



ESP8266 Memory Map

This page describes the physical memory layout of the ESP8266 family.

Reset Vector

The reset vector is 40000080h, which maps to internal ROM.

Memory Layout

Address	Name	Size	Type	R/W	Description
00000000h			Exc		Causes fault when reading.
20000000h			No		Unmapped, repeated pattern of 00 80 00 00.
3FF00000h	dport0	1000h	I/O	RW?	Memory-mapped I/O, repeated every 100h.
3FF10000h			No		Unmapped, zeroes.
3FF20000h	?		?	RW?	Unidentified data.
3FF30000h			No		Unmapped, zeroes.
3FFC0000h	?	20000h	?	RW?	uint32 mapping to the address it is located at. What is this?
3FFE0000h			No		Unmapped, zeroes.
3FFE8000h	dram0	14000h	RAM	RW	User data RAM. Available to applications.
3FFFC000h		4000h	RAM		ETS system data RAM.
40000000h	brom?	10000h	ROM	RW?	Internal ROM. May be writable somehow, but details unknown.
40010000h			No		Zeroes. Assumed to be unmapped.
40100000h	iram1	8000h	RAM	RW	Instruction RAM. Used by bootloader to load SPI Flash <40000h.
40108000h			?		Zeroes.
40140000h			?		Repeated pattern of 59 31 d8 ec.
40200000h			?		SPI Flash is mapped here. Mapping hardware has 32KBytes cache (Unconfirmed). Non-cached code runs 12...13 times slower than code from IRAM. Cached code runs as fast as from IRAM. This area is readable as data with aligned 4-byte reads.
40300000h			?		Unmapped, repeated pattern of 00 80 00 00.
60000000h	?	1000h	I/O	RW?	Uncached I/O
60001000h	?	800h	?	RW?	Uncached I/O
60001800h	?	800h	?	RW?	Uncached. Mapped to 60001000h?
60002000h			Exc		Causes fault when reading.
70000000h		90000000h	No		Unmapped, repeated pattern of 00 80 00 00.

SPI Flash ROM Layout (without OTA upgrades)			
Address	Size	Name	Description
00000h	248k	app.v6.flash.bin	User application
3E000h	8k	masterdevicekey.bin	OTA device key. Unconfirmed: Not used without OTA
40000h	240k	app.v6.irom0text.bin	SDK libraries
7C000h	8k	espinitdata_default.bin	Default configuration, see note below
7E000h	8k	blank.bin	Filled with FFh. May be WiFi configuration

This is for ESP IoT SDK version 0.8 and above. Note that the address at which the default configuration is stored is always 16K from the end of the Flash memory. The address given in the table above is for a 512KB Flash chip.

SPI Flash ROM Layout (with OTA upgrades)

Address	Size	Name	Description
00000h	4k	boot.bin	Bootloader
01000h	64k	app.v6.flash1.bin	User application, slot 1
11000h	180k	app.v6.irom0text1.bin	SDK libraries, slot 1
3E000h	8k	masterdevicekey.bin	OTA device key
40000h	4k		Unused
41000h	64k	app.v6.flash1.bin	User application, slot 2
51000h	180k	app.v6.irom0text1.bin	SDK libraries, slot 2
7C000h	8k	espinitdata_default.bin	Default configuration, see note below
7E000h	8k	blank.bin	Filled with FFh. May be WiFi configuration.

This is for ESP IoT SDK version 0.8 and above, supporting OTA upgrades. Note that the address at which the default configuration is stored is always 16K from the end of the Flash memory. The address given in the table above is for a 512KB Flash chip.

Exception Vectors

Address	Name
40000010h	DebugException
40000020h	NMIException
40000030h	KernelException
40000050h	UserException
40000070h	DoubleException
40000080h	Reset

Memory-Mapped I/O Registers

Base Address	Size	Name	Description
3FF00000h–	?	dport0	3FF00014h is a clock setting register. See note 2.

Base Address	Size	Name	Description
60000000h	80h	uart0	The UART0 config registers, see <code>examples/IoTDemo/include/drivers/uartregister.h</code>
60000100h	100h	spi1	Unconfirmed: The SPI controller registers, see <code>examples/IoTDemo/include/driver/spiregister.h</code>
60000200h	100h	spi0	Unconfirmed: The SPI controller registers, see <code>examples/IoTDemo/include/driver/spiregister.h</code>
60000300h	74h	gpio	Unconfirmed: The timer config registers, see <code>include/eaglesoc.h</code> <code>60000600h</code> <code>28h</code> <code>timer</code> Unconfirmed: The timer config registers, see <code>include/eaglesoc.h</code>
60000700h	A4h	rtc	Unconfirmed: The RTC config registers, see <code>include/eaglesoc.h</code> <code>60000800h</code> <code>44h</code> <code>iomux</code> The IO MUX config registers, see <code>include/eaglesoc.h</code> . See note 3.
60000F00h	80h	uart1	

1. Most of the memory-mapped I/O Registers live in 60000000h–60000FFFh.
2. Setting bit 0 changes the CPU to 160 MHz mode. Clearing it means 80 MHz. Note that you need to call `osupdatecpufrequency(int freqinmhz)` when changing the bit. Probably to calibrate timers. The UART divisor is not updated automatically, so you also have to call `uartdivmodify(uartno, clockfreqinhz / baudrateinbaud)`.
3. iomux Pin Registers (60000804h–60000843h)

```

31      24      16      8      0
----- -ffff--- ----- ud--UDEe
      \- Function          ||  |||\- Output Enable
                          ||  |||\- Output Enable during sleep
                          ||  |||\- Pull-down during sleep
                          ||  ||\- Pull-up during sleep
                          ||  \-\ Pull-down
                          |\-\ Pull-down
                          \-\ Pull-up

```

References

- Forum post Memory Layout, p274 [<http://www.esp8266.com/viewtopic.php?f=5&t=9&start=30#p274>]
- Forum post Firmware Dump, p263 [<http://www.esp8266.com/viewtopic.php?f=6&t=39&start=10#p263>]
- Forum post Memory Layout, p889 [<http://www.esp8266.com/viewtopic.php?f=5&t=9&start=50#p889>]
- Forum post Cloud update documentation, p2486 [<http://www.esp8266.com/viewtopic.php?f=5&t=454#p2486>]
- `lx106-rc-2010.1/xtensa-elf/lib/xt2000-rt/memmap.xmm`
- `espiotsdkv0.9.3/examples/IoTDemo/ssl/app/espconn_ssl.c`
- Github melalheart: ESP SDK linker scripts [<https://github.com/metalheart/esp8266/tree/master/ld>]