



Comentario técnico: CTC-107
 Componente: **M5Stack como herramienta de desarrollo para ESP32 con Mongoose-OS**
 Autor: Sergio R. Caprile, Senior R&D Engineer

Revisiones	Fecha	Comentarios
0	29/09/20	

El M5Stack no es sólo un simpático juguete para hobbyistas, puede ser una interesante herramienta de desarrollo para el ESP32 y fundamentalmente una plataforma a la que podemos rápidamente agregar componentes y obtener el hardware necesario para disponer de una maqueta, una prueba de concepto, sin tener que reinventar la rueda con cada prospecto de desarrollo y reutilizando nuestro código existente. Veremos aquí cómo podemos hacer esto con la ayuda de Mongoose-OS.

Descripción del M5Stack



El M5Stack cuenta con un display color de 2" con resolución de 320x240 píxeles, tres botones, conectores Grove¹, batería de 500mAh y antena WiFi, todo en un gabinete cómodo y pequeño. El display nos permite además tener la posibilidad de visualización del estado de nuestra maqueta sin tener que recurrir a desarrollos o conectividades adicionales ajenas a la prueba en sí, es decir, si la aplicación requiere o no un display, ya disponemos de uno funcionando. El módulo principal (M5Core) cuenta con un conector Grove (el conector A), el cual transporta el bus I²C.²

El módulo secundario (M5GO Bottom) cuenta con dos conectores Grove (B y C), ambos pueden ser utilizados como I/O o en particular los pines del conector B pueden asignarse a ADC y DAC y los del conector C a la UART.³ Este módulo termina a su vez el bus I²C con los resistores apropiados, y permite extenderlo mediante pogo pins en la parte inferior, recibiendo un módulo adicional para cargar la batería (M5GO Charger).

Conectores					
Grove A	rojo	SCL	SDA	5V	GND
Grove B	negro	GPIO36(ADC1ch0)	GPIO26(DAC2)	5V	GND
Grove C	azul	GPIO16(RxD2)	GPIO17(TxD2)	5V	GND

Botones			
GPIO	GPIO39	GPIO38	GPIO37
Botón	A(1)	B(2)	C (3)

El M5GO Bottom incluye además una barra de diez LEDs RGB del tipo "neopixel", que controlamos mediante el GPIO15 .

1 https://wiki.seeedstudio.com/Grove_System/

2 Si bien podríamos mirar para otro lado y usarlo como I/O, esto no es aconsejable pues dentro del M5Stack hay varios periféricos I²C conectados a este bus.

3 Los pines coinciden con los asignados por defecto para la UART2, pero pueden asignarse igualmente a la UART1.

Adicionalmente, el M5Core incluye un parlante y su amplificador de 1W, que podemos controlar mediante el GPIO25 (DAC1); mientras que el M5GO Bottom incluye un micrófono y su preamplificador, cuya señal podemos muestrear mediante el GPIO34 (ADC1, canal 6) .

Si bien es mayormente utilizado para la programación, disponemos de un puerto USB tipo C con un CP2104 conectado a la UART0 del ESP32.

En caso que necesitemos más puertos de I/O o se trate de un desarrollo algo más involucrado, podemos conectarnos a una serie de pines (compartidos) del ESP32 mediante un conector que los distribuye entre los módulos y en este caso tenemos accesible retirando el M5GO Bottom.

Como nota de color, disponemos además de una IMU de 9-ejes conectada al bus I²C, que no abordamos aquí.

La desinformación técnica es abundante,⁴ a tal efecto hemos condensado los aspectos fundamentales en un apéndice al final de este documento.

Configuración

Configuramos en el archivo YAML que describe el proyecto:

```
libs:  
  - origin: https://github.com/CikaElectronica/m5stack_wrapper
```

Como vemos, sólo hace falta indicar que usamos la library de M5Stack desarrollada en Cika.⁵

Operación

Un simpático (y tal vez molesto) detalle del M5Stack es su botón rojo de doble función que opera como reset y como encendido/apagado. Mientras estemos conectados por el puerto USB, es decir, “desarrollando”, este botón ubicado próximo a dicho conector opera con la función de reset. Cuando estamos “operando”, es decir, sin conexión USB, un simple toque produce el reset del dispositivo, mientras que para apagarlo debemos realizar dos toques sucesivos. Para encenderlo, lo haremos con un toque.

Recordemos que además de lo anterior, si pensamos utilizar WiFi, deberemos configurar las credenciales para conectarnos por WiFi a nuestra red (SSID y clave).

En los archivos adjuntos disponemos de un ejemplo en detalle. El mismo muestra en pantalla el estado de la conexión, cuenta las presiones de los botones, y muestra la hora obtenida mediante SNTP⁶. Mostramos además como enviar y recibir mensajes por el puerto serie y una elemental RPC a modo de disponer de un esqueleto sobre el cual montar nuestra prueba de concepto rápidamente.

Desarrollo de nuestra aplicación

Describimos brevemente la API disponible. Para más información podemos consultar el código fuente de la library en el repositorio Github.⁷

Desarrollo en mJS

Incluimos la API de nuestro wrapper declarando `load('api_m5stack.js')`

Leemos los botones mediante instrucciones de lectura de GPIO, los pines los hemos comodamente incluido dentro del objeto JSON M5Stack como BTN1 a BTN3: `M5Stack.BTN1, M5Stack.BTN2, M5Stack.BTN3`

⁴ En páginas web se disemina mucha información redundante por doquier, incluyendo mezclar acotaciones innecesarias entre datos fundamentales y notas de aplicación con el esquemático.

⁵ Hemos desarrollado un wrapper a modo de facilitar el uso del soporte ya existente en Mongoose-OS para los periféricos contenidos en el M5Stack. El mismo puede ser aprovechado desde JavaScript (mJS).

⁶ Para información sobre el uso de SNTP, consultar el [CTC-097](#)

⁷ https://github.com/CikaElectronica/m5stack_wrapper

Inicializamos el display llamando a `M5stack.displayInit()`. Esto enciende el backlight, podemos controlarlo manualmente para ahorrar consumo llamando a `M5stack.backlightOn()` y `M5stack.backlightOff()`

Borramos el contenido del display llamando a `M5stack.cls()`

Imprimimos textos llamando a `M5stack.print(columna, M5stack.line(fila), 'texto')`

Imprimimos textos centrados llamando a `M5stack.printCentered(columna, M5stack.line(fila), 'texto');` esto comienza a imprimir en una columna tal que el centro del texto se ubica en la columna especificada

Seleccionamos el color del texto o del fondo llamando a `M5stack.setFgColor(color)` y `M5stack.setBgColor(color)` respectivamente. El parámetro corresponde a una especificación *RGB565* y por comodidad hemos definido los colores principales, por ejemplo `M5stack.BLACK`. Podemos usar las definiciones de la library del ILI9341 pero tengamos presente que hay que invertir los bits, por ejemplo `M5stack.setFgColor(~ILI9341.CYAN)`

Configuramos el tipo de letra llamando a `M5stack.setFont(tipo)`, hay cuatro tipografías disponibles de 9, 12, 18 y 24 puntos.

Inicializamos los LEDs y obtenemos un handler para operar sobre ellos mediante:

```
let strip = M5stack.pixelInit();
```

a continuación podemos operar sobre los LEDs llamando a `strip.setPixel(led, r, g, b)` y mostrarlo llamando a `strip.show()`. Borramos la barra llamando a `strip.clear()`

Para una operación elemental de la UART, proveemos un callback que se ejecutará cada vez que haya nuevos datos en el buffer de recepción o se libere espacio en el buffer de transmisión:

```
UART.setDispatcher(uartNo, function(uartNo) {
  if (UART.readAvail(uartNo) > 0) {
    // Received new data
    let data = UART.read(uartNo);
    let ra = data.length; // account for more data arriving between readAvail() and read()
  }
}, null);
```

Habilitamos la recepción llamando a `UART.setRxEnabled(uartNo, true)`

Enviamos datos llamando a `UART.write(uartNo, data)`

Desarrollo avanzado, o en C

En caso de necesitar mayor funcionalidad de la provista, o desarrollar en C, remitimos a las libraries originales sobre las que hemos construido el wrapper, con las debidas consideraciones:

- *Display color*
 - Mongoose-OS provee soporte para el ILI9341 conectado mediante SPI.⁸ El M5Stack (al momento de escribir este texto) emplea el ILI9342C, los colores se ven invertidos a menos que hagamos una simple inversión de los bits que enviamos al seleccionar el color.⁹
- *I²C*
 - Mongoose-OS provee soporte para un bus I²C;¹⁰ pero además es probable que ya exista un driver para el periférico que queremos conectar.
- *Barra de LEDs RGB*
 - Mongoose-OS provee soporte para controlar LEDs del tipo “neopixel”.¹¹
- *UART*
 - Mongoose-OS provee soporte para las UARTs del ESP32.¹²

⁸ <https://mongoose-os.com/docs/mongoose-os/api/drivers/ili9341-spi.md>

⁹ Para más información, [consultar el código fuente](#)

¹⁰ <https://mongoose-os.com/docs/mongoose-os/api/core/i2c.md>

¹¹ <https://mongoose-os.com/docs/mongoose-os/api/drivers/neopixel.md>

¹² https://mongoose-os.com/docs/mongoose-os/api/core/mgos_uart.h.md

Apéndice: Especificaciones del M5Stack

Flash Memory	16MB
PSRAM ¹³	4MB
Display	2", 320x240 TFT LCD, ILI9342C, 853nit max.
Speaker	1W-0928
IMU	BMM150 + MPU6886
Battery	3.7V 500 mAh
Antenna	2.4GHz 3D Antenna
M5 Core	CPU, LCD, card, IMU, speaker, buttons
M5GO Bottom	battery, microphone, LED bar
M5GO Charger	battery charger

SPI bus

ESP32 pin	GPIO23	GPIO19	GPIO18	GPIO14	GPIO27	GPIO33	GPIO32	GPIO4
ILI9342C	MOSI/MISO		CLK	CS	D/C	RST	Backlight	
TF Card	MOSI	MISO	CLK					CS

I²C bus

Compartido por el conector Grove A, un IP5306 modificado (power management), la IMU, el M5GO Bottom, y mediante pogo pins el M5GO Charger. La alimentación del bus es de 5V.

<i>I²C addresses</i>		
IP5306	0x75	versión modificada ad hoc de un chip con funciones de power management
MPU6886	0x68	6-Axis MotionTracking Sensor
BMM150	0x10	3-Axis Geomagnetic Sensor

Nota: la IMU ha cambiado a lo largo de las diferentes revisiones, la información brindada corresponde a la más reciente al momento de escribir este texto y concuerda con lo disponible en Cika en dicho momento.¹⁴

Nota: el controlador de display también tiene variantes, las primeras versiones no invertían el color, por lo tanto es común encontrar demos que en nuestros dispositivos presentan los colores invertidos.

Esquemáticos

El correspondiente al M5Core y al bus de interconexión de módulos (M-bus) lo encontramos en la documentación¹⁵, mientras que del M5GO Bottom podemos observar con paciencia una imagen en algo similar¹⁶.

13 PSRAM (Pseudo-Static RAM) en el ESP32 refiere a memoria RAM estática en un bus SPI (al igual que la flash) utilizable además de la RAM interna al chip. Excepto para desarrolladores avanzados, esto corresponde al sistema operativo.

14 Original: MPU9250 ; siguiente: SH200Q+BMM150 ; siguiente: se reemplaza el SH200Q por el MPU6886

15 [https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/schematic/Core/M5-Core-Schematic\(20171206\).pdf](https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/schematic/Core/M5-Core-Schematic(20171206).pdf), el más reciente al

momento de escribir este texto, según se lo refiere aquí: <https://docs.m5stack.com/#/en/core/fire?id=schematic>

16 https://docs.m5stack.com/#/en/base/m5go_bottom?id=schematic