

# Memória EEPROM

## Objetivo:

- Dados que forem alterados devem persistir mesmo que a placa resete.
- Manter dados de timers e limiares:
  - Timers
    - Keep Alive
    - Warn DutyCycle
    - Warn Tx
  - Limiares
    - De bateria
    - De luminosidade
    - De movimento (Opção as serem escolhida).
- Contar números de falhas ocorridas.
- Flag indicando que se quer gravar na serial as chaves.

## Solução:

- Gravar dados na memória não volátil EEPROM, memória que permite ler e escrever e (sobrescrever).
- Procurar endereço de memória EEPROM do STM32.
  - 0x8080000
- Na primeira vez que é ligada, a EEPROM estará vazia (apagar pelo J-link ou ST-link). Cuidado: esse não necessita selecionar o setor para apagar.
- Na primeira vez será colocado os dados padrões de tempos, limiares e uma flag de início, na primeira memória EEPROM.
- Toda vez que algum dado for alterado no firmware será gravado na EEPROM no endereço correspondente.

## Disposição da memória

Target memory (Entire flash chip) \*

Address: 0x8080000 x1 x2 x4

Address	0	4	8	C	ASCII
8080000	FEFAE4B3	000249F0	00007530	0001D4C0	.....I...u0...
8080010	00002710	000007D0	00000001	00000000	.....'.....
8080020	0000FC09	00050FB9	0004BB68	00000000	.....h.....
8080030	00000010	00000100	00000000	00000000	.....
8080040	00000000	00000000	00000000	00000000	.....
8080050	00000000	00000000	00000000	00000000	.....
8080060	00000000	00000000	00000000	00000000	.....

- **Endereço 0x8080000: FLAG** de início **0xFEFAE4B3**. Se flag está escrita na memória indica que já ocorreu gravação inicial.

- **Endereço 0x8080004: Keep Alive.** Valor em milisegundos.
- **Endereço 0x8080008: Warn Tx (tempo de envio dentro do estado de alerta).** Valor em milisegundos.
- **Endereço 0x808000C: Warn DutyCycle (tempo do estado de alerta).** Valor em milisegundos.
- **Endereço 0x8080010: Limiar Luminosidade.** Valor em lux.
- **Endereço 0x8080014: Limiar Bateria.** Valor em centiVolts. Deve-se dividir esse valor por 100 para obter então o valor em volts.
- **Endereço 0x8080018: Tipo de limiar do movimento.** Composto de 3 bytes:
  - **Primeiro** byte: Representa o valor do limiar de movimento representado em segundos. Dado o valor em hexadecimal “*val*”.
    - A conta é feita por:  $(val - 1) * 0.625 = \text{resultado em segundos}$ .
    - Vai de 0 (0x00) segundos até 39,375 (0x3F) segundos.
  - **Segundo** byte: informam o limiar de detecção de queda livre da seguinte forma, a priori utiliza somente 3 bits:

bbb (binário)	Limite	bbb (binário)	Limite
0001	156 mg	0101	344 mg
0010	219 mg	0110	406 mg
0011	250 mg	0111	469 mg
0100	312 mg	1000	500 mg

- **Terceiro** byte: Representa o ângulo limite.
  - Enumeração entre 1 e 4:
    - ✦ 1 → limiar de 80º
    - ✦ 2 → limiar de 70º
    - ✦ 3 → limiar de 60º
    - ✦ 4 → limiar de 50º
- **Endereço 0x8080020: Código de produto** (5 dígitos em decimais transformado em hexas).
- **Endereço 0x8080024: Número de série.** (6 dígitos em decimais transformado em hexas).
- **Endereço 0x808028: Corresponda data de fabricação.** São 6 dígitos decimais, 2 primeiros são o dia, 2 seguintes o mês e os dois últimos o ano.
- **Endereço 0x8080030: Versão de hardware.**
- **Endereço 0x8080034: Versão de software.**
- **Endereço 0x8080040: Contador de falhas.** Nesse caso, recebe-se uma palavra completa (4 bytes) e trata cada byte de forma separada.

Endereço -> 8080060  
Valor -> 00000000

- **Byte 1** – Conta quantas falhas do tipo – “usage\_fault”. Pode ir de 0 até 0xFF – 255.
  - **Byte 2** - Conta quantas falhas do tipo na memória – “mem\_fault”. Pode ir de 0 até 0xFF – 255.
  - **Byte 3** – Conta quantas falhas do tipo barramento – “bus\_fault”. Pode ir de 0 até 0xFF – 255.
  - **Byte 4** – Conta quantas falhas do tipo hardware – “hard\_fault”. Pode ir de 0 até 0xFF – 255.
- **Endereço 0x8080050:** Contém flag para usar valores passados pela serial. Caso tenha que usar os valores seriais, a o endereço deve ser preenchido com o valor: **0x5EF23312**, caso contrário deve estar zerado.