Sistemi Embedded 2017/2018

Laboratorio 1: GPIO

Sulla schedina di sviluppo:

- Il pulsante "P3.7" è collegato al pin 7 della porta GPIO 3 attraverso il jumper J1. La pressione del pulsante risulta in uno "0".
- Il Led verde "P1.6" è collegato al pin 6 della porta GPIO 1 attraverso il jumper J3. Il Led si accende se l'uscita viene portata a "1".

Si consideri il seguente programma in Assembly, in cui sono stati rimossi i commenti al codice nel Main Loop:

```
$NOMOD51                    ; Ask Keil not to define 8051 registers
$INCLUDE (C8051F020.INC)    ; SFR definitions for the C8051F020

;-------------------------------------------------------------------------------
; Reset and Interrupt Vectors
cseg at 000h                ; Define an absolute code segment located at 0x00
ljmp   Reset                ; On reset, jump to label "Reset"

;-------------------------------------------------------------------------------
; Code segment "Main"
Main segment CODE           ; The assembler chooses the location of the code segment
rseg Main                   ; Switch to this code segment
using 0                     ; Specify register bank

Reset:
       clr EA               ; Disable all interrupts
       mov WDTCN, #0DEh     ; Disable the watchdog
       mov WDTCN, #0ADh     ;
       anl OSCICN, #014h    ; Disable missing clock detector
                            ; and set internal osc at 2 MHz as the clock source
       mov XBR0, #000h      ; Set and enable the crossbar
       mov XBR1, #000h      ;
       mov XBR2, #040h      ;
       orl P1MDOUT, #040h   ; Set P1.6 to push-pull

MainLoop:
       jnb P3.7, LedOn
       clr P1.6
       jmp MainLoop

LedOn:
       setb P1.6
       jmp MainLoop

;-------------------------------------------------------------------------------
; End of file
end
```

Creare un nuovo progetto nell'ambiente di sviluppo, trascrivere questo codice e compilare.

Se la compilazione va a buon fine, nel log di compilazione dovrebbe comparire questo messaggio, che tra le altre cose indica la dimensione del programma compilato (37 bytes):

```
Program Size: data=8.0 xdata=0 code=37
LINK/LOCATE RUN COMPLETE.  0 WARNING(S),  0 ERROR(S)
```

Caricare il programma compilato nella memoria del microcontrollore ed eseguire il codice passo-passo (modalità debug) in emulazione sulla schedina di sviluppo.

Cercare una risposta a queste domande:

- A cosa serve questo programma?
- E' davvero necessario impostare P1.6 come push-pull? Perché?
- Quanti e quali interrupt sono usati?
- In che modo viene rilevato lo stato del pulsante P3.7 (interrupt o polling)?
- Qual è il ritardo massimo (a spanne ma non troppo) che può intercorrere tra la pressione del pulsante e la gestione dell'evento da parte del sistema?
- E quello minimo?

Una volta trovata risposta a queste domande, usare il modello sottostante per scrivere un programma equivalente in C:

```
#include <c8051f020.h>    // SFR definitions

// Other definitions
sbit Button = P3^7;
sbit Led = P1^6;

void init (void) {
       EA = 0;
       // . . .
}

void main (void) {
       init();
       while(1){
              // . . .
       }
}
```

- Quali vantaggi e svantaggi riuscite a individuare nell'uso del C rispetto all'Assembly?