



Nota de Aplicación: CAN-100

Título: **Grabación de datos en flash con Holtek ARM HT32F**

Autor: Sergio R. Caprile, Senior R&D Engineer

Revisiones	Fecha	Comentarios
0	23/03/12	

La flash del micro está dividida en segmentos de 1KB. Cada uno de ellos puede borrarse por separado y cada palabra dentro de este segmento puede escribirse mediante un conjunto de registros del FMC (Flash Memory Controller). Un segmento especial de flash contiene los bits de configuración de protección, lo que en otras arquitecturas se conoce como option byte, option register, fuses, etcétera. Es posible utilizar el resto de este segmento para almacenar datos de calibración, de configuración, o lo que el developer considere oportuno. Realizaremos esto en esta nota.

Para operar sobre la flash disponemos de un registro donde escribimos la operación a realizar, otro donde indicamos la dirección de memoria, y otro donde escribimos el dato que queremos escribir. Un flag de estado nos indica cuándo ha concluido la operación.

Por ejemplo, podemos borrar la flash correspondiente al “option-byte” mediante la siguiente función:

```
void FLASH_erase(void)
{
    HT_FLASH->OCMR = FLASH_CMD_PAGEERASE;
    HT_FLASH->TADR = 0x1FF00000;
    HT_FLASH->OPCR = FLASH_SEND_MAIN;
    while((HT_FLASH->OPCR & FLASH_READY) != FLASH_READY);
}

```

De modo similar, podemos escribir una determinada cantidad de datos en el mismo segmento mediante esta función:

```
void FLASH_write(uint32_t *faddr, uint32_t *addr, int len)
{
    if(len&3)
        len=(len>>2)+1;
    else len>>=2;
    HT_FLASH->OCMR = FLASH_CMD_PROGRAM;
    while(len--){
        HT_FLASH->TADR = (uint32_t)faddr++;
        HT_FLASH->WRDR = *(addr++);
        HT_FLASH->OPCR = FLASH_SEND_MAIN;
        while((HT_FLASH->OPCR & FLASH_READY) != FLASH_READY);
    }
}

```

La función recibe una cantidad de bytes y escribe una cantidad de words.

Para poder grabar algo en ese segmento de flash, debemos primero salvar el contenido del option-byte, borrar el segmento y luego volver a escribir el option-byte y grabar los datos. Para esto, definimos una estructura para poder almacenar el option-byte, y otra que nos permita tener un objeto de configuración que guardar y recuperar todo junto:

```
typedef struct {
    int data;
    uint8_t wasted3bytes;
} myconfig_t;

myconfig_t myconfig;

typedef struct {

```

CAN-100, Grabación de datos en flash con Holtek ARM HT32F

```
    uint32_t data[9];
} optbyte_t;

void saveconfig(void)
{
    optbyte_t saveopt;

    saveopt=(*(optbyte_t *)0x1FF00000);          // salva option byte en stack
    FLASH_erase();                             // borra página (segmento)
    // vuelve a grabar option-byte
    FLASH_write((uint32_t *)0x1FF00000, (uint32_t *)&saveopt, sizeof(optbyte_t));
    // graba datos del usuario
    FLASH_write((uint32_t *)0x1FF00050, (uint32_t *)&myconfig, sizeof(myconfig));
    HT_FLASH->OPCR = FLASH_IDLE;
}
}
```

Finalmente, llamando a esta función podemos guardar en flash un grupo de datos de más de 900 bytes. La flash está en el mapa de memoria, por lo cual podemos leerla directamente o copiarla a RAM para poder modificarla. Para accederla deberemos armar un puntero con la dirección física (y usarlo):

```
myconfig=(*(myconfig_t *)0x1FF00050);
```