

VITOR AMADEU SOUZA

Programando o

PIC32 em C

Com base no XC32 e MPLAB X

© 2014 by Cerne Tecnologia e Treinamento Ltda.

© 2014 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Esta apostila publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Outubro de 2014

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: Cerne Tecnologia e Treinamento

E-mail da Empresa: cerne@cerne-tec.com.br

Home Page: www.cerne-tec.com.br.com.br

Atendimento ao Consumidor: sac@cerne-tec.com.br

Contato com o Autor: vitor@cerne-tec.com.br

Dedicatória

Como nos meus outros
livros, dedico este livro a minha querida esposa Renata Leal.

***“Uma civilização é julgada pelo tratamento que
dispensa às minorias.”***

Gandhi

Prefácio

A Microchip vem lançando novidades a cada dia e a linha PIC32 faz parte deste enorme leque. Estes novos Cis, que vem equipados com recursos de microcontroladores de 32 bits estão cada dia mais fazendo parte dos novos projetos eletrônicos. Este livro, chamado de *Uma Nova Tecnologia* vem para apresentar ao leitor a família PIC32 de microcontroladores e a linguagem C, com base no compilador XC32 desenvolvido pela própria Microchip. O PIC32 estudado é o PIC32MX320F032H em encapsulamento TQFP de 64 pinos. O livro está organizado de forma que o leitor encontrará informações importantes sobre esta família, pois os exemplos e explicações estão desenvolvidos de forma a permitir que o leitor vá galgando, da primeira a última página no desenvolvimento de projetos eletrônicos com os recursos de hardware e software propostos no PIC32 e no XC32.

Sumário

Capítulo 1 – Material Utilizado no Livro	10
1.1 Conhecendo as Ferramentas usadas no livro.....	10
Capítulo 2 - Conhecendo o MPLAB x	12
2.1 Introdução	12
Capítulo 3 – O Microcontrolador PIC32MX320F032H	21
3.1 Entendendo o PIC32	21
3.2 Características do PIC32MX320F032H	27
Capítulo 4 – Entendendo e Configurando os Configurations Bits.....	29
4.1 Os Configurations Bits do PIC32MX320F032H	29
4.2 PLL Input Divider.....	30
4.3 PLL Multiplier.....	36
4.4 System PLL Output Clock Divider	39
4.5 Oscillator Selection Bits	43
4.5.1 XT, HS, EC	44
4.5.2XTPLL, HSPLL, ECPLL.....	47
4.5.3 FRC	47
4.5.4 FRC/16.....	49
4.5.5 FRCDIV	50
4.5.6 FRCPLL.....	52
4.5.7 LPRC	53
4.5.8 SOSC.....	54
4.6 Secondary Oscillator Enable.....	55
4.7 Internal / External Switch Over	56
4.8 Primary Oscillator Configuration.....	56
4.9 CLKO Output Signal	56
4.10 Peripheral Clock Divisor	57

4.11 Clock Switching and Monitor Selection	60
4.12 Watchdog Timer Enable e Postscaler	62
4.13 ICE/ICD Comm Channel Select	62
4.14 Boot Flash Write Protect	64
4.15 Code Protect	64
4.16 Configurando o Configuration Bits no Programa C	65
4.17 Configuration Bits Usados no Livro	80
4.18 Exercícios	82
Capítulo 5 – Linguagem C	86
5.1 Introdução	86
5.2 Variáveis	86
5.3 Constantes	95
5.4 Vetores	97
5.5 Matrizes	101
5.6 Declaração If	104
5.7 Declaração Switch	110
5.8 Comando For	116
5.9 Comando While	121
5.10 Comando Do While	124
5.11 Outros Operadores da Linguagem	125
5.11.1 Operador Aritmético	125
5.11.2 Operador de Bit	128
5.11.3 Operador Lógico	132
5.13 Exercícios	133
Capítulo 6 – Funções do XC32	134
6.1 Funções Matemáticas	134
6.2 Manipulação de Caracteres	146

Capítulo 7 – Funções do Usuário	155
7.1 Introdução	155
7.2 Exercícios	161
Capítulo 8 – Explorando os I/Os do PIC32	162
8.1 Introdução	162
8.2 Registradores de Configurações de I/O	163
8.2.1 PORTE	164
8.2.2 PORTG, PORTB, PORTF, PORTC e PORTD	177
8.3 Desafios	190
Capítulo 9 – Timers de 16 Bits.....	191
9.1 Introdução	191
9.2 Timer1	193
9.3 Timer2, Timer3, Timer4 e Timer5	210
Capítulo 10 – Timers de 32 Bits	218
10.1 Introdução	218
Capítulo 11 – Compare	226
11.1 Introdução	226
11.2 Modo 000, Compare Desabilitado	230
11.3 Modo 001, Pino OC1 Inicia em 0 e Salta para 1 em Evento de Comparação	230
11.4 Modo 010, Pino OC1 Inicia em 1 e Salta para 0 em Evento de Comparação	233
11.5 Modo 011, Pino OC1 é Invertido em Evento de Comparação	235
11.6 Modo 100, Modo de Geração de Pulso Simples	237
11.7 Modo 101, Modo de Geração de Pulso Continua	240
11.8 Modo 110, Modo PWM sem Pino de Fault	242
11.9 Modo 111, Modo PWM com Pino de Fault	245

Capítulo 12 – UART.....	249
12.1 Introdução	249
12.2 Registradores Utilizados	252
Capítulo 13 – Interrupções.....	271
13.1 Introdução	271
13.2 Interrupção Externa 0	275
13.3 Interrupção Externa 1	282
13.4 Interrupção Externa 2	295
13.5 Interrupção Externa 3	291
13.6 Interrupção Externa 4	295
13.7 Interrupção de Timer1	302
13.8 Interrupção de Timer2, Timer3, Timer4 e Timer5.....	306
13.9 Interrupção de Mudança de Estado.....	314
13.10 Interrupção Input Capture 1,2,3,4 e 5	320
13.11 Interrupção de Output Compare 1, 2, 3, 4 e 5	328
13.12 Interrupção de Transmissão e Recepção Serial.....	335
13.13 Interrupção de Conversão AD.....	339
Capítulo 14 – Capture	342
14.1 Introdução	342
14.2 000 – Modo de Captura Desabilitado	346
14.3 001 – Modo de Detecção de Borda de Subida ou Descida.....	346
14.4 010 – Modo de Captura na Borda de Descida.....	349
14.5 011 – Modo de Captura na Borda de Subida	351
14.6 100 – Modo de Captura na Quarta Borda de Subida	353
14.7 101 – Modo de Captura na Décima-Sexta Borda de Subida.....	356
14.8 110 – Modo de Captura Simples.....	358
14.9 111 – Modo de Interrupção.....	360
14.10 Capture2, Capture3, Capture4 e Capture5	360

Capítulo 15 – Modo de Tensão de Referência	366
15.1 Introdução	366
Capítulo 16 – Comparador	372
16.1 Introdução	372
Capítulo 17 – Conversor AD	385
17.1 Introdução	385
Apêndice 1 – Resposta dos Exercícios	419
Apêndice 2 – Esquema Elétrico do Kit Cerne PIC32.....	423

Capítulo 1 – Material Utilizado no Livro

Neste capítulo é abordado o software e o hardware adotados para os experimentos propostos no livro.

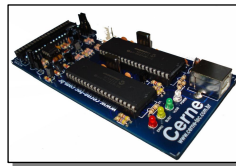
1.1 Conhecendo as Ferramentas Usadas no Livro

Neste livro é usado como ambiente de desenvolvimento no PC o software MPLAB X Versão 2.20 ou superior além do compilador XC32. Ambos podem ser baixados através do site do próprio fabricante, cujo endereço é www.microchip.com. Após o download, proceda a instalação de ambos os programas no seu PC. O XC32 será utilizado em conjunto com o MPLAB X no decorrer deste livro.

Em relação ao hardware para as experiências, serão utilizados o kit didático e o gravador da Cerne Tecnologia, que podem ser visualizados abaixo:



Kit Didático Cerne PIC32



Gravador e Debugador USB para PIC

O gravador funciona em conjunto com o kit didático para gravar o kit PIC32 através da porta USB. Ambos podem ser adquiridos no site www.cerne-tec.com.br.

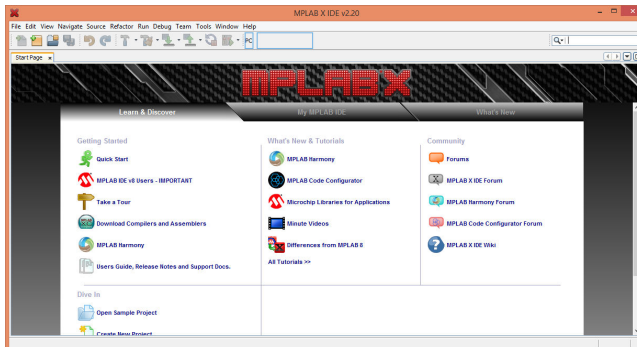
O objetivo do livro é auxiliar o estudante no aprendizado da parte teórica através de cada capítulo apresentado e a seguir poder comprovar o seu funcionamento com a compilação do código e em seguida a gravá-lo na placa didática.

Capítulo 2 – Conhecendo o MPLAB X

Neste capítulo é abordado como criar projetos e compilar usando o software MPLAB da Microchip.

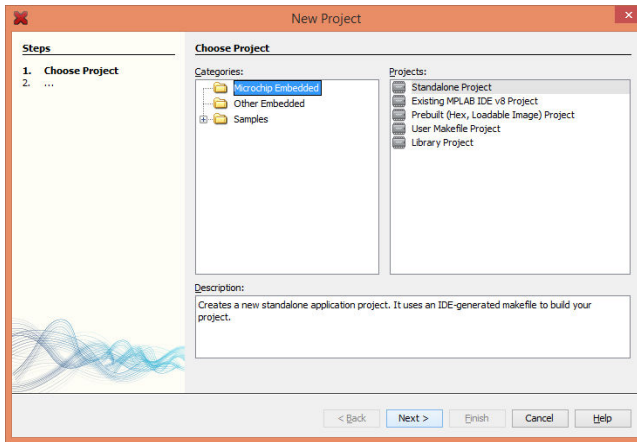
2.1 Introdução

Após o download e instalação do MPLAB X e XC32, inicialize o MPLAB X, a seguinte tela será apresentada.

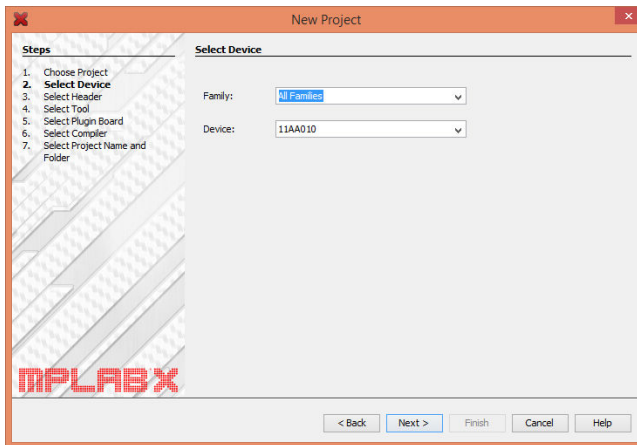


Obs: A versão 2.20 do MPLAB X foi utilizada nesta literatura. Acredita-se que os passos aqui apresentados venham a modificar-se muito pouco nas versões mais recentes.

Para criar um novo projeto, vá em File -> New Project. Neste momento, a seguinte tela surgirá:

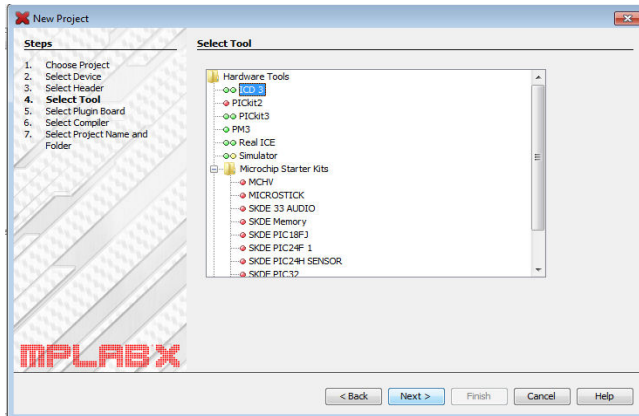


Clique em Next. A seguinte tela será apresentada.

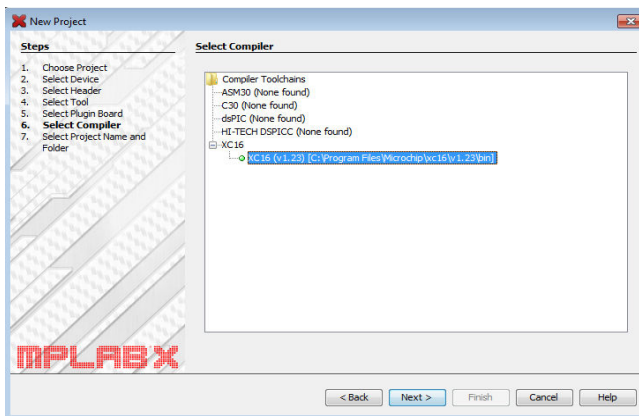


Em family, escolha a opção 32-bit MCUs (PIC32) e em device, escolha o modelo

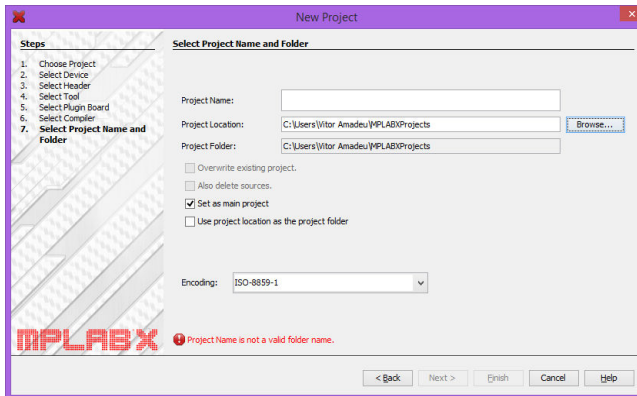
PIC32MX320F032H. Em seguida, pressione Next. A seguinte tela será carregada.



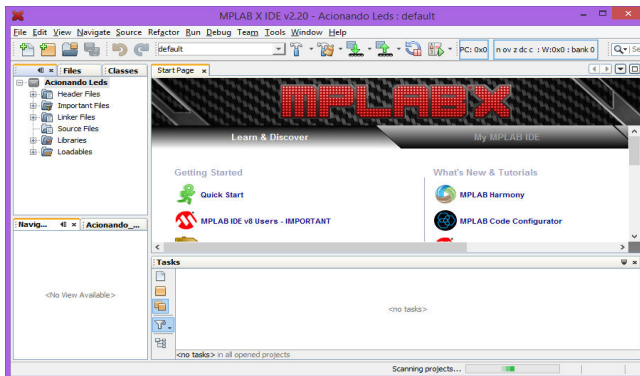
Esta tela seleciona o gravador que será usado para gravar o PIC. Selecione o modelo PICKit 2, já que o gravador GPPIC USB é compatível com este modelo. Em seguida, pressione Next.



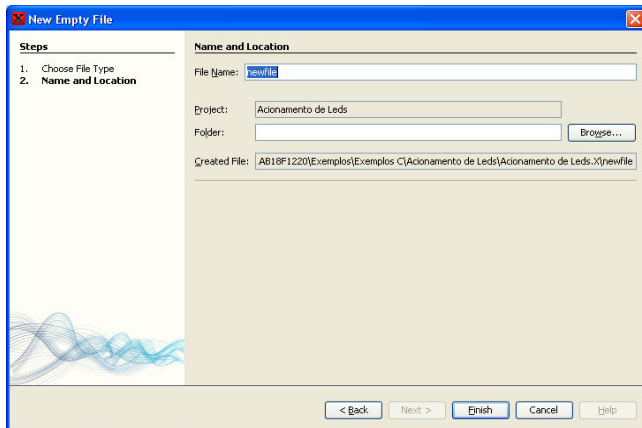
Nesta tela dever-se selecionar o compilador que será usado, selecione neste caso a opção XC32. Pressione o botão Next.



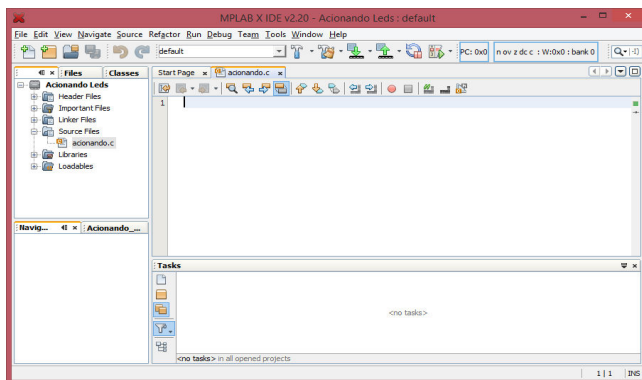
Em *Project Name* informe o nome do projeto. Em *Project Location*, onde o mesmo ficará salvo. Feito isso, pressione o botão Finish. A tela ficará da seguinte forma:



Dê um clique com o botão direito do mouse sobre a opção **Source Files** e em seguida escolha **New -> Empty File**. A seguinte tela surgirá:



Dê um nome ao arquivo fonte e coloque a extensão **.c** e em seguida, clique no botão **Finish**. Teremos a seguinte tela:



Observe que o arquivo criado é o arquivo fonte, onde será desenvolvido os projetos para programar o PIC32. Digite o seguinte código, que é a estrutura básica de um programa C:

```
#include <p32xxxx.h>
#include <sys/attribs.h>

#pragma config FPLLIDIV=DIV_4
#pragma config FPLLMUL=MUL_16
#pragma config FPLLLODIV=DIV_2
#pragma config FNOSC=PRIPLL
#pragma config FSOSCEN=OFF
#pragma config POSCMOD=HS
#pragma config OSCIOFNC=OFF
#pragma config FPBDIV=DIV_1
#pragma config FCKSM=CSDCMD
#pragma config FWDTEN=OFF
#pragma config ICESEL=ICS_PGx1
#pragma config BWP=ON
#pragma config DEBUG=ON

main()
{

}
```

O código apresentado anteriormente é a estrutura básica de um programa em C. A seguir uma análise do mesmo.

```
#include <p32xxxx.h>
//Aqui é informado o processador que
//será utilizado pelo projeto

#pragma config FPLLIDIV=DIV_4
#pragma config FPLLMUL=MUL_16
#pragma config FPLLODIV=DIV_2
#pragma config FNOSC=PRIPLL
#pragma config FSOSCEN=OFF
#pragma config POSCMOD=HS
#pragma config OSCIOFNC=OFF
#pragma config FPBDIV=DIV_1
#pragma config FCKSM=CSDCMD
#pragma config FWDTEN=OFF
#pragma config ICESEL=ICS_PGx1
#pragma config BWP=ON
#pragma config DEBUG=ON

//Configurações básicas do fusível

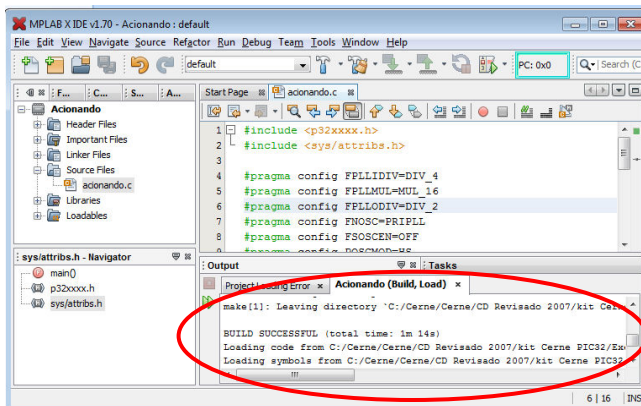
void main(void)
{
    //Esse aqui é o bloco principal
    //de programa.
    //Este é o único
    //bloco que o C precisa para
    //funcionar.
}

//Note que tudo que vem precedido // é
//chamado de comentário do programa e não
//é interpretado pelo compilador
```

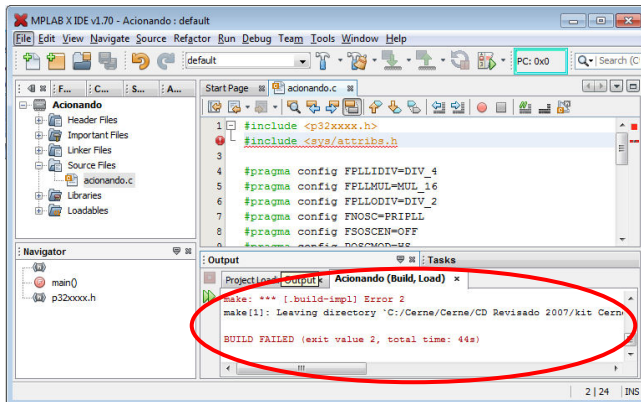
Abaixo um exemplo de comentário de várias linhas:

```
/*  Hoje  
    está  
    quente  
*/
```

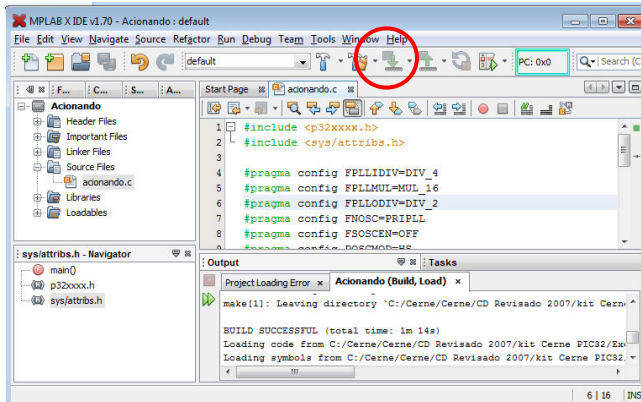
Para compilar o projeto vá no menu *Run -> Build Main Project*. Caso a compilação ocorra sem encontrar erros, o compilador irá informar ao término a seguinte mensagem:



Caso ocorra algum erro, o compilador irá reportar com a mensagem abaixo:



Pode-se compilar e já gravar o microcontrolador diretamente, usando a estrutura de hardware apresentada anteriormente. Para isso, pressione o botão marcado abaixo:



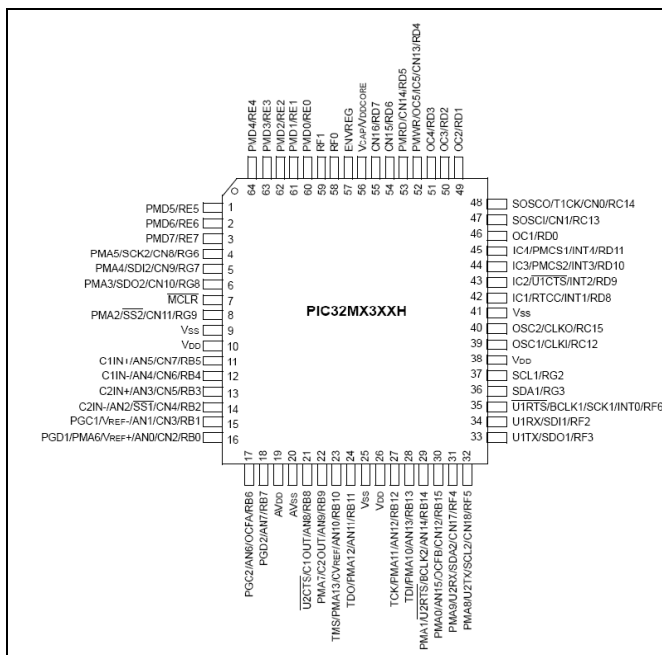
Capítulo 3 – O Microcontrolador

PIC32MX320F032H

Neste capítulo serão estudados os aspectos principais do microcontrolador PIC32MX320F032H.

3.1 Entendendo o PIC32

Os tópicos principais do PIC32, microcontrolador de 32 bits da Microchip, são mostrados mas sem muito aprofundamento pois são vistos nos próximos capítulos. Inicialmente, veja a pinagem deste microcontrolador de 64 pinos e as suas funcionalidades:



Observe que vários pinos têm mais de uma função, como por exemplo, o pino 1 que é PMD5/RE5. Vamos analisar todos os pinos na função básica de I/O (Input e Output ou Entrada e Saída) do mesmo, por exemplo, considere o pino 1 apenas como RE5, pois as funções especiais serão explicadas no decorrer do livro evitando assim que o aluno se confunda nesta parte introdutória. Um pino de I/O é todo aquele que se configurado como entrada pode ler o nível 1 ou 0 e se configurado como saída pode impor o nível 0 ou 1.

Pinos de 1 a 16 do PIC32MX320F032H		
Pino	Nome	Função
1	RE5	Pino de I/O do Microcontrolador
2	RE6	Pino de I/O do Microcontrolador
3	RE7	Pino de I/O do Microcontrolador
4	RG6	Pino de I/O do Microcontrolador
5	RG7	Pino de I/O do Microcontrolador
6	RG8	Pino de I/O do Microcontrolador
7	MCLR	Pino de Reset do Microcontrolador, ou seja, o mesmo é reinicializado se este pino for levado à zero (aterrado).
8	RG9	Pino de I/O do Microcontrolador
9	VSS	GND do Microcontrolador
10	VDD	Alimentação positiva, este chip é alimentado com uma tensão que pode variar de 2,5V até 3,6V
11	RB5	Pino de I/O do Microcontrolador
12	RB4	Pino de I/O do Microcontrolador
13	RB3	Pino de I/O do Microcontrolador
14	RB2	Pino de I/O do Microcontrolador
15	RB1	Pino de I/O do Microcontrolador
16	RB0	Pino de I/O do Microcontrolador

Veja que neste lado do microcontrolador estão presentes 13 pinos de I/O divididos através dos PORTS (Portas de Acesso) RE5, RE6 e RE7 que fazem parte do PORTE do microcontrolador. Os pinos RG6, RG7 e RG8 são do

PORTG do microcontrolador e os pinos RB5, RB4, RB3, RB2, RB1 e RB0 do PORTB do microcontrolador. Os pinos de alimentação também estão presentes nesta parte do chip, através dos pinos 9 (VSS ou GND) e 10 (VDD ou, por exemplo, 3,3V que será a tensão que a ser utilizada. As outras partes que compõem a pinagem deste microcontrolador podem observadas abaixo:

Pinos de 17 a 32 do PIC32MX320F032H		
Pino	Nome	Função
17	RB6	Pino de I/O do Microcontrolador
18	RB7	Pino de I/O do Microcontrolador
19	AVDD	Alimentação Positiva do Conversor Analógico -> Digital
20	AVSS	GND do Conversor Analógico -> Digital
21	RB8	Pino de I/O do Microcontrolador
22	RB9	Pino de I/O do Microcontrolador
23	RB10	Pino de I/O do Microcontrolador
24	RB11	Pino de I/O do Microcontrolador
25	VSS	GND do Microcontrolador
26	VDD	Alimentação positiva, este chip é alimentado com uma tensão que pode variar de 2,5V até 3,6V
27	RB12	Pino de I/O do Microcontrolador
28	RB13	Pino de I/O do Microcontrolador
29	RB14	Pino de I/O do Microcontrolador
30	RB15	Pino de I/O do Microcontrolador
31	RF4	Pino de I/O do Microcontrolador
32	RF5	Pino de I/O do Microcontrolador

Nesta parte do microcontrolador são ao todo 12 pinos de I/O divididos através dos pinos RB6, RB7, RB8, RB9, RB10, RB11, RB12, RB13, RB14 e RB15 que fazem parte do PORTB do microcontrolador. Encontram-se também os pinos RF4 e RF5 pertencentes ao PORTF do chip. Os pinos de alimentação também estão disponíveis através dos pinos 25 (VSS) e 26 (VDD). A alimentação do conversor analógico ->

digital fica disponível nos pinos 19 (AVDD) e 20 (AVSS). Agora analise os pinos 33 até 48:

Pinos de 33 a 48 do PIC32MX320F032H		
Pino	Nome	Função
33	RF3	Pino de I/O do Microcontrolador
34	RF2	Pino de I/O do Microcontrolador
35	RF6	Pino de I/O do Microcontrolador
36	RG3	Pino de I/O do Microcontrolador
37	RG2	Pino de I/O do Microcontrolador
38	VDD	Alimentação positiva, este chip é alimentado com uma tensão que pode variar de 2,5V até 3,6V
39	OSC1	Entrada para oscilador
40	OSC2	Saída para oscilador
41	VSS	GND do Microcontrolador
42	RD8	Pino de I/O do Microcontrolador
43	RD9	Pino de I/O do Microcontrolador
44	RD10	Pino de I/O do Microcontrolador
45	RD11	Pino de I/O do Microcontrolador
46	RD0	Pino de I/O do Microcontrolador
47	RC13	Pino de I/O do Microcontrolador
48	RC14	Pino de I/O do Microcontrolador

Neste lado do microcontrolador há 12 pinos de I/O divididos da seguinte forma: RF3, RF2, RF6 que fazem parte do PORTF do microcontrolador. Existem também os pinos RG3 e RG2 do PORTG. Os pinos RD8, RD9, RD10, RD11 e RD0 do PORTD. Os pinos RC13 e RC14 do PORTC. A alimentação está disponível nos pinos 38 (VDD) e 41 (GND ou VSS). Está disponível nos pinos 39 (OSC1) e 40 (OSC2) a entrada para o oscilador externo que fica ligado ao microcontrolador. Veja agora a última parte do microcontrolador, que vai dos pinos 49 a 64:

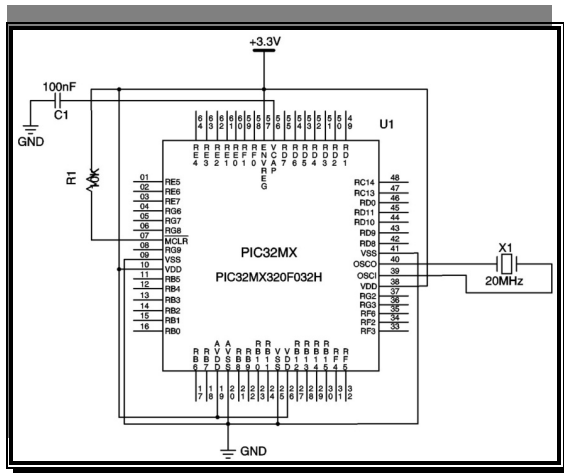
Pinos de 49 a 64 do PIC32MX320F032H		
Pino	Nome	Função
49	RD1	Pino de I/O do Microcontrolador
50	RD2	Pino de I/O do Microcontrolador
51	RD3	Pino de I/O do Microcontrolador
52	RD4	Pino de I/O do Microcontrolador
53	RD5	Pino de I/O do Microcontrolador
54	RD6	Pino de I/O do Microcontrolador
55	RD7	Pino de I/O do Microcontrolador
56	VDDCORE	Entrada de tensão para a tensão de núcleo
57	ENVREG	Habilitação do regulador interno
58	RF0	Pino de I/O do Microcontrolador
59	RF1	Pino de I/O do Microcontrolador
60	RE0	Pino de I/O do Microcontrolador
61	RE1	Pino de I/O do Microcontrolador
62	RE2	Pino de I/O do Microcontrolador
63	RE3	Pino de I/O do Microcontrolador
64	RE4	Pino de I/O do Microcontrolador

São ao todo nesta parte do microcontrolador 14 pinos de I/O e estão disponíveis através dos pinos RD1, RD2, RD3, RD4, RD4, RD6 e RD7 que fazem parte do PORTD. Temos também os pinos RF0 e RF1 que fazem parte do PORTF. Os pinos RE0, RE1, RE2, RE3 e RE4 que compõem o PORTE do microcontrolador. O pino ENVREG é um pino de controle que ao ser colocado em nível 1 (3,3V, por exemplo) habilita o regulador interno ao chip para alimentar o núcleo (core) do microcontrolador. Normalmente este pino é deixado em nível alto e coloca-se apenas um capacitor de 100nF conectado ao pino VDDCORE para dar maior estabilidade a tensão do núcleo do microcontrolador.

Na tabela abaixo é mostrado quantos PORTS tem disponíveis neste chip e quantos pinos o mesmo disponibiliza em seu encapsulamento:

Quantidade de PORTS			
Pinos	Nome	PORT	Total de Pinos
1, 2, 3, 60, 61, 62, 63 e 64	RE5, RE6, RE7, RE0, RE1, RE2, RE3, RE4	PORTE	8
4, 5, 6, 8, 36 e 37	RG6, RG7, RG8, RG9, RG3 e RG2	PORTG	6
11, 12, 13, 14, 15, 16, 17, 18, 21, 22, 23, 24, 27, 28, 29 e 30	RB5, RB4, RB3, RB2, RB1, RB0, RB6, RB7, RB8, RB9, RB10, RB11, RB12, RB13, RB14 e RB15	PORTB	16
31, 32, 33, 34, 35, 58 e 59	RF4, RF5, RF3, RF2, RF6, RF0 e RF1	PORTF	7
39, 40, 47 e 48	RC12, RC15, RC13 e RC14	PORTC	4
42, 43, 44, 45, 46, 49, 50, 51, 52, 53, 54 e 55	RD8, RD9, RD10, RD11, RD0, RD1, RD2, RD3, RD4, RD5, RD6 e RD7	PORTD	12
Total de Pinos de I/O			53

Observe que dos 64 pinos disponíveis, tem-se 53 pinos para conectar dispositivos digitais como botões, leds, displays e etc. (as funções especiais dos pinos serão verificadas nos próximos capítulos). O hardware básico para que o PIC32MX320032H funcione é o que está apresentado na figura abaixo:



Note que os VDD devem ser ligados em 3,3V, por exemplo, e todos os VSS no GND (Terra da Fonte). Nos pinos OSCO e OSCI é ligado um oscilador, como por exemplo, de 20 MHz. O pino MCLR fica em nível alto, com o uso de um resistor de 10K de pull-up ligado a 3,3V. O pino ENVREG fica em nível alto também para que possa habilitar o regulador interno de 2,5V para o funcionamento do núcleo do chip (core). No pino VDDCORE basta colocar um capacitor de 100nF a fim de manter estável a tensão de funcionamento interna.

3.2 Características do PIC32MX320F032H

Outras características importantes deste microcontrolador:

PIC32MX320F032F	
Pinos	64
Memória de Programa	32 kB
Memória de Dados	8 kB
Velocidade Máxima de Processamento	40 MHz
Timers/Capture/Compare	5
EUSART/SPI/I ² C	2
Entradas Analógicas	16
Comparadores	2

A memória de programa é referente ao valor máximo em Kb (kilo Bytes) que seu programa deve ter para caber dentro deste chip tendo neste caso 32 kB sendo Flash, ou seja, não apaga na ausência de energia porque é apagada e escrita eletricamente. A memória de dados refere-se à memória volátil do microcontrolador, aquela que na ausência de energia é perdido seu conteúdo, porém tem um fator importante que é o tempo de acesso da mesma que é bem rápido. Esta memória tem a capacidade de 8kB (kilo Bytes) interno a este chip. No item velocidade máxima, podemos observar em qual velocidade máxima nosso microcontrolador pode funcionar, sendo para este modelo no máximo 40 MHz, porém outros modelos podem funcionar até 80 MHz como, por exemplo, o modelo PIC32MX320F128H. Os outros recursos como Timers, EUSART e entradas analógicas, por exemplo, serão explicados nos capítulos vindouros com mais detalhes, por enquanto não se preocupe com eles.

Capítulo 4 – Entendendo e Configurando os Configurations Bits

Entender os configurations bits é fundamental para poder trabalhar corretamente com os microcontroladores. Neste capítulo começará o estudo.

4.1 Os Configurations Bits do PIC32MX320F032H

Entender a programação dos configurations bits é essencial para funcionamento correto do microcontrolador. Ao todo, possui os seguintes configurations bits:

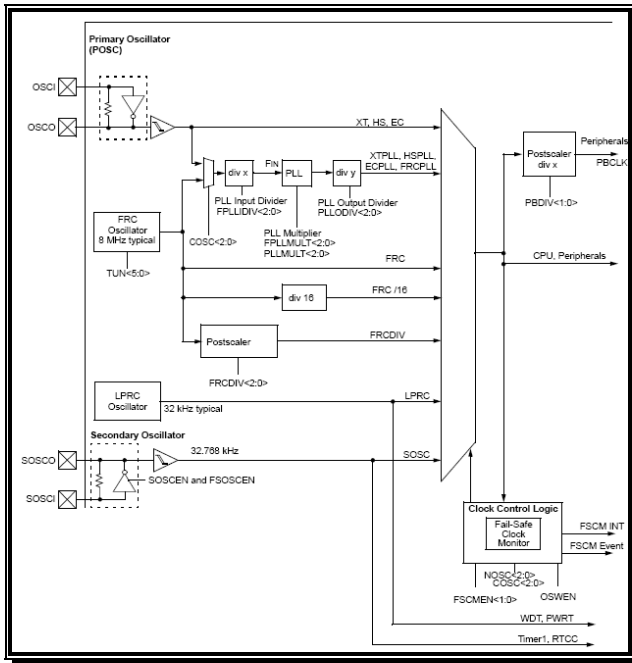
Configurations Bits
PLL Input Divider
PLL Multiplier
System PLL Output Clock Divider
Oscillator Selection Bits
Secondary Oscillator Enable
Internal/External Switch Over
Primary Oscillator Configuration
CLKO Output Signal Active on the OSCO Pin
Peripheral Clock Divisor
Clock Switching and Monitor Selection
Watchdog Timer Postscaler
Watchdog Timer Enable
ICE/ICD Comm Channel Select
Boot Flash Write Protect
Code Protect

Tabela de Configurações do PIC32MX320032H

Para acessar e mudar esta configuração manualmente no MPLAB X, vá ao menu *Window -> PIC Memory Views -> Configuration Bits*.

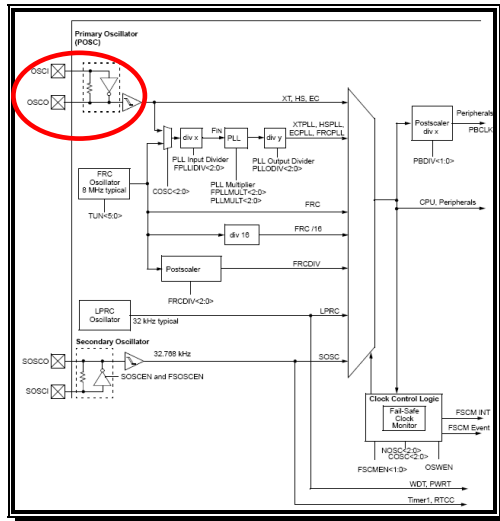
4.2 PLL Input Divider

A PLL (Phase-Locked Loop) é um circuito interno ao Microcontrolador que permite dividir a frequência de entrada caso seja utilizado os modos de oscilação XTPLL, HSPLL, ECPLL ou FRCPLL, ou seja, a PLL esteja habilitada. No caso do bloco PLL Input Divider, a função dele é dividir a frequência de entrada do chip. Na figura abaixo está organizado o bloco de configuração de clock do microcontrolador:

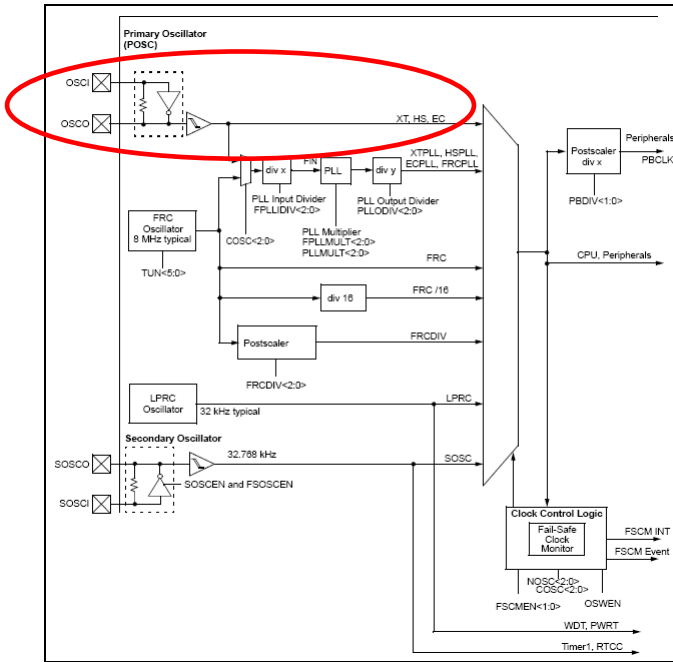


Sistema de Clock do Microcontrolador

Note que tudo começa nas entradas OSCO e OSCI onde deverá ficar conectado um cristal externo. A marcação abaixo demonstra as entradas de ligação OSCO e OSCI no sistema de clock assim como no microcontrolador:



Entrada de Clock do Microcontrolador



Caminho do circuito quando o bloco PLL não for utilizado

Normalmente não é usado o bloco PLL quando o cristal que está conectado externamente é alto e não há a necessidade de usar o bloco PLL para aumentar o clock proveniente do cristal. Se for utilizar o bloco PLL note que na figura abaixo este bloco está marcado:

frequência de 5 MHz que está dentro da faixa que é de 4 MHz a 5 MHz.



Obs.: Em breve será apresentado um modo de fazer a mesma configuração, porém diretamente no código C.

- 1) Em um projeto, será usado um clock de entrada de 4 MHz e o modo PLL. Informe qual o valor indicado para o funcionamento correto do bloco PLL na opção PLL Input Divider.

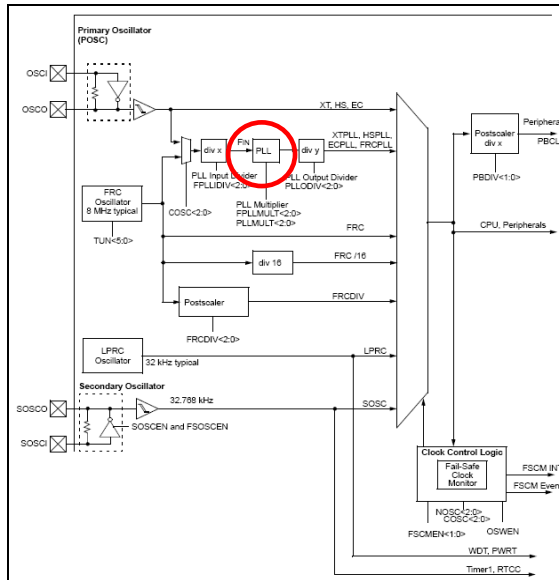
R: Neste caso a melhor escolha seria a opção 1x Divider, pois a divisão de 4MHz por 1 é igual a 4 MHz que fica na faixa de 4MHz a 5 MHz para o funcionamento correto do bloco PLL.

- 2) Em um projeto, será usado um clock de entrada de 25 MHz e será usado o modo PLL. Informe qual o valor indicado para o funcionamento correto do bloco PLL na opção PLL Input Divider.

R: Neste caso podemos usar duas escolhas no PLL Input Divider, sendo elas a opção 5x Divider e 6x Divider. Na primeira opção teremos um clock de 5 MHz ($25 \text{ MHz}/5$) e na segunda, um clock de 4,16 MHz ($25\text{MHz}/6$). Para facilitar o cálculo do clock, recomendo que a primeira opção seja usada, pois note que o valor foi inteiro (5MHz) facilitando assim o cálculo final do clock da máquina.

4.3 PLL Multiplier

Este bloco vem em seguida do bloco PLL Input Divider, como marcado abaixo:



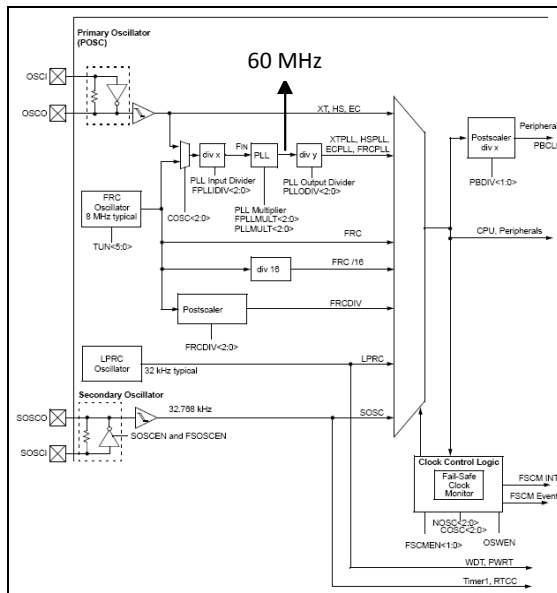
Opções de Divisão do Bloco PLL Multiplier

A função deste bloco é permitir que a frequência proveniente do PLL Input Divider (é uma frequência que deve estar na faixa de 4 MHz a 5 MHz) seja multiplicada obtendo-se assim uma frequência maior. Veja que a frequência de entrada pode ser multiplicada por 15x Multiplier até 24x Multiplier. A seguir alguns exemplos que irão facilitar o entendimento por parte do leitor.

- 1) Em um projeto, será usado um clock de entrada de 4 MHz e será usado o modo PLL. Sabendo que o PLL Input Divider está configurado para 1x já que o clock de

entrada é de 4 MHz, informe qual a frequência de saída do bloco PLL Multiplier sabendo que o mesmo está configurado para 15x Multiplier.

R: Multiplique a frequência de saída do PLL Input Divider por 15x, obtendo-se assim a frequência de 60 MHz. Ou seja, teríamos no ponto marcado abaixo tal frequência (note que esta ainda não é a frequência no qual o microcontrolador irá funcionar):

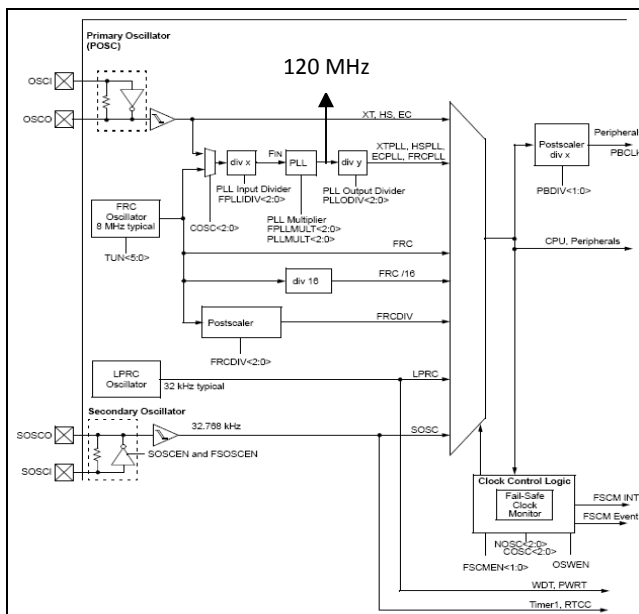


Saída de Frequência do Bloco PLL Multiplier

2) Em um projeto, será usado um clock de entrada de 20 MHz e será usado o modo PLL. Sabendo que o PLL Input Divider está configurado para 4x já que o clock de entrada é de 20 MHz e $20\text{MHz}/4$ dá o clock de 5 MHz que está na faixa de funcionamento do PLL Multiplier,

informe qual a frequência de saída do bloco PLL Multiplier sabendo que o mesmo está configurado para 24x Multiplier.

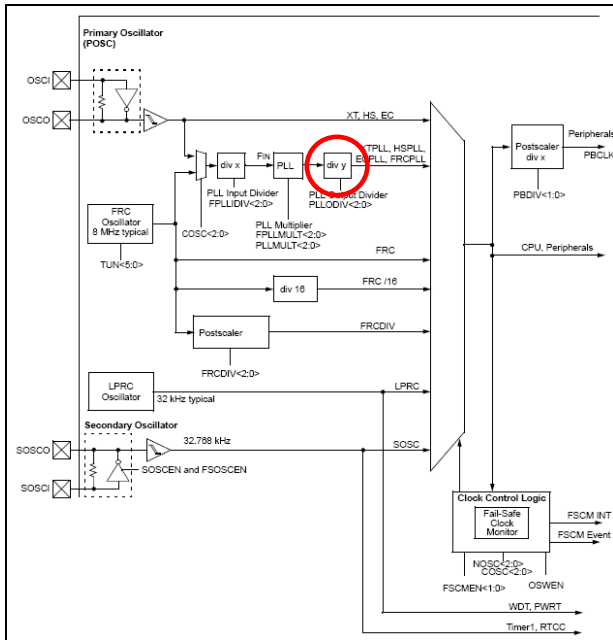
R: Multiplique a frequência de saída do PLL Input Divider por 24x, obtendo-se assim a frequência de 120 MHz. Ou seja, tem-se no ponto marcado abaixo tal frequência (note que esta ainda não é a frequência no qual o microcontrolador irá funcionar):



Saída de Freqüência do Bloco PLL Multiplier

4.4 System PLL Output Clock Divider

Este bloco fica logo a seguir do bloco PLL Multiplier e a função dele é dividir a frequência oriunda deste bloco. Tal bloco está marcado abaixo com o nome div y:

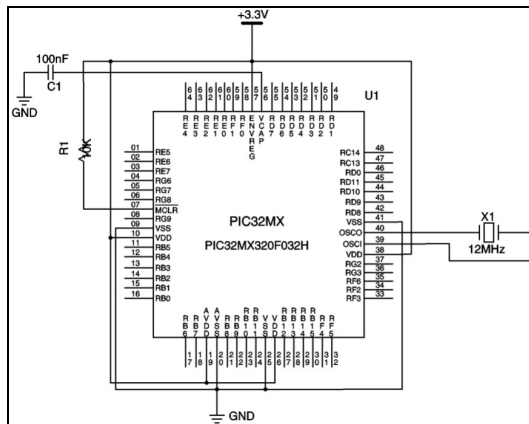


Bloco System PLL Output Clock Divider

A frequência de saída deste bloco irá determinar a velocidade de processamento da CPU do PIC32. Neste caso, devemos ter cuidado para não causar um overclock no PIC (aumento de frequência), pois neste modelo, a frequência máxima de funcionamento é de 40 MHz (outros modelos podem chegar a 80 MHz, consulte neste caso o datasheet).

Alguns exemplos para entender melhor de forma completa a configuração do sistema de clock do PIC32.

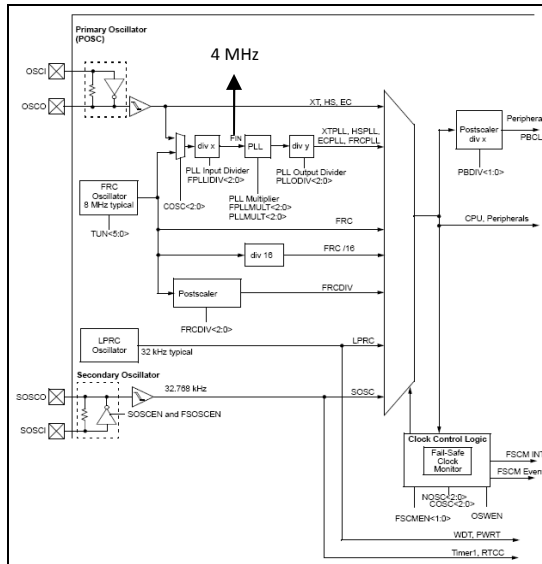
- 1) De acordo com o esquema apresentado abaixo, determine o clock que a CPU irá funcionar sabendo:
 - a) O PLL Input Divider está configurado para 3x Divider;
 - b) O PLL Multiplier está configurado para 20x Multiplier;
 - c) O PLL System Divider Clock está configurado para 2x PLL Divider;
 - d) A opção HSPLL será usada.



Esquema Elétrico Proposto

Bom, vamos iniciar nosso cálculo. No esquema acima é composto de um cristal de 12 MHz. A configuração do bloco PLL Divider foi feita para dividir a frequência de entrada, neste caso 12 MHz por 3, neste caso há após o bloco PLL Divider a frequência de 4 MHz. Na figura abaixo é possível visualizar com mais clareza:

Frequência
de Entrada:
12 MHz



Frequência de saída do bloco PLL Input Divider

Continuando com a análise, tem-se a seguir um bloco PLL Multiplier configurado para 20x, sendo assim na saída deste bloco a frequência é de 80 MHz que é justamente 4 MHz multiplicado por 20, acompanhe na figura abaixo:

Frequência
de Entrada:
12 MHz

