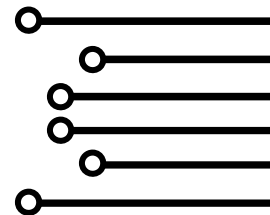
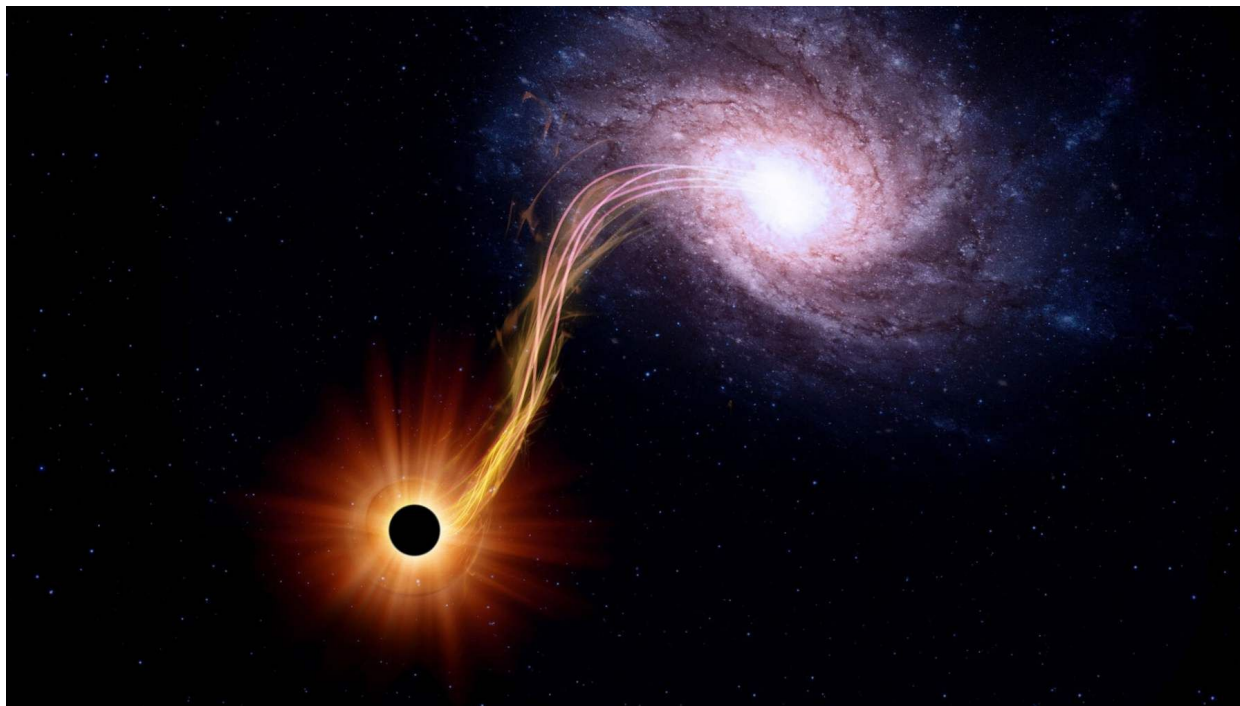




C Embebido



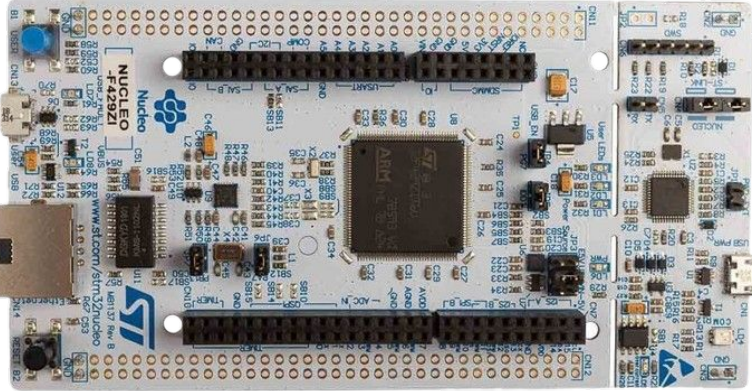


¿ANTES DEL HOLA MUNDO QUE HABÍA?

Startup Code

```
main.c startup_stm324290tx.s system_stm3240xx.c
25 */
26
27 .syntax unified
28 .cpu cortex-m4
29 .fpu softvfp
30 .thumb
31
32.global g_pfnVectors
33.global Default_Handler
34
35/* start address for the initialization values of the .data section.
36defined in linker script */
37.word _sidata
38/* start address for the .data section. defined in linker script */
39.word _sdata
40/* end address for the .data section. defined in linker script */
41.word _edata
42/* start address for the .bss section. defined in linker script */
43.word _sbss
44/* end address for the .bss section. defined in linker script */
45.word _ebss
46/* stack used for SystemInit_ExtMemCtl; always internal RAM used */
47
48/**
49 * @brief This is the code that gets called when the processor first
```

- Código de arranque / Código de Inicio.
- Se utiliza para configurar registros de hardware, establecer la pila y el punto de entrada para el programa principal.
- Normalmente se encuentra .S
- Depende mucho de la herramienta a usar.



```
C:\msys5> startup_tm32429zbx X C:\system_tm32429zbx
25 */
26
27 .syntax unified
28 .cpu cortex-m4
29 .fpu softvfp
30 .thumb
31
32.global _g_pfnVectors
33.global Default_Handler
34
35/* start address for the initialization values of the .data section.
36defined in linker script */
37.word _sidata
38/* start address for the .data section, defined in linker script */
39.word _sdata
40/* end address for the .data section, defined in linker script */
41.word _edata
42/* start address for the .bss section, defined in linker script */
43.word _sbss
44/* end address for the .bss section, defined in linker script */
45.word _ebss
46/* stack used for SystemInit_ExtMemCtl; always internal RAM used */
47
48/**
49 * @brief This is the code that gets called when the processor first
```

Veamos el StartUp Code

Reset

Table 7.3 CMSIS-Core Exception Definitions

Exception Number	Exception Type	CMSIS-Core Enumeration (IRQn)	CMSIS-Core Enumeration Value	Exception Handler Name
1	Reset	-	-	Reset_Handler
2	NMI	NonMaskableInt_IRQn	-14	NMI_Handler
3	Hard Fault	HardFault_IRQn	-13	HardFault_Handler
4	MemManage Fault	MemoryManagement_IRQn	-12	MemManage_Handler
5	Bus Fault	BusFault_IRQn	-11	BusFault_Handler
6	Usage Fault	UsageFault_IRQn	-10	UsageFault_Handler
11	SVC	SVC_IRQn	-5	SVC_Handler
12	Debug Monitor	DebugMonitor_IRQn	-4	DebugMon_Handler
14	PendSV	PendSV_IRQn	-2	PendSV_Handler
15	SYSTICK	SysTick_IRQn	-1	SysTick_Handler
16	Interrupt #0	(device-specific)	0	(device-specific)
17	Interrupt #1 - #239	(device-specific)	1 to 239	(device-specific)

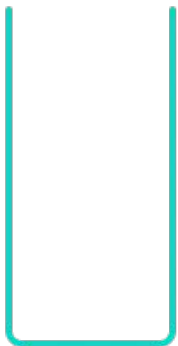
Todo programa inicia en el Vector Reset



Stack (Pila)

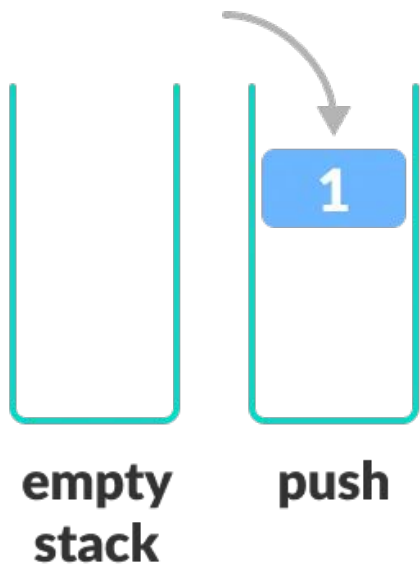
Stack

- Es un lugar de la memoria donde se guardan temporalmente datos mientras se ejecutan.



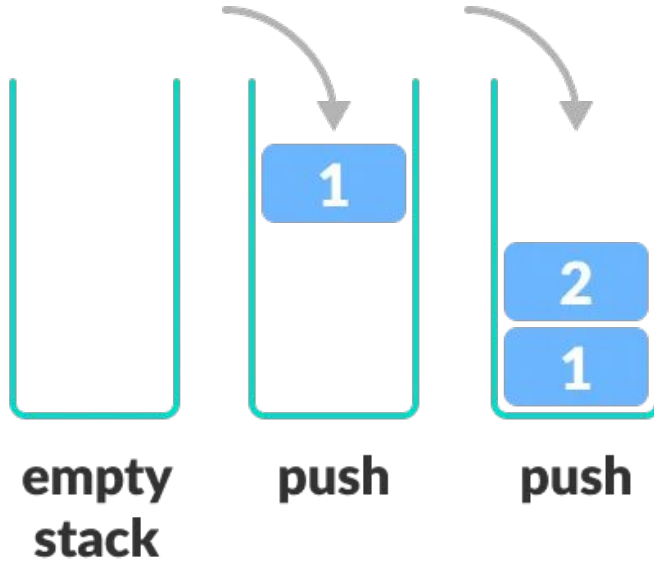
**empty
stack**

Stack



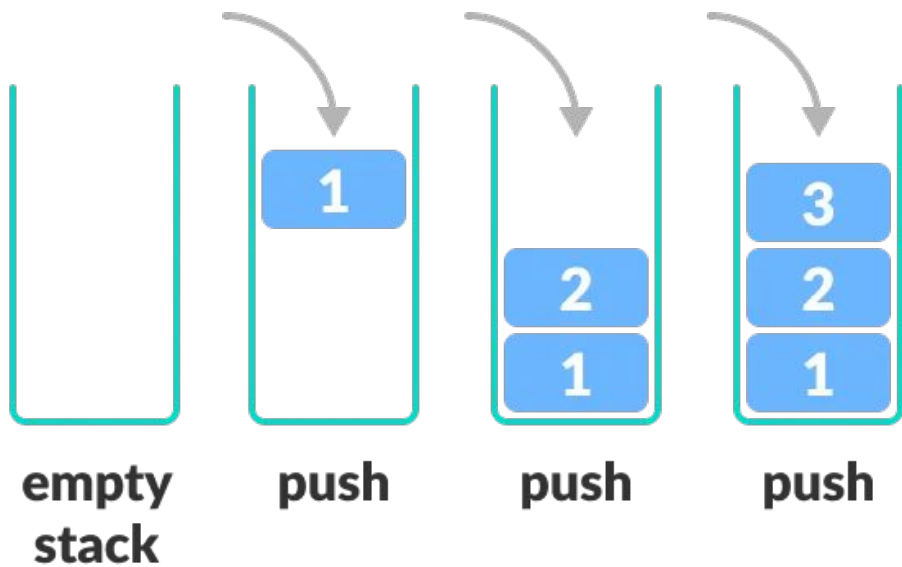
- Push: agrega un elemento en lo más alto de la pila.

Stack

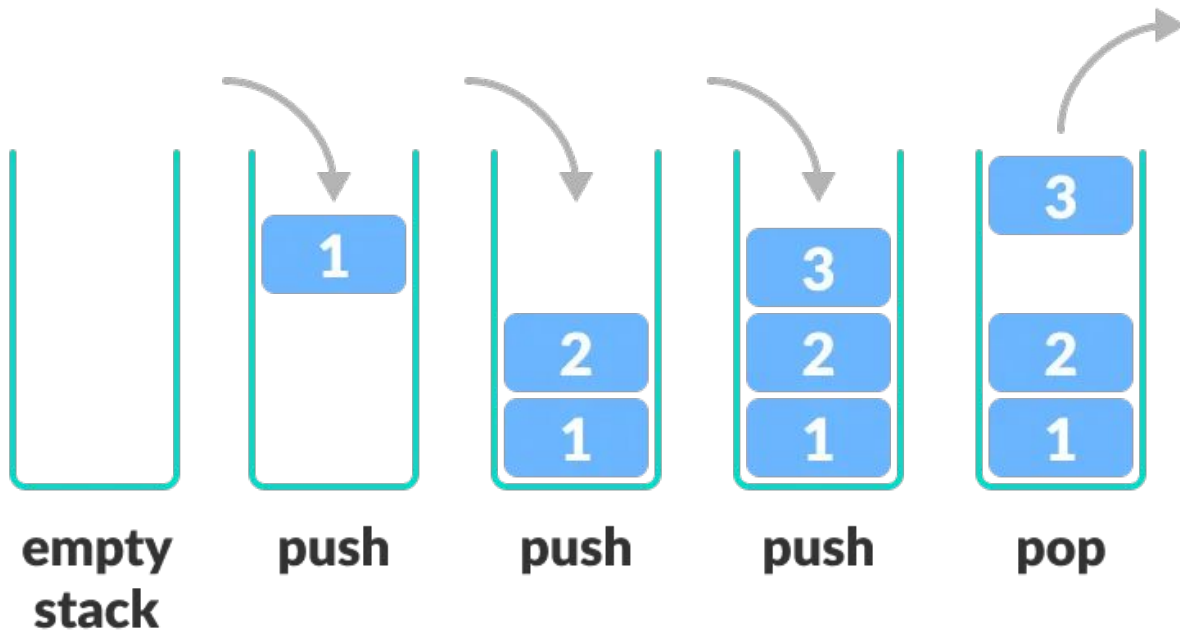


- Push: agrega un elemento en lo más alto de la pila.

Stack

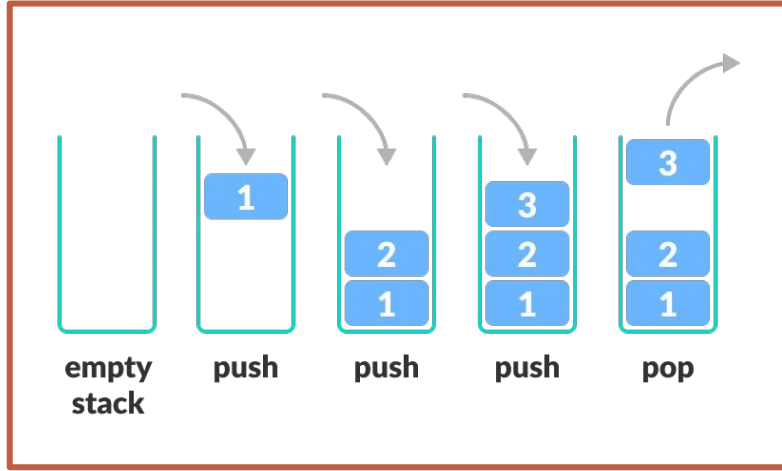


Stack



- Pop: retira un elemento de la pila.

Stack



- Almacenamiento temporal de datos, cuando una función se está ejecutando.
- Direcciones de retorno.
- Almacenar variables locales.
- Para mantener el estado del CPU y registrar valores en caso una interrupción.



Stack Pointer



Gracias

@welstheory

hola@welstheory.com

+51 918 899 684

