

Kit de desenvolvimento IOnet

KDPIC4550

Manual de Utilização

Sumário

1.	INTRODUÇÃO	1
1.1	Objetivo	1
1.2	Etapas.....	1
1.3	Requisitos/conhecimentos prévios desejáveis	2
2.	HARDWARE.....	3
2.1	O Layout.....	3
2.2	O Circuito.....	4
3.	SOFTWARES NECESSÁRIOS.....	6
3.1	Instalação dos softwares	6
3.1.1	Instalação do MPLAB X IDE e MPLAB XC8.....	6
4.	MPLAB X IDE: VISÃO GERAL	8
4.1	Funcionalidades	8
4.2	Conceito.....	8
4.3	Principais elementos da interface	9
5.	MPLAB X IDE : PROGRAMAÇÃO EM LINGUAGEM ASSEMBLY PARA O KIT	10
5.1	Criando o projeto	10
5.2	Criando o arquivo de código.....	14
5.3	Compilando o projeto.....	16
6.	MPLAB X IDE: PROGRAMAÇÃO EM LINGUAGEM “C” PARA O KIT.....	17
6.1	Criando o projeto	17
6.2	Criando o arquivo de código.....	21
6.3	Compilando o projeto.....	25
7.	GRAVANDO O CÓDIGO COMPILADO NO KIT	26
7.1	Ativando o modo de programação do kit	26
7.2	Realizando a gravação do Kit.....	27
8.	BIBLIOGRAFIA	30
9.	NOTA DA IONET	31

Índice de Figuras

Figura 1.1: Fluxograma básico de utilização do Kit.....	2
Figura 2.1: Vista Superior do Kit.....	3
Figura 2.2: Circuito eletrônico do Kit.....	5
Figura 4.1: MPLAB X IDE – Janela principal.....	9
Figura 5.1: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 1.....	10
Figura 5.2: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 2.....	11
Figura 5.3: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 3.....	11
Figura 5.4: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 4.....	12
Figura 5.5: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 5.....	13
Figura 5.6: MPLAB X IDE -Projeto criado em linguagem Assembly.....	13
Figura 5.7: MPLAB X IDE - Criando o arquivo de código Assembly– Passo 1.....	14
Figura 5.8: MPLAB X IDE - Criando o arquivo de código Assembly– Passo 2.....	15
Figura 5.9: MPLAB X IDE - Arquivo de código Assembly criado.....	15
Figura 5.10: MPLAB X IDE – Compilando o projeto.....	16
Figura 6.1: MPLAB X IDE - Criando um projeto em linguagem “C” – Passo 1.....	17
Figura 6.2: MPLAB X IDE - Criando um projeto em linguagem “C” – Passo 2.....	18
Figura 6.3: MPLAB X IDE - Criando um projeto em linguagem “C” – Passo 3.....	18
Figura 6.4: MPLAB X IDE - Criando um projeto em linguagem “C” – Passo 4.....	19
Figura 6.5: MPLAB X IDE - Criando um projeto em linguagem “C” – Passo 5.....	20
Figura 6.6: MPLAB X IDE -Projeto criado em linguagem “C”.....	21
Figura 6.7: MPLAB X IDE - Criando o arquivo de código em “C” – Passo 1.....	22
Figura 6.8: MPLAB X IDE - Criando o arquivo de código em “C” – Passo 2.....	22
Figura 6.9: MPLAB X IDE - Arquivo de código em “C” criado.....	23
Figura 6.10: MPLAB X IDE – Abrindo as propriedades do projeto.....	24
Figura 6.11: MPLAB X IDE – Abrindo as configurações do XC8 Linker.....	24
Figura 6.12: MPLAB X IDE – Compilando o projeto.....	25
Figura 7.1: Microchip USB HID Bootloader.....	26
Figura 7.2: Microchip USB HID Bootloader - Kit em modo de autoprogramação.....	27
Figura 7.3: Microchip USB HID Bootloader – Seleccionando o arquivo” .hex”.....	28
Figura 7.4: Microchip USB HID Bootloader – Arquivo ”.hex” selecionado e pronto para gravar.	28
Figura 7.5: Microchip USB HID Bootloader – Gravação realizada.....	29

1. INTRODUÇÃO

O Kit de desenvolvimento IOnet KDPIC4550 é uma excelente ferramenta para iniciar os estudos dos microcontroladores da linha PIC18 da *Microchip*®.

Contendo microcontrolador PIC18F4550, *leds*, botões, potenciômetro e conectores, é possível desenvolver programas, tanto em linguagem *Assembly* quanto em linguagem “C”, para explorar dezenas de periféricos presentes neste microcontrolador. Tudo isso com um Kit compacto e funcional.

No Kit é possível explorar recursos de:

- Configuração e utilização de pinos de I/O digitais;
- Conversores analógicos para digital;
- Comunicação USB;
- PWM, etc.

Usando os pinos de I/O presente em dois conectores, é possível conectar módulos e interfaces para realizar uma infinidade de experimentos, tais como:

- Teclado matricial;
- Matriz de *leds*;
- Manipulação de *display* de cristal líquido (LCD);
- Manipulação de *display* gráfico (GLCD e TFT);
- Comunicação serial (USART, SPI, I2C, etc.);
- Comunicação *Ethernet*, entre outros;

1.1 Objetivo

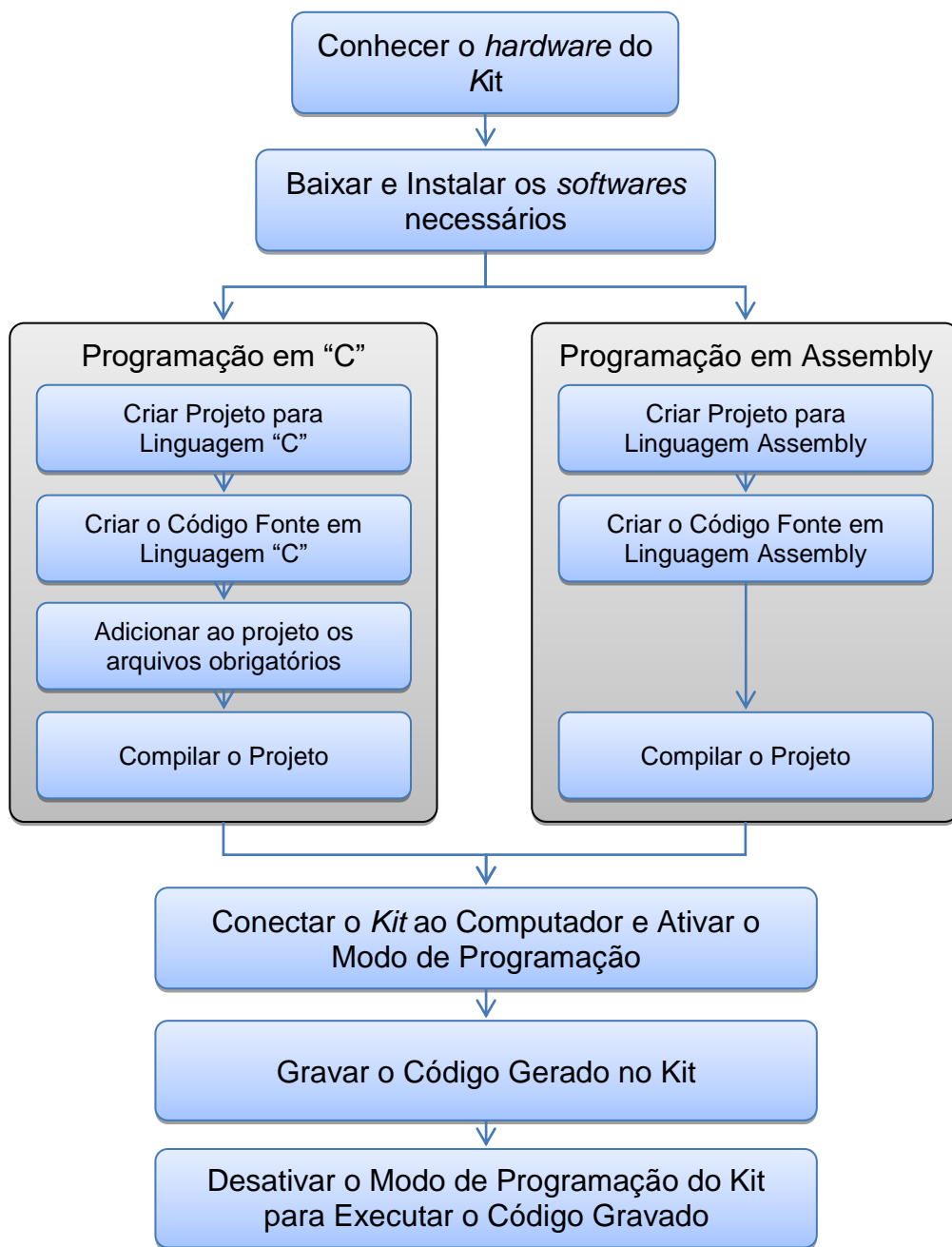
Este breve manual busca fornecer as informações básicas para iniciar o estudo em sistemas microcontrolados utilizando o Kit de desenvolvimento IOnet KDPIC4550.

Destaca-se a explanação sobre o seu *hardware* bem como os *softwares* necessários para realizar a Edição, Compilação e Gravação do Kit.

1.2 Etapas

Visando ordenar as etapas, para a melhor utilização do Kit, a Figura 1.1 apresenta um fluxograma básico a ser seguido pelo utilizador.

Figura 1.1: Fluxograma básico de utilização do Kit.



1.3 Requisitos/conhecimentos prévios desejáveis

Para melhor utilização do *Kit*, alguns requisitos/conhecimentos são necessários:

- Conhecimento básico em Eletrônica digital e analógica;
- Familiaridade com Linguagens de Programação, preferencialmente Linguagem “C” e *Assembly*.

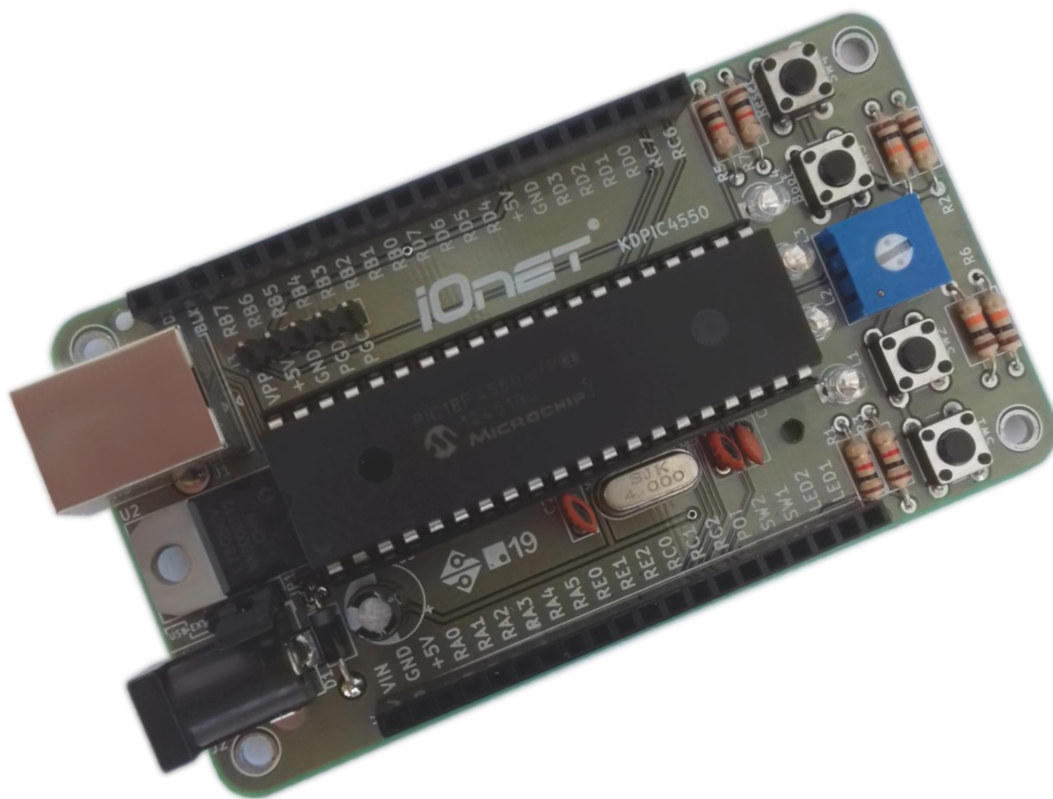
2. HARDWARE

Este capítulo irá fornecer as informações básicas para que se possam explorar os periféricos presentes no *Kit* e apresentar o seu circuito. Isto permitirá também que o utilizador realize conexões com módulos e periféricos externos com maior segurança.

2.1 O Layout

Na Figura 2.1 é apresentada a vista superior do *Kit*.

Figura 2.1: Vista Superior do Kit.



Descrição dos principais elementos identificados no do *Kit*:

- **J1** – Porta USB para conectar o *Kit* ao computador seja para gravação ou executar interfaces customizadas;
- **J2** – *Jack* para conexão de alimentação externa entre 9 a 12 Vcc;
- **J3** – Conector para gravação *Microchip In-Circuit Serial Programming* - ICSP;
- **JP1** – *Jumper* de seleção da forma de alimentação do *Kit*, sendo via porta USB ou via conector Vin (alimentação externa);
- **U1** – microcontrolador PIC18F4550;

- **U2** – Regulador de tensão com saída de 5Vcc;
- **Y1** – Cristal de quartzo de 4MHZ para *clock* do microcontrolador;
- **SW1 e SW2** – Botões de uso geral, tendo interligação tipo “pull-down” e acesso via conector J4;
- **SW3/Boot** – Botão de ativação do modo de programação e/ou botão de uso geral;
- **SW4/RESET** – Botão de *reset*;
- **L1, L2 e L3** – *Leds* de uso geral, interligado a resistores limitadores de corrente e acesso via conector P1;
- **L4** – *Led* de indicação de alimentação presente
- **POT** – Potenciômetro com seus terminais de extremidade conectados a VCC e GND e terminal central com acesso via conector J4;
- **J4 e J5** – Conectores para acesso aos pinos do microcontrolador, alimentação, *leds*, botões e potenciômetro tendo identificação pino a pino.

2.2 O Circuito

Na Figura 2.2 é apresentado o circuito eletrônico do Kit.



3. SOFTWARES NECESSÁRIOS

Para realizar a programação do microcontrolador PIC18F4550 presente no *Kit de desenvolvimento IOnet KDPIC4550* é necessário os seguintes *softwares*:

- **MPLAB IDE** – este é um ambiente integrado de desenvolvimento, onde é possível realizar a edição e compilação de programas. É uma ferramenta gratuita que pode ser ‘baixada’ diretamente do site da *Microchip*® (possui integrado o compilador *Assembly*).

Esta ferramenta permite também a integração de outros compiladores da própria *Microchip*® ou de terceiros. Possui as versões: MPLAB IDE 8.xx interface clássica e, o MPLAB X IDE baseado no *Netbeans* contendo excelentes recursos para edição de programas.

- **MPLAB XC8** – compilador “C” para microcontroladores de 8 bits da *Microchip*® que pode ser integrado ao MPLAB IDE.
- **USB BootLoader** – ferramenta para realizar a carga do programa compilado no microcontrolador do *Kit* via porta USB. É também gratuita e pode ser ‘baixada’ do site da *Microchip*®. Pode ser utilizada sem nenhuma restrição em microcontrolador PIC18F4550 que possua pré-gravado o HID *bootload*.

Nota: Este manual fornece as informações básica para o desenvolvimento dos programas com o Kit, usando o MPLAB X IDE, uma nova e promissora ferramenta da Microchip®. No entanto, a versão MPLAB IDE 8.xx é totalmente compatível com o Kit.

3.1 Instalação dos softwares

Todos os *softwares* possuem assistente padrão *Windows* de instalação, exceto o HIDBootLoader que não precisa ser instalado.

De posse dos *softwares necessários*, basta executar o instalador e seguir com a instalação padrão. Quando for solicitado o caminho da instalação, sugere-se instalar na raiz da unidade “C:” de seu computador.

3.1.1 Instalação do MPLAB X IDE e MPLAB XC8

Antes de iniciar a instalação do MPLAB X IDE certifique-se de que seu computador tenha instalado o *Java Runtime Environment (JRE)*, pois essa versão do MPLAB foi concebida com o conceito de multiplataforma.

Caso seu computador não possua o JRE, faça o download no site java.com e o instale.

Instale primeiramente o MPLAB X IDE e após concluída sua instalação, instale o MPLAB XC8. Instalando nesta ordem, ao terminar o próprio MPLAB XC8 irá realizar a sua configuração dentro do MPLAB X IDE.

Durante o processo de instalação, de preferência as opções default que são indicadas em passo da instalação, isto fará com que tudo funcione adequadamente.

4. MPLAB X IDE: VISÃO GERAL

4.1 Funcionalidades

O MPLAB X IDE é uma nova ferramenta da *Microchip®* que contempla funcionalidades extremamente úteis ao programador, uma vez que reúne em um só *software* os recursos de:

- Edição de programas;
- Compilação;
- Simulação;
- Gravação;
- Depuração;
- Integração com compiladores de outros fabricantes.
- *Plug-ins*, entre outros.

4.2 Conceito

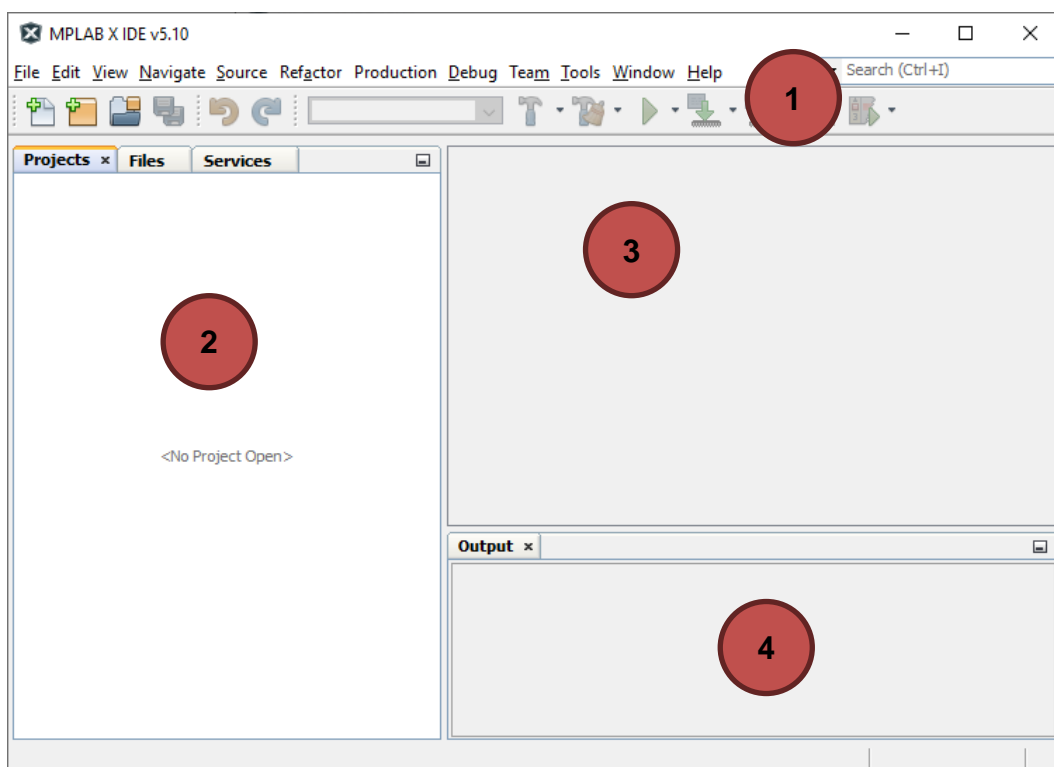
O MPLAB X IDE usa o conceito *workspace* para organizar todos os arquivos de código e configuração de compiladores estruturados de forma bastante intuitiva, onde é possível manter diferentes projetos abertos ao mesmo tempo, mas cada um com sua própria configuração.

Tanto programas em linguagem *Assembly* quanto em linguagem “C” podem ser elaborados usando o MPLAB X IDE.

4.3 Principais elementos da interface

Ao executar o MPLAB X IDE, geralmente via menu Windows *Iniciar>Todos os programas>Microchip®>MPLAB X IDE>MPLAB X IDE*, é aberto sua janela principal. Na Figura 4.1 apresentamos esta janela e, em seguida o detalhamento dos elementos principais que a constitui.

Figura 4.1: MPLAB X IDE – Janela principal.



Detalhes da Janela principal do MPLAB X IDE:

- **1 – Barra de menus e ícones atalhos:** permite acesso a todas as funcionalidades do *software*, organizados por categorias, seguindo os padrões de *software Windows*;
- **2 – Lista de Projetos:** relaciona todos os projetos abertos no ambiente de trabalho;
- **3 – Edição de código:** permite editar o código fonte do projeto, possui formataadores dinâmicos de códigos para facilitar a visualização de diferentes tipos de elementos do código;
- **4 – Janela de *Output*:** fornece as informações referentes à compilação, tais como: sucesso da compilação, erros, pontos de atenção, etc.

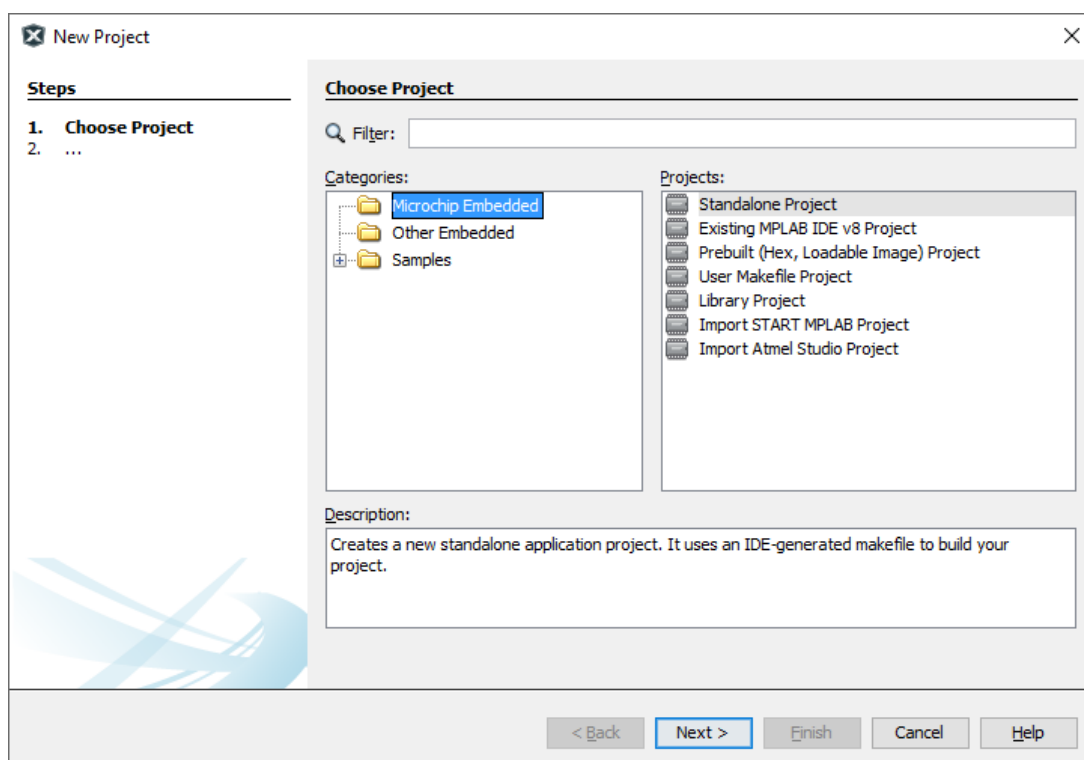
5. MPLAB X IDE : PROGRAMAÇÃO EM LINGUAGEM ASSEMBLY PARA O KIT

5.1 Criando o projeto

Estando com o MPLAB X IDE aberto, clique no menu “File > New Project...” e será aberta a janela do assistente de criação de novo projeto (Figura 5.1).

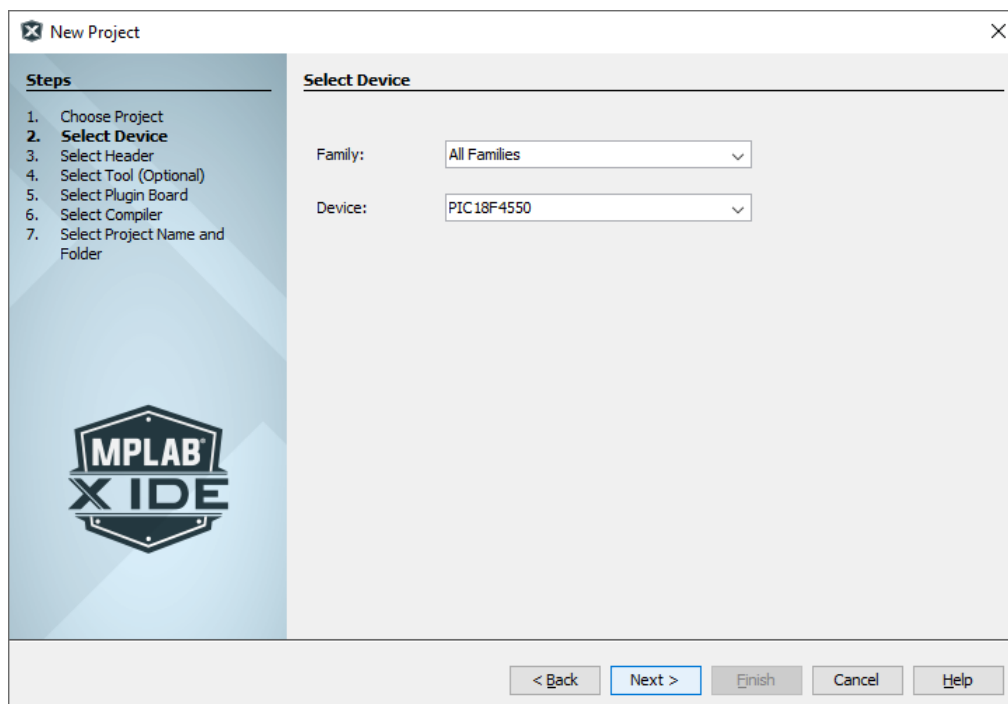
Nesta janela selecione em “Categories:” a opção “Microchip® Embedded”. E em “Projects:” selecione “Standalone Project”. Clique no botão “Next >”.

Figura 5.1: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 1.



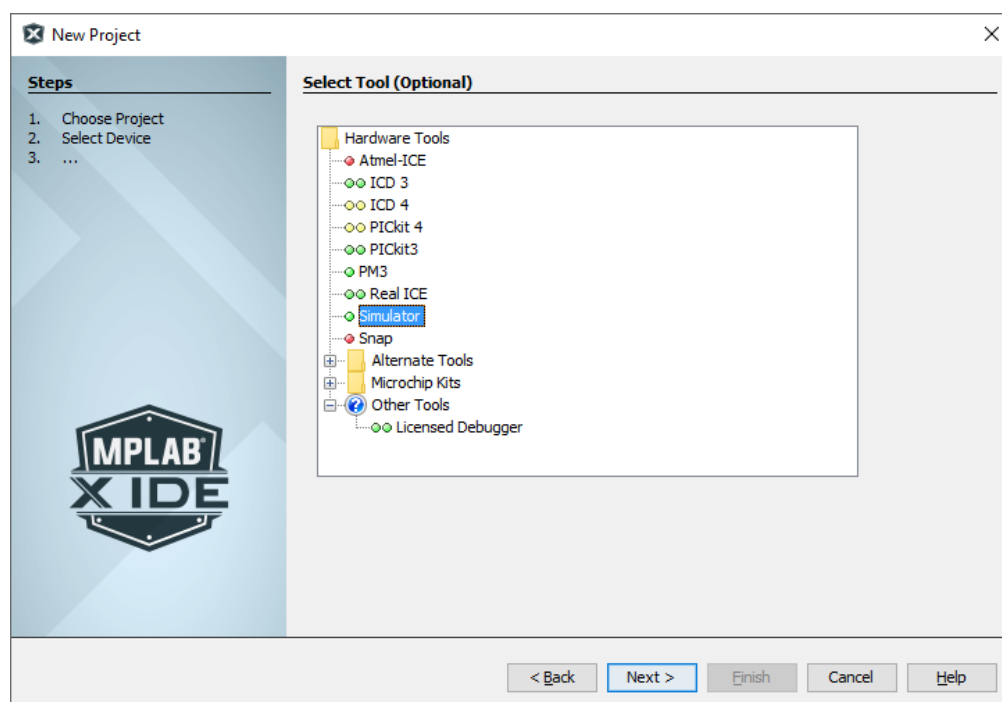
Na nova janela (Figura 5.2), selecione no *combo box* “Family:” a opção “Advanced 8-bit MCUS (PIC18)” e, em “Device:” a opção “PIC28F4550”. Clique no botão “Next>”.

Figura 5.2: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 2.



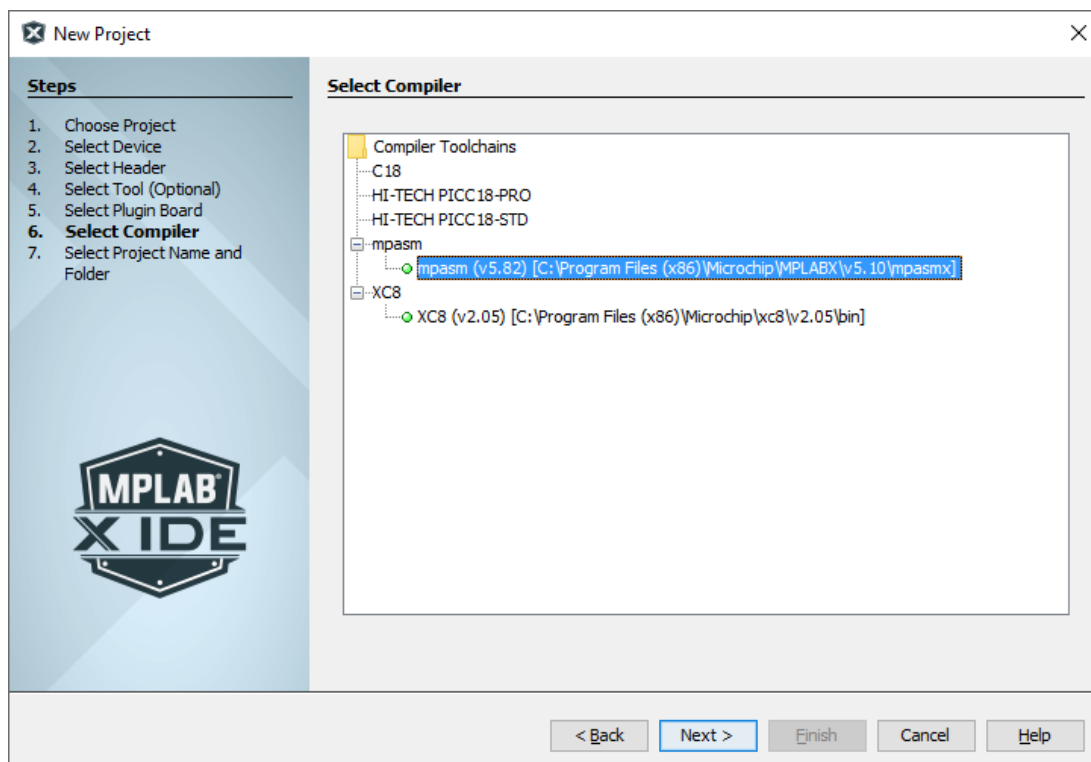
Em “Select Tool” (Figura 5.3), selecione a opção “Simulator” e clique no botão “Next>”.

Figura 5.3: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 3.



Na janela “Select Compiler” é apresentada toda a relação de compiladores instalados no computador. Selecione “mpasm (v5.82)[diretório de sua instalação]” para ser utilizado o compilador *Assembly* (Figura 5.4). Clique no botão “Next>”.

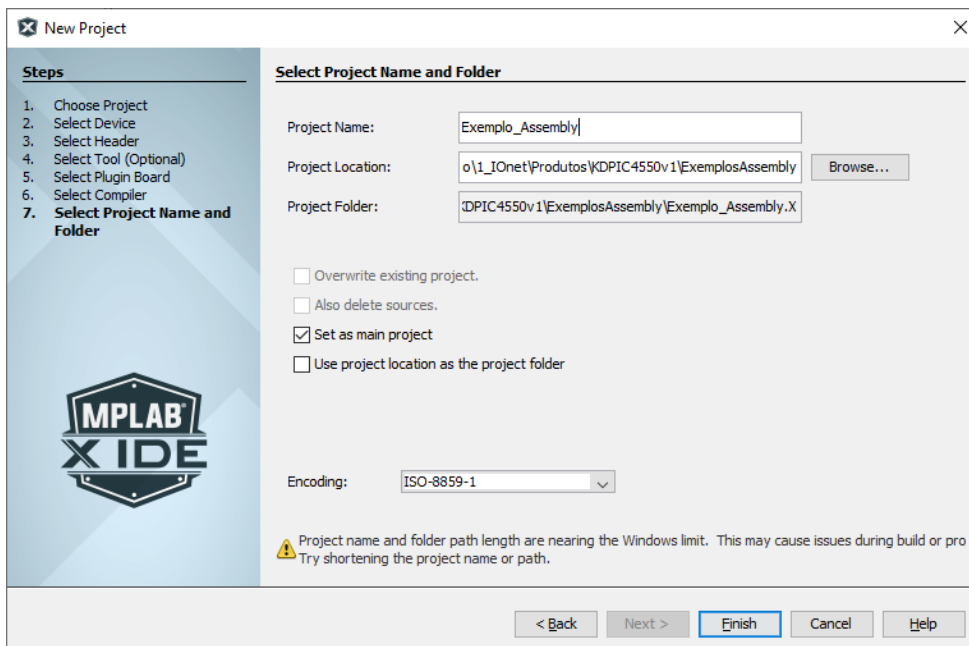
Figura 5.4: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 4.



Na próxima janela (Figura 5.5), campo “Project Name:” digite o nome do projeto que está sendo criado. Clique no botão “Browse...” e selecione o diretório no qual será armazenado o seu projeto, ou digite o caminho no campo “Project Location”.

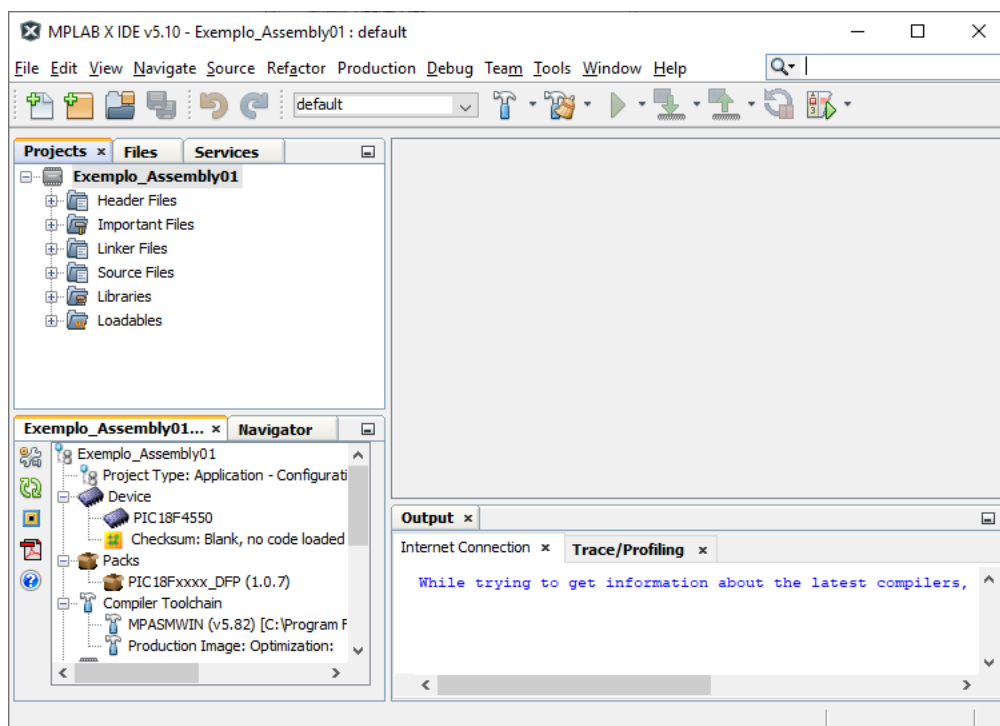
Selecione a opção “Set as main project” para tornar o seu novo projeto como o principal no MPLAB X IDE no momento da compilação. Clique no botão “Finish” para concluir a criação do seu projeto.

Figura 5.5: MPLAB X IDE - Criando um projeto em linguagem Assembly – Passo 5.



Observe que na lista de projetos da janela principal do MPLAB X IDE o projeto criado é exibido bem como sua estrutura de configuração (Figura 5.6).

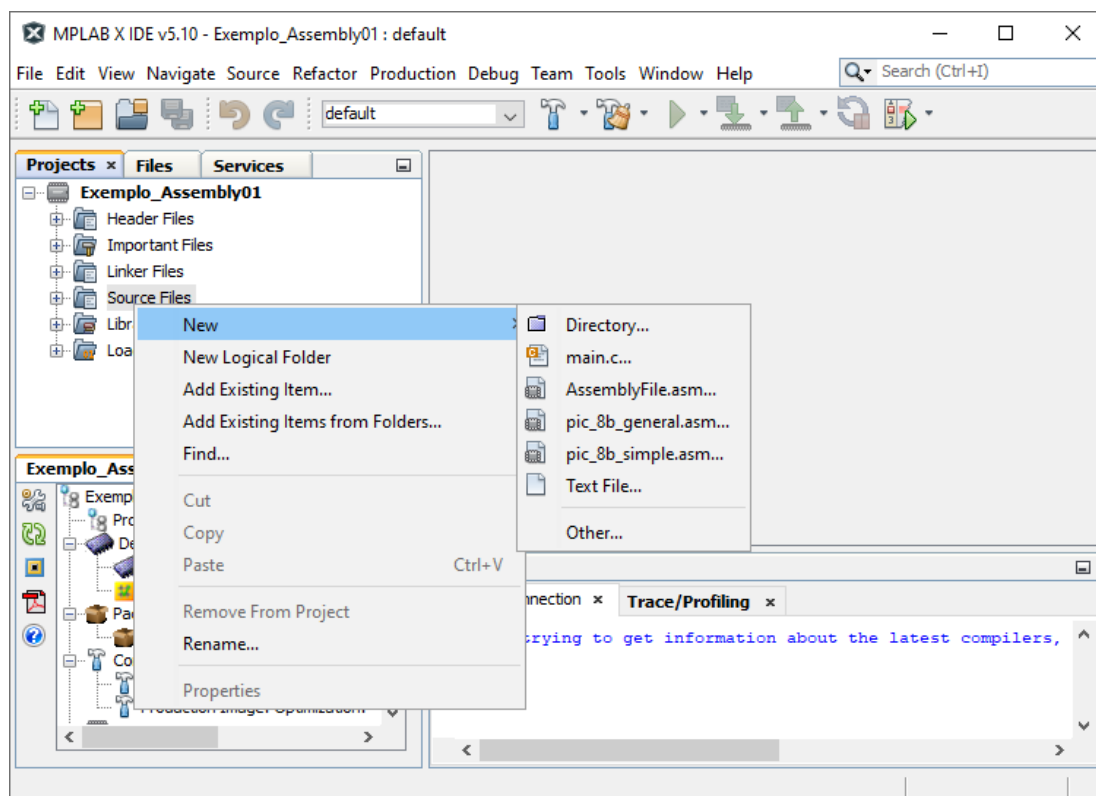
Figura 5.6: MPLAB X IDE –Projeto criado em linguagem Assembly.



5.2 Criando o arquivo de código

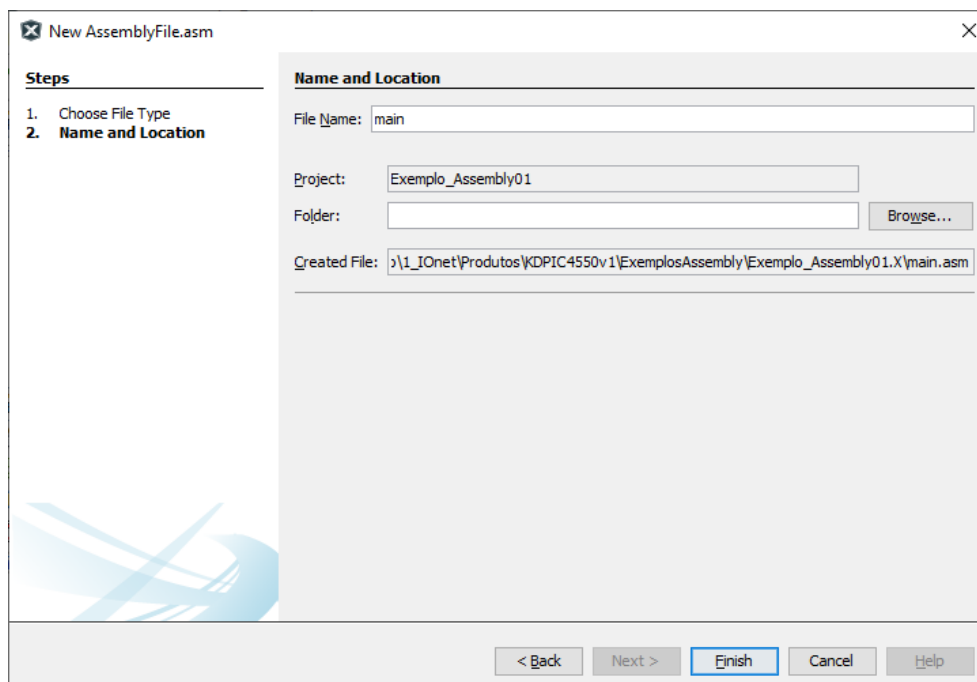
Com o projeto criado, agora se pode definir o código fonte em *Assembly* clicando com o botão direito do *mouse* na opção “Source Files” do menu flutuante. Clique em “New -> AssemblyFile.asm...” (Figura 5.7). Lembre-se de que o arquivo de código fonte em *Assembly* possui a extensão “.asm”.

Figura 5.7: MPLAB X IDE - Criando o arquivo de código Assembly– Passo 1.



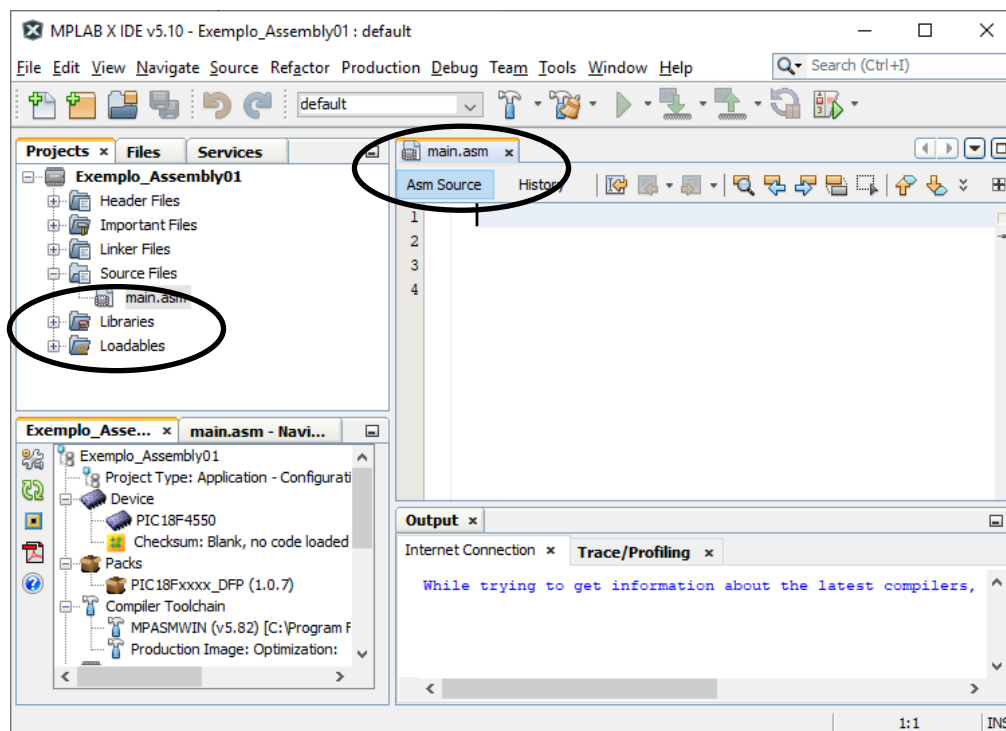
Uma nova janela será aberta (Figura 5.8). Digite no campo “File Name” o nome do arquivo com a extensão “.asm”. Clique no botão “Finish”.

Figura 5.8: MPLAB X IDE - Criando o arquivo de código Assembly- Passo 2.



Observe que o arquivo de código *Assembly* aparece na estrutura do projeto e também aberto para edição (Figura 5.9).

Figura 5.9: MPLAB X IDE - Arquivo de código Assembly criado.

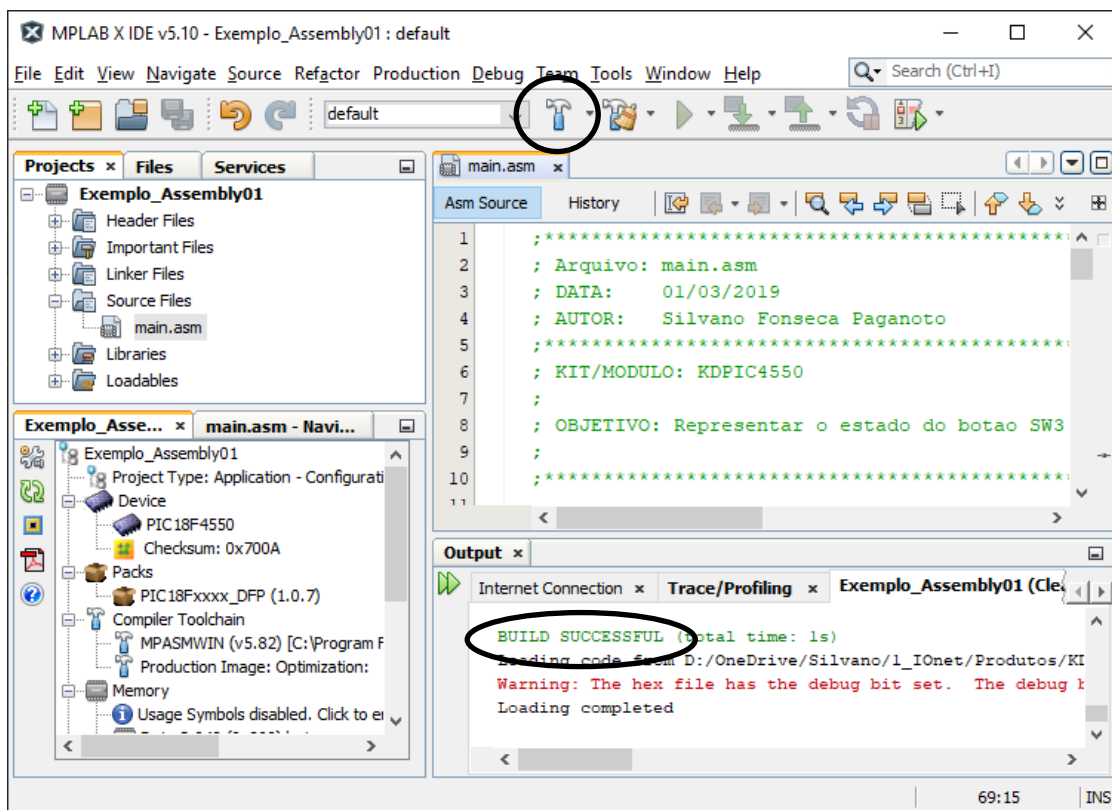


5.3 Compilando o projeto

Uma vez com o código fonte digitado, para compilá-lo, basta clicar no ícone “Build Main Project”.

Caso o programa não tenha nenhum problema de sintaxe, será exibida a mensagem ‘BUILD SUCCESSFUL’ na janela “Output” (Figura 5.10), indicando que a compilação foi bem sucedida.

Figura 5.10: MPLAB X IDE – Compilando o projeto.



Após a compilação é criado um arquivo com extensão “.hex” dentro de “[pasta do projeto]\dist\default\production”. Este arquivo é o que deverá ser carregado para o Kit via programa HIDBootLoader.

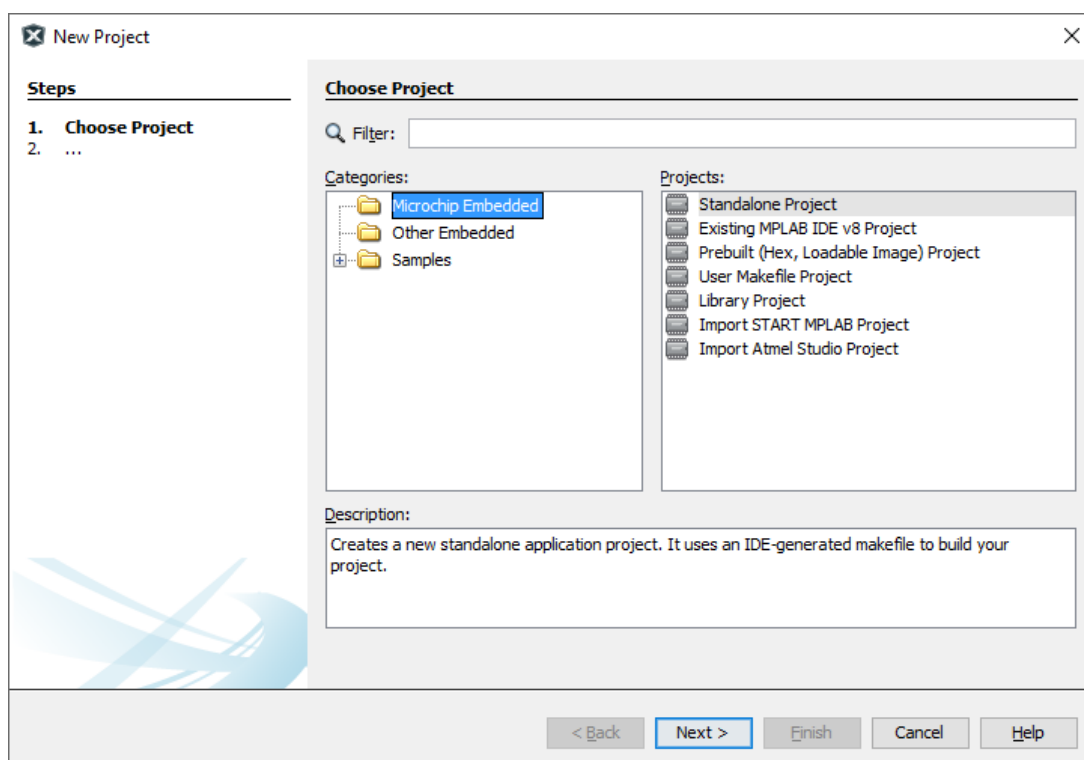
6. MPLAB X IDE: PROGRAMAÇÃO EM LINGUAGEM “C” PARA O KIT

6.1 Criando o projeto

Estando com o MPLAB X IDE aberto, clique no menu “File > New Project...” e será aberta a janela do assistente de criação de novo projeto (Figura 6.1).

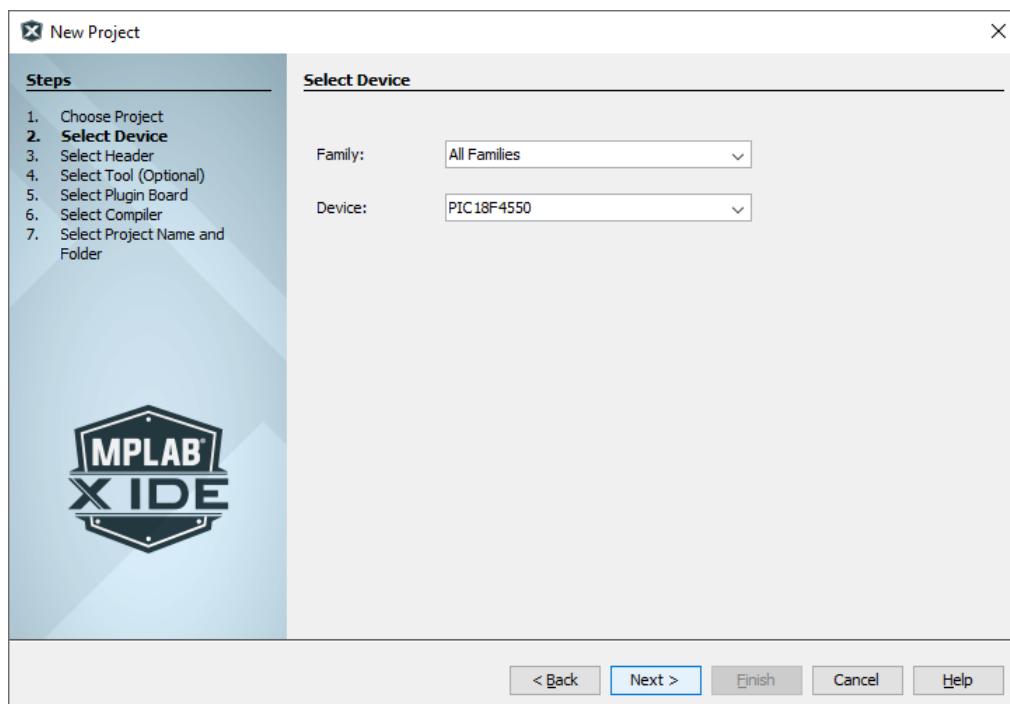
Nesta janela selecione em “Categories:” a opção “Microchip® Embedded” e em “Projects:” selecione “Standalone Project”. Clique no botão “Next >”.

Figura 6.1: MPLAB X IDE - Criando um projeto em linguagem “C” – Passo 1.



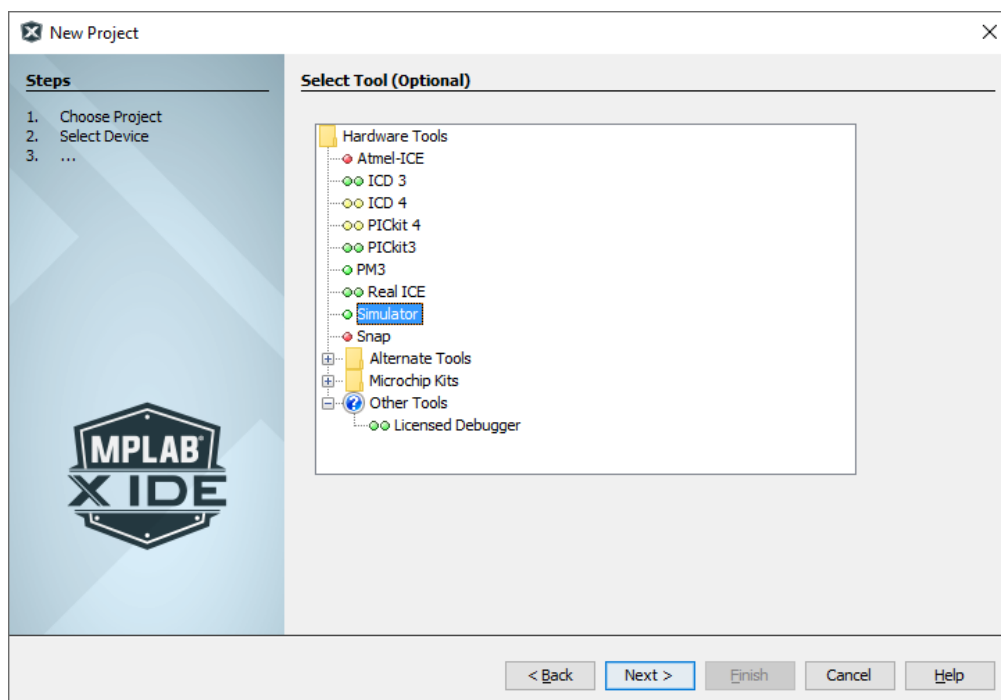
Na nova janela (Figura 6.2), selecione no *combo box* “Family:” a opção “Advanced 8-bit MCUS (PIC18)” e, em “Device:” a opção “PIC28F4550”. Clique no botão “Next>”.

Figura 6.2: MPLAB X IDE - Criando um projeto em linguagem "C" – Passo 2.



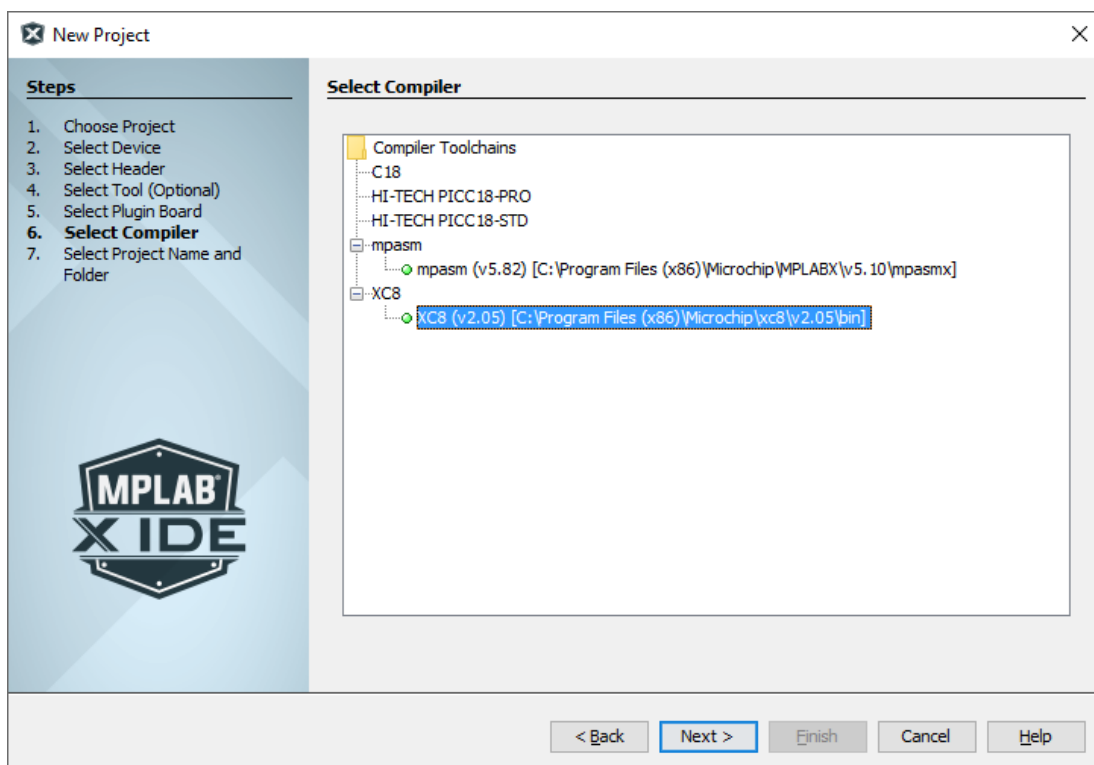
Em "Select Tool" (Figura 6.3), selecione a opção "Simulator" e clique no botão "Next>".

Figura 6.3: MPLAB X IDE - Criando um projeto em linguagem "C" – Passo 3.



Na janela “Select Compiler” é apresentada toda a relação de compiladores instalados no computador. Selecione “C18(v3.36)[diretório de sua instalação]” para ser utilizado o compilador “C” da Microchip® para a família PIC18 (Figura 6.4). Clique no botão “Next>”.

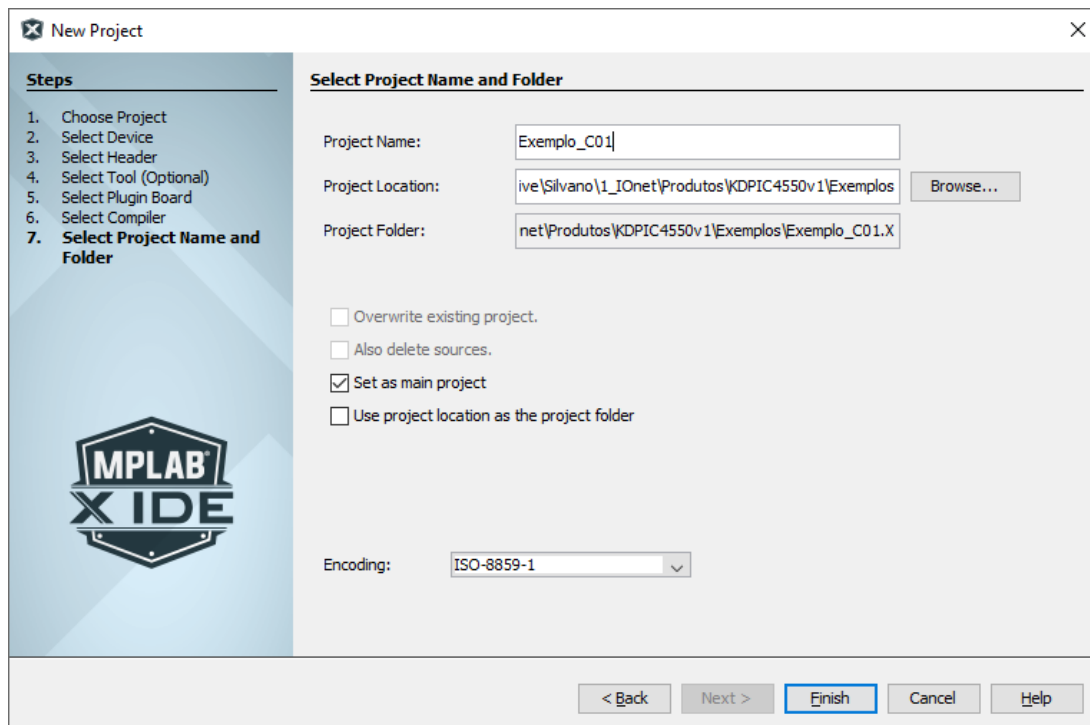
Figura 6.4: MPLAB X IDE - Criando um projeto em linguagem “C” – Passo 4.



Na próxima janela (Figura 5.5), em “Project Name:” digite o nome do projeto que está sendo criado. Clique no botão “Browse...” e selecione o diretório no qual será armazenado o seu projeto, ou digite o caminho no campo “Project Location”.

Selecione a opção “Set as main project” para tornar o seu novo projeto como o principal no MPLAB X IDE no momento da compilação e, clique no botão “Finish” para concluir a criação do seu projeto.

Figura 6.5: MPLAB X IDE - Criando um projeto em linguagem "C" – Passo 5.

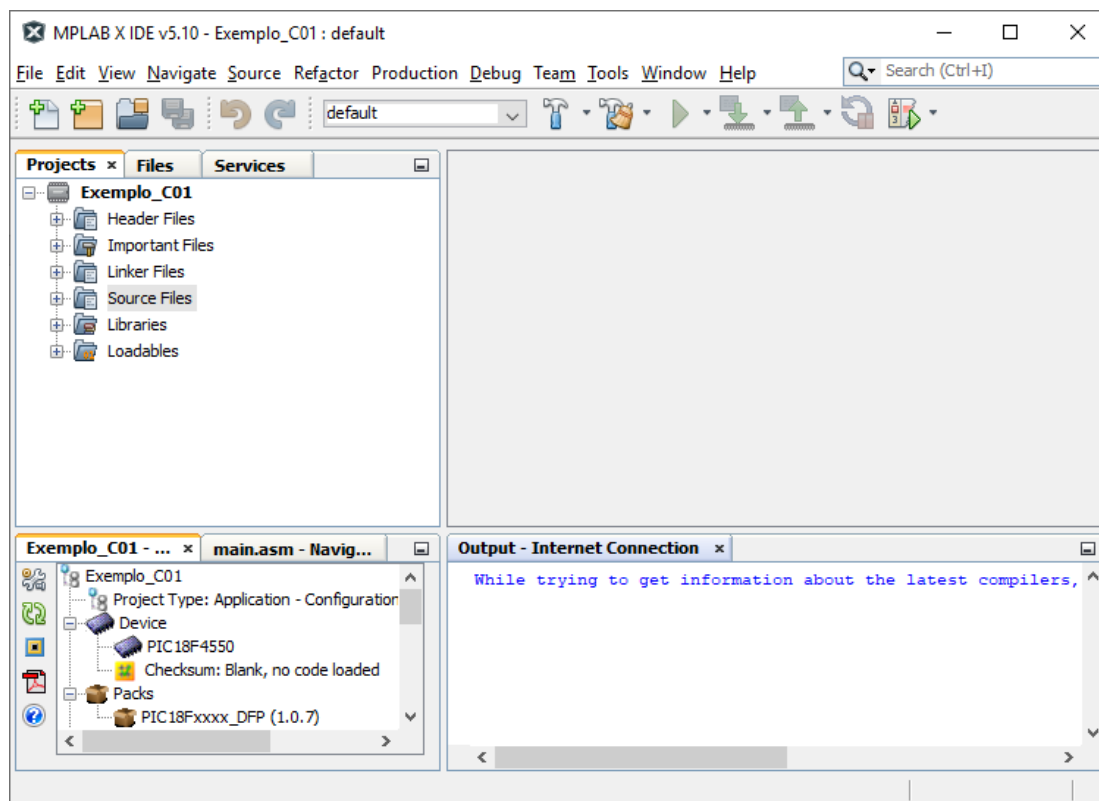


Observe que na lista de projetos da janela principal do MPLAB X IDE o projeto criado é exibido bem como sua estrutura de configuração (Figura 6.6).

Na pasta onde foi criado o seu projeto o mesmo aparece como uma pasta "NomeDoProjeto.X".

Caso seja necessário mudar o projeto de pasta, basta copiar o mesmo para o novo local e toda a configuração, bem como arquivos de código que estão nela.

Figura 6.6: MPLAB X IDE –Projeto criado em linguagem “C”.

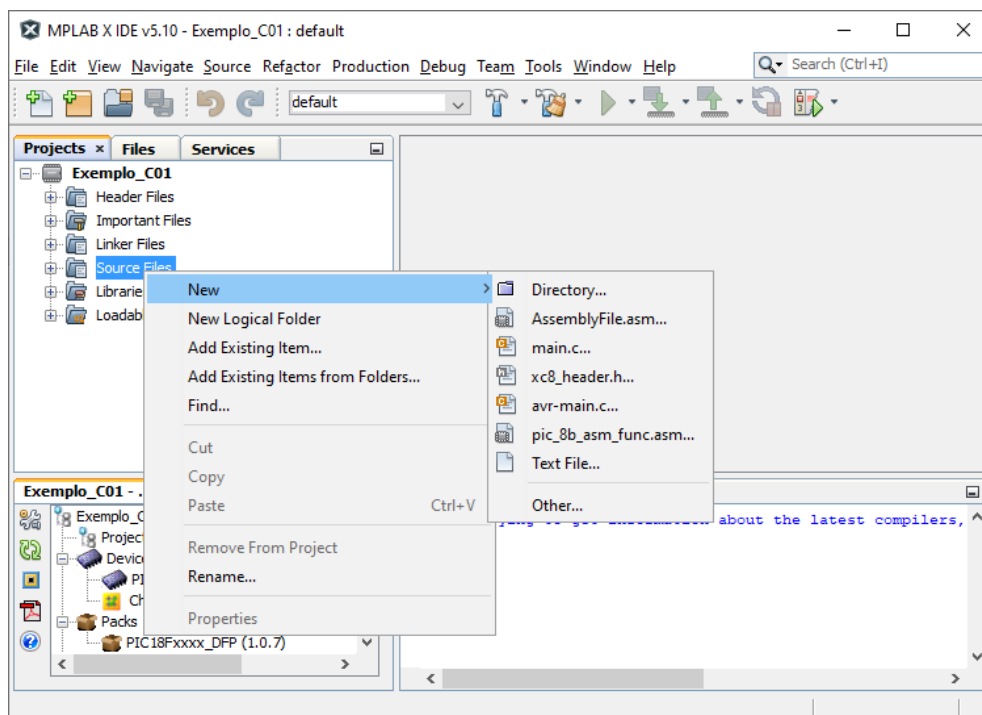


6.2 Criando o arquivo de código

Com o projeto criado, agora se pode definir o código fonte em “C” clicando com o botão direito do mouse na opção “Source Files” do menu flutuante. Clique em “New -> main.c...” (Figura 6.7).

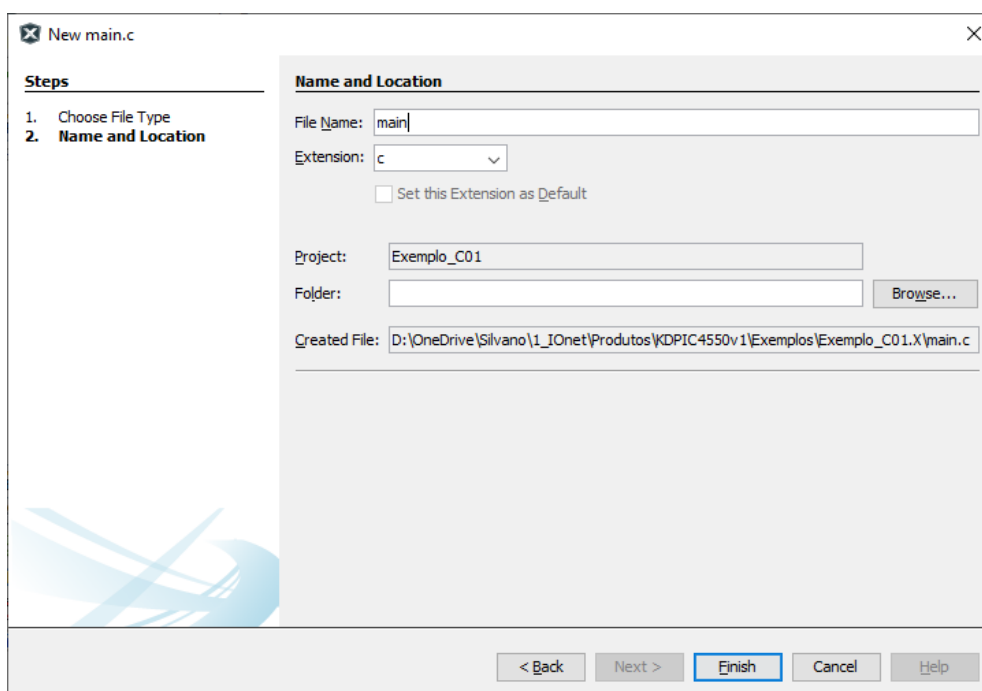
Lembre-se de que o arquivo de código fonte em “C” possui extensão “.c” e o arquivo de cabeçalho possui extensão “.h”.

Figura 6.7: MPLAB X IDE - Criando o arquivo de código em "C" – Passo 1.



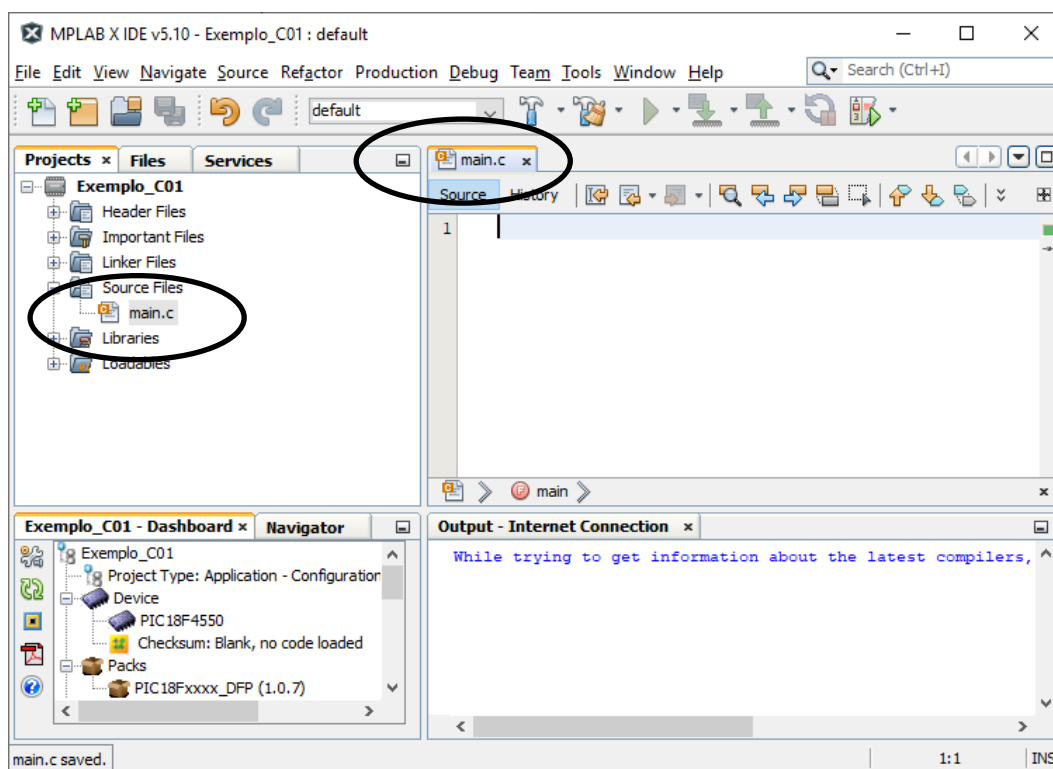
Uma nova janela é aberta (Figura 6.8). Digite o nome do arquivo no campo "File Name" e clique no botão "Finish".

Figura 6.8: MPLAB X IDE - Criando o arquivo de código em "C" – Passo 2.



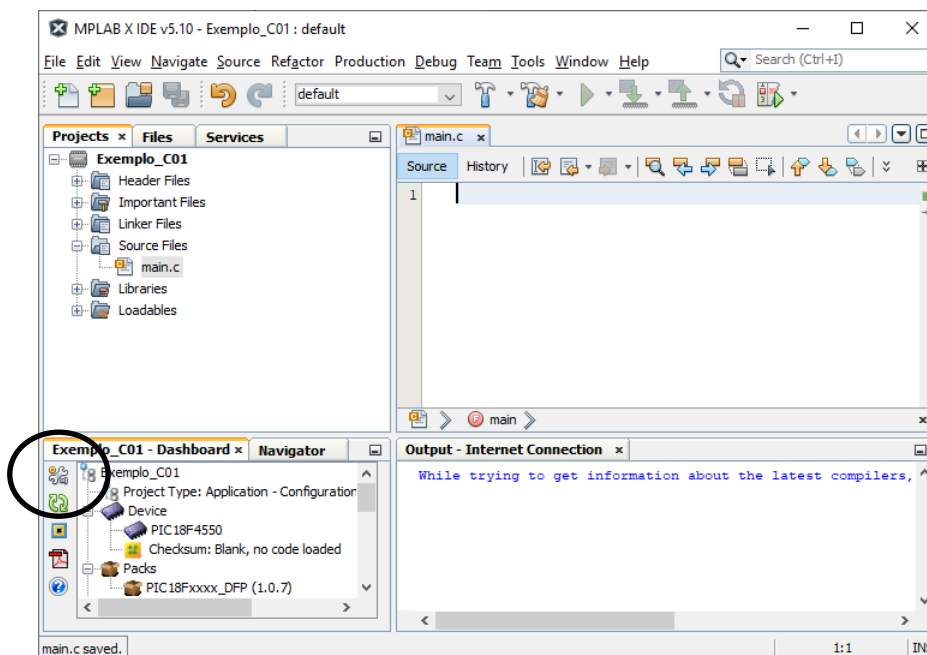
Observe que o arquivo de código “C” aparece na estrutura do projeto e também é aberto para edição (Figura 6.9).

Figura 6.9: MPLAB X IDE - Arquivo de código em “C” criado.



Como o kit possui o *bootload* é necessário incluir um parâmetro de compilação, para isto, acesse as propriedades do projeto clicando em “Project Properties” (Figura 6.10).

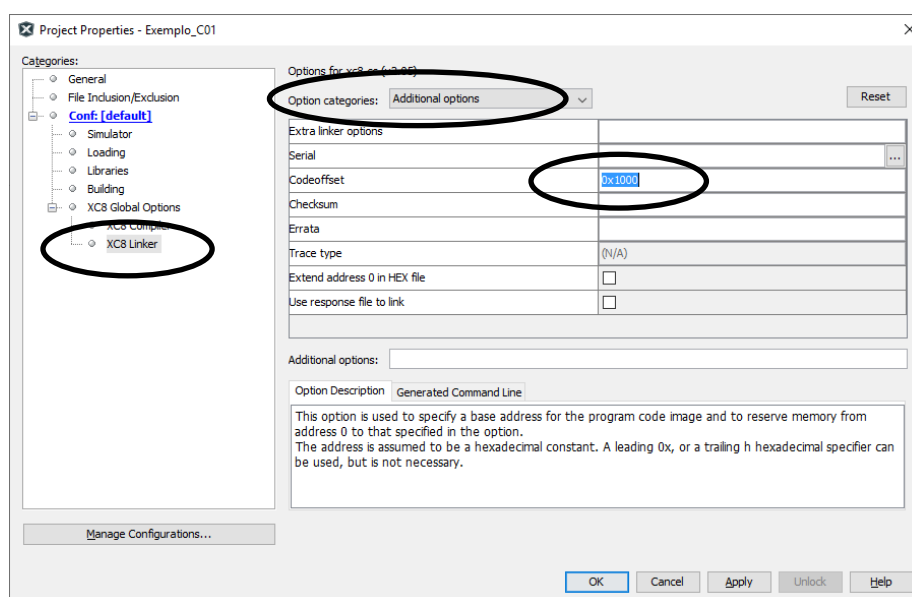
Figura 6.10: MPLAB X IDE – Abrindo as propriedades do projeto.



Na janela “Project Properties” clique “XC8 Linker” e em “Option Categories” selecione a opção “Additional options” e digite o valor “0x1000” no campo “Codeoffset” (Figura 6.11).

Alterado este valor, clique no botão “Apply” e em seguida no botão “Ok” para fechar esta janela. Feito isto o código gerado ao compilar o projeto estará adequado para ser carregado no kit usando o software “Microchip USB HID Bootloader”.

Figura 6.11: MPLAB X IDE – Abrindo as configurações do XC8 Linker.

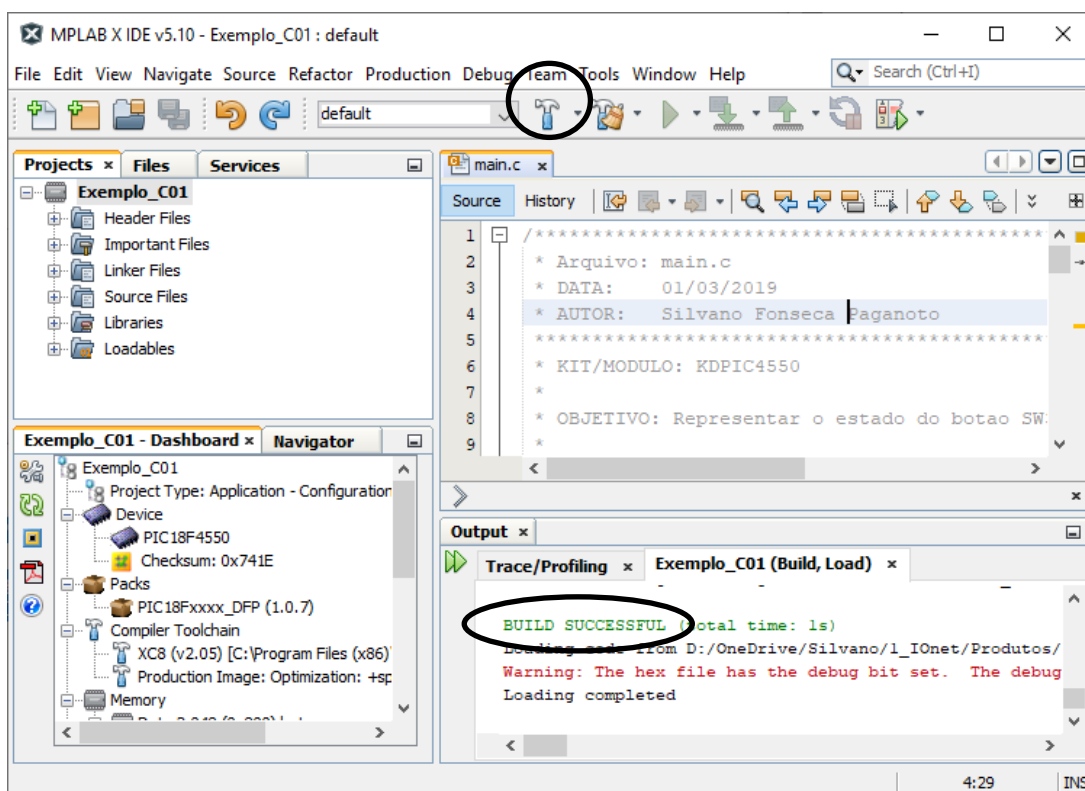


6.3 Compilando o projeto

Uma vez com o código fonte digitado, para compilá-lo, basta clicar no ícone “Build Main Project”.

Caso o programa não tenha nenhum problema de sintaxe, será exibida a mensagem ‘BUILD SUCCESSFUL’ na janela “Output” (Figura 6.12), indicando que a compilação foi bem sucedida.

Figura 6.12: MPLAB X IDE – Compilando o projeto.



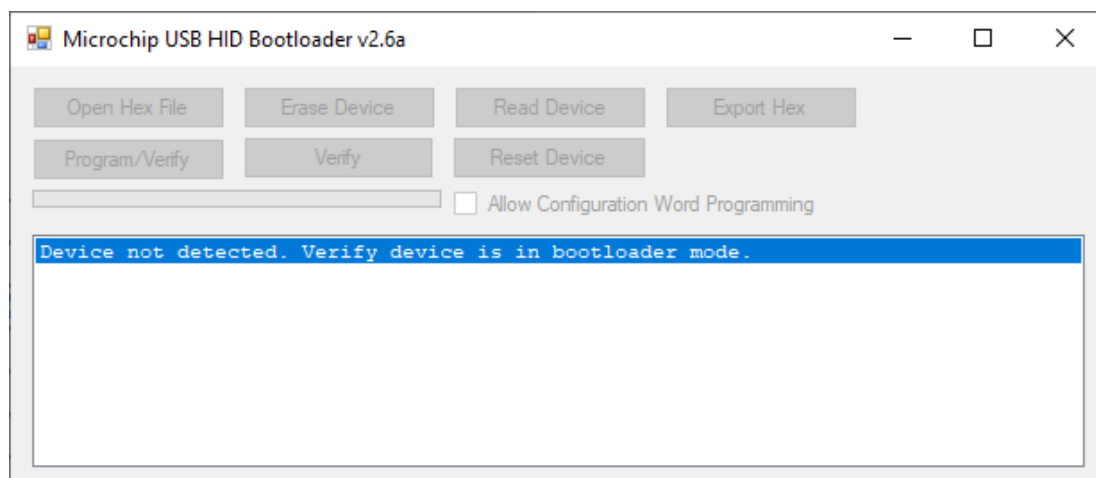
Após a compilação é criado um arquivo com extensão “.hex” dentro de “[pasta do projeto]\dist\default\production”. Este arquivo é o que deverá ser carregado para o Kit via programa HIDBootLoader.

7. GRAVANDO O CÓDIGO COMPILADO NO KIT

Uma vez compilado o código fonte, é gerado o arquivo com extensão “.hex” na “[pasta do projeto]\dist\default\production” de seu projeto.

Este arquivo é o que deverá ser transferido para o *Kit*. A forma de realizar esta transferência ou gravação do microcontrolador é por meio do “Microchip USB HID Bootloader”. Este programa pode ser baixado no site da Microchip®, versão v2.6a, utilizada neste manual (Figura 7.1).

Figura 7.1: Microchip USB HID Bootloader.



O “Microchip USB HID Bootloader” não necessita de instalação, mas geralmente é fornecido em um arquivo compactado (.zip), no qual está contido o arquivo executável e as bibliotecas necessárias para o seu perfeito funcionamento.

Assim, é necessário descompactá-lo em uma pasta de sua escolha e, se preferir, criar um atalho para *desktop* para facilitar seu uso.

7.1 Ativando o modo de programação do kit

O *Kit* possui gravado em uma área específica da memória do microcontrolador, um mecanismo de autoprogramação chamado “bootload”. Este recurso permite que o microcontrolador receba um programa sem a necessidade de um gravador externo.

Para acessar o mecanismo de autoprogramação, ou seja, entrar no modo programação do *Kit*, siga os passos abaixo:

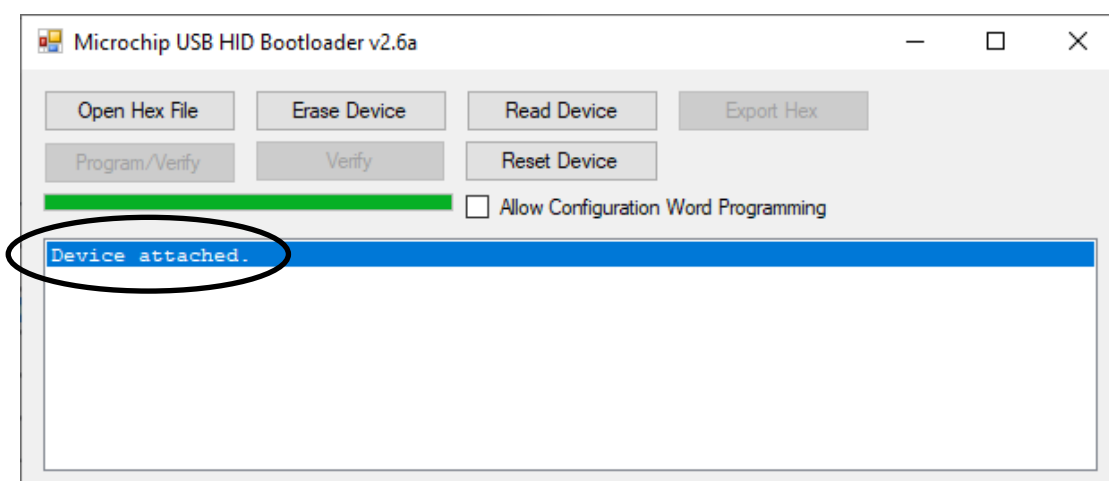
1. Execute o software “Microchip USB HID Bootloader”. Note que os botões ficaram desabilitados até que o *Kit* esteja conectado e em modo de programação;
2. Ligue a alimentação do *Kit*. Caso esteja usando uma fonte externa para alimentar o *Kit*, posicione o *jumper* “JP1” em “EXT”. Caso contrário, posicione o *jumper* “JP1” em “USB”. Assim, o próprio cabo USB alimentará o *Kit*;

3. Conecte o *Kit* ao computador utilizando um cabo USB;
4. Pressione e mantenha pressionado o botão “SW4/Reset”, em seguida, pressione e mantenha pressionado o botão “SW3/Boot”;
5. Libere o botão “SW4/Reset” e em seguida libere o botão “SW3/Boot”.

Após executar estes passos, o *Kit* estará no modo autoprogramação e no “Microchip USB HID Bootloader” indicará “Device attached” (Figura 7.2).

Observação: se durante a energização do *Kit* se o botão “SW3/Boot” estiver pressionado, o *Kit* também entrará no modo de autoprogramação!

Figura 7.2: Microchip USB HID Bootloader - Kit em modo de autoprogramação

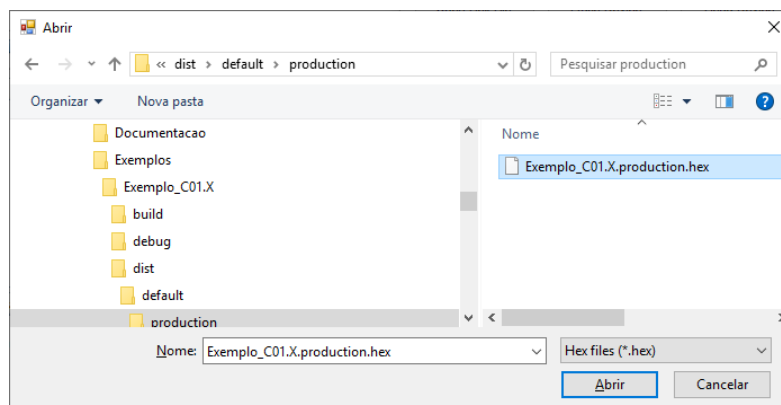


Para sair do modo de autoprogramação, basta pressionar brevemente o botão “SW4/Reset” do *Kit* ou no botão “Reset Device” do “Microchip USB HID Bootloader”.

7.2 Realizando a gravação do Kit

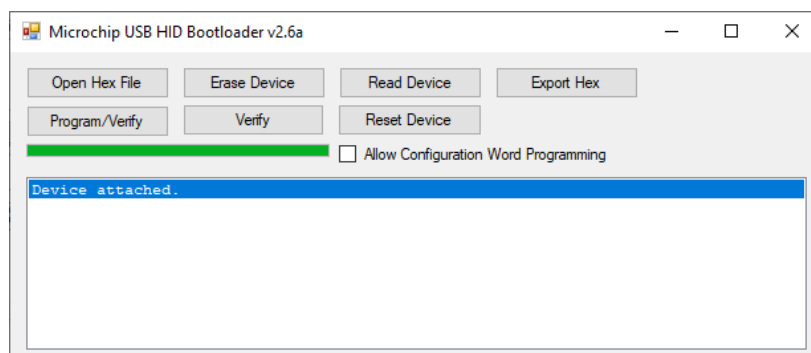
Estando com o *Kit* no modo de autoprogramação e conectado ao computador, ao executar o “Microchip USB HID Bootloader”, clique no botão “Open Hex File”. Na nova janela aberta, selecione o arquivo com extensão “.hex” contido na “[pasta do projeto]\dist\default\production” de seu projeto (Figura 7.3). Clique no botão “Abrir”.

Figura 7.3: Microchip USB HID Bootloader – Selecionando o arquivo “.hex”.



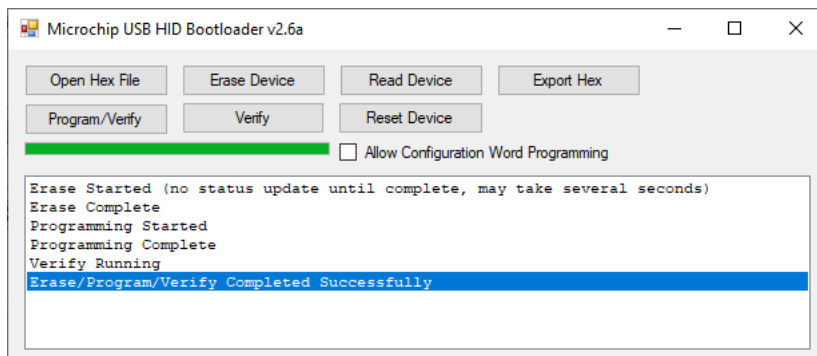
Note que na janela do “Microchip USB HID Bootloader” o botão “Program/Verify” é habilitado (Figura 7.4).

Figura 7.4: Microchip USB HID Bootloader – Arquivo “.hex” selecionado e pronto para gravar.



Para realizar a gravação, clique no botão “Program/Verify”, quando estiver concluída a gravação, será exibida a mensagem “Erase/Program/Verify Completed Successfully” indicando que a gravação foi realizada com sucesso (Figura 7.5).

Figura 7.5: Microchip USB HID Bootloader – Gravação realizada



Neste momento, o microcontrolador já está gravado com o seu programa, mas ainda o mesmo não está em execução, pois o *Kit* ainda está no modo de autoprogramação.

Para sair do modo de autoprogramação e executar o programa carregado, clique no botão “Reset Device”. Também é possível sair do modo de autoprogramação e executar o programa carregado pressionando brevemente o botão “Reset” presente no *Kit*.

Observação: quando o kit não está no modo de autoprogramação, o “Microchip USB HID Bootloader” indica que o *Kit* está desconectado, mesmo ele estando fisicamente ligado ao computador.

8. BIBLIOGRAFIA

MICROCHIP®. **PIC18F2455/2550/4455/4550 Data sheet**. Disponível em:

<<http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>>. Acesso em: 14 abr. 2012.

MICROCHIP®. **MPLAB X IDE Platform Essentials**. Contêm manuais e vídeos sobre a ferramenta bem como download da mesma. Disponível

em: <<http://www.microchip.com/pagehandler/en-us/family/mplabx>>. Acesso em: 14 abr. 2012.

MICROCHIP®. **USB Framework for PIC18, PIC24 & PIC32**. Contem informação e código fonte sobre como estabelecer comunicação USB com Microcontroladores PIC. Disponível em: <

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en537044>. Acesso em: 14 abr. 2012.

PEREIRA, Fábio. **Microcontrolador PIC18 Detalhado - Hardware e Software**. 1a.edição, São Paulo: Ed. Érica, 2010.

9. NOTA DA IONET

Obrigado por ter adquirido um produto da *IONet*. Este breve manual foi escrito para ajudar o utilizador a desfrutar da melhor forma possível o *Kit* de desenvolvimento.

Leia-o atentamente antes da utilização e guarde-o em local acessível para posterior consulta.

Caso encontre alguma incoerência ou erro, contate-nos.