

Microcontrolador PIC no GNU/Linux

Este artigo descreve como implementar um ambiente de programação para microcontroladores [Microchip PIC](#), no Sistema Operacional GNU/Linux.



- [Introdução](#)
- [Desenvolvimento em Assembly](#)
- [Desenvolvimento em C](#)
- [Desenvolvimento em JAL](#)
- [Desenvolvimento em Outras Linguagens](#)
- [Programador \(Hardware e Software\)](#)
- [Simulador](#)
- [IDE para Desenvolvimento](#)
- [Screenshots da IDE PikLab](#)
- [Créditos](#)
- [Referências e Bibliografia](#)

Introdução [\[Voltar para o Sumário\]](#)

Tradicionalmente, um desenvolvedor de software para PIC's utiliza o [MPLAB](#), em conjunto com o assembler MPASM ou com um compilador C fornecido pela Microchip (ou por terceiros), disponíveis somente em versões comerciais e compatíveis somente com o Microsoft Windows.

Os problemas na migração para o GNU/Linux iniciam com a ~~falta de suporte oficial da Microchip para o GNU/Linux~~: atualmente existe apenas um software programador em linha de comando (para uso com o PICKit 2).

*** Atualização (jun/2012):** A Microchip está lançando a IDE [MPLAB X](#), que traz um ambiente de desenvolvimento baseado no [NetBeans](#), e um conjunto de compiladores (comerciais), para suportar os Sistemas Operacionais Microsoft Windows, GNU/Linux e Apple MacOS X.

Entretanto, as comunidades *open-source* desenvolvem algumas ferramentas que podem ser usadas para essa finalidade; muitas delas estão listadas no site [GNU PIC](#). Aqui serão apresentadas algumas dessas ferramentas.

Desenvolvimento em Assembly [\[Voltar para o Sumário\]](#)

A melhor opção para programação em Assembly é utilizar o [GPUTILS](#), que é um conjunto de programas semelhantes aos fornecidos pela Microchip (adicionando ainda um disassembler). É totalmente *open-source*, e compatível com os programas oficiais, fornecendo até mesmo outras funcionalidades adicionais.

Nas distribuições derivadas do Debian (como é o caso do Ubuntu), o repositório oficial de pacotes fornece o GPUTILS para a instalação através do APT-GET:

```
sudo apt-get install gputils
```

Em outras distribuições, os repositórios oficiais também podem fornecer o pacote gputils, mas também há a opção de se realizar o download dos fontes diretamente do [site do GPUTILS](#) e compilar manualmente.

As ferramentas disponibilizadas pelo GPUTILS são:

- Assembler: **gpasm**
- Linker: **gplink**
- Disassembler: **gpdasm**
- Outras ferramentas: **gplib, gpstrip, gpvc, gpvo**

Para desenvolver em Assembly, pode-se usar um editor de textos qualquer, juntamente com o GPUTILS. Para projetos maiores, a ferramenta GNU Make pode ser usada, através do uso de Makefiles.

Uma opção melhor do que essa é utilizar um ambiente de desenvolvimento (IDE), que já fornece um editor de textos com destaque de sintaxe, um gerenciador de projetos e até mesmo algumas ferramentas para transferir o programa para o microcontrolador. Essa opção será vista [mais adiante](#) neste artigo.

Desenvolvimento em C [\[Voltar para o Sumário\]](#)

Existem versões comerciais de alguns compiladores C compatíveis com o GNU/Linux, além da possibilidade de rodar compiladores para Microsoft Windows através do *Wine*.

Os compiladores comerciais mais usados (alguns deles com versões para GNU/Linux) são: [HI-TECH Software](#), [Source BOOST](#), [CCS Compilers](#), [MPLAB XC Compilers](#).

No mundo *open-source*, há atualmente duas alternativas viáveis:

- [SDCC](#) (*Small Device C Compiler*);
- [CPIK](#) (The C compiler for PIC18 devices).

O **SDCC** (*Small Device C Compiler*) é um compilador designado para microcontroladores/microprocessadores Intel MCS51, Freescale HC08 e Zilog Z80, mas tem um port específico para Microchip PIC.

O problema é que esse port não tem o mesmo ritmo de desenvolvimento do projeto SDCC como um todo, e está constantemente num status "*work in progress*", segundo seus mantenedores.

O SDCC tem um suporte relativamente bom para os PIC's de 16 bits (que são os da série 18Fxxxx), um suporte bastante limitado (e pouco funcional) para os PIC's de 14 bits (da série 16Fxxx) e nenhum suporte para outras séries de PIC's (10, 12, 17, 24, dsPIC).

Ele pode ser instalado no Debian / Ubuntu através do repositório oficial, com o APT-GET:

```
sudo apt-get install sdcc-nf sdcc-doc-nf
```

Em outras distribuições, os repositórios oficiais também podem fornecer o pacote sdcc (ou sdcc-nf), mas também há a opção de se realizar o download dos fontes diretamente do [site do SDCC](#) e compilar manualmente.

O **CPIK** é um compilador desenvolvido por Alain Gibaud, e recentemente teve seu código-fonte disponibilizado sob a licença GPL. Seu suporte se limita aos PIC's de 16 bits (da série 18Fxxxx) e nem todos seus recursos ainda estão funcionais e estáveis.

Para instalar o CPIK, deve-se seguir as instruções descritas no site do projeto: [CPIK Downloads](#).

Para desenvolver em C, pode-se usar um editor de textos qualquer, juntamente com um dos compiladores. Para projetos maiores, a ferramenta GNU Make pode ser usada, através do uso de Makefiles.

[Mais adiante](#) neste artigo, será vista a opção de se utilizar um ambiente de desenvolvimento (IDE) , fornecendo um editor de textos com destaque de sintaxe, um gerenciador de projetos e até mesmo algumas ferramentas para transferir o programa para o microcontrolador.

Desenvolvimento em JAL [\[Voltar para o Sumário\]](#)

[JAL](#) é um acrônimo para "*Just Another Language*". É uma linguagem de fácil aprendizado, semelhante a Pascal, destinada ao desenvolvimento com microcontroladores Microchip PIC.

Atualmente, o compilador JAL mais usado é o [JALv2](#), que é uma reescrita completa do compilador original criado por Wouter van Ooijen, com diversos melhoramentos e otimizações.

O compilador JALv2 suporta uma variedade de PIC's, desde os PIC10xx até os PIC18Fxxxx.

Para facilitar o desenvolvimento com o JALv2, há um projeto denominado [jallib](#), que consiste num conjunto de bibliotecas desenvolvidas e testadas pela comunidade (ADC, PWM, UART, USB, I2C, LCD, FAT32, PATA, etc...).

A maneira mais fácil de utilizar o compilador JALv2 é realizar o download de um pacote completo no [site do projeto jallib](#), que já contém o binário do compilador (para Microsoft Windows, GNU/Linux ou Apple MacOS X), o conjunto completo de bibliotecas, documentação, e vários programas-exemplo (*samples*), escritos em JAL.

[Mais adiante](#) neste artigo, será vista a opção de se utilizar um ambiente de desenvolvimento (IDE) , fornecendo um editor de textos com destaque de sintaxe, um gerenciador de projetos e até mesmo algumas ferramentas para transferir o programa para o microcontrolador.

No artigo [Ambiente de Desenvolvimento para Microchip PIC, com Eclipse, JAL, e GNU/Linux](#) há um roteiro de como preparar um ambiente de desenvolvimento para microcontroladores Microchip PIC, usando [Eclipse](#), linguagem JAL, e GNU/Linux, com suporte ao programador PICkit 2 (ou ao [PK2C](#)).

Desenvolvimento em Outras Linguagens [\[Voltar para o Sumário\]](#)

Além dos tradicionais Assembly e C, alguns fabricantes (como o [Source BOOST](#)) disponibilizam compiladores comerciais que permitem o desenvolvimento em outras linguagens, como BASIC, Pascal, C++, etc.

Programador [\[Voltar para o Sumário\]](#)

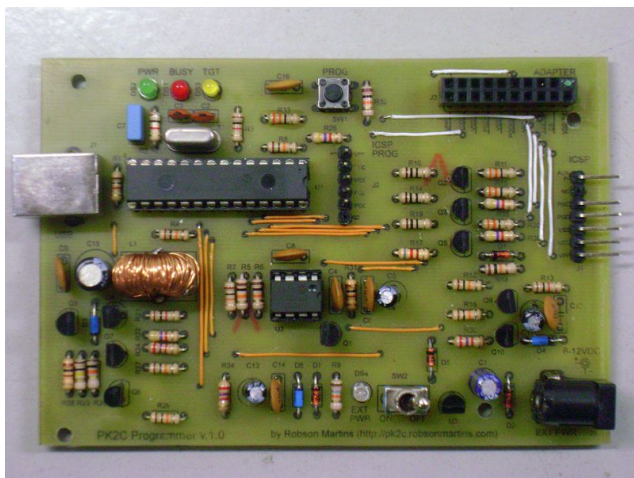
Para transferir o programa (arquivo HEX) gerado pelo assembler ou pelo compilador, é necessária uma placa (*hardware*) de gravação, juntamente com um software apropriado. Normalmente, os desenvolvedores possuem kits prontos adquiridos através da Microchip ou de seus representantes (como PicStart, ICD, PICkit). Além disso, há algumas opções livres na Internet, que possibilitam montar o seu próprio gravador (são disponibilizados diagramas esquemáticos, layouts de placas e até mesmo descrições do funcionamento).

Aqui mesmo neste site, há duas opções de programador:

- [PK2C \(PICkit 2 Clone\)](#), um clone do PICkit 2 original da Microchip, que funciona conectado à porta USB do PC, e suporta a gravação de PIC's de

diferentes famílias e memórias E2PROM seriais, além de outros recursos.

- [MPSP \(Microchip PIC Serial Programmer\)](#), que funciona conectado à porta serial do PC e permite a gravação de vários dispositivos (PIC) diferentes, além de memórias E2PROM seriais.



Programador PK2C



Programador MPSP

O *software* usado para gravação deve ser compatível com o *hardware* programador, algumas opções existentes são *open-source* e podem ser vistas nas páginas:

[PK2C - Softwares suportados](#) e [MPSP - Softwares suportados](#).

Simulador [\[Voltar para o Sumário\]](#)

Como simulador, existem basicamente duas opções para o GNU/Linux: uma é o uso da nova IDE *freeware* da Microchip ([MPLAB X](#)), que apesar de gratuita, é proprietária.

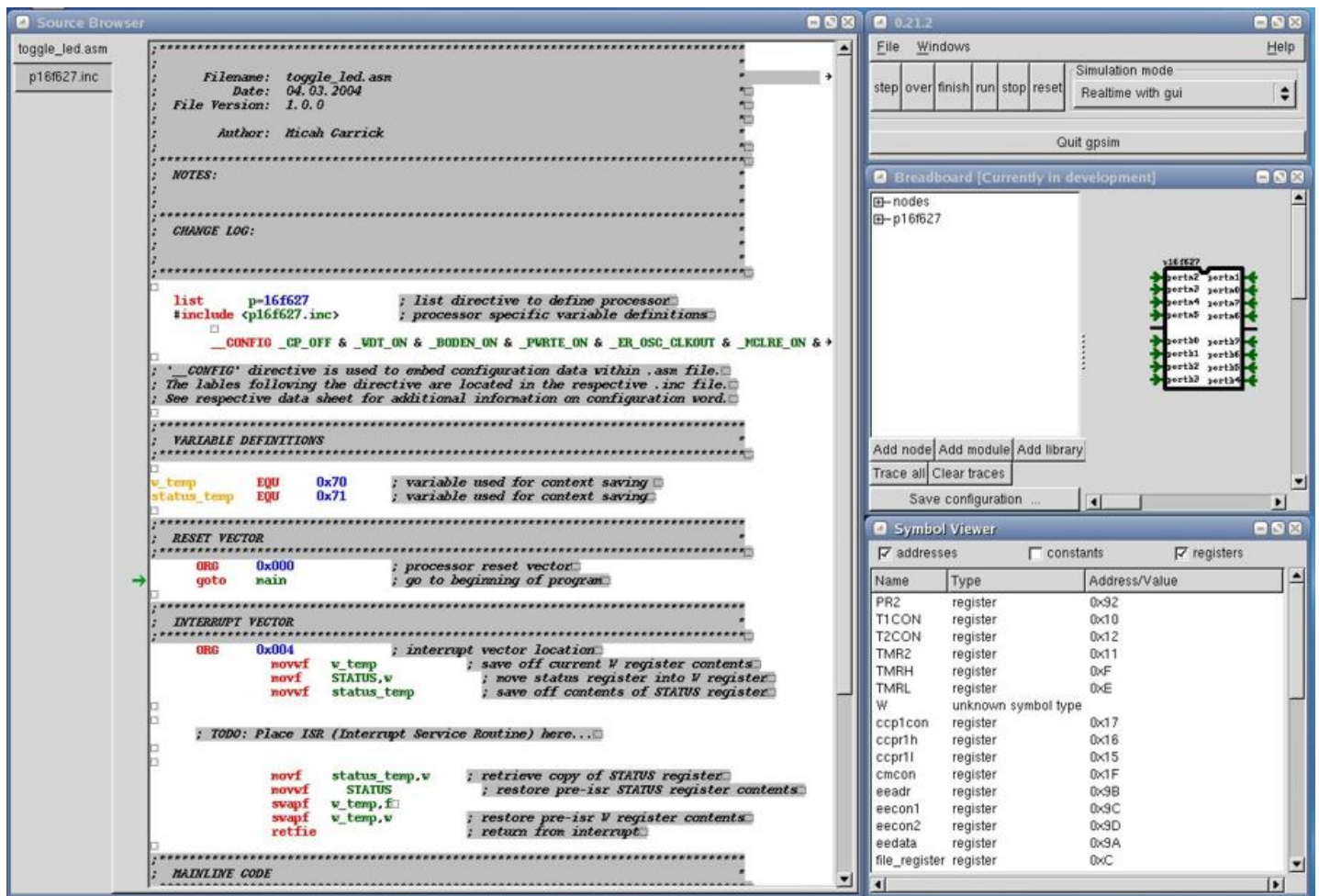
A outra opção é o [GPSIM](#), que é totalmente *open-source*.

Para instalar o GPSIM no Debian / Ubuntu, use o repositório oficial através do APT-GET:

```
sudo apt-get install gpsim gpsim-doc gpsim-lcd gpsim-led gpsim-logic
```

Em outras distribuições, os repositórios oficiais também podem fornecer os pacotes `gpsim`, mas também há a opção de se realizar o download dos fontes diretamente do [site do GPSIM](#) e compilar manualmente.

A grande limitação do GPSIM é o suporte a apenas alguns modelos de PIC.



GPSIM em ação

IDE para Desenvolvimento [\[Voltar para o Sumário\]](#)

Existem algumas IDE's para desenvolvimento de PIC's no GNU/Linux. Algumas delas são proprietárias, outras *freeware* (como o [MPLAB X](#)), outras totalmente *open-source*.

Aqui serão abordadas algumas opções de IDE *open-source*.

• PikDev

A primeira é a IDE [PikDev](#), que fornece um editor de textos poderoso, integração com as ferramentas GPUTILS para programação em Assembly e possibilidade de gravar PIC's com *hardwares* alternativos via porta serial ou paralela (atualmente ele suporta também o PICKIT 2, fornecido pela Microchip, e seus clones). O PikDev não está nos repositórios oficiais da maioria das distribuições, devendo ser instalado via download de um pacote binário, ou então compilado manualmente.

• PikLab

Uma melhor alternativa ao PikDev é o [PikLab](#), que é um *fork* do projeto PikDev, porém adiciona os recursos:

- Integração com vários compiladores (incluindo os comerciais, até mesmo via *Wine*), permitindo programação em Assembly (usando GPUTILS), C (incluindo o SDCC), C++, BASIC, JAL e outras linguagens.
- Integração com o GPSIM (simulador de PIC's).
- Suporte a gravadores diretos (via porta serial ou paralela) e gravadores oficiais, como ICD, PicStart, PICKIT (via porta serial ou USB).
- Depuração (debugger) com o ICD ou GPSIM.
- Além de um editor de textos com destaque de sintaxe e um poderoso gerenciador de projetos.

Para instalar o PikLab no Debian / Ubuntu, use o repositório oficial através do APT-GET:

```
sudo apt-get install piklab
```

Em outras distribuições, os repositórios oficiais também podem fornecer os pacotes *piklab*, mas também há a opção de se realizar o download dos fontes diretamente do [site do PikLab](#) e compilar manualmente.

• Eclipse

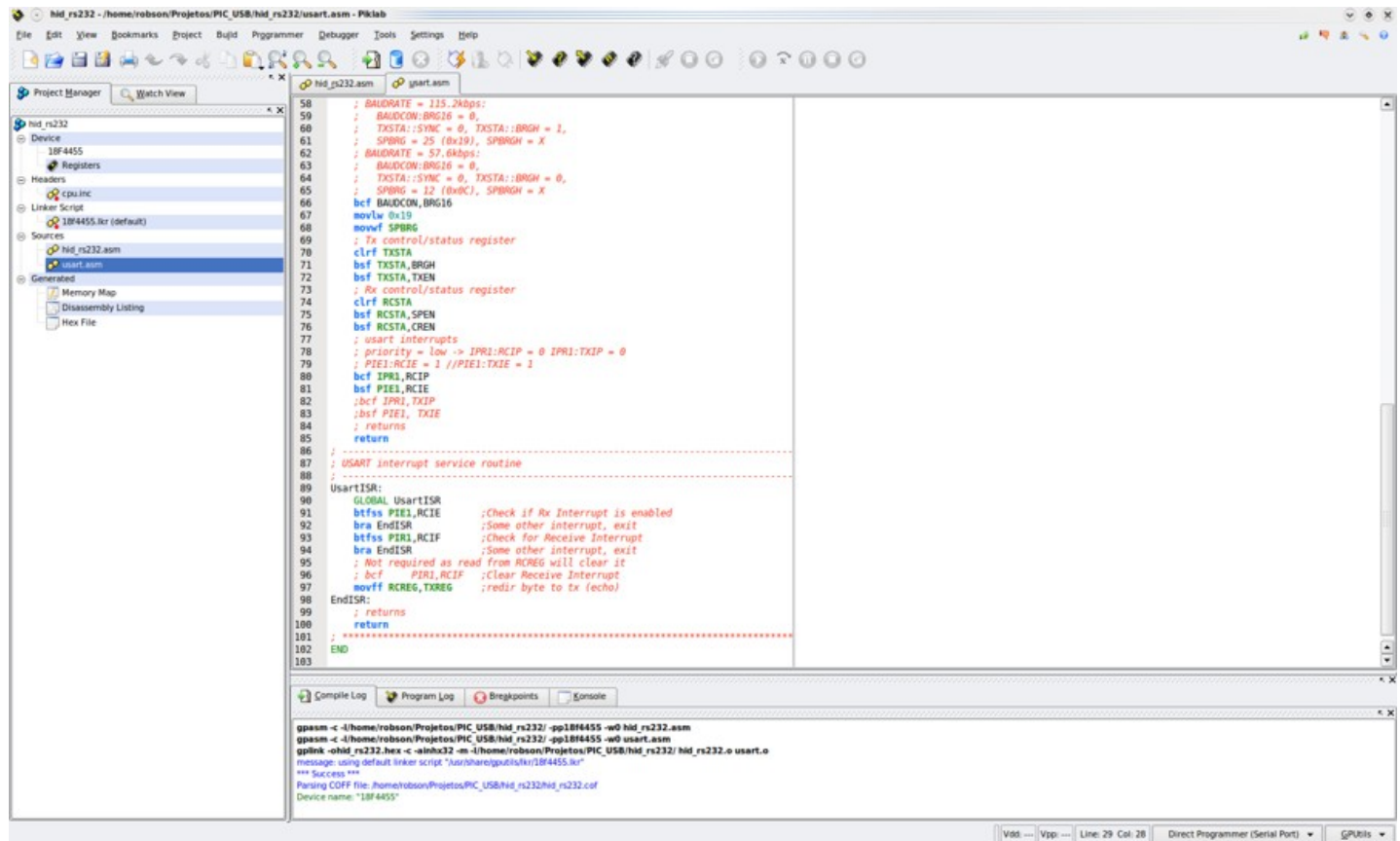
Outra opção de IDE para o desenvolvimento de PIC's (usando linguagem JAL), é o [Eclipse](#), uma IDE *open-source* bastante conhecida no mundo dos programadores Java.

Neste caso, é usado um [plug-in](#) do Eclipse para suportar a sintaxe JAL, em conjunto com o compilador (JALv2) e as bibliotecas do projeto [jallib](#). Esse plug-in também suporta a gravação de dispositivos através do programador PICKit 2 (e seus clones).

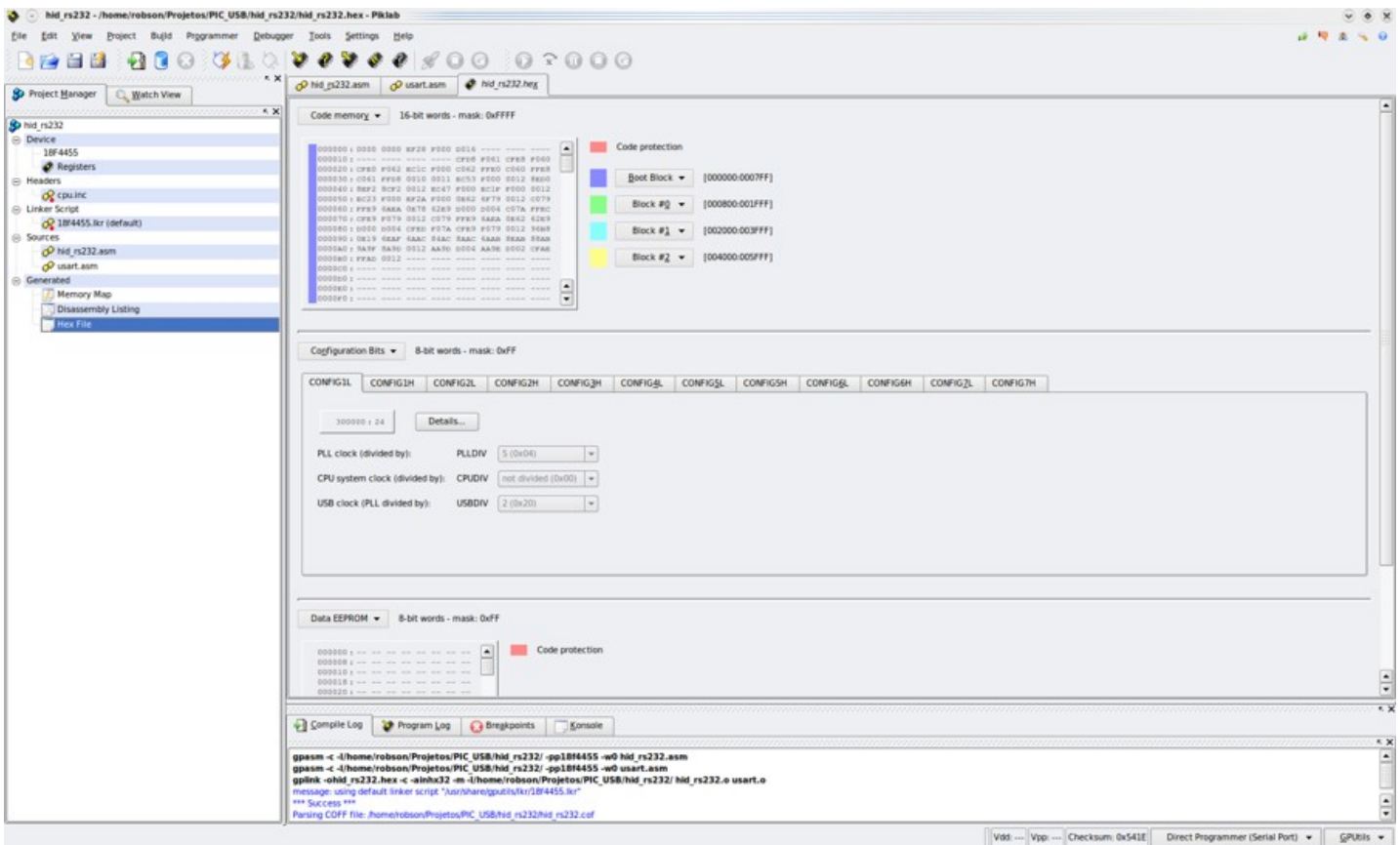
Esta é uma ótima opção *multiplataforma* para o desenvolvimento de PIC's em linguagem JAL, pois a IDE Eclipse e o compilador JALv2 podem rodar satisfatoriamente em Microsoft Windows, GNU/Linux ou Apple MacOS X.

No artigo [Ambiente de Desenvolvimento para Microchip PIC, com Eclipse, JAL, e GNU/Linux](#) há um roteiro de como preparar um ambiente de desenvolvimento para microcontroladores Microchip PIC, usando Eclipse, linguagem JAL, e GNU/Linux, com suporte ao programador PICKit 2 (ou ao PK2C).

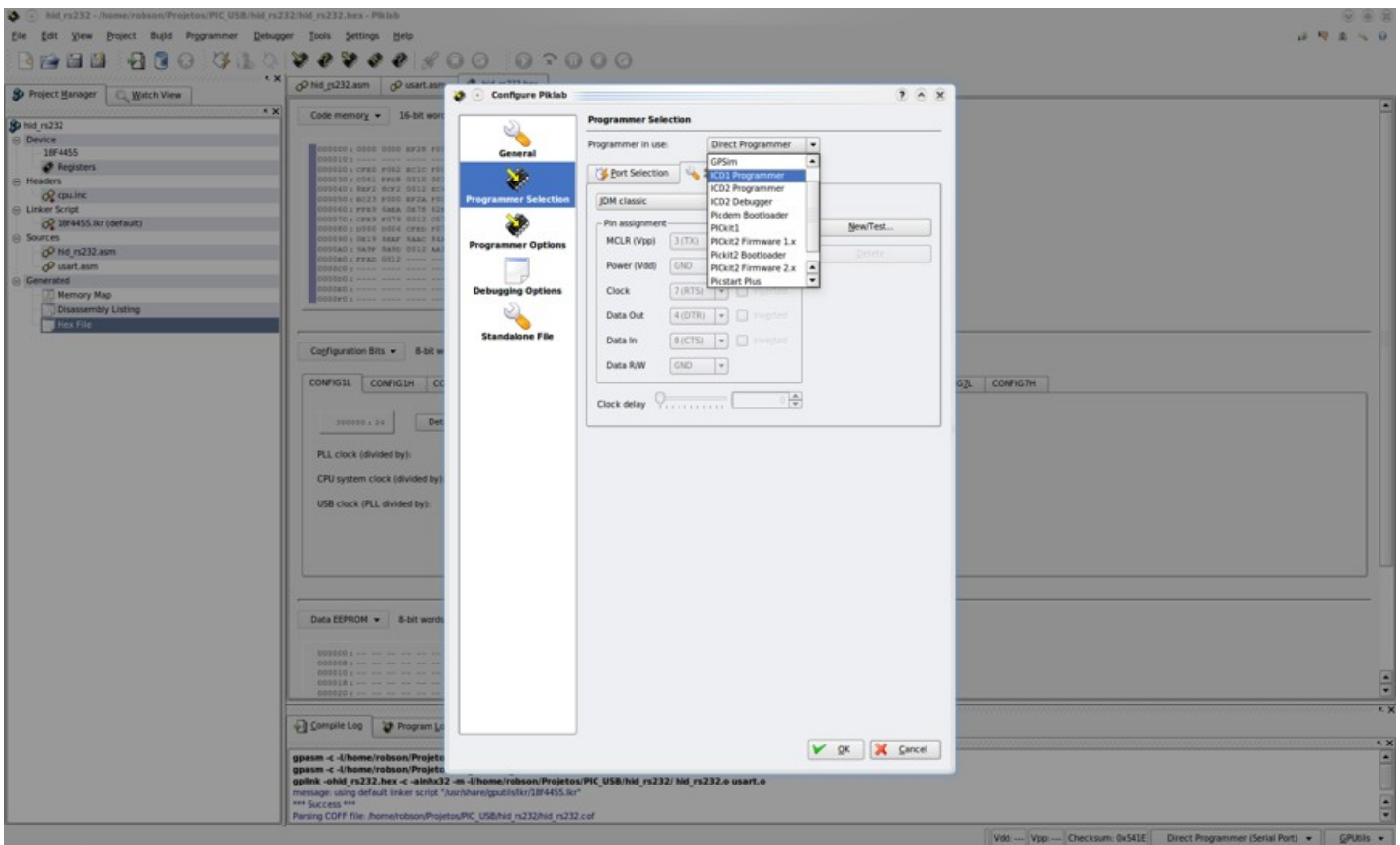
Screenshots do PikLab [\[Voltar para o Sumário\]](#)



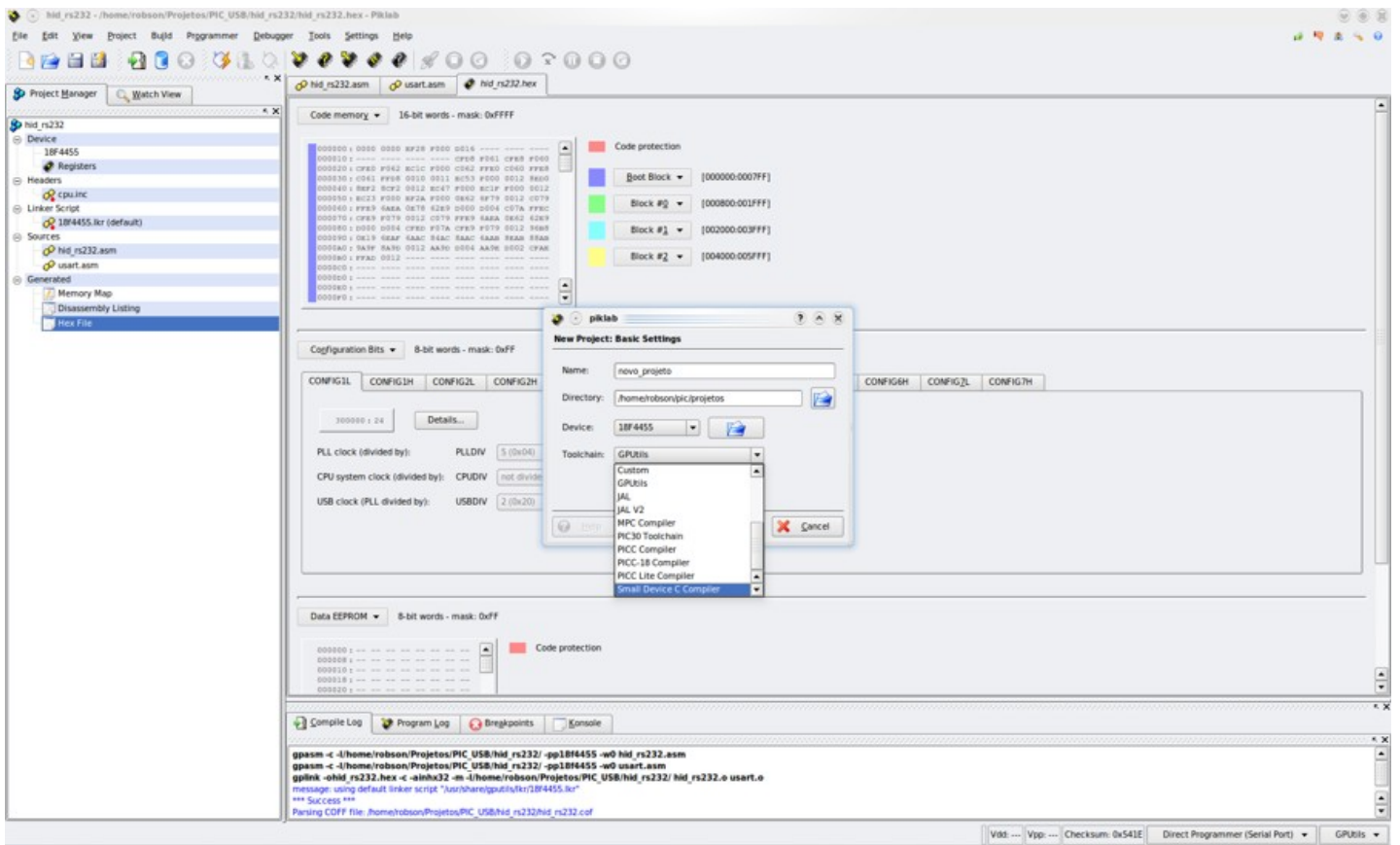
Edição de arquivo no PikLab



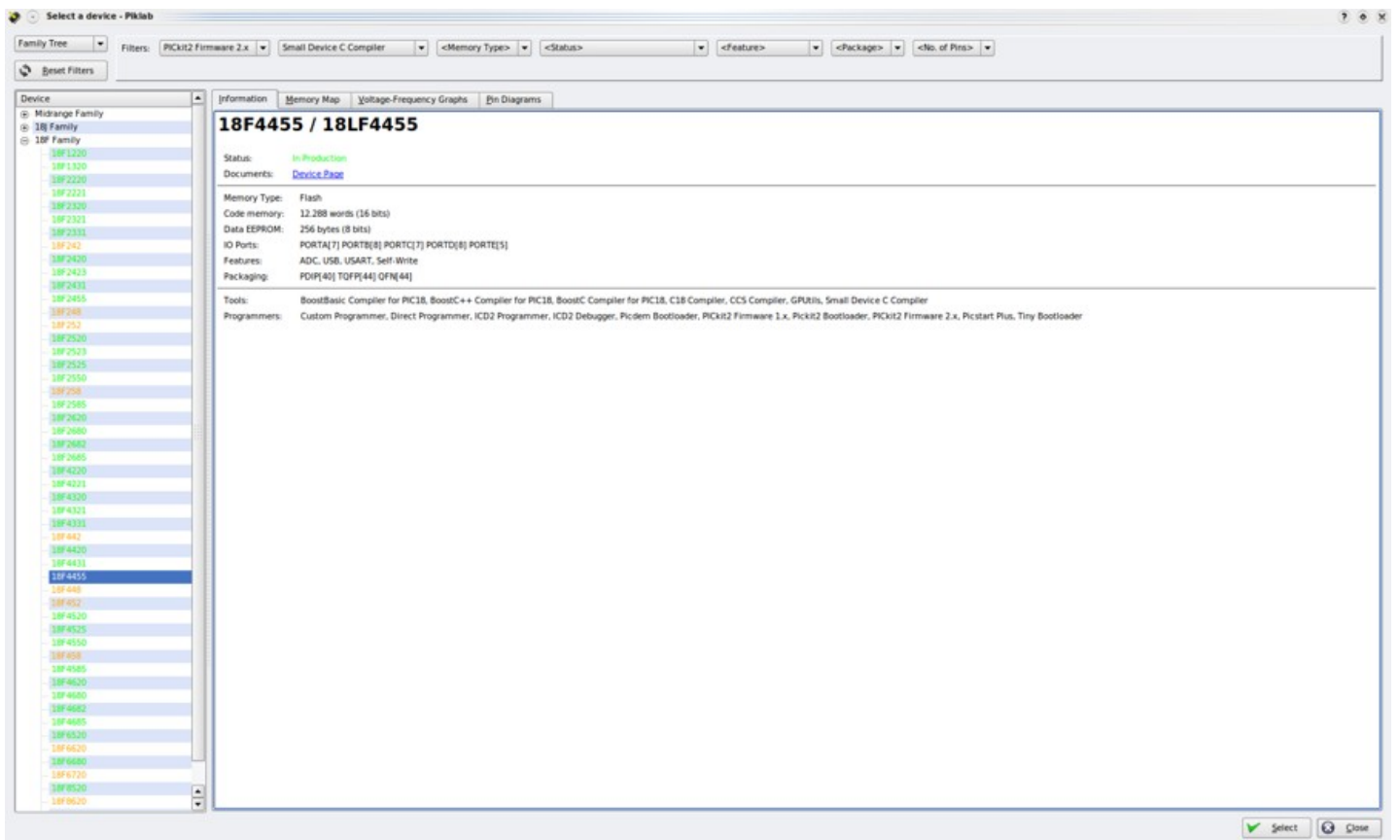
Arquivo HEX gerado pelo PikLab



Seleção de Gravador (Programmer) no PikLab



Seleção de Compilador (Toolchain) no PikLab



Informações sobre o modelo do PIC selecionado no PikLab

Créditos [\[Voltar para o Sumário\]](#)

Robson Martins

<http://www.robsonmartins.com>

Todos os autores e fabricantes aqui citados, com seus devidos créditos.

As marcas citadas podem ser registradas e são propriedade de seus fabricantes / detentores.

O conteúdo deste artigo pode ser livremente distribuído, desde que os nomes dos autores sejam mantidos. É proibida a comercialização ou apropriação de qualquer conteúdo aqui descrito.

Referências e Bibliografia: [\[Voltar para o Sumário\]](#)

Dattalo, Scott. **GNUPIC Page**.

<http://www.gnupic.org/> [site is down]

Microchip. **MPLAB IDE**.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002

Microchip. **MPLAB X**.

<http://www.microchip.com/mplabx/>.

Microchip. **MPLAB XC Compilers**.

<http://www.microchip.com/mplabxc/>.

HI-TECH Software. **HI-TECH C Compilers**.

<http://www.htsoft.com/>

SourceBoost Technologies. **SourceBoost Compilers**.

<http://www.sourceboost.com/>

Custom Computer Services, Inc. **CCS Compilers**.

<http://www.ccsinfo.com/>

Sourceforge.net. **GNU PIC Utilities**.

<http://gputils.sourceforge.net/>

Sourceforge.net. **Small Device C Compiler**.

<http://sdcc.sourceforge.net/>

Gibaud, Alain. **CPIK: The C compiler for PIC18 devices**.

<http://pikdev.free.fr/download-cpik.php3>

Just Another Language v2 (JALv2).

<http://www.casadeyork.com/jalv2/>

JAL Eclipse Plugin.

<https://code.google.com/p/jal-eclipse-plug-in/>

jallib - JAL V2 Libraries

<http://justanotherlanguage.org/>

Piklab - IDE for PIC microcontrollers.

<http://piklab.sourceforge.net/>

Pikdev - IDE for PIC microcontrollers.

<http://pikdev.free.fr/>

Sourceforge.net. **GPSIM**.

<http://gpsim.sourceforge.net/>

Microchip. **PICKit 2 Development Programmer/Debugger**.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023805

Martins, Robson. **Microchip PIC USB Programmer - PICKit 2 Clone (PK2C)**.

<http://www.robsonmartins.com/eletr/pk2c/>

Martins, Robson. **Microchip PIC Serial Programmer (MPSP)**.

<http://www.robsonmartins.com/eletr/mpsp/>

The Eclipse Foundation. **Eclipse IDE**.

<http://www.eclipse.org/>

Oracle. **NetBeans IDE**.

<http://netbeans.org/>

Microchip Forums. **"PICKit 2 stand-alone app device file update (1.62)" topic**.

<http://www.microchip.com/forums/m487219-print.aspx>

Todos os sites foram visitados em 09/06/2012.

[\[Voltar para o Sumário\]](#)