

Mestrado Integrado em Engenharia Electrónica Industrial e Computadores

Microprocessadores

2018/2019

Guia 1

Técnicas de programação *Debounce* e *Serial IO*

Vitor Silva

vsilva@dei.uminho.pt

José Mendes

jose.mendes@dei.uminho.pt

Departamento de Electrónica Industrial

Universidade do Minho

Enunciado

Implemente um programa em linguagem C para o microcontrolador C8051F388, que permite interagir com os botões de pressão PB1 e PB2, escrever no display de sete segmentos e enviar a receber parâmetros através da porta série. O programa deve incrementar (PB1) / decrementar (PB2) um valor em memória e apresentar o seu resultado no display. Adicionalmente, este deve ainda incrementar/decrementar o mesmo valor, quando recebe pela porta série os caracteres 'i'/'I' e 'd'/'D' respetivamente, mantendo sempre atualizado o valor apresentado no display e enviando-o através da porta série.

Nota:

- A. A leitura das teclas deve ser feita por *polling* ou com o uso de um timer, sem *debounce*.
- B. Comece por usar a UART0 com o timer 1 no baudrate 115200.
- C. Altere o exercício para a UART1 com o gerador de baudrate interno.

Exercício

Apresente o algoritmo em texto descrevendo a sequência de operações a realizar bem como o conjunto de variáveis de memória, tipo e conjunto de periféricos usados. Elabore um pequeno texto com a configuração de cada um dos periféricos onde apresenta o conteúdo dos registos de configuração. O exercício termina com a tradução do algoritmo texto em instruções C, deve apresentar o programa na sua versão final.

Elementos a entregar

Algoritmo de texto, conjunto de variáveis de memória, configuração de periféricos e ficheiro .c contendo o programa na sua versão final.

Keil C51 e KIT8051 Hello World Cheat Sheet

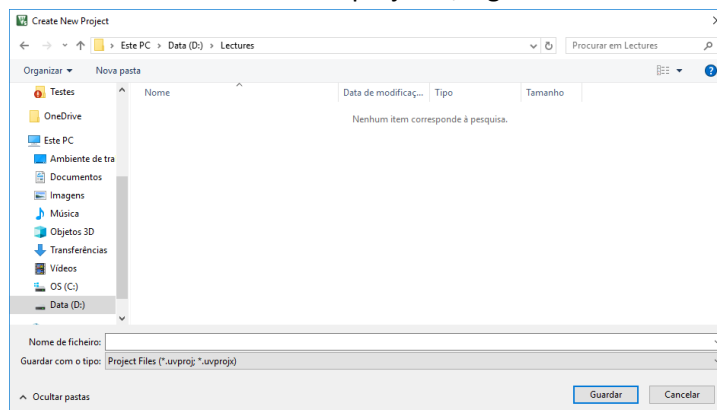


Nesta secção podem encontrar um pequeno guia para introduzir o Keil C51 para programar o microcontrolador C8051F388 em C.

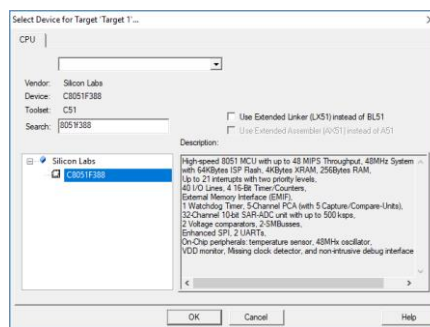
1. Criar o Projecto para trabalhar em C

1.1. Abrir o Keil uVision

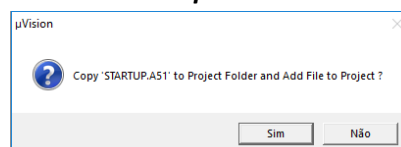
1.2. Na barra de menu, ir a Project e seleccionar New uVision Project. Aparece a seguinte janela, e em **File name** colocar o nome do projecto, e.g. *helloworld* e clicar em **Guardar**.



1.3. Na janela seguinte, seleccionar o microcontrolador C8051F388 e clicar no **OK**

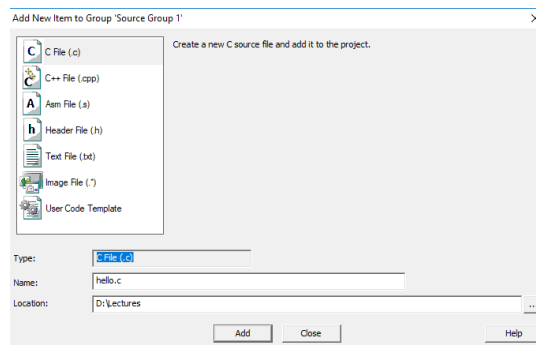


1.4. Aparece a seguinte janela, seleccionando **Não**. O compilador usa este código antes de invocar a função *main*, apenas não estará visível no projecto. (Se copiarmos o ficheiro, temos uma oportunidade para alterar o *startup* do micro em caso de necessidade).



2. Adicionar o ficheiro C

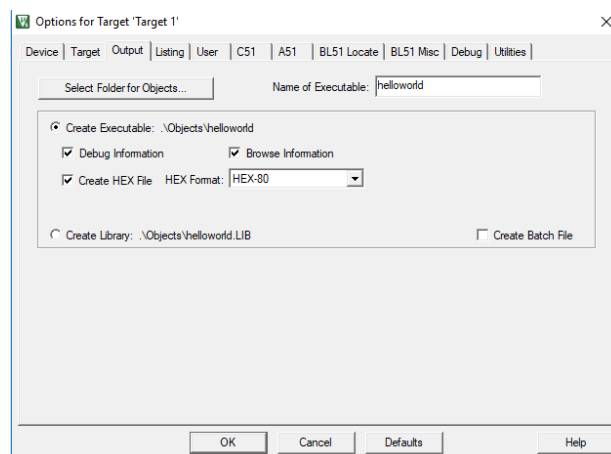
- 2.1. Expandir as pastas na **Project Window**. No menu de contexto (tecla direita do rato) de **'Source Group 1'** seleccionar **'Add New Item to Group 'Source Group 1'...**



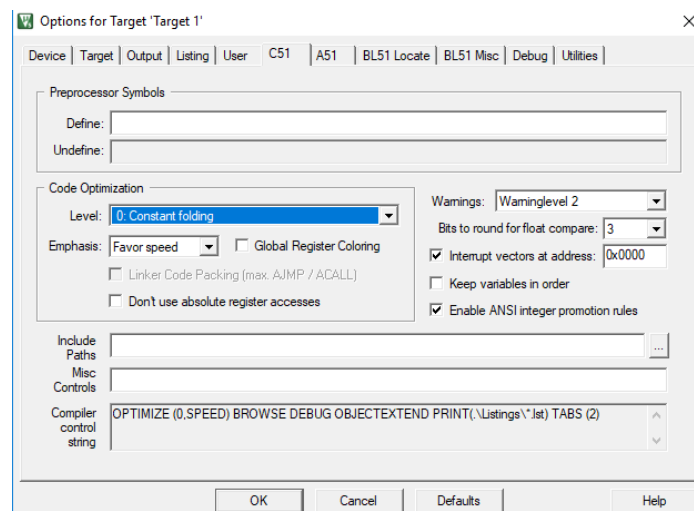
- 2.2. Seleccionar **C File (.c)** e atribuir um nome em **Name** e clicar no **Add**.
- 2.3. Editar o ficheiro com um código de exemplo mínimo para testar o funcionamento:

3. Configurações: Compilador e Flash Programming

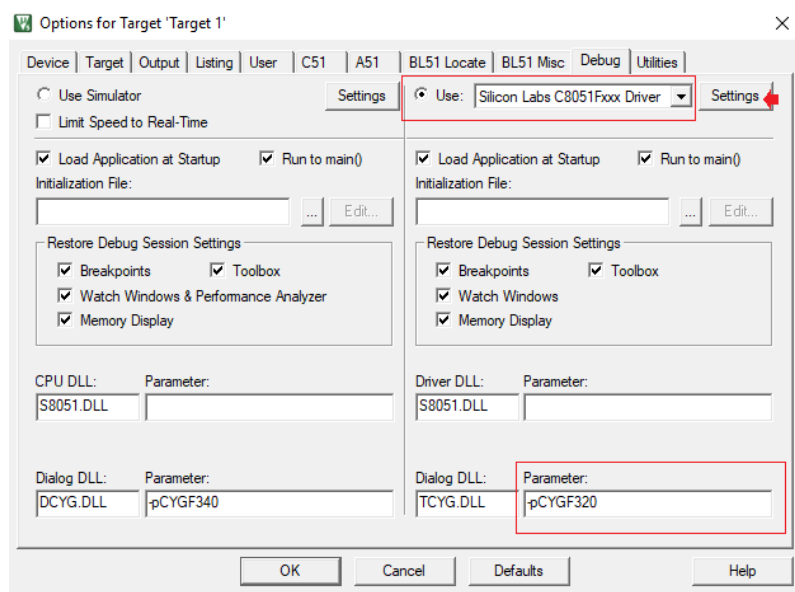
- 3.1. Na barra de menu, ir a Project e seleccionar **Options for Target 'Target 1'**.
- 3.2. Em **Output**, seleccionar **Create HEX File** e **HEX-80**



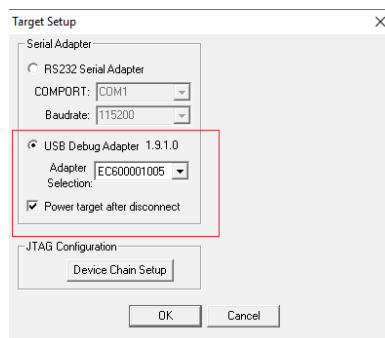
- 3.3. Em **C51** colocar o nível de optimização em 0: **Constant folding**



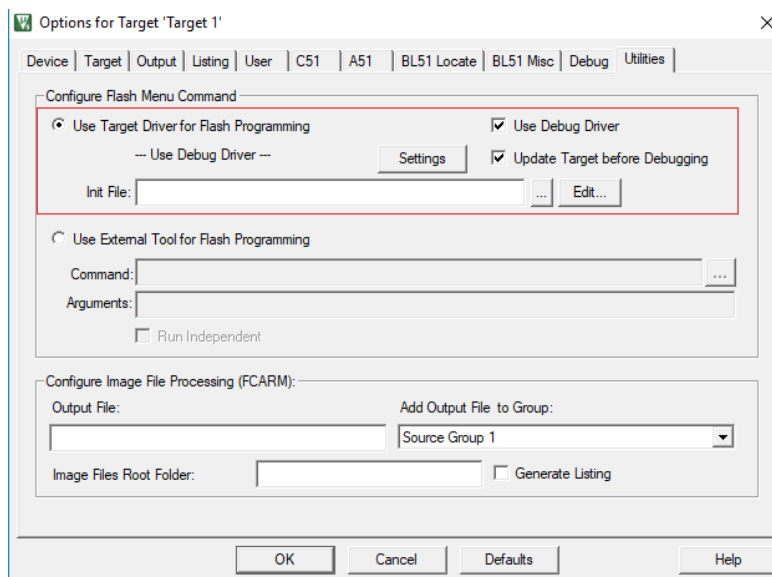
3.4. Em **Debug**, seleccionar a opção: **Use:** e escolher a entrada **Silicon Labs C8051Fxxx Friver**. Altere o parâmetro **Dialog DLL** para **-pCYGF320**.



3.5. Em **Settings**, seleccionar **USB Debug Adapter** conforme a figura:

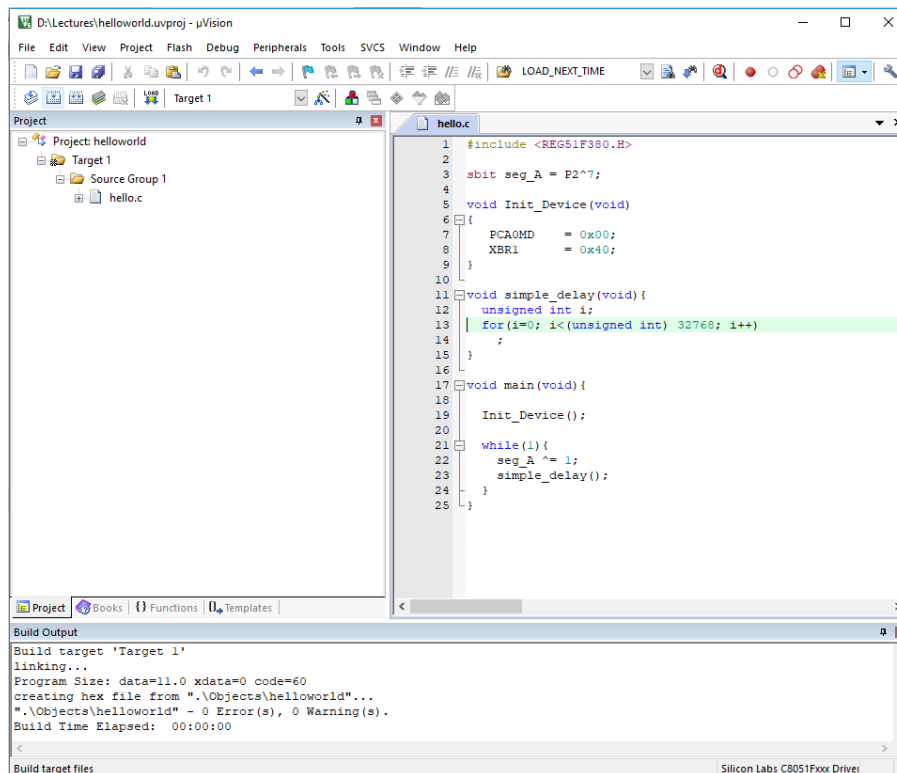


3.6. Em **Utilities** seleccionar: **Use Target Driver for flash programming**, conforme a figura seguinte.



4. Compilar e testar

4.1 Na barra de ferramentas, carregar em **Build**.



4.2 Clicar em **Load**, **Sim** para *Erase* e **Ok** para programar.

O segmento A do BCD deve ficar a piscar.

5. Hello World!

```
g1.c u1.c
1  #include <REG51F380.H>
2  #include <stdio.h>
3
4  void print_u0(char* p_message);
5
6  void Oscillator_Init() {
7      PCA0MD = 0;
8      FLSCCL   = 0x90;
9      CLKSEL   = 0x03;
10 }
11
12 void UART_Init() {
13     XBR0      = 0x01;
14     XBR1      = 0x40;
15     SCON0     = 0x10;
16     TMOD      = 0x20;
17     CKCON     = 0x08;
18     TH1       = 0x30;
19     TR1 = 1;
20     TI0 = 1;
21 }
22
23 void main (void) {
24
25     char message[] = "Hello World\r\n";
26
27     Oscillator_Init();
28
29     UART_Init();
30
31     // just u0
32     printf("Hello 8051\n");
33
34     print_u0(message);
35
36     while(1);
37 }
```

6. UART 1 com Baudrate interno

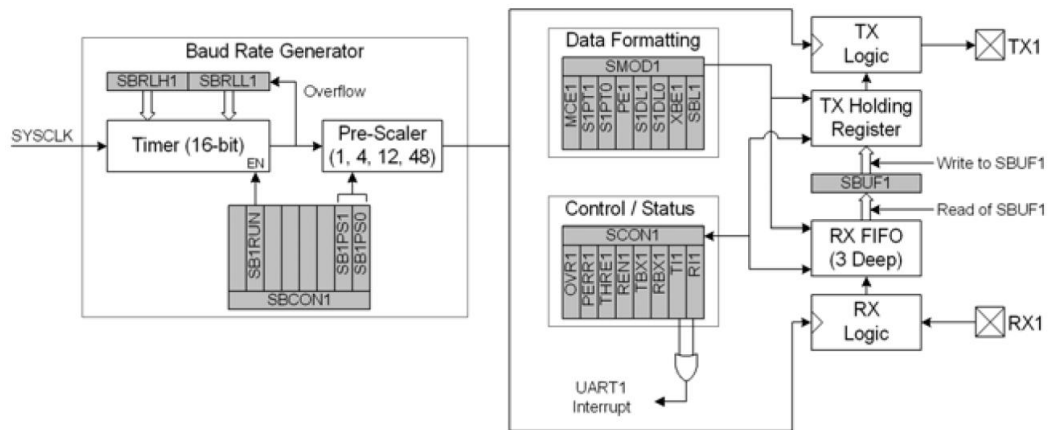


Figure 23.1. UART1 Block Diagram

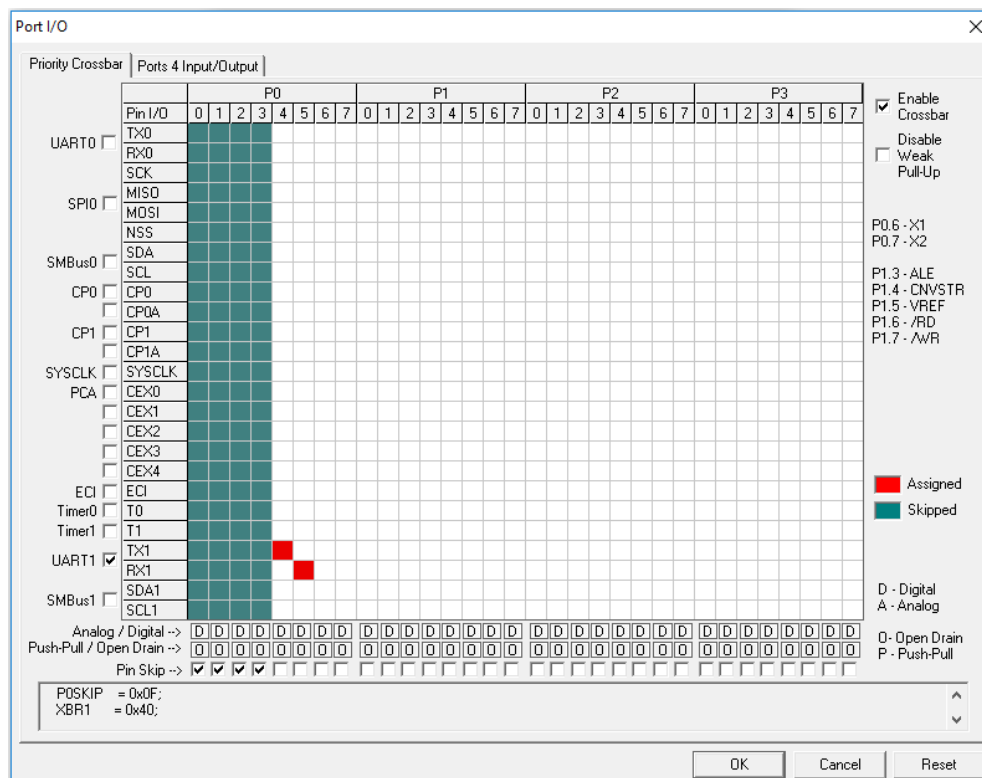


Table 23.1. Baud Rate Generator Settings for Standard Baud Rates

	Target Baud Rate (bps)	Actual Baud Rate (bps)	Baud Rate Error	Oscillator Divide Factor	SB1PS[1:0] (Prescaler Bits)	Reload Value in SBRLH1:SBRL1
SYSCLK = 12 MHz	230400	230769	0.16%	52	11	0xFFE6
	115200	115385	0.16%	104	11	0xFFCC
	57600	57692	0.16%	208	11	0xFF98
	28800	28846	0.16%	416	11	0xFF30
	14400	14388	0.08%	834	11	0xFE5F
	9600	9600	0.0%	1250	11	0xFD8F
	2400	2400	0.0%	5000	11	0xF63C
	1200	1200	0.0%	10000	11	0xEC78
SYSCLK = 24 MHz	230400	230769	0.16%	104	11	0xFFCC
	115200	115385	0.16%	208	11	0xFF98
	57600	57692	0.16%	416	11	0xFF30
	28800	28777	0.08%	834	11	0xFE5F
	14400	14406	0.04%	1666	11	0xFCBF
	9600	9600	0.0%	2500	11	0xFB1E
	2400	2400	0.0%	10000	11	0xEC78
	1200	1200	0.0%	20000	11	0xD8F0
SYSCLK = 48 MHz	230400	230769	0.16%	208	11	0xFF98
	115200	115385	0.16%	416	11	0xFF30
	57600	57554	0.08%	834	11	0xFE5F
	28800	28812	0.04%	1666	11	0xFCBF
	14400	14397	0.02%	3334	11	0xF97D
	9600	9600	0.0%	5000	11	0xF63C
	2400	2400	0.0%	20000	11	0xD8F0
	1200	1200	0.0%	40000	11	0xB1E0

$$\text{Baud Rate} = \frac{\text{SYSCLK}}{(65536 - (\text{SBRLH1:SBRL1}))} \times \frac{1}{2} \times \frac{1}{\text{Prescaler}}$$

Equation 23.1. UART1 Baud Rate


```

6 //Uart1
7 //-----
8 // SCON 1
9 //-----
10 // 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
11 //-----
12 // OVR1 | PERR1 | THRE1 | REN1 | TBX1 | RBX1 | TI1 | RI1
13 //-----
14
15 #define B_REN1 6
16 #define B_TI1 1
17 #define B_RI1 0
18
19 //-----
20 // SBCON 1
21 //-----
22 // 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
23 //-----
24 // - | SBRUN | | SB1PS1|SB1PS0
25 //-----
26
27 #define B_SBRUN 6
28 #define B_SB1PS1 1
29 #define B_SB1PS0 0
30
31 void UART_I_Init(){
32
33 //Crossbar
34 POSKIP = 0x0F;
35 XBR1 = 0x40;
36 XBR2 = 0x01;
37
38 //Baudrate 115200
39 SBRL1 = 0x30;
40 SBRLH1 = 0xFF;
41
42 //SBCON1 = 0x43;
43 SBCON1 = (1<<B_SB1PS1) | (1<<B_SB1PS0);
44 SBCON1 |= (1<< B_SBRUN);
45
46 SCON1 |= (1<<B_TI1);
47 }
48

```