



## PicsimLab\_0\_7\_0

Luis Claudio Gambôa Lopes <lcgamboa@yahoo.com>

Download: [github](#) or [sourceforge](#)

February 4, 2018

# Contents

<b>I</b>	<b>Manual em Português</b>	<b>3</b>
1	Introdução	6
2	Interface	7
3	Placas	10
4	Comunicação Serial	16
5	Depuração Integrada com o MPLABX (PIC)	20
6	Integração com a IDE do Arduino (ATMEGA)	21
7	Depuração com o avr-gdb (ATMEGA)	22
8	Osciloscópio	23
9	Partes Avulsas	24
10	Como Fazer (How To's)	31
<b>II</b>	<b>English Manual</b>	<b>32</b>
1	Introduction	35
2	Interface	36
3	Boards	39
4	Serial Communication	45
5	MPLABX Integrated Debug (PIC)	49
6	Arduino IDE Integration (ATMEGA)	50

<i>CONTENTS</i>	<i>2</i>
<b>7 avr-gdb Debug (ATMEGA)</b>	<b>51</b>
<b>8 Oscilloscope</b>	<b>52</b>
<b>9 Spare Parts</b>	<b>53</b>
<b>10 How To's</b>	<b>60</b>
<b>III License</b>	<b>61</b>

## **Parte I**

# **Manual em Português**

# Table of Contents

<b>1</b>	<b>Introdução</b>	<b>6</b>
<b>2</b>	<b>Interface</b>	<b>7</b>
2.1	Janela Principal . . . . .	7
2.2	Comandos . . . . .	9
<b>3</b>	<b>Placas</b>	<b>10</b>
3.1	Características da Placa 1 . . . . .	10
3.2	Características da Placa 2 . . . . .	11
3.3	Características da Placa 3 . . . . .	12
3.4	Características da Placa 4 . . . . .	13
3.5	Características da Placa 5 . . . . .	14
<b>4</b>	<b>Comunicação Serial</b>	<b>16</b>
4.1	Instalação e Configuração do com0com (Windows) . . . . .	16
4.2	Instalação e Configuração do tty0tty (Linux) . . . . .	18
<b>5</b>	<b>Depuração Integrada com o MPLABX (PIC)</b>	<b>20</b>
<b>6</b>	<b>Integração com a IDE do Arduino (ATMEGA)</b>	<b>21</b>
<b>7</b>	<b>Depuração com o avr-gdb (ATMEGA)</b>	<b>22</b>
<b>8</b>	<b>Osciloscópio</b>	<b>23</b>

<i>TABLE OF CONTENTS</i>	5
<b>9 Partes Avulsas</b>	<b>24</b>
9.1 Servo Motor . . . . .	26
9.2 Step Motor . . . . .	26
9.3 Push Buttons . . . . .	27
9.4 Switchs . . . . .	28
9.5 LEDs . . . . .	28
9.6 Potentiometers . . . . .	29
9.7 RGB LED . . . . .	30
<b>10 Como Fazer (How To's)</b>	<b>31</b>

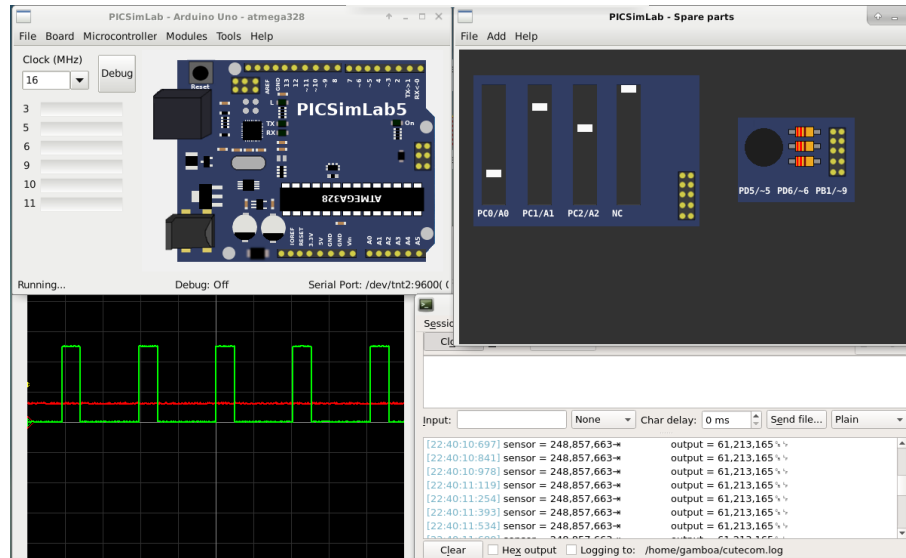
---

# Capítulo 1

## Introdução

PICSimLab significa PIC Simulator Laboratory

O PICSimLab é um emulador de tempo real de placas de desenvolvimento com integração de depuração com o MPLABX/avr-gdb. O PICSimLab suporta os microcontroladores do [picsim](#) ( PIC16F84, PIC16F628, PIC16F648, PIC16F777, PIC16F877A, PIC18F452, PIC18F4520, PIC18F4550 e PIC18F4620) e o microcontrolador do [si-mavr](#) (ATMEGA328). O PICSimLab tem integração com as IDE MPLABX/Arduino para programação dos microcontroladores das placas.



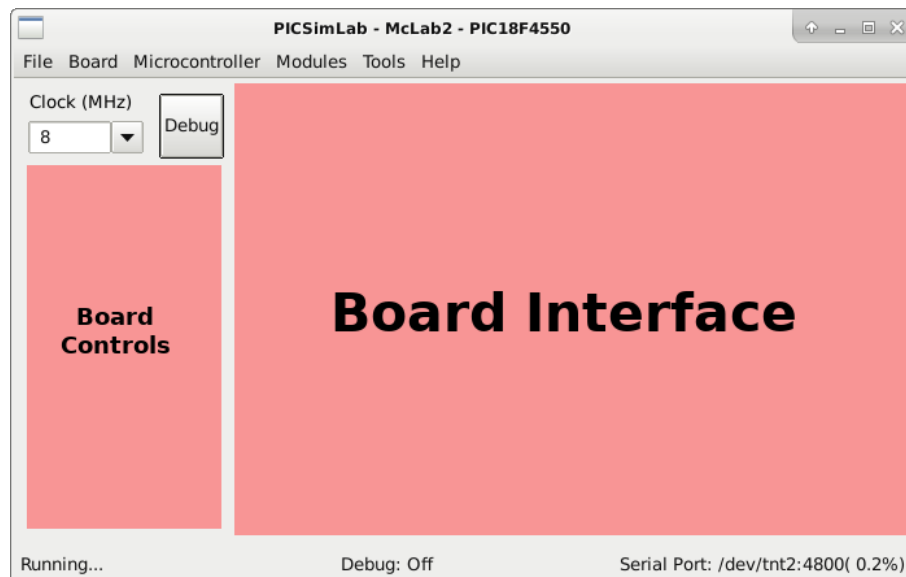
## Capítulo 2

# Interface

### 2.1 Janela Principal

A janela principal é composta de um menu, uma barra de status, um combobox de seleção de frequência, um botão liga/desliga para acionar a depuração (debug), alguns controles específicos da placa e parte de interface da placa em si.

No título da janela é mostrado o nome do simulador PICSimLab, seguido da placa e do microcontrolador em uso.



O combobox de seleção de frequência altera diretamente a velocidade de trabalho do microcontrolador, quando o label “Clock (MHz)” fica em vermelho indica que o computador não está sendo capaz de executar o programa em tempo real para o clock

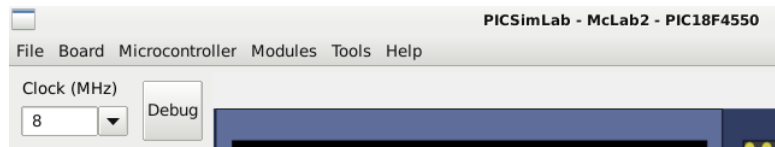


selecionado. Neste caso a simulação pode apresentar alguma diferença do esperado e a carga da CPU ser aumentada.

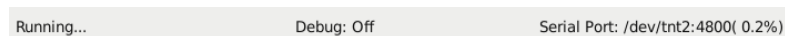
O botão liga/desliga para acionar a depuração serve para habilitar o suporte à depuração, com o suporte ativo há uma carga maior de simulação.

Os menus e suas funções são listados abaixo:

- File
  - Load Hex - Carrega arquivo .hex
  - Reload Last - Recarrega último arquivo .hex utilizado
  - Configure - Abre a janela de configuração
  - Save Workspace - Salva todas as configurações atuais do workspace em um arquivo .pzw
  - Load Workspace - Carrega as configurações salvas de uma arquivo .pzw
  - Exit
- Board
  - 1 McLab1 - Escolhe a placa 1
  - 2 K16F - Escolhe a placa 2
  - 3 McLab2 - Escolhe a placa 3
  - 4 PICGenios - Escolhe a placa 4
  - 5 Arduino - Escolhe a placa 5
- Microcontroller
  - xxxxx - Seleciona o microcontrolador a ser utilizado (depende da placa selecionada)
- Modules
  - Oscilloscope - Abre a janela do osciloscópio
  - Spare parts - Abre a janela de peças avulsas
- Tools
  - Serial Term - Abre o terminal serial [Cutecom](#)
- Help
  - Contents - Abre a janela de Ajuda
  - Examples - Carrega exemplos
  - About - Mostra mensagem de versão e autor



Na primeira parte da barra de status é mostrado o estado da simulação, na parte do meio o estado do suporte a depuração e na última parte o nome da porta serial utilizada, sua velocidade padrão e o erro em relação a velocidade real configurada no microcontrolador.



## 2.2 Comandos

Sobre a área de interface da placa é possível interagir de algumas formas:

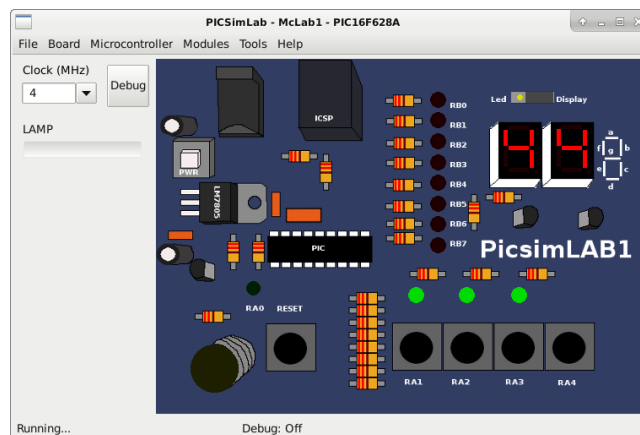
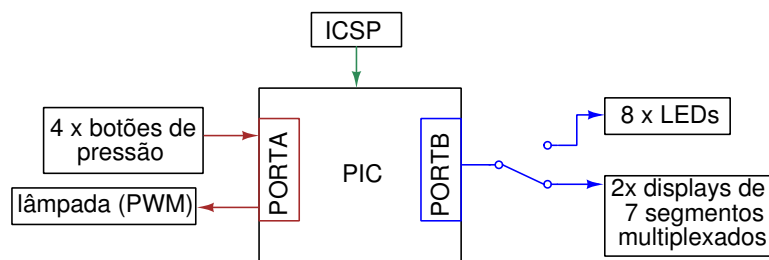
- Clique no conector ICSP para carregar um arquivo .hex.
- Clique no botão PWR para ligar/desligar o emulador.
- Os botões podem ser acionados pelo mouse ou pelas teclas 1, 2, 3 ...

## Capítulo 3

# Placas

### 3.1 Características da Placa 1

Emula a placa de desenvolvimento McLab1 da Labtools que utiliza um PIC16F84, PIC16F628 ou PIC16F648.



Esquemático da placa 1.

Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSimLab.

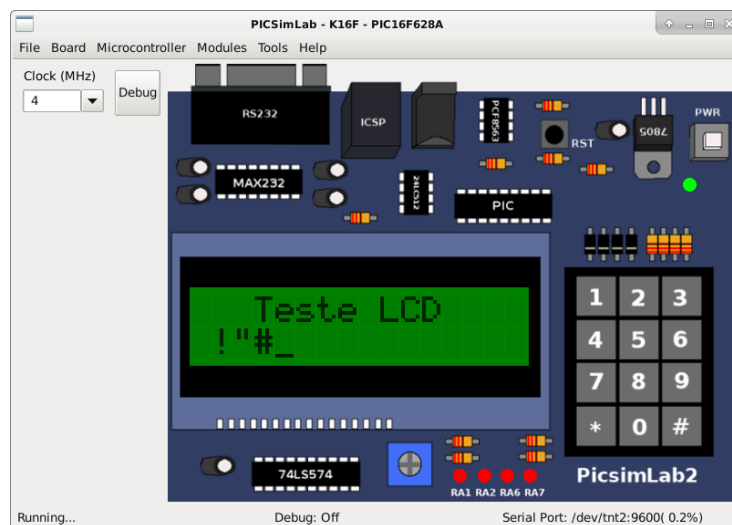
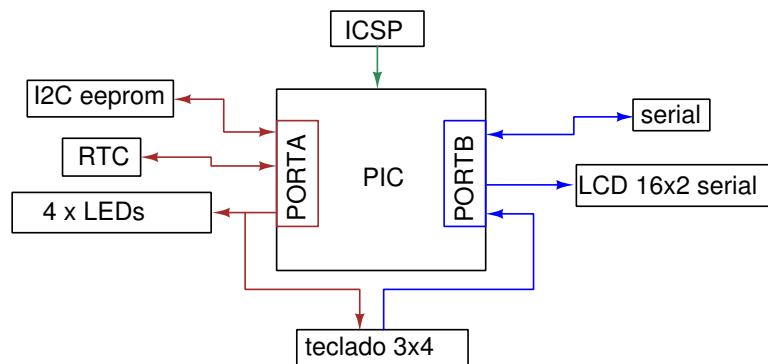
O código fonte de exemplo da placa PICSimLab1 usando o **MPLABX** e o compilador **XC8** está no diretório: [src/board\\_1/](#).

Compra do kit McLab1, manual e exemplos na área de download [www.mosaico.com.br](http://www.mosaico.com.br)

O hardware e a utilização do kit também é descrita no livro **Desbravando o PIC - Ampliado e Atualizado para PIC 16F628A** da editora **Erica** (ISBN: 978-85-7194-867-9).

## 3.2 Características da Placa 2

Emula uma placa didática desenvolvida pelo autor que utiliza um PIC16F84, PIC16F628 ou PIC16F648.



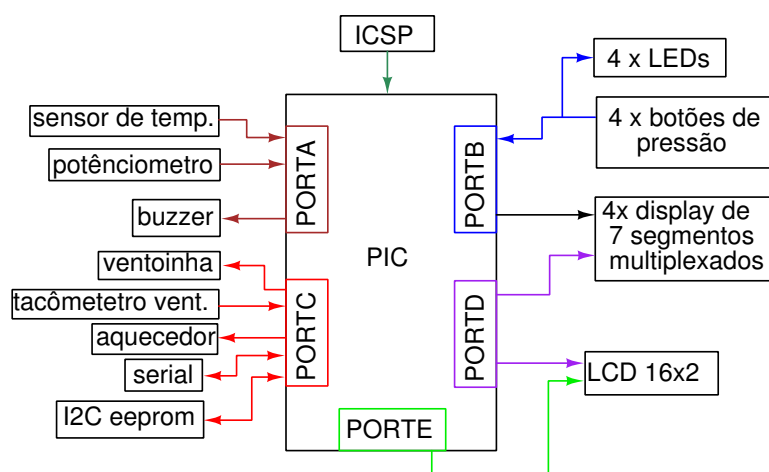
Esquemático da placa 2.

Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSimLab.

O código fonte de exemplo da placa PICSimLab2 usando o **MPLABX** e o compilador **XC8** está no diretório: [src/board\\_2/](#).

### 3.3 Características da Placa 3

Emula a placa de desenvolvimento McLab2 da Labtools que utiliza um PIC16F777, PIC16F877A, PIC18F452, PIC18F4520, PIC18F4550 ou PIC18F4620.



Esquemático da placa 3.

Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSIMLab.

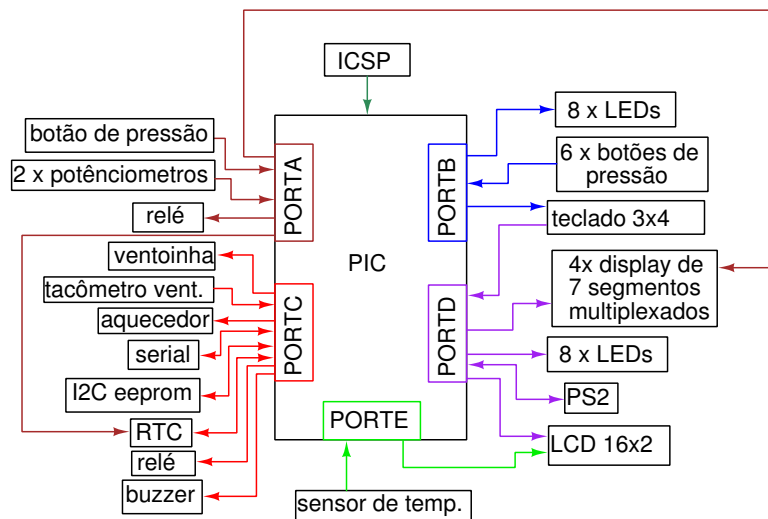
O código fonte de exemplo da placa PICSIMLab3 usando o **MPLABX** e o compilador **XC8** está no diretório: [src/board\\_3/](#).

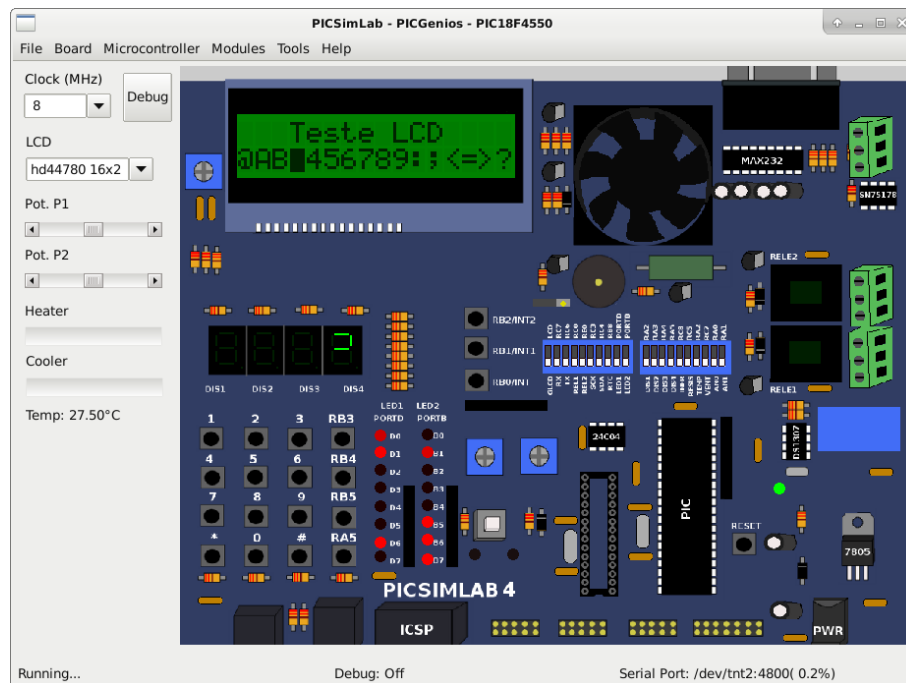
Compra do kit McLab2, manual e exemplos na área de download [www.mosaico.com.br](http://www.mosaico.com.br)

O hardware e a utilização do kit também é descrita no livro **Conectando o PIC - Recursos Avançados** da editora **Erica** (ISBN: 978-85-7194-737-5).

### 3.4 Características da Placa 4

Emula a placa de desenvolvimento PICGenios PIC18F e PIC16F Microchip da microgenios que utiliza um PIC16F777, PIC16F877A, PIC18F452, PIC18F4520, PIC18F4550 ou PIC18F4620.





Esquemático da placa 4.

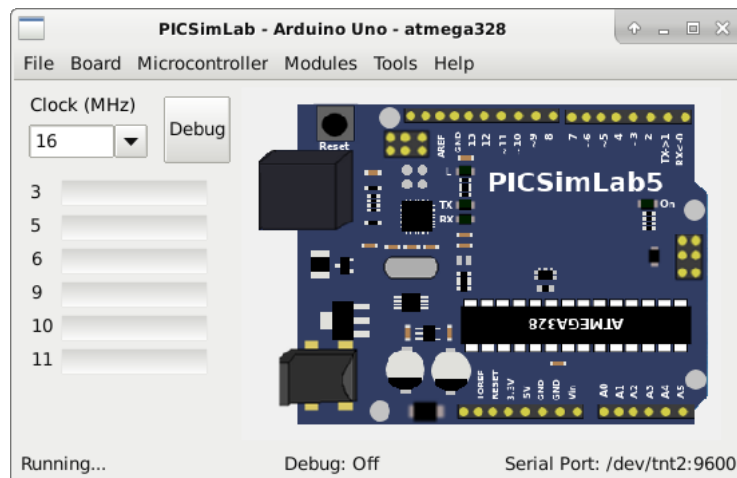
Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSIMLab.

O código fonte de exemplo da placa PICSIMLab4 usando o [MPLABX](#) e o compilador [XC8](#) está no diretório: [src/board\\_4/](#).

Compra do kit PICGenios PIC18F e PIC16F Microchip e manual em [www.microgenios.com](http://www.microgenios.com)

### 3.5 Características da Placa 5

Emula a placa de desenvolvimento Arduino Uno que utiliza um microcontrolador AT-MEGA328.



Esquemático da placa 5.

Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSimLab.

O código fonte de exemplo da placa PICSimLab5 usando a IDE [Arduino com o avr-gcc](#) está no diretório: [src/board\\_5/](#).

Mais informações sobre o Arduino em [www.arduino.cc](http://www.arduino.cc)



## Capítulo 4

# Comunicação Serial

Para utilizar o a porta serial do simulador, instale um emulador NULL-MODEM:

- Windows: com0com <http://sourceforge.net/projects/com0com/>
- Linux: tty0tty <https://github.com/lcgamboa/tty0tty>

Para comunicação o PICSimLab deve ficar conectado em uma porta do emulador NULL-MODEM e o outro aplicativo conectado na outra porta. Exemplos de configuração ligando o PICSimLab ao [Cutecom](#) para comunicação serial:

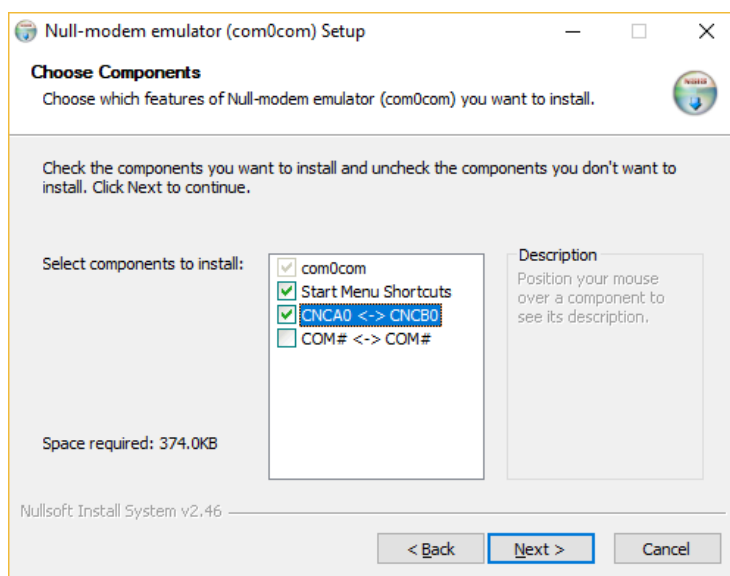
OS	porta PICSimLab	porta Cutecom	NULL-Modem prog.	Conexão
Windows	com1	com2	com0com	com1<=>com2
Linux	/dev/tnt2	/dev/tnt3	tty0tty	/dev/tnt2<=>/dev/tnt3

### 4.1 Instalação e Configuração do com0com (Windows)

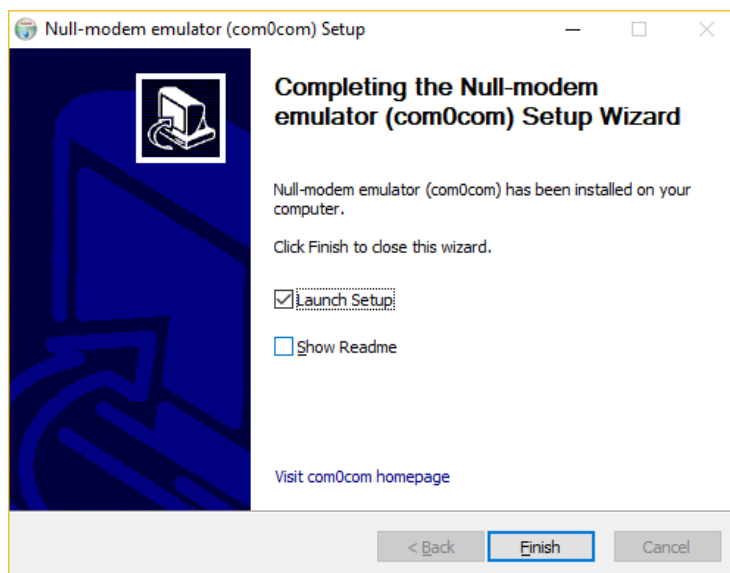
Faça o download da versão assinada do [com0com](#).

Descompacte o arquivo .zip baixado e execute o instalador específico de seu sistema operacional, x86 para windows 32 bits ou x64 para windows 64 bits.

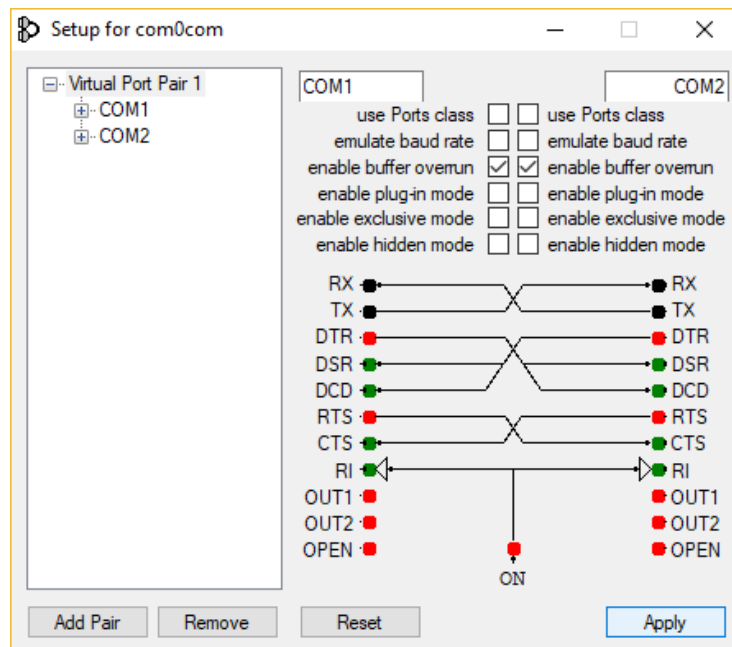
Configure a janela “choose components” como a figura abaixo:



Na última janela da configuração, marque a opção “Launch setup”:



Na janela do setup, troque os nomes das portas para COM1, COM2, COM3.... Marque apenas a opção “enable buffer overrun” nas duas portas, clique no botão “Apply” e feche o setup. Na configuração mostrada na figura abaixo, as portas COM1 e COM2 formam uma conexão NULL-MODEM, onde uma porta deve ser utilizada pelo PICSimLab e outra pela aplicação com comunicação serial.



## 4.2 Instalação e Configuração do tty0tty (Linux)

Faça o download do [tty0tty](https://github.com/0x00b00b/tty0tty). Descompacte a pasta baixada.

Abra um terminal e entre na pasta `tty0tty/module` e digite os comandos na sequência:

```
sudo apt-get update
sudo apt-get -y upgrade
sudo apt-get -y install gcc make linux-headers-`uname -r`
make
sudo make install
```

O usuário deve estar no grupo **dialout** para poder acessar as portas. Para adicionar seu usuário ao grupo **dialout** use o comando:

```
sudo usermod -a -G dialout your_user_name
```

depois disso é necessário fazer logout e login para as permissões do grupo tenham efeito.

Depois de instalado, o módulo cria 8 portas interconectadas da seguinte forma:

```
/dev/tnt0 <=> /dev/tnt1
/dev/tnt2 <=> /dev/tnt3
/dev/tnt4 <=> /dev/tnt5
/dev/tnt6 <=> /dev/tnt7
```

a conexão entre cada par é da forma:

```
TX    ->  RX
RX    <-   TX
RTS   ->  CTS
CTS   <-   RTS
DSR   <-   DTR
CD    <-   DTR
DTR   ->  DSR
DTR   ->  CD
```

Qualquer par de portas formam uma conexão NULL-MODEM, onde uma porta deve ser utilizada pelo PICSimLab e outra pela aplicação com comunicação serial.

## Capítulo 5

# Depuração Integrada com o MPLABX (PIC)

Para utilizar o IDE [MPLABX](#) para depurar e programar o PicsimLab, basta instalar o plugin [com-picsim-picsimlab.nbm](#) no MPLABX.

O plugin se conecta ao Picsimlab através de um socket TCP na porta 1234, permita o acesso no firewall.

[Tutorial: Como usar o MPLABX para programar e depurar o PICsimLab \(Inglês\)](#)

## Capítulo 6

# Integração com a IDE do Arduino (ATMEGA)

Para utilização integrada com a IDE do Arduino, basta fazer a configuração da porta serial como explicado na seção 4 e carregar o bootloader do Arduino. O bootloader pode ser carregado pelo menu “Help->Examples” escolhendo o arquivo ATmegaBOOT\_168\_atmega328.pzw.

No windows, considerando o com0com fazendo uma conexão NULL-MODEM entre a porta COM1 e COM2, basta conectar o PICSimLab na porta COM1 e a IDE Arduino na porta COM2 ou vice-versa.

No Linux o funcionamento é o mesmo, mas utilizando por exemplo as portas /dev/tnt2 e /dev/tnt3.

No Linux para as portas virtuais serem detectadas no Arduino é necessário substituir a biblioteca lib/liblistSerials.j.so do Arduino por uma com suporte a detecção das portas do tty0tty, que pode se baixada no link [listSerialC com suporte ao tty0tty](#).

## Capítulo 7

# Depuração com o avr-gdb (ATMEGA)

Com o suporte a depuração habilitado é possível utilizar o avr-gdb para depurar o código utilizado no simulador. Utilize o avr-gdb com o arquivo .elf como parâmetro:

```
avr-gdb arquivo_compilado.elf
```

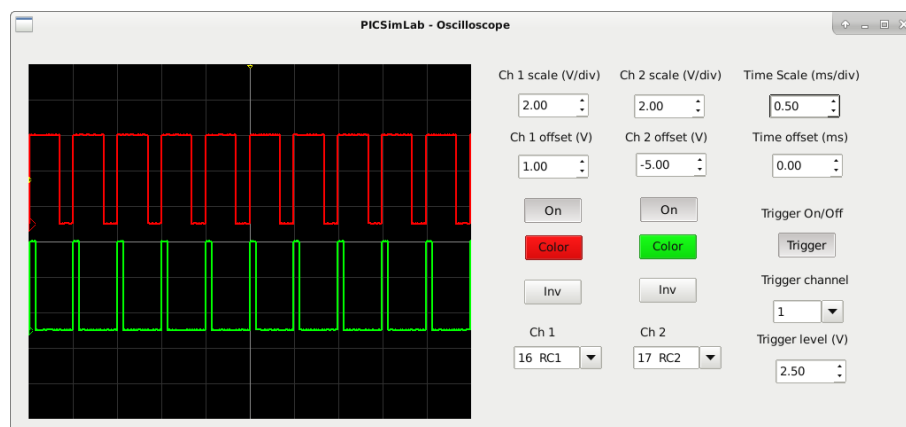
e o comando abaixo para se conectar:

```
target remote localhost:1234
```

## Capítulo 8

# Osciloscópio

O PICSimLab possui um osciloscópio básico de dois canais que pode ser utilizado para ver o sinal em qualquer pino do microcontrolador. O osciloscópio pode ser acessado pelo menu “Modules->Oscilloscope”.





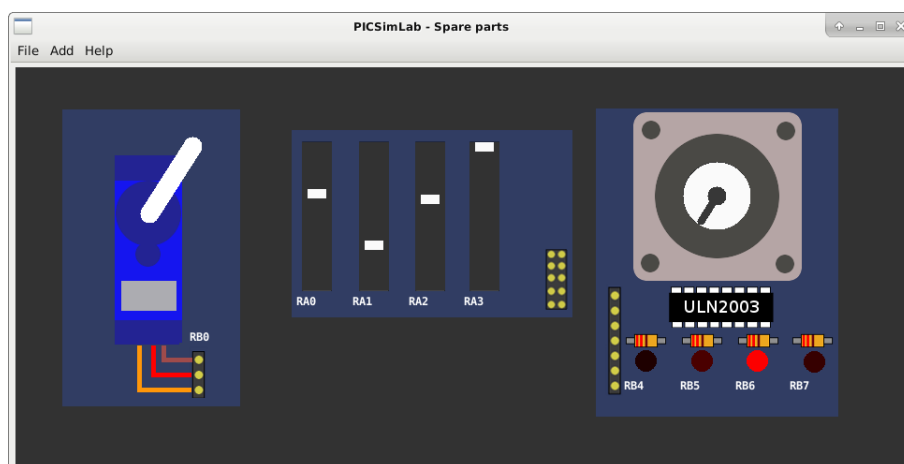
## Capítulo 9

# Partes Avulsas

O PICsimLab possui uma janela que permite a ligação de partes avulsas ao microcontrolador, ela pode ser acessada pelo menu “Modules->Spare parts”.

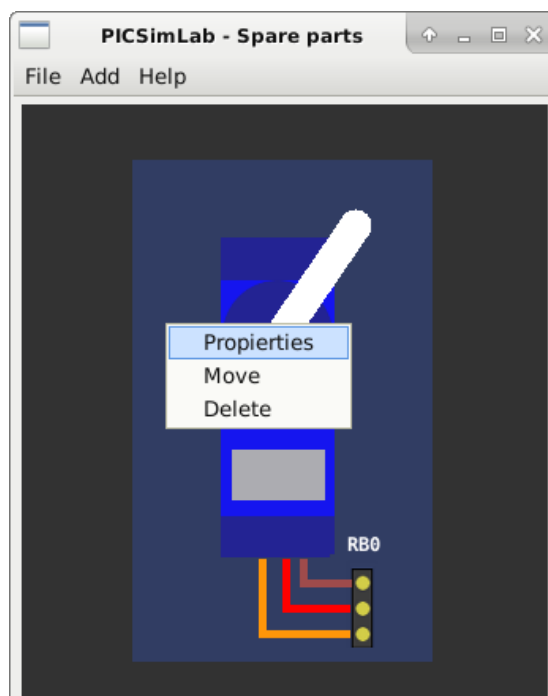
A janela principal possui o menu com as seguintes funções:

- File
  - Save configuration - Salva as configurações atuais das partes avulsas em um arquivo .pcf
  - Load configuration - Carrega as configurações de uma arquivo .pcf
- Add
  - Servo Motor - Adiciona um servo motor
  - Step Motor - Adiciona um motor de passo
  - Push Buttons - Adiciona 8 botões de pressionar
  - Switchs - Adiciona 8 chaves
  - LEDs - Adiciona 8 LEDs vermelhos
  - Potentiometers - Adiciona 4 potenciômetros
  - RGB LED - Adiciona 1 LED RGB
- Help
  - Contents - Abre a janela de Ajuda
  - About - Mostra mensagem de versão e autor



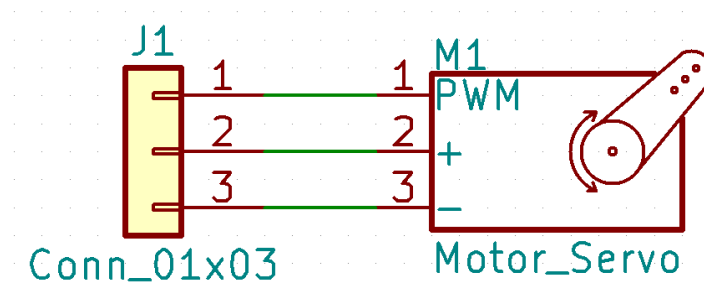
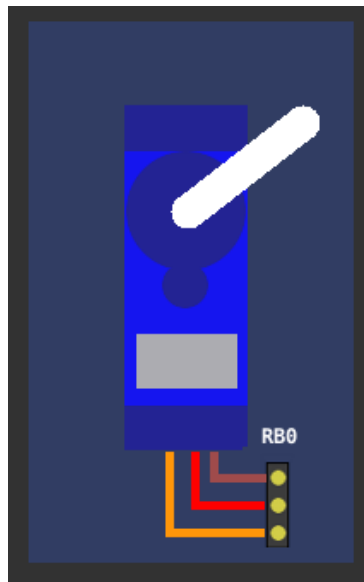
Depois de adicionado a parte, com um clique com o botão direito do mouse é possível acessar o menu das opções da parte com as opções:

- Properties - Abre a janela de configuração das conexões
- Move - Desbloqueia a parte para movimentação
- Delete - Remove a parte



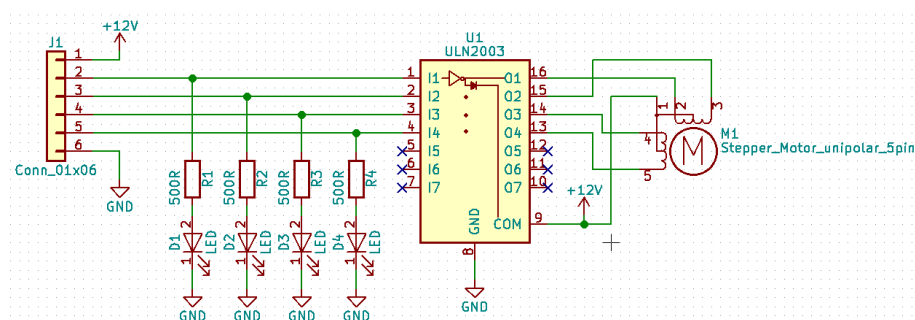
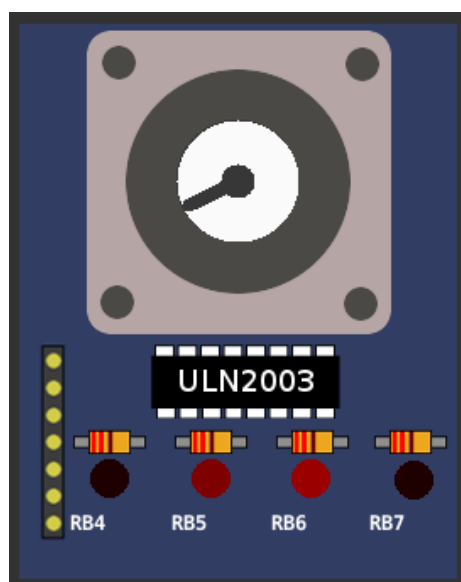
## 9.1 Servo Motor

O servo motor é um componente que deve ser acionado com um pulso de largura variável de 1ms a 2ms a cada 20 ms. Um pulso de 1ms posiciona o servo a  $-90^\circ$ , um de 1,5ms a  $0^\circ$  e um de 2ms a  $90^\circ$ .



## 9.2 Step Motor

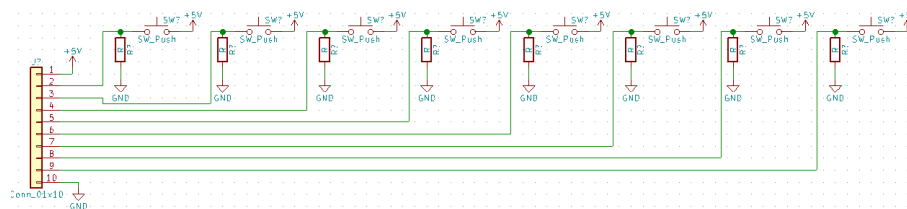
O motor de passo é um componente com 4 bobinas que devem ser acionadas na ordem correta para fazer o deslocamento do rotor. Cada passo do motor é de  $1.8^\circ$ .



### 9.3 Push Buttons

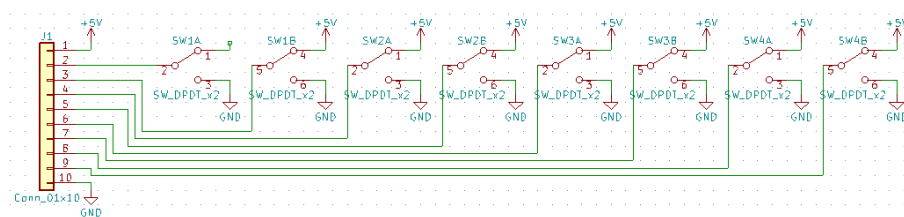
Esta parte é formada por 8 botões de pressão. Quando pressionado a saída vai para nível lógico “1”.





## 9.4 Switchs

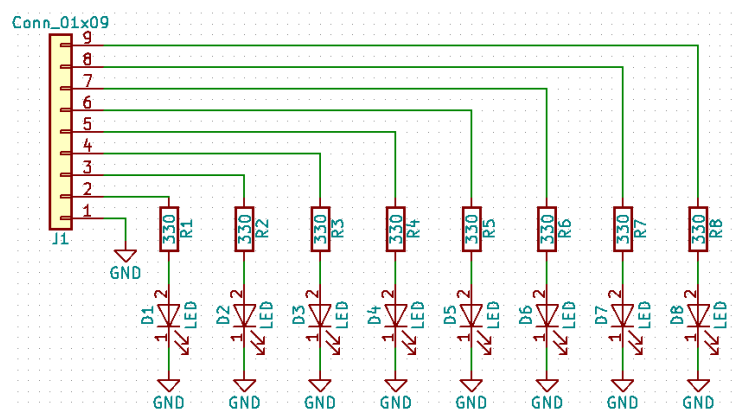
Esta parte é formada por 8 chaves com posição ligado ou desligado (0 ou 1).



## 9.5 LEDs

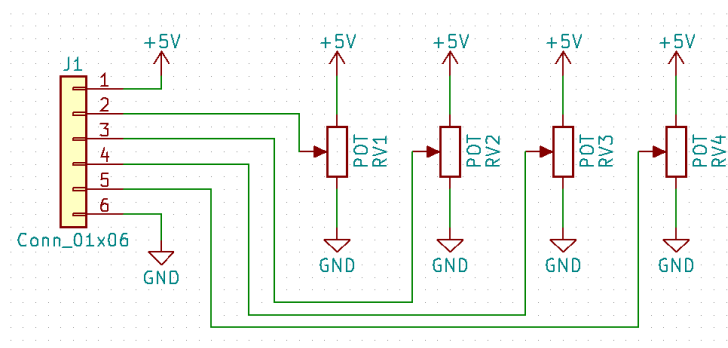
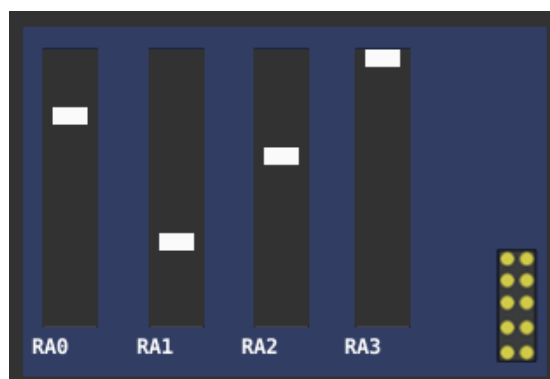
Essa parte é uma barra de 8 LEDs vermelhos independentes.





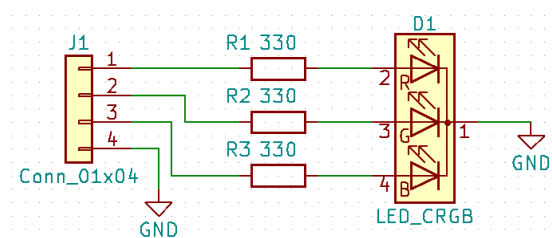
## 9.6 Potentiometers

Essa parte é formada por 4 potenciômetros ligados entre 0 e 5 Volts, a saída está ligada ao cursor e varia dentro dessa faixa de tensão.



## 9.7 RGB LED

Essa parte é formada por um LED RGB de 4 terminais. Cada cor pode ser acionada de forma independente. Utilizando PWM é possível gerar varias cores através da combinação da 3 cores primárias.



## Capítulo 10

# Como Fazer (How To's)

- [How to use MPLABX to program and debug PICsimLab.](#)
- [\(Deprecated\) How to Compile PICsimLab and Create New Boards.](#)



**Part II**

**English Manual**

# Table of Contents

---

<b>1</b>	<b>Introduction</b>	<b>35</b>
<b>2</b>	<b>Interface</b>	<b>36</b>
2.1	Main Window . . . . .	36
2.2	Commands . . . . .	38
<b>3</b>	<b>Boards</b>	<b>39</b>
3.1	Features of Board 1 . . . . .	39
3.2	Features of Board 2 . . . . .	40
3.3	Features of Board 3 . . . . .	41
3.4	Features of Board 4 . . . . .	42
3.5	Features of Board 5 . . . . .	43
<b>4</b>	<b>Serial Communication</b>	<b>45</b>
4.1	Com0com Installation and Configuration(Windows) . . . . .	45
4.2	tty0tty Installation and Configuration (Linux) . . . . .	47
<b>5</b>	<b>MPLABX Integrated Debug (PIC)</b>	<b>49</b>
<b>6</b>	<b>Arduino IDE Integration (ATMEGA)</b>	<b>50</b>
<b>7</b>	<b>avr-gdb Debug (ATMEGA)</b>	<b>51</b>
<b>8</b>	<b>Oscilloscope</b>	<b>52</b>

<i>TABLE OF CONTENTS</i>	34
<b>9 Spare Parts</b>	<b>53</b>
9.1 Servo Motor . . . . .	55
9.2 Step Motor . . . . .	55
9.3 Push Buttons . . . . .	56
9.4 Switchs . . . . .	57
9.5 LEDs . . . . .	57
9.6 Potentiometers . . . . .	58
9.7 RGB LED . . . . .	59
<b>10 How To's</b>	<b>60</b>

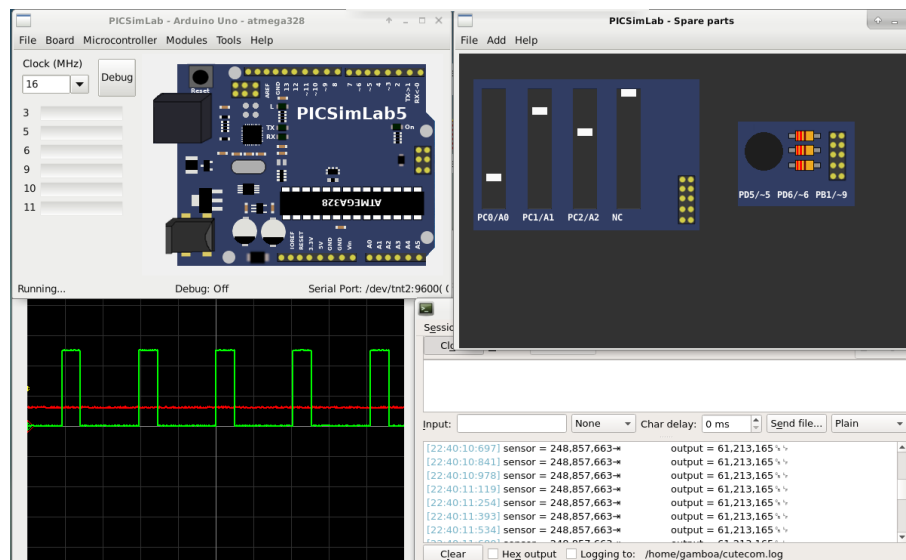
---

# Chapter 1

## Introduction

PICSimLab means PIC Simulator Laboratory

PICSimLab is a realtime emulator of development boards with integrated MPLABX/avr-gdb debugger. PICSimLab supports [picsim](#) microcontrollers (PIC16F84, PIC16F628, PIC16F648, PIC16F777, PIC16F877A, PIC18F452, PIC18F4520, PIC18F4550 and PIC18F4620) and [simavr](#) microcontroller (ATMEGA328). PICSimLab have integration with MPLABX/Arduino IDE for programming the boards microcontrollers.



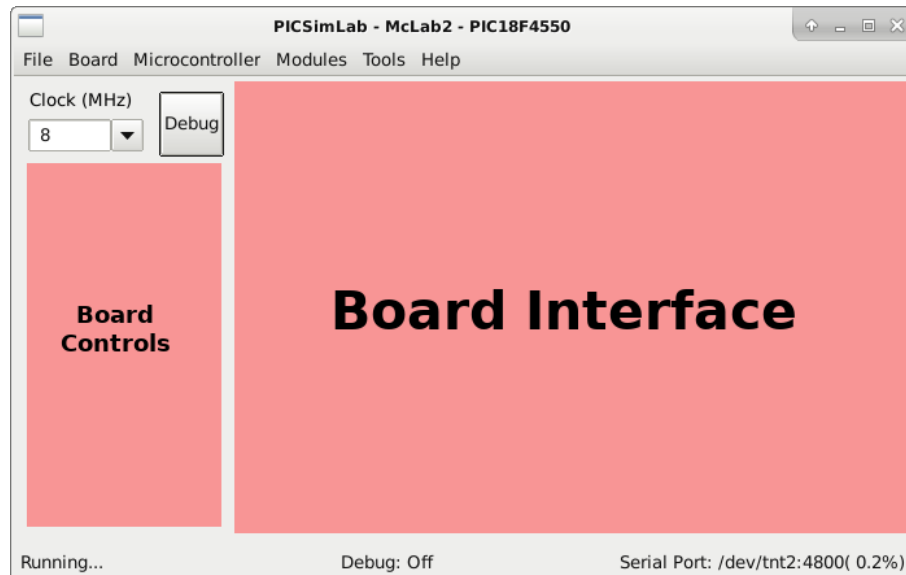
## Chapter 2

# Interface

### 2.1 Main Window

The main window consists of a menu, a status bar, a frequency selection combobox, an on/off button to trigger debugging, some board-specific controls and the part of the board interface itself.

In the title of the window is shown the name of the simulator PICSimLab, followed by the board and the microcontroller in use.



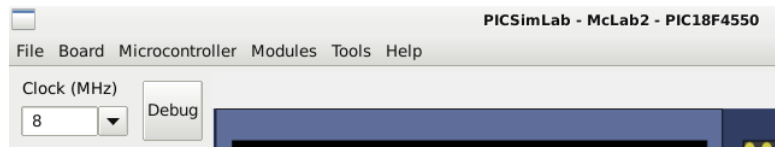
The frequency selection combobox directly changes the working speed of the microcontroller, when the “Clock (MHz)” label goes red indicates that the computer is not being able to run the program in real time for the selected clock. In this case

the simulation may present some difference than expected and the CPU load will be increased.

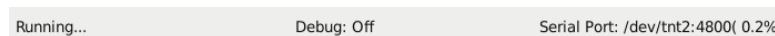
The on/off button to enable debugging is used to enable debugging support, with the active support there is a higher simulation load.

The menus and their functions are listed below:

- File
  - Load Hex - Load .hex files
  - Reload Last - Reload the last used .hex file
  - Configure - Open the configuration windows
  - Save Workspace - Saves all current workspace settings to a .pzw file
  - Load Workspace - Loads saved settings from a .pzw file
  - Exit
- Board
  - 1 McLab1 - choose board 1
  - 2 K16F - Choose board 2
  - 3 McLab2 - Choose board 3
  - 4 PICGenios - Choose board 4
  - 5 Arduino - Choose board 5
- Microcontroller
  - xxxxx - Selects the microcontroller to be used (depends on the selected board)
- Modules
  - Oscilloscope - Open the oscilloscope window
  - Spare parts - Open the spare parts window
- Tools
  - Serial Term - Open the serial terminal Cütecom
- Help
  - Contents - Open the Help window
  - Examples - Load the examples
  - About - Show message about author and version



The first part of the status bar shows the state of the simulation, in the middle part the status of the debug support and in the last part the name of the serial port used, its default speed and the error in relation to the real speed configured in the microcontroller.



## 2.2 Commands

On the interface area of the board it is possible to interact in some ways:

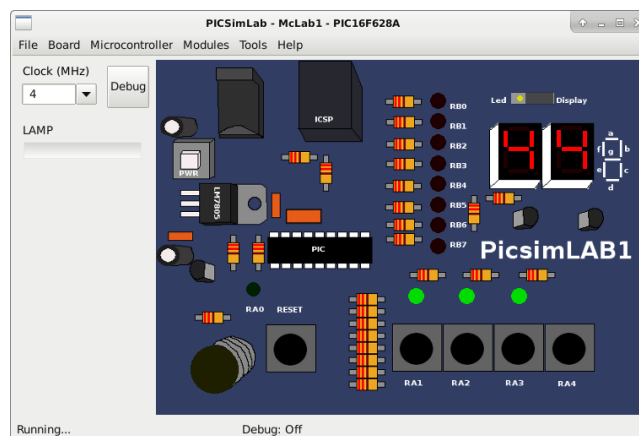
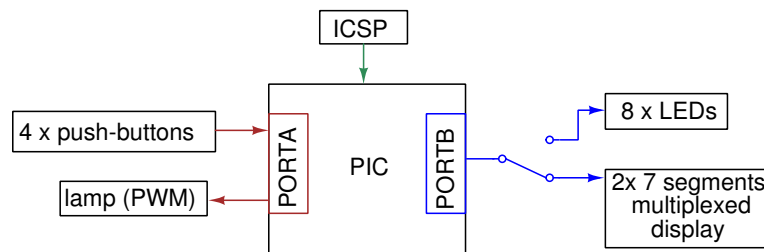
- Click in ICSP connector to load an .hex file.
- Click in PWR button to ON/OFF the emulator..
- The buttons can be activated through mouse or keys 1, 2, 3 e 4.

## Chapter 3

# Boards

### 3.1 Features of Board 1

It emulates the Labtools development board McLab1 that uses one PIC16F84, PIC16F628 or PIC16F648.



[Board 1 schematics.](#)



The code examples can be loaded in PICSimLab menu **Help->Examples**.

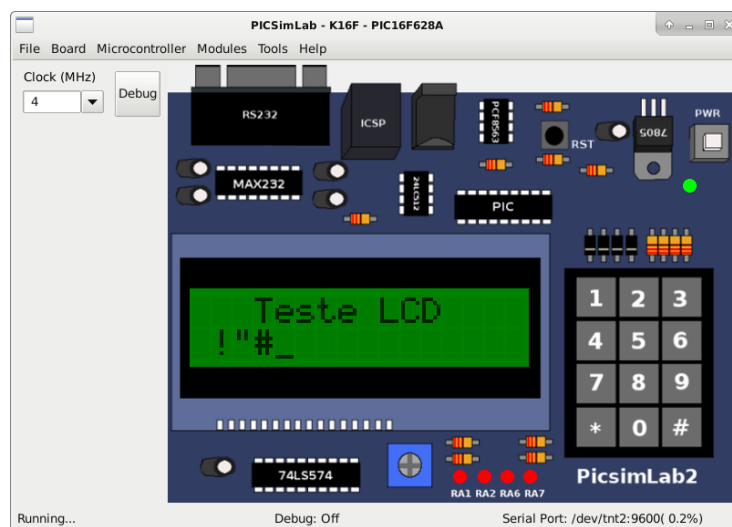
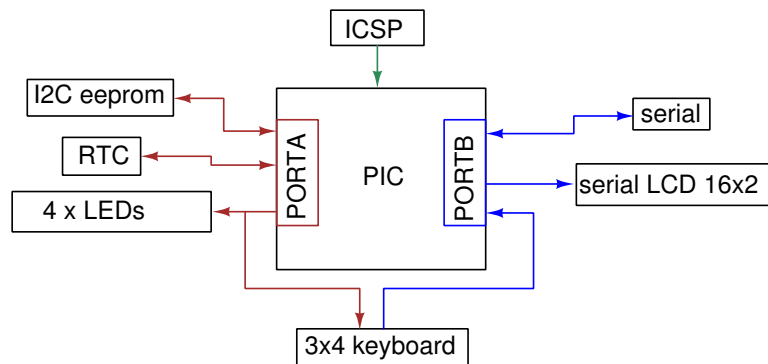
The source code of PICSimLab1 example using **MPLABX** and **XC8** compiler are in the folder: [src/board\\_1/](#).

To buy McLab1 kit, download manual and examples you can go to [www.mosaico.com.br](http://www.mosaico.com.br)

The hardware and the use of kit are described in the book **Desbravando o PIC - Ampliado e Atualizado para PIC 16F628A** of [Erica publisher](#) (ISBN: 978-85-7194-867-9).

## 3.2 Features of Board 2

It emulates an didactic board developed by author that uses one PIC16F84, PIC16F628 or PIC16F648.



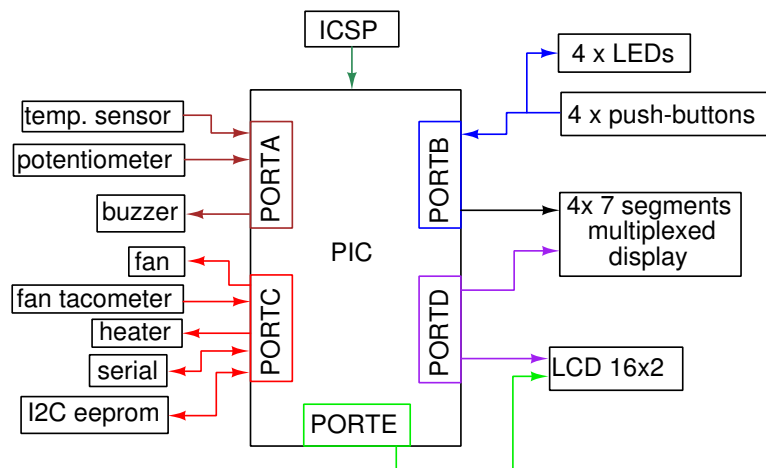
[Board 2 schematics.](#)

The code examples can be loaded in PICSIMLab menu **Help->Examples**.

The source code of PICSIMLab2 example using **MPLABX** and **XC8** compiler are in the folder: [src/board\\_2/](#).

### 3.3 Features of Board 3

It emulates the Labtools development board McLab2 that uses one PIC16F777, PIC16F877A, PIC18F452, PIC18F4520, PIC18F4550 or PIC18F4620.



[Board 3 schematics.](#)

The code examples can be loaded in PICSIMLab menu **Help->Examples**.

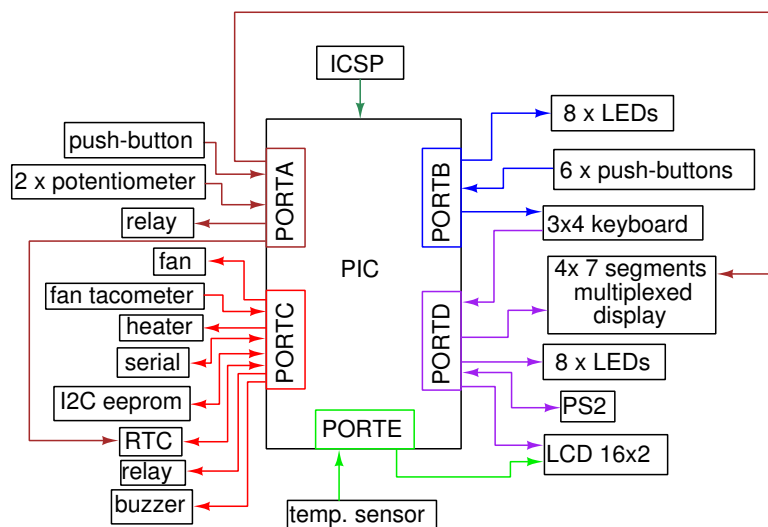
The source code of PICSimLab3 example using [MPLABX](#) and [XC8](#) compiler are in the folder: [src/board\\_3/](#).

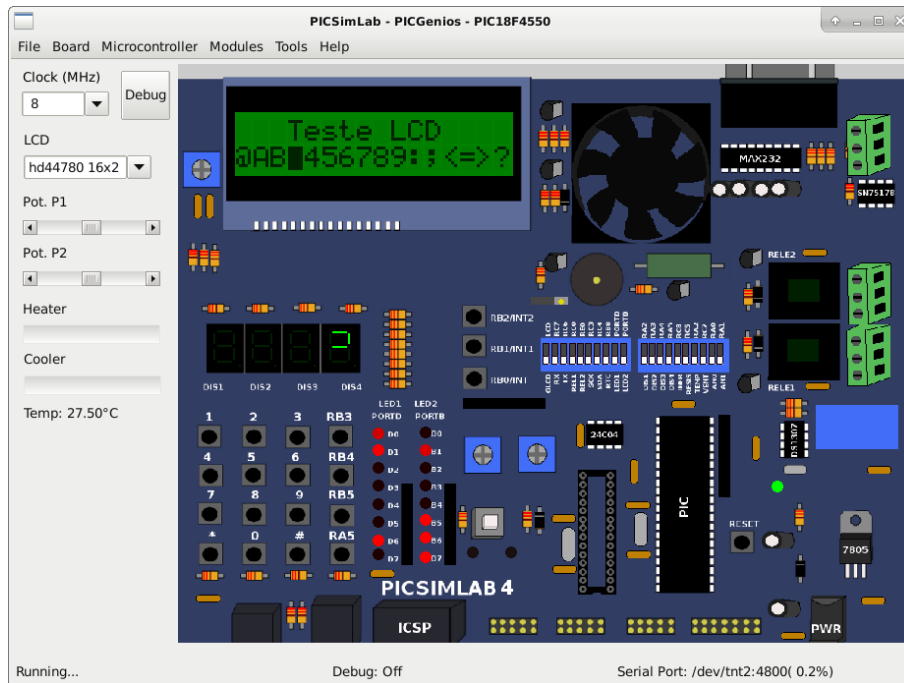
To buy McLab2 kit, download manual and examples you can go to [www.mosaico.com.br](http://www.mosaico.com.br)

The hardware and the use of kit are described in the book **Conectando o PIC - Recursos Avançados** of [Erica publisher](#) (ISBN: 978-85-7194-737-5).

### 3.4 Features of Board 4

It emulates the microgenius development board PICGenios PIC18F e PIC16F Microchip that uses one PIC16F777, PIC16F877A, PIC18F452, PIC18F4520, PIC18F4550 or PIC18F4620.





Board 4 schematics.

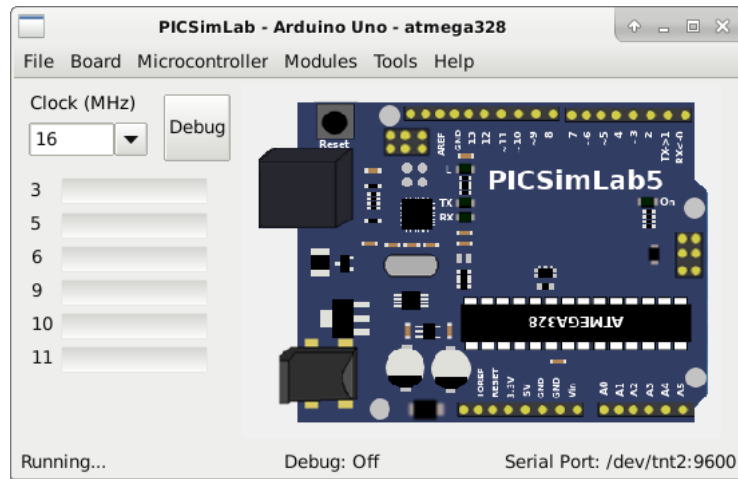
The code examples can be loaded in PICSIMLAB menu **Help->Examples**.

The source code of PICSIMLAB4 example using **MPLABX** and **XC8** compiler are in the folder: [src/board\\_4/](#).

To buy PICGenios PIC18F and PIC16F Microchip kit and download manual [www.microgenios.com](http://www.microgenios.com).

### 3.5 Features of Board 5

It emulates the Arduino Uno development board that uses one ATMEGA328 micro-controller.



[Board 5 schematics.](#)

The code examples can be loaded in PICSimLab menu **Help->Examples**.

The source code of PICSimLab5 examples using the [Arduino IDE with avr-gcc](#) are in the folder: [src/board\\_5/](#).

More informations about the Arduino in [www.arduino.cc](http://www.arduino.cc)

## Chapter 4

# Serial Communication

To use the simulator serial port, install a NULL-MODEM emulator:

- Windows: com0com <http://sourceforge.net/projects/com0com/>
- Linux: tty0tty <https://github.com/lcgamboa/tty0tty>

For communication the PICSimLab should be connected in one port of the NULL-MODEM emulator and the other application connected in the other port. Configuration examples linking PICSimLab to [Cutecom](#) for serial communication:

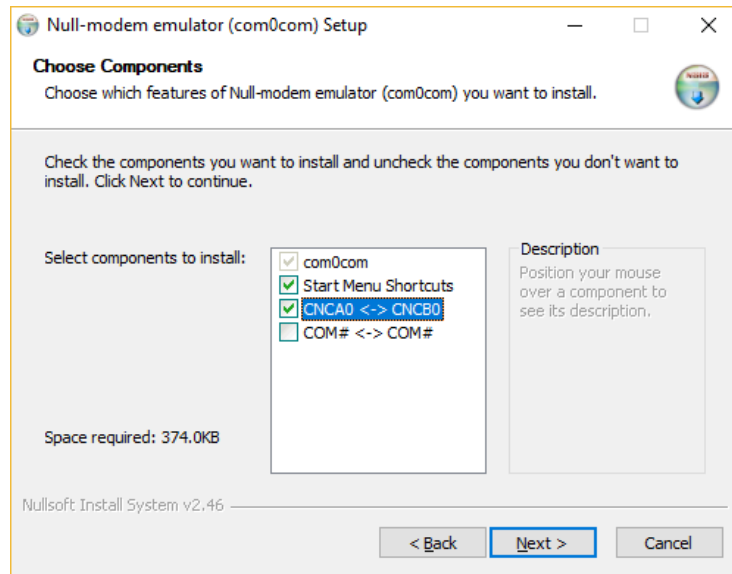
OS	PicsimLab port	Cutecom port	NULL-Modem prog.	Connection
Windows	com1	com2	com0com	com1<=>com2
Linux	/dev/tnt2	/dev/tnt3	tty0tty	/dev/tnt2<=>/dev/tnt3

### 4.1 Com0com Installation and Configuration(Windows)

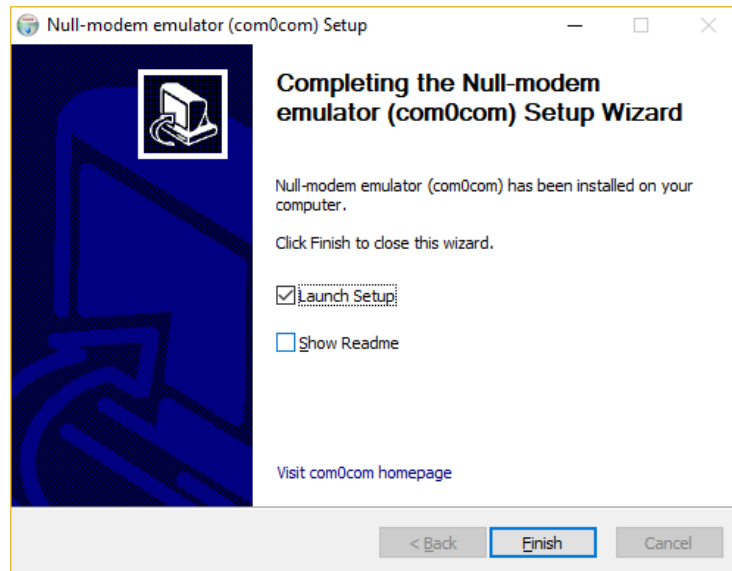
Download the signed version of [com0com](#).

Unzip the downloaded .zip file and run the specific installer of your operating system, x86 for windows 32-bit or x64 for windows 64-bit.

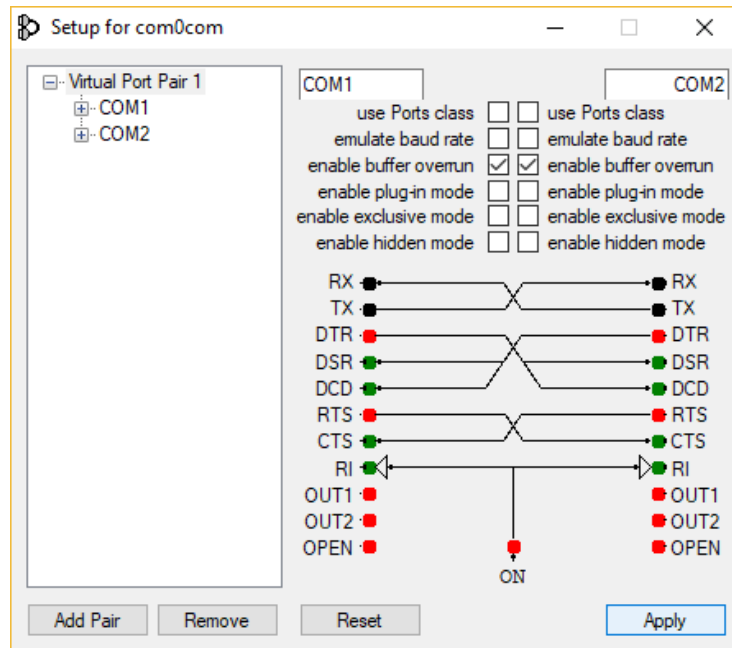
Configure the “choose components” window as the figure below:



In the last configuration window, check the “Launch setup” option:



In the setup window, change the port names to COM1, COM2, COM3 .... Just check the “enable buffer overrun” option on the two ports, click in the “Apply” button and close the setup. In the configuration shown in the figure below, the COM1 and COM2 ports form a NULL-MODEM connection, where one port must be used by the PICSimLab and another by the application with serial communication.



## 4.2 tty0tty Installation and Configuration (Linux)

Download the [href https://github.com/lcgamboa/tty0tty/archive/master.zip](https://github.com/lcgamboa/tty0tty/archive/master.zip) tty0tty. Unzip the downloaded folder.

Open a terminal and enter in the `tty0tty/module/` folder and enter the following commands:

```
sudo apt-get update
sudo apt-get -y upgrade
sudo apt-get -y install gcc make linux-headers-`uname -r`
make
sudo make install
```

The user must be in the **dialout** group to access the ports. To add your user to **dialout** group use the command:

```
sudo usermod -a -G dialout your_user_name
```

after this is necessary logout and login to group permissions take effect.

Once installed, the module creates 8 interconnected ports as follows:

```
/dev/tnt0 <=> /dev/tnt1
/dev/tnt2 <=> /dev/tnt3
/dev/tnt4 <=> /dev/tnt5
/dev/tnt6 <=> /dev/tnt7
```



the connection between each pair is of the form:

```
TX    ->  RX
RX    <-   TX
RTS   ->  CTS
CTS   <-   RTS
DSR   <-   DTR
CD    <-   DTR
DTR   ->  DSR
DTR   ->  CD
```

Any pair of ports form a NULL-MODEM connection, where one port must be used by the PICSimLab and another by the application with serial communication.

## Chapter 5

# MPLABX Integrated Debug (PIC)

To use the [MPLABX](#) IDE for debug and program the PicsimLab, install the plugin [com-picsim-picsimlab.nbm](#) in MPLABX.

The plugin connect to Picsimlab through a TCP socket using port 1234, and you have to allow the access in the firewall.

[Tutorial: how to use MPLABX to program and debug PICsimLab.](#)

## Chapter 6

# Arduino IDE Integration (ATMEGA)

For integrated use with the Arduino IDE, simply configure the serial port as explained in the section 4 and load the Arduino bootloader. The bootloader can be loaded from the “Help->Examples” menu by choosing the file ATmegaBOOT\_168\_atmega328.pzw.

In Windows, considering com0com making a NULL-MODEM connection between COM1 and COM2, simply connect the PICSimLab on the COM1 port and the Arduino IDE on the COM2 port or vice versa.

On Linux the operation is the same, but using for example the ports /dev/tnt2 and /dev/tnt3.

In Linux for the virtual ports to be detected in Arduino it is necessary to replace the library lib/liblistSerials.j.so of the Arduino with a version which support the detection of tty0tty ports, that can be downloaded in the link [listSerialC with tty0tty support](#).

## Chapter 7

# avr-gdb Debug (ATMEGA)

With debug support enabled you can use avr-gdb to debug the code used in the simulator. Use avr-gdb with the .elf file as the parameter:

```
avr-gdb compiled_file.elf
```

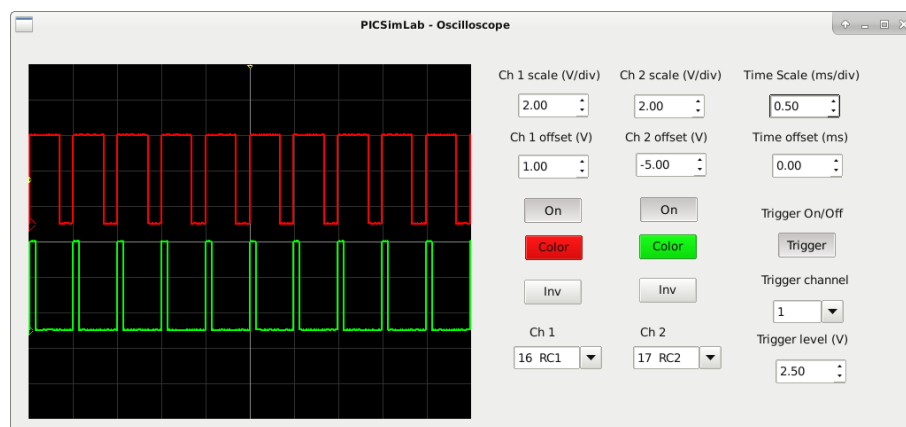
and the command below to connect:

```
target remote localhost:1234
```

## Chapter 8

# Oscilloscope

The PICSimLab has a basic two-channel oscilloscope that can be used to view the signal on any pin of the microcontroller. The oscilloscope can be accessed through the “Modules->Oscilloscope” menu.



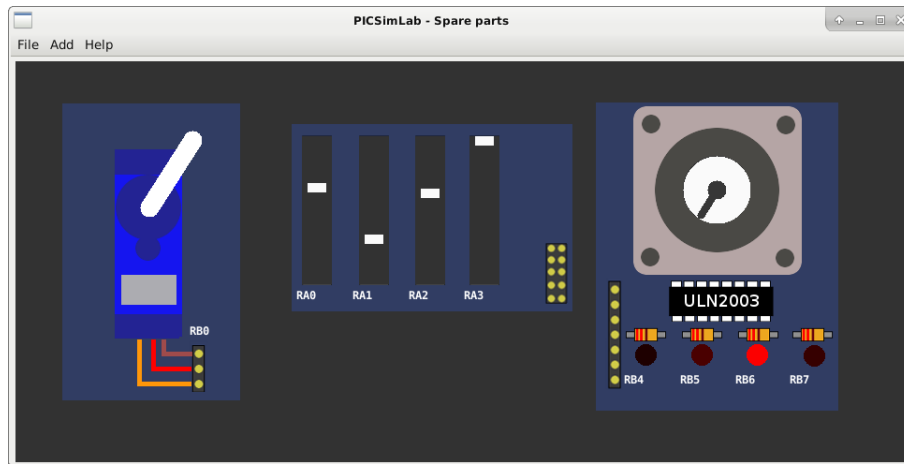
## Chapter 9

# Spare Parts

The PICSimLab has a window that allows the connection of spare parts to the micro-controller, it can be accessed through the menu “ Modules-> Spare parts ”.

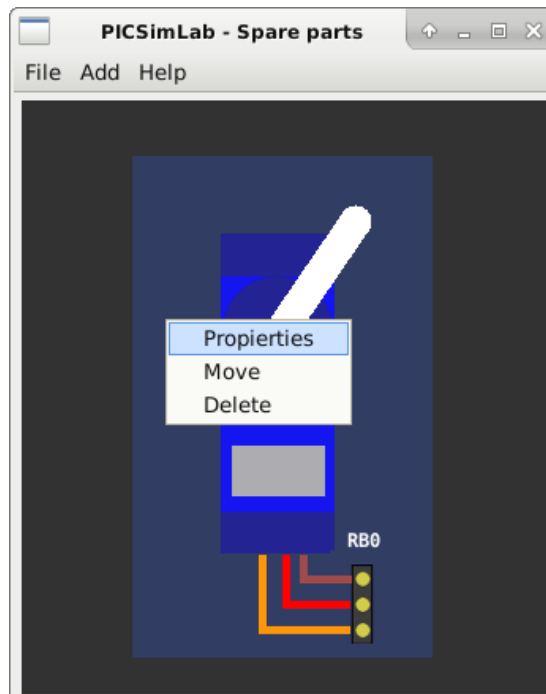
The main window has the menu with the following functions:

- File
  - Save configuration - Saves the current settings of the spare parts to an .pcf file
  - Load configuration - Loads the settings of an .pcf file
- Add
  - Servo Motor - Adds a servo motor
  - Step Motor - Adds a step motor
  - Push Buttons - Adds 8 push buttons
  - Switchs - Adds 8 switchs
  - LEDs - Adds 8 red LEDs
  - Potentiometers - Adds 4 potentiometers
  - RGB LED - Adds 1 RGB LED
- Help
  - Contents - Open Help window
  - About - Show message about author and version



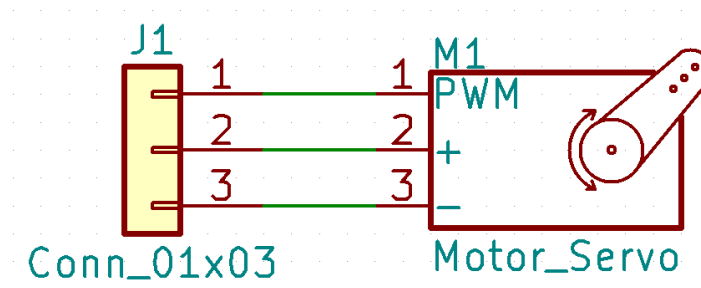
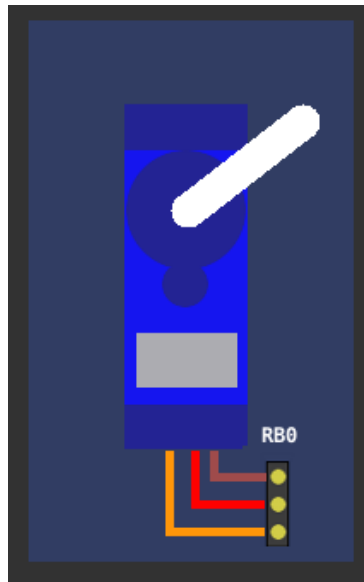
After adding the part, with a right click of the mouse you can access the options menu of the part with the options:

- Properties - Opens the connection settings window
- Move - Unlocks the part to move
- Delete - Remove part



## 9.1 Servo Motor

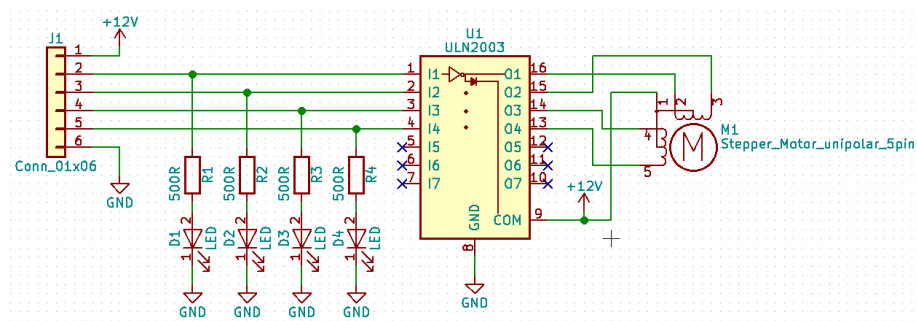
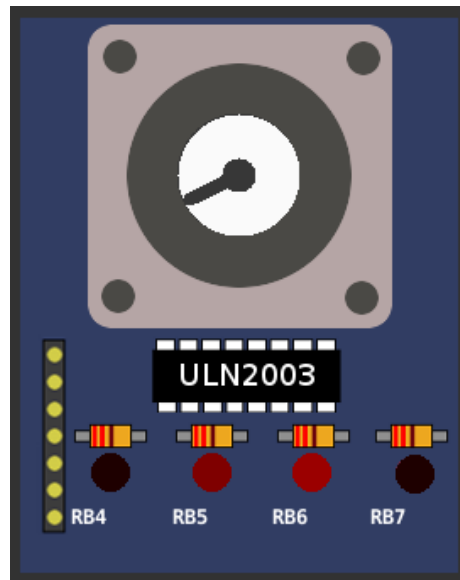
The servo motor is a component that must be activated with a pulse of variable width from 1ms to 2ms every 20 ms. A pulse of 1ms positions the servo at  $-90^\circ$ , one from 1.5ms to  $0^\circ$  and one from 2ms to  $90^\circ$ .



## 9.2 Step Motor

The stepper motor is a component with 4 coils that must be driven in the correct order to rotate the rotor. Each step of the motor is  $1.8^\circ$ .

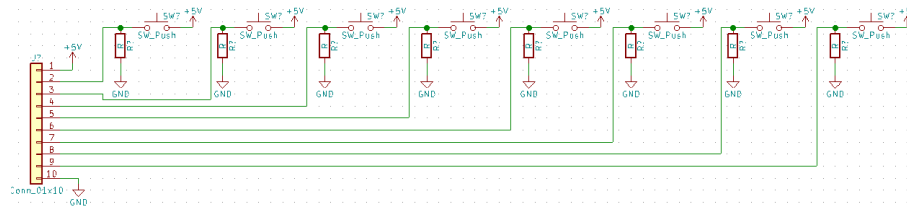




### 9.3 Push Buttons

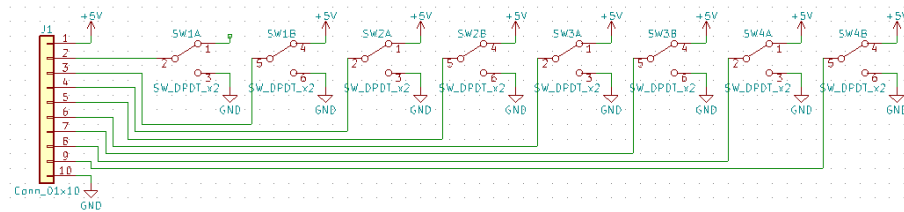
This part consists of 8 push buttons. When pressed the output goes to logic level “1”.





## 9.4 Switchs

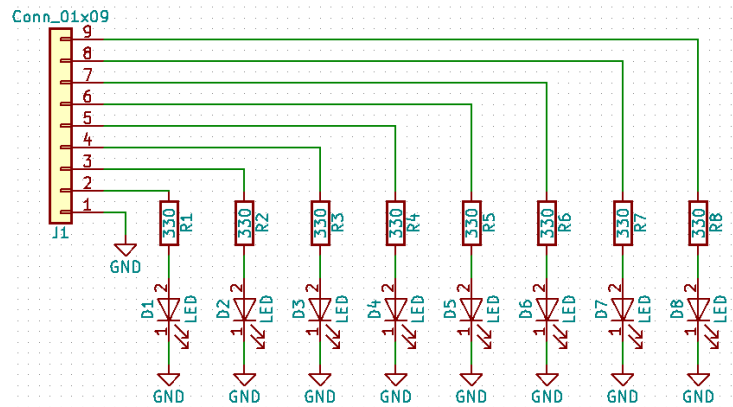
This part consists of 8 keys with on or off position (0 or 1).



## 9.5 LEDs

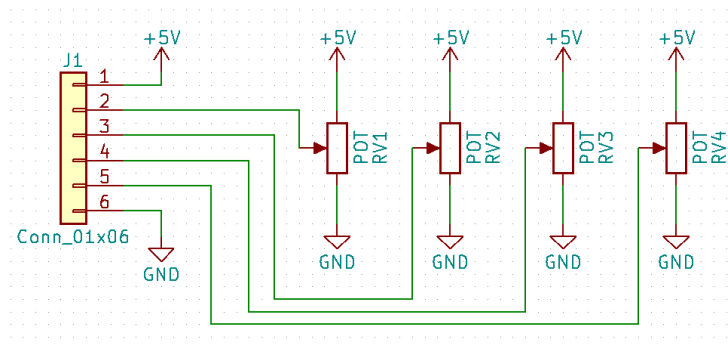
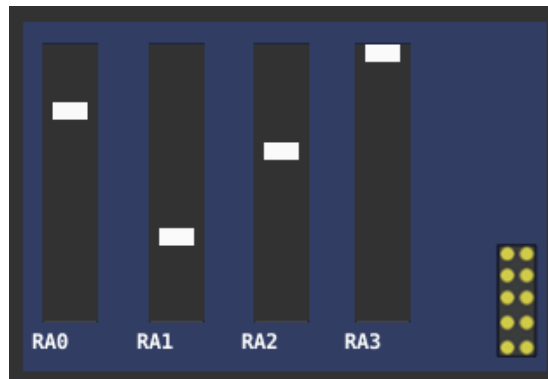
This part is a bar of 8 independent red LEDs.





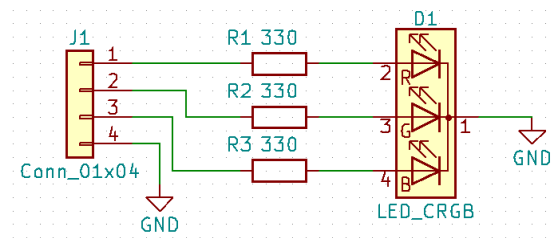
## 9.6 Potentiometers

This part is formed by 4 potentiometers connected between 0 and 5 volts, the output is connected to the cursor and varies within this voltage range.



## 9.7 RGB LED

This part consists of a 4-pin RGB LED. Each color can be triggered independently. Using PWM it is possible to generate several colors by combining the 3 primary colors.



## Chapter 10

## How To's

- [How to use MPLABX to program and debug PICsimLab.](#)
- [\(Deprecated\) How to Compile PICsimLab and Create New Boards.](#)

# **Part III**

# **License**

Copyright © 2018 Luis Claudio Gambôa Lopes <lcgamboa@yahoo.com>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.