraded development board provides plenty of hardware resources, such as LCD screen, speaker, Led bar, 16M flash, microphone and more. Light but not least.

All M5stack development board can be programmed through Arduino IDE, WebIDE UIFlow, Micropython, and Blockly, simplifying the development process for those requiring a joint hardware and software solution. Not only does M5stack have far-reaching IoT applications in industry, agriculture, and home, but it also empowers students to learn to code in STEM classrooms.

### Notice:

The newly-produced M5Core replaces the screen with better display performance and higher viewing angle, so it has some compatibility problems with the old Arduino library. When using the old library for screen driving, it will produce reverse color display. You can open the Arduino. The library management option will upgrade your M5Stack library to the latest version (after 0.2.8) to solve this problem.



## Product Features

- 5V DC power supply

- 16 MByte flash
- BMM150 + MPU6886
- Speaker, 3 Buttons, Color LCD(320\*240), 1 Reset
- 2.4G Antenna: Proant 440
- TF card slot (16G Maximum size)
- Extendable Pins & Holes
- Grove Port
- M-Bus Socket & Pins
- Development Platform [UIFlow](#), [MicroPython](#), [Arduino](#)
- Product Size: 54.2mm x 54.2mm x 17.9mm
- Product weight: 159g

## ESP32 Features

---

- 240 MHz dual core Tensilica LX6 microcontroller with 600 DMIPS
- Integrated 520 KB SRAM
- Integrated 802.11b/g/n HT40 Wi-Fi transceiver, baseband, stack and LWIP
- Integrated dual mode Bluetooth (classic and BLE)
- Hall sensor
- 10x capacitive touch interface
- 32 kHz crystal oscillator
- PWM/timer input/output available on every GPIO pin
- SDIO master/slave 50MHz
- SD-card interface support

## M5GO Bottom

[Click to view details parameters](#)

## PinMap

---

### Peripherals Pin Map

#### LCD & TF card

LCD : 320x240 TF card Maximum size 16GB

ESP32 Chip	GPIO23	GPIO19	GPIO18	GPIO14	GPIO27	GPIO33	GPIO32	GPIO4
ILI9342C	MOSI/MISO	/	CLK	CS	DC	RST	BL	
TF Card	MOSI	MISO	CLK					CS

#### Button & Speaker

ESP32 Chip	GPIO39	GPIO38	GPIO37	GPIO25
Button Pin	BUTTON A	BUTTON B	BUTTON C	
Speaker				Speaker Pin

#### GROVE Port A & IP5306

We've use the customized I2C version of IP5306, on power management.

---

GROVE A	SCL	SDA	5V	GND
---------	-----	-----	----	-----

---

IP5306	SCL	SDA	5V	GND
--------	-----	-----	----	-----

---

**IP5306 charging/discharging, Voltage parameter**

---

charging	discharging
0.00 ~ 3.40V -> 0%	4.20 ~ 4.07V -> 100%
3.40 ~ 3.61V -> 25%	4.07 ~ 3.81V -> 75%
3.61 ~ 3.88V -> 50%	3.81 ~ 3.55V -> 50%
3.88 ~ 4.12V -> 75%	3.55 ~ 3.33V -> 25%
4.12 ~ / -> 100%	3.33 ~ 0.00V -> 0%

---

**6-Axis MotionTracking Sensor MPU6886**

*MPU6886 I2C address 0x68*

---

ESP32 Chip	GPIO22	GPIO21	5V	GND
------------	--------	--------	----	-----

---

MPU6886	SCL	SDA	5V	GND
---------	-----	-----	----	-----

---

**3-Axis Geomagnetic Sensor BMM150**

*BMM150 I2C address 0x10*

---

ESP32 Chip	GPIO22	GPIO21	5V	GND
------------	--------	--------	----	-----

---

BMM150	SCL	SDA	5V	GND
--------	-----	-----	----	-----

---

**M5GO Base Port**

**GROVE Port B**

---

ESP32 Chip	GPIO36	GPIO26	5V	GND
------------	--------	--------	----	-----

---

GROVE B	GPIO36	GPIO26	5V	GND
---------	--------	--------	----	-----

---

**GROVE Port C**

---

ESP32 Chip	GPIO16	GPIO17	5V	GND
------------	--------	--------	----	-----

---

GROVE C	RXD	TXD	5V	GND
---------	-----	-----	----	-----

---

**LED Bar & Micphone & Speaker**

---

ESP32 Chip	GPIO15	GPIO34	GPIO25
------------	--------	--------	--------

---

LED Bar	SIG Pin
---------	---------

---

MIC	MIC Pin
-----	---------

---

## PARAMETER

Resources	Parameter
ESP32	240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth
Flash Memory	16MB
Power Input	5V @ 500mA
Port	TypeC x 1, GROVE(I2C+I/O+UART) x 1
IPS Screen	2 inch, 320x240 Colorful TFT LCD, ILI9342C
Speaker	1W-0928
MEMS	BMM150 + MPU6886
Battery	500 mAh @ 3.7V
Operating Temperature	32°F to 104°F ( 0°C to 40°C )
Size	54 x 54 x 21 mm
Case Material	Plastic ( PC )

**Notice1: M5PORT EXPLAIN** You can identify the port name and function by its color, red is PortA(21/22) mainly used for I2C, black is PortB(26/36) which can be used for DA/AD, Singel-bus communication, Blue is PortC(16/17) can be used for Uart. Correspondingly, most of the M5 Units have the Port with matched color for specify which port it should go in on the M5Core. Those port identification is a convenience for UIFlow (Blockly) Users. For advanced using ,you can do you own customization, since most of the PIN on ESP32 are remapping-able. Unfortunately, PortA(red) can not be used as analog read in. It refers to GPIO 21 & 22 from ESP32, which doesn't have AD channel alternatives:

- ADC1(8 channels attached to GPIOs 32-39)
- ADC2(10 channels attached to GPIOs 0, 2, 4, 12-15, 25-27)

To use AD read function : 1, Use Dupont cable refers to the pins on the side which can be used as an AD channel. 2, Get a M5GO bottom, which comes with a PortB. 3, Get a PbHUB and connect it with PortA, then you can have 6 PortBs. For more information about Pin assignment and Pin Remapping, Please refer to EPS32 Datasheet

## Package Includes:

- 1x M5GO
- 1x ENV Unit
- 1x GROVE Cable
- 1x Type-C USB(20cm)
- 1x User Manual



## Related Link

---

- **Datasheet**

[ESP32](#)  
[MPU6886](#)  
[BMM150](#)

- **Register Manual**

[IP5306](#)

## Schematic

---

**Schematic - [BASIC](#)**

## Version Change

---

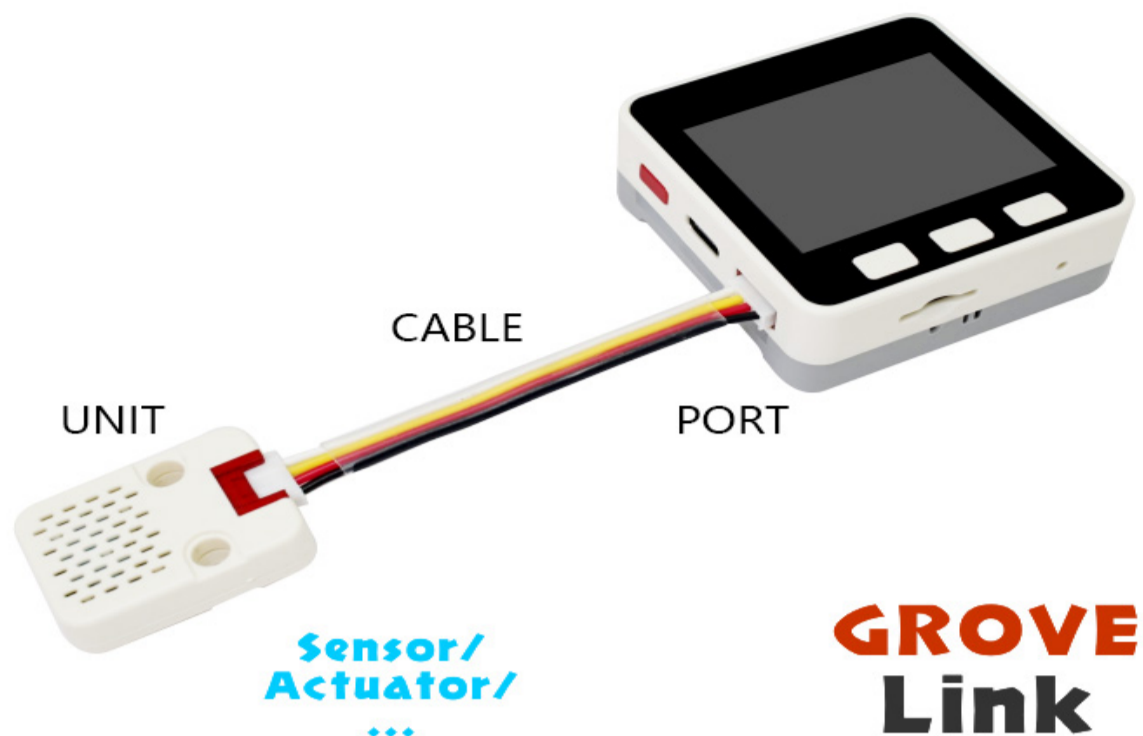
Release Date	Product Change
2018.4	Initial public release
2019.6	MPU9250 changed to MPU6886+BMM150
2019.7	TN screen changed to IPS screen
2019.11	Battery capacity changed from 600mAh to 500mAh

## Example

---

### 1. Arduino IDE

M5GO + ENV to get weather data:



But before compiling this program, please install `Adafruit BMP280 Library` and copy `Adafruit_Sensor.h` to `C:\Users\\Documents\Arduino\libraries\Adafruit_BMP280_Library` first.

```
arduino

/*
   Install Adafruit BMP280 Library first.
*/
#include <M5Stack.h>
#include "DHT12.h"
#include <Wire.h> //The DHT12 uses I2C communication.
#include "Adafruit_Sensor.h"
#include <Adafruit_BMP280.h>

// new two objects
DHT12 dht12;
Adafruit_BMP280 bme;

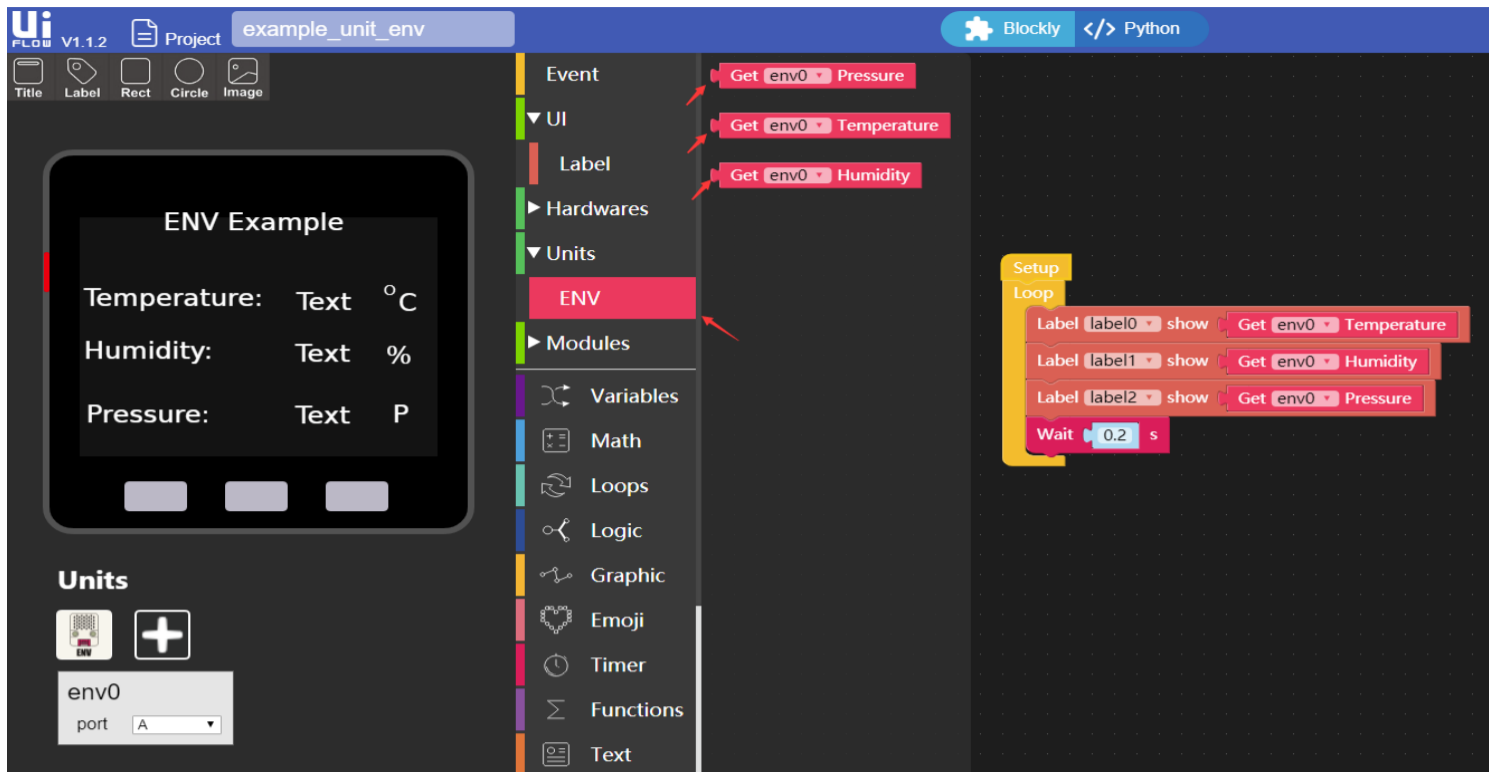
// initialization
M5.begin();
Wire.begin();
bme.begin();

// read data
float tmp = dht12.readTemperature();
float hum = dht12.readHumidity();
float pressure = bme.readPressure();
```

More examples are [here](#)

## 2. UIFlow

For more details, click [here](#)



The screenshot displays the M5Stack UI Designer interface for a project named "example\_unit\_env". The interface is divided into several sections:

- Top Bar:** Shows the version "V1.1.2", the project name "example\_unit\_env", and the "Blockly" editor mode with a "Python" toggle.
- Left Panel (UI Preview):** Displays a mobile device screen titled "ENV Example" with three text labels: "Temperature: Text °C", "Humidity: Text %", and "Pressure: Text P". Below the screen is a "Units" section with a "+" icon and a dropdown menu for "env0" set to "A".
- Center Panel (Blockly Script):** Contains a "Setup" block followed by a "Loop" block. The "Loop" block contains three "Label show" blocks connected to "Get env0" blocks for "Temperature", "Humidity", and "Pressure", followed by a "Wait 0.2 s" block.
- Right Panel (Component Palette):** Lists various components such as "Event", "UI", "Label", "Hardwares", "Units", "Modules", "Variables", "Math", "Loops", "Logic", "Graphic", "Emoji", "Timer", "Functions", and "Text". The "ENV" component is highlighted in red.