PicsimLab_0_8_2

Luis Claudio Gambôa Lopes <lcgamboa@yahoo.com>

Download: github

August 23, 2020

Contents

Part I

English Manual

Table of Contents

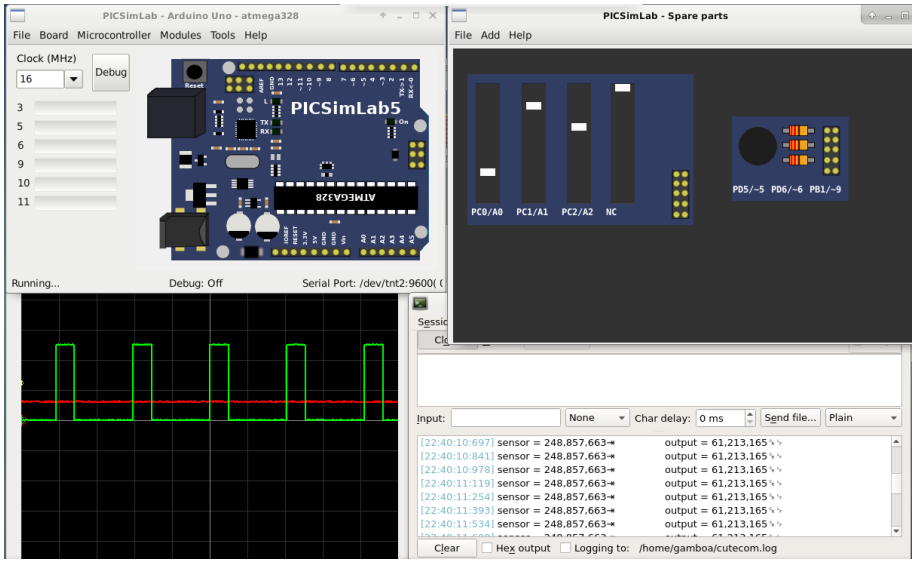
Chapter 1

Introduction

PICSimLab means PIC Simulator Laboratory

PICSimLab is a realtime emulator of development boards with integrated MPLABX/avr-gdb debugger. PICSimLab supports some picsim microcontrollers and some simavr microcontrollers. PICSimLab have integration with MPLABX/Arduino IDE for programming the boards microcontrollers.

The experimental version supports uCsim, gpsim and qemu-stm32 simulators in addition to the stable ones.



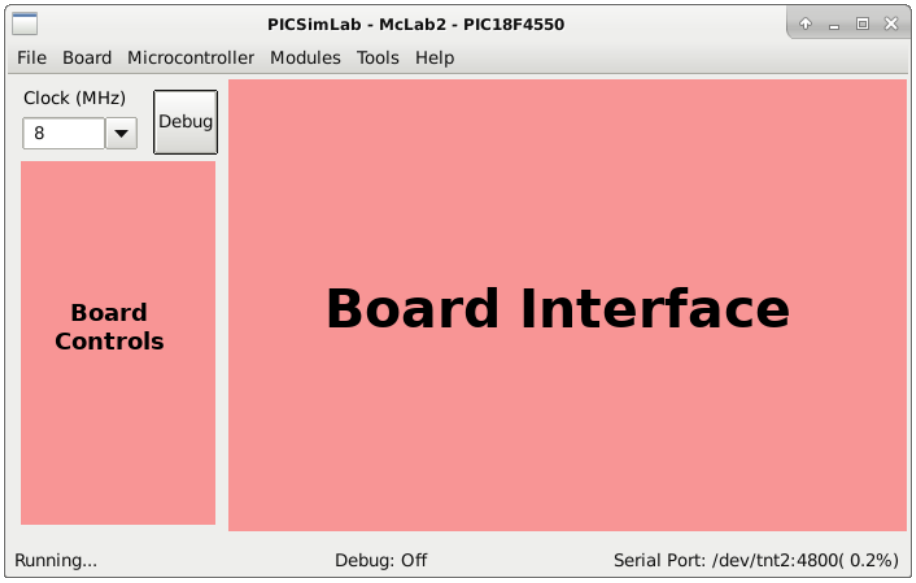
Chapter 2

Simulator 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

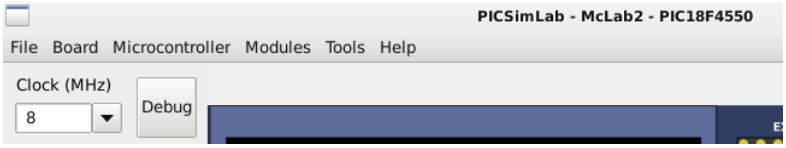
the simulation may present some difference than expected and the CPU load will be increased.

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
  - Save Hex - Save memory in a .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
  - Breadboard - Choose board Breadboard
  - McLab1 - Choose board McLab1
  - K16F - Choose board K16F
  - McLab2 - Choose board McLab2
  - PICGenios - Choose board PICGenios
  - Arduino Uno - Choose board Arduino Uno
- 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 Terminal - Open the serial terminal Cutecom
  - Serial Remote Tank - Open the remote tank simulator
  - Esp8266 Modem Simulator - Open the Esp8266 Modem Simulator
  - Arduino bootloader - Load microcontroller with Arduino serial bootloader
- Help
  - Contents - Open the Help window

- Board - Open the Board Help window
- Examples - Load the examples
- About Board - Show message about author and version of board
- About PICSimLab - Show message about author and version of PICSimLab



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

PICSimLab currently supports five backend simulators. The stable version supports picsim and simavr. The experimental version supports uCsim, gpsim and qemu-stm32 in addition to the stable ones.

The Figure 3.1 shows which cards are based on which backend simulator:

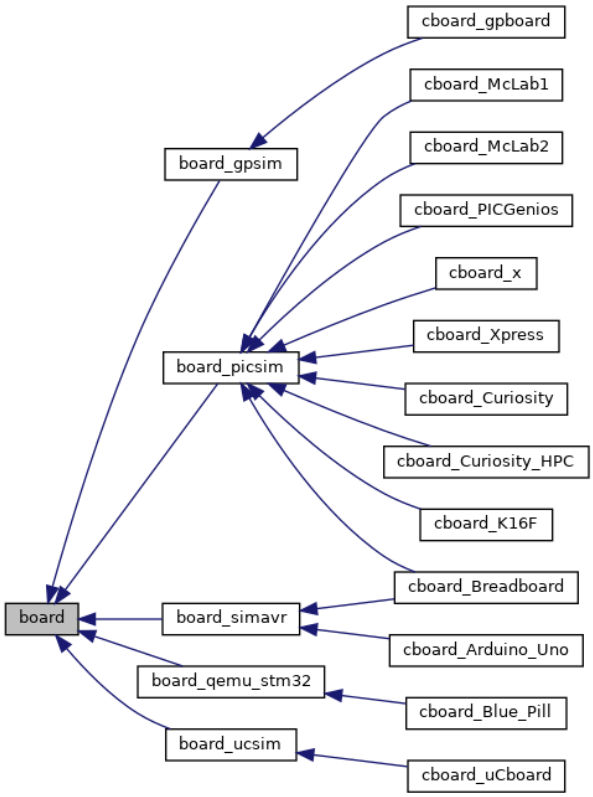


Figure 3.1: Boards backend simulators

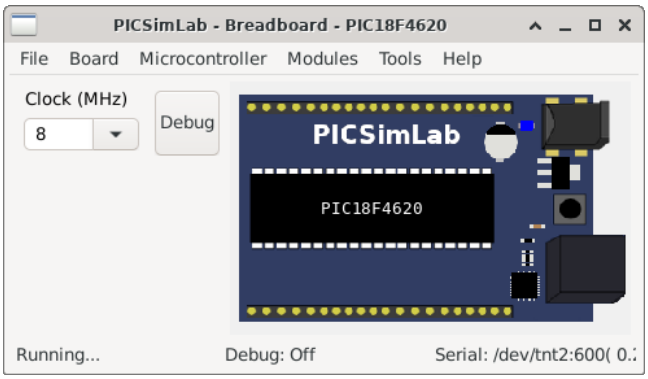
The below table show the supported debug interface of each simulator:

Backend	Debug Support
picsim	MPLABX Integrated Debug (see Chapter 6)
simavr	MPLABX Integrated Debug (see Chapter 6) and remote avr-gdb (see Chapter 8)
qemu-stm32	remote arm-gdb (see Chapter 9)
uCsim	uCsim remote console (telnet) (see Chapter 10)
gpsim	none yet

3.1 Features of Board Breadboard

TODO

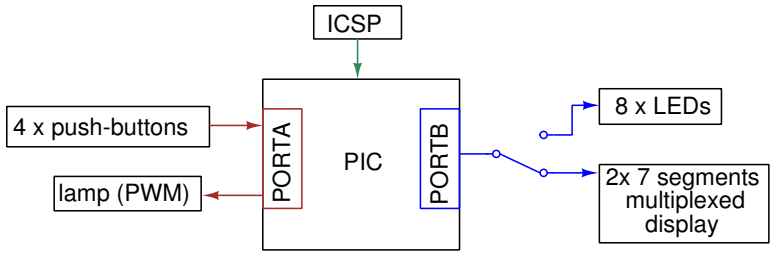
It is a generic board only with reset, serial and crystal circuits and support to multiple microcontrollers.

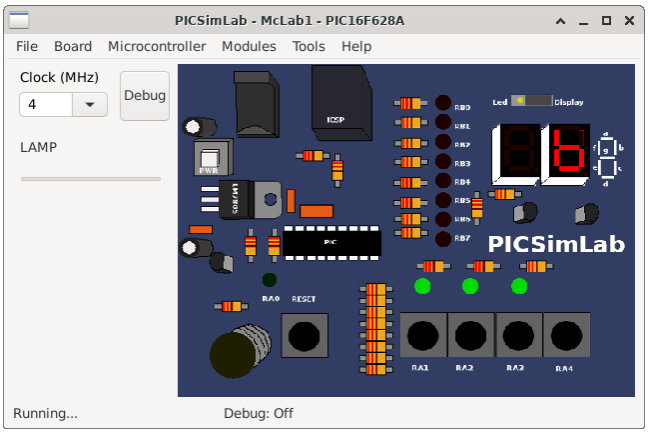


Examples

3.2 Features of Board McLab1

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





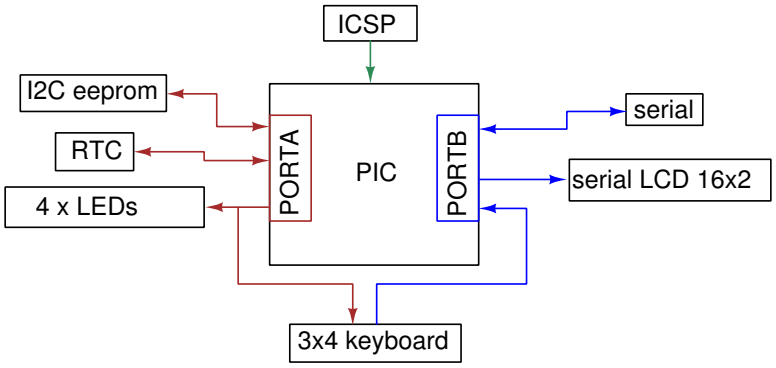
Board McLab1 schematics.

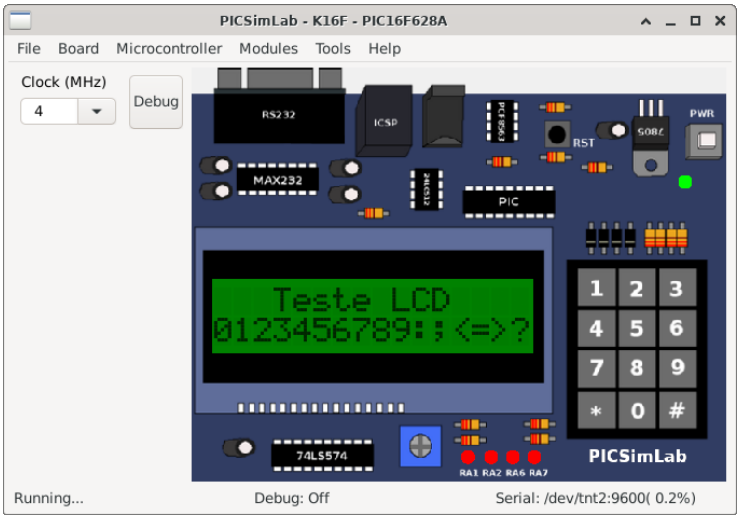
The code examples can be loaded in PICSimLab menu **Help->Examples**.

The source code of board McLab1 examples using MPLABX and XC8 compiler are in the link: board_McLab1.

3.3 Features of Board K16F

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





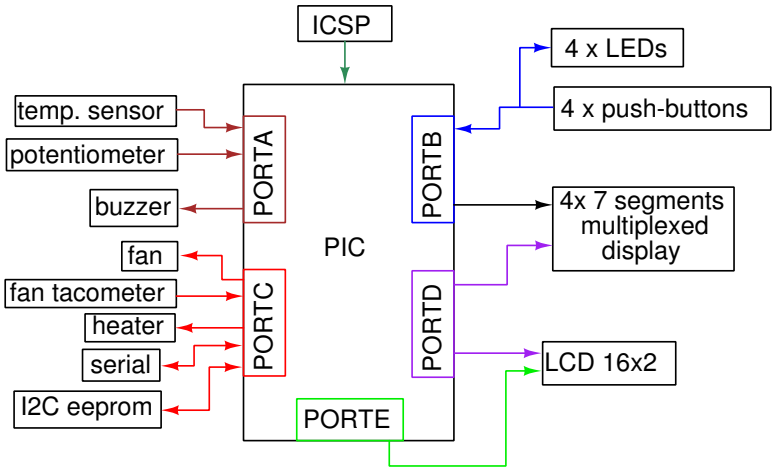
Board K16F schematics.

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

The source code of board K16F examples using MPLABX and XC8 compiler are in the link: board_K16F.

3.4 Features of Board McLab2

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



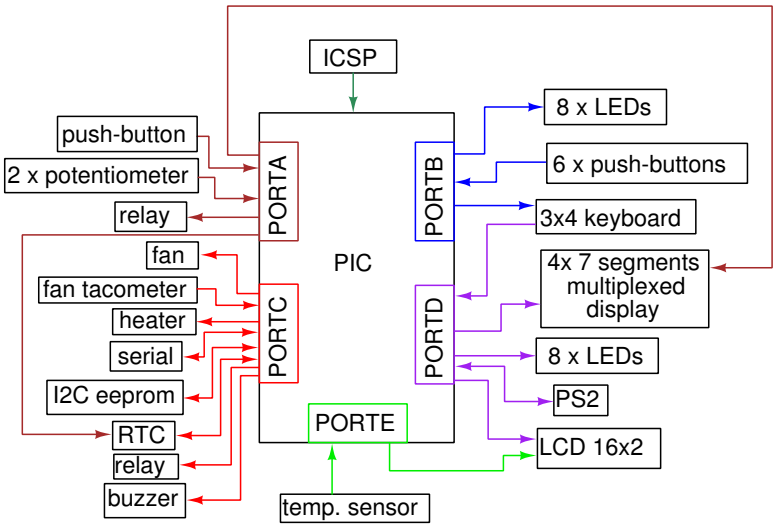
Board McLab2 schematics.

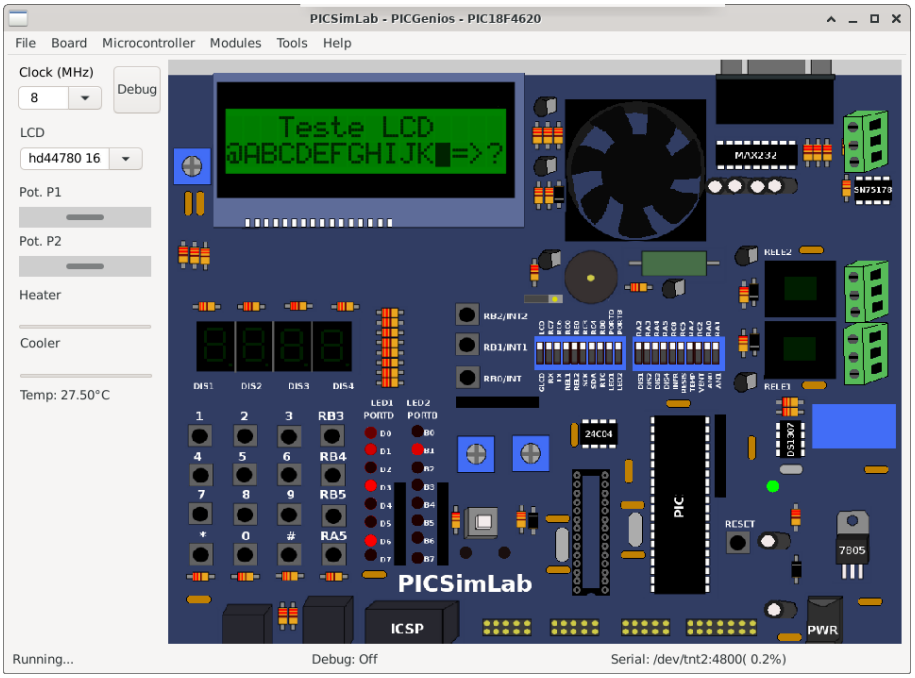
The code examples can be loaded in PICSimLab menu **Help->Examples**.

The source code of board McLab2 examples using MPLABX and XC8 compiler are in the link: board_McLab2.

3.5 Features of Board PICGenios

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





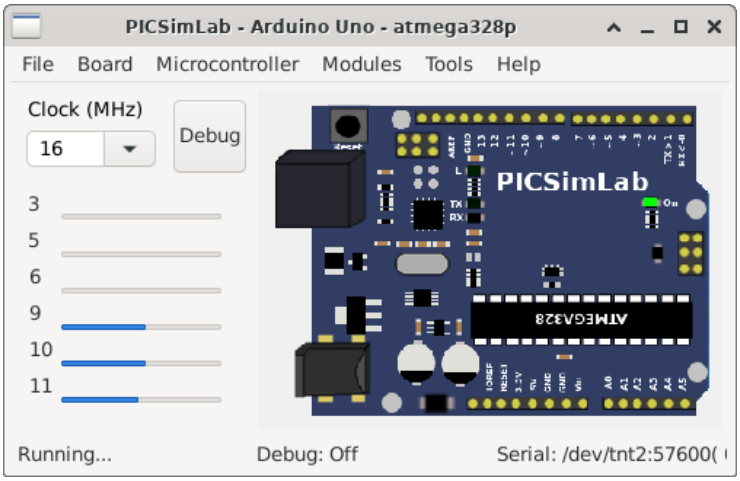
Board PICGenios schematics.

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

The source code of board PICGenios examples using MPLABX and XC8 compiler are in the link: board_PICGenios.

3.6 Features of Board Arduino Uno

It emulates the Arduino Uno development board that uses one ATMEGA328 microcontroller.



Board Arduino Uno schematics.

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

The source code of board Arduino Uno examples using the Arduino IDE with avr-gcc are in the link: board_Arduino_Uno.

More informations about the Arduino in www.arduino.cc

Chapter 4

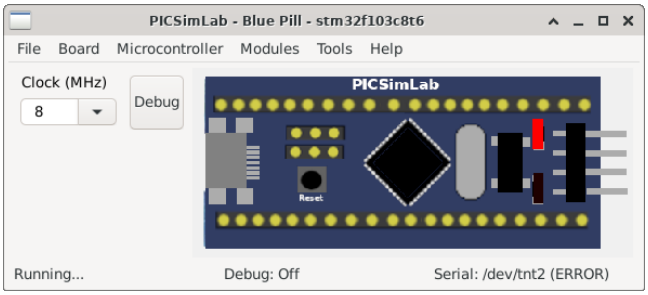
Experimental Boards

Boards in the experimental phase. Probably with some bugs and missing features.

4.1 Features of Board Blue Pill

TODO

It is a generic board only with reset, serial and crystal circuits and support to stm32f103c8t6 microcontroller of qemu-stm32.

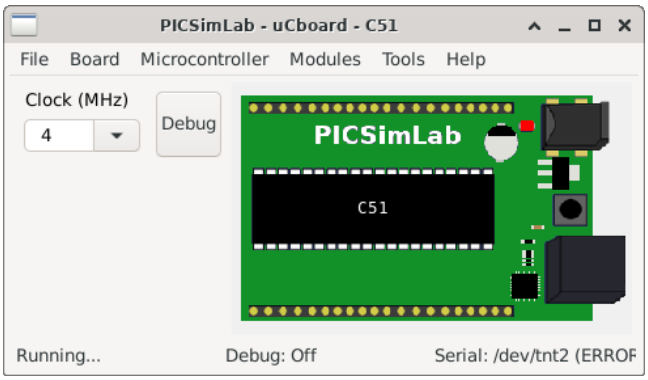


Examples

4.2 Features of Board uCboard

TODO

It is a generic board only with reset, serial and crystal circuits and support to multiple microcontrollers (initially C51, Z80 and STM8S103)of uCsim.

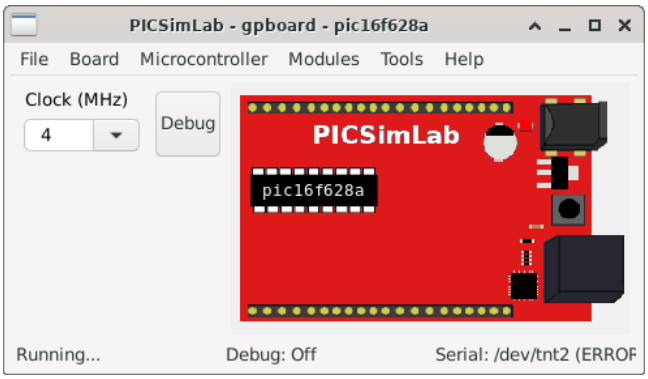


Examples

4.3 Features of Board gpboard

TODO

It is a generic board only with reset, serial and crystal circuits and support to multiple microcontrollers of gpsim.

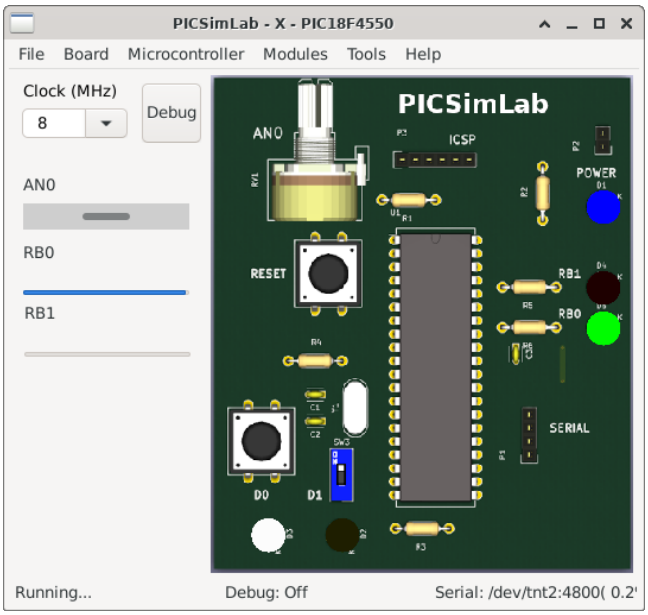


Examples

4.4 Features of Board X

TODO

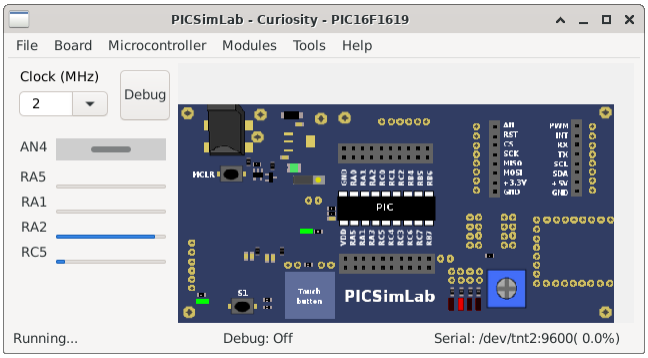
It is a generic board, used as example in “how to make a PICSimLab board”.



Examples

4.5 Features of Board Curiosity

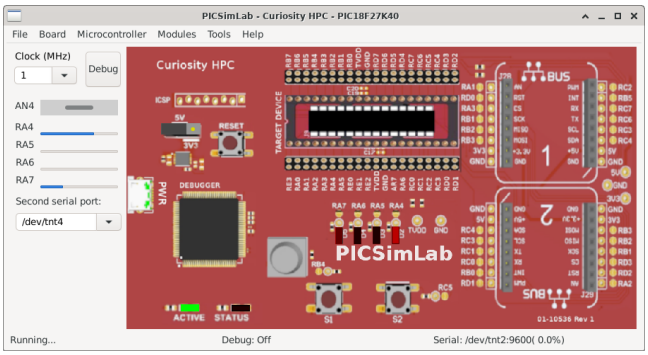
TODO



Examples

4.6 Features of Board Curiosity HPC

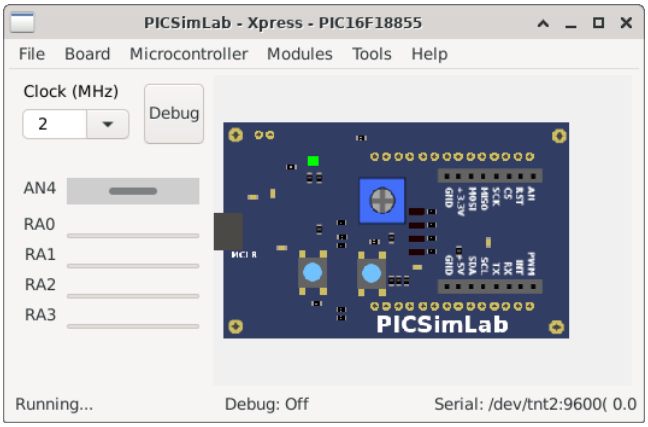
TODO



Examples

4.7 Features of Board Xpress

TODO



Examples

Chapter 5

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:

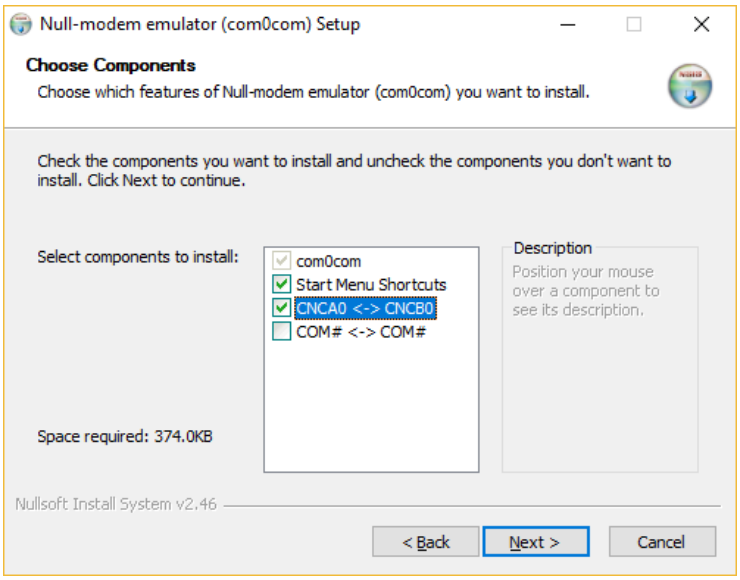
OS	PicsimLab port	Cutecom port	NULL-Modem prog.	Connection
Windows	com1	com2	com0com	com1<=>com2
Linux	/dev/tnt2	/dev/tnt3	tty0tty	/dev/tnt2<=>/dev/tnt3

5.1 Com0com Installation and Configuration(Windows)

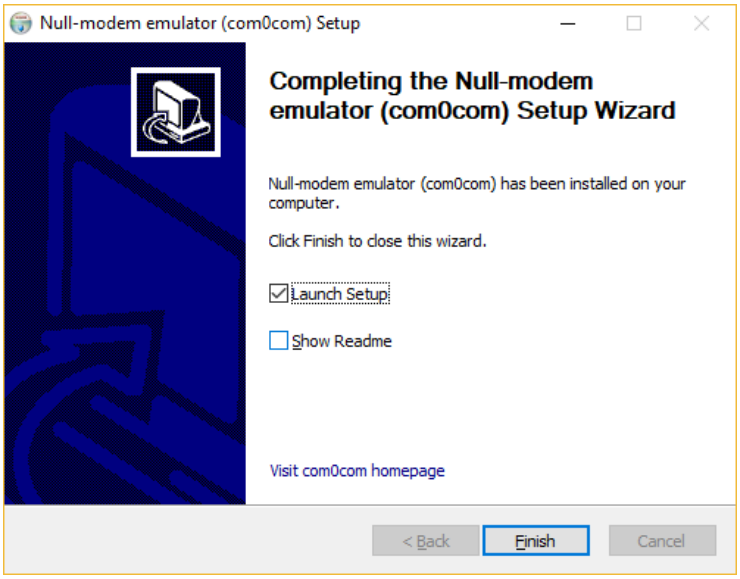
Download the signed version of com0com.

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.

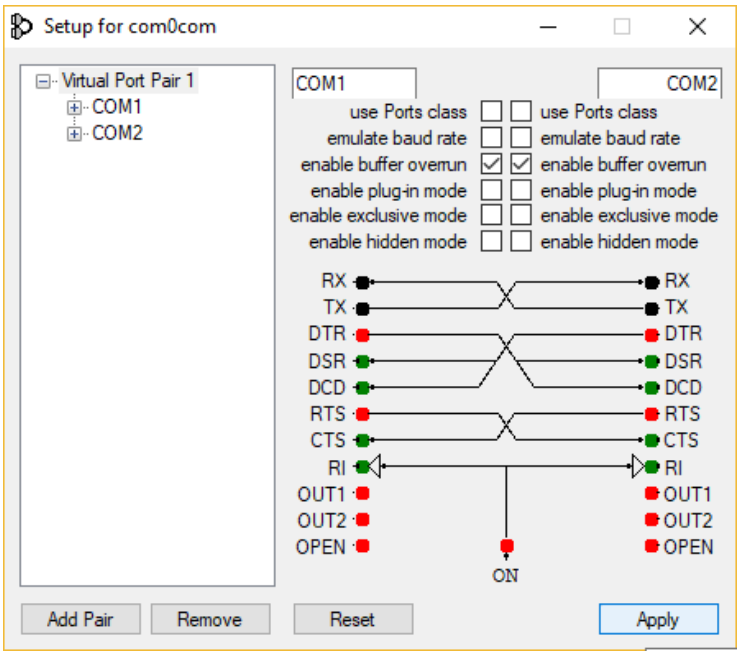
Configure the “choose components” window as the figure below:



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.



5.2 tty0tty Installation and Configuration (Linux)

Download the href https://github.com/lcgamboa/tty0tty/archive/master.zip tty0tyy. 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 6

MPLABX Integrated Debug (picsim and simavr)

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 (or other defined in configuration window), and you have to allow the access in the firewall.

Tutorial: how to use MPLABX to program and debug PICsimLab.

It's possible import and debug a Arduino sketch into MPLABX using the Arduino import plugin.

Chapter 7

Arduino IDE Integration (simavr)

For integrated use with the Arduino IDE, simply configure the serial port as explained in the section 5 and load the Arduino bootloader. The bootloader can be loaded from the "Tools->Arduino bootloader" menu.

In Windows, considering com0com making a NULL-MODEM connection between COM1 and COM2, simply connect the PICSimLab on the COM1 port (defined in configuration window) 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/liblistSerialsj.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 8

avr-gdb Debug (simavr)

With debug support enabled you can use avr-gdb to debug the code used in the simulator. Use the configuration window to choose between MDB (MPLABX) or GDB to debug AVR microcontrollers.

Use avr-gdb with the .elf file as the parameter:

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

and the command below to connect (1234 is the default port):

```
target remote localhost:1234
```

Graphic debug mode can be made using eclipse IDE with Sloeber Arduino plugin.

Chapter 9

arm-gdb Debug (qemu-stm32)

With debug support enabled you can use arm-none-eabi-gdb (or gdb-multiarch) to debug the code used in the simulator.

Use arm-none-eabi-gdb with the .elf file as the parameter:

```
arm-none-eabi-gdb compiled_file.elf
```

and the command below to connect (1234 is the default port):

```
target remote localhost:1234
```

Graphic debug mode can be made using eclipse IDE with Eclipse Embedded CDT.

Chapter 10

uCsim Debug

The uCsim debug console can be accessed with the telnet (1234 is the default port):

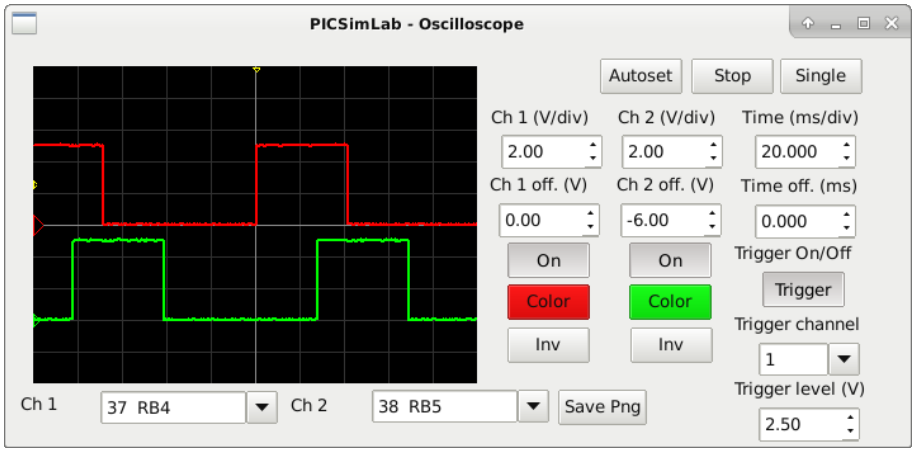
```
telnet localhost 1234
```

All uCsim commands are supported.
For windows users putty telnet client is a good option to access the uCsim console.

Chapter 11

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 12

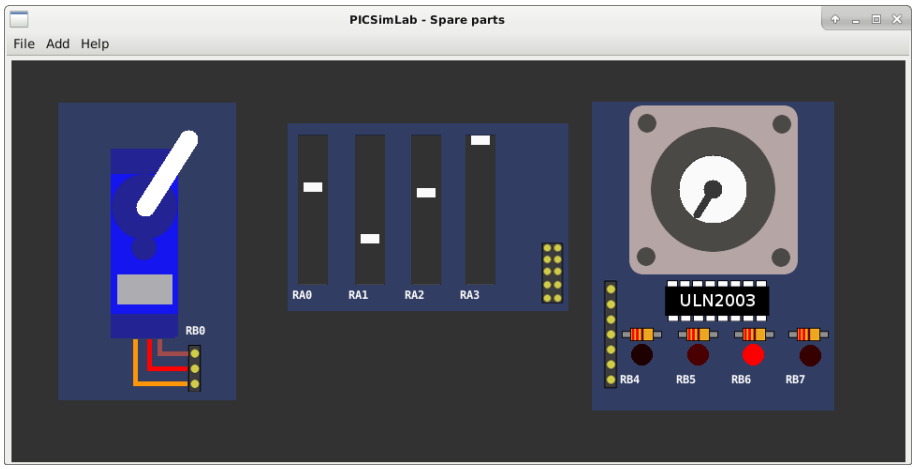
Spare Parts

The PICSimLab has a window that allows the connection of spare parts to the microcontroller, 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
  - 7 Segments Display - TODO
  - Buzzer - TODO
  - D. Transfer Function - TODO
  - Gamepad - Adds a gamepad
  - IO 74xx595 - TODO
  - IO MCP23S17 - TODO
  - IO PCF8574 - TODO
  - Keypad - TODO
  - LCD hd44780 - Adds a text display hd44780
  - LCD pcf8833 - Adds a color graphic display pcf8833
  - LCD pcd8544 - Adds a monochrome graphic display pcd8544
  - LED Matrix - TODO
  - LEDs - Adds 8 red LEDs
  - MEM 24CXXX - TODO
  - Potentiometers - Adds 4 potentiometers

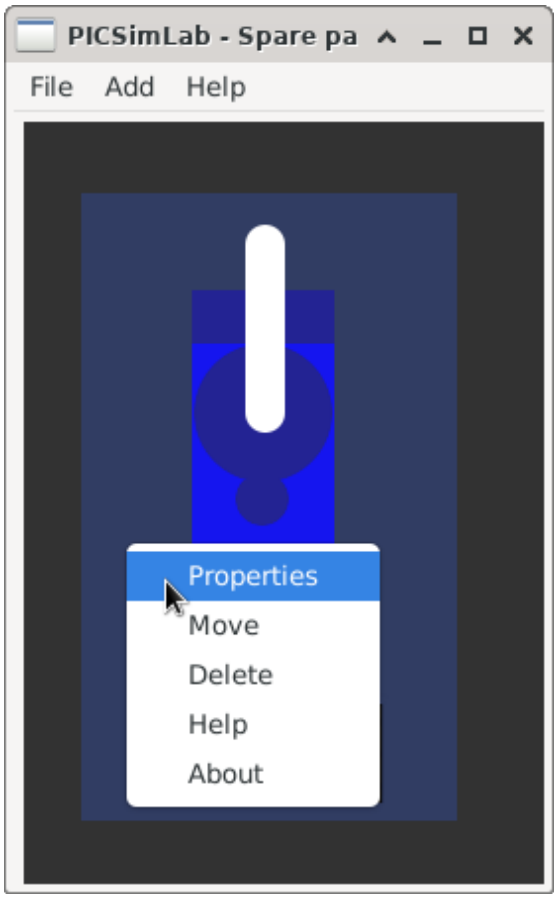
  - Push Buttons - Adds 8 push buttons
  - Push Buttons (Analogic) - TODO
  - RGB LED - Adds 1 RGB LED
  - RTC ds1307 - TODO
  - RTC pfc8563 - TODO
  - Servo Motor - Adds a servo motor
  - Signal Generator - TODO
  - Step Motor - Adds a step motor
  - Switchs - Adds 8 switchs
  - Temperature System - TODO
  - VCD Dump - TODO
  - VCD Dump (Analogic) - TODO
- 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:

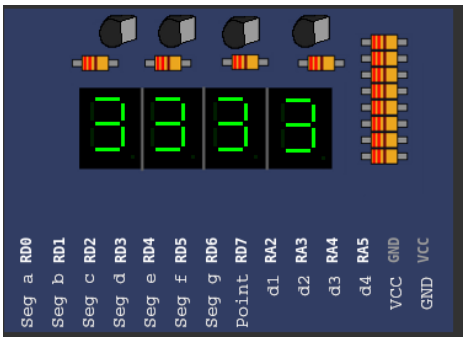
- Properties - Opens the connection settings window
- Move - Unlocks the part to move
- Delete - Remove part
- Help - Open Help window of part

- About - Show message about author and version of part



12.1 7 Segments Display

TODO



Examples

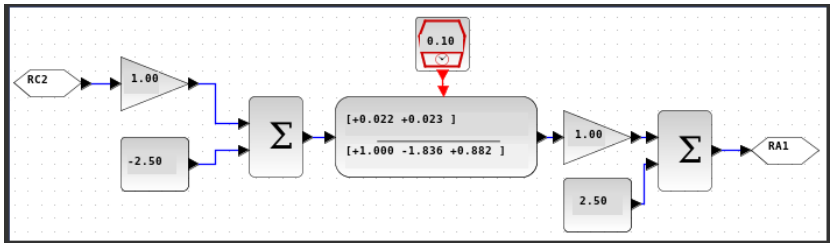
12.2 Buzzer

TODO

Examples

12.3 D. Transfer Function

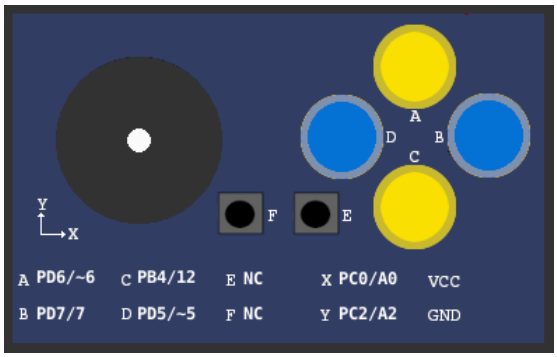
TODO

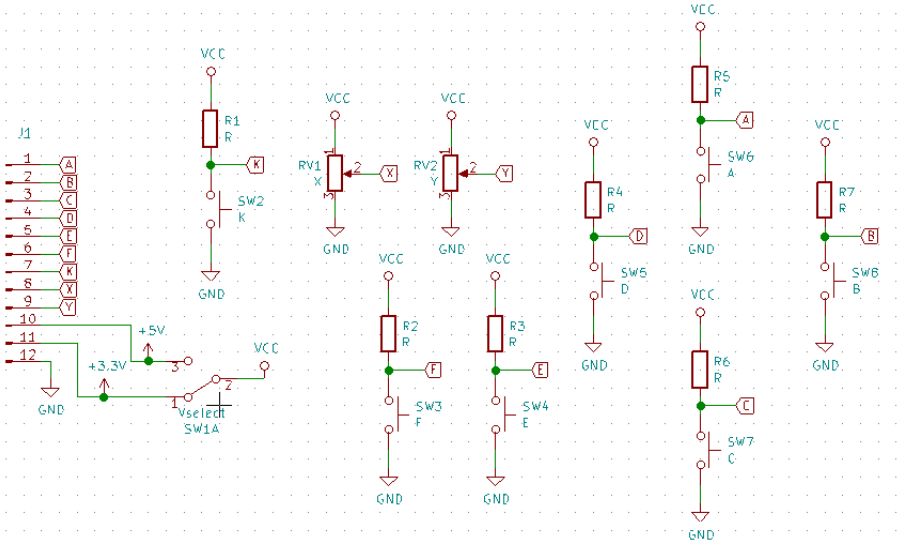


Examples

12.4 Gamepad

This part is a gamepad with two analog axis and 7 push buttons.

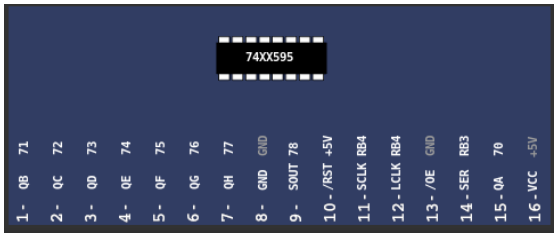




Examples

12.5 IO 74xx595

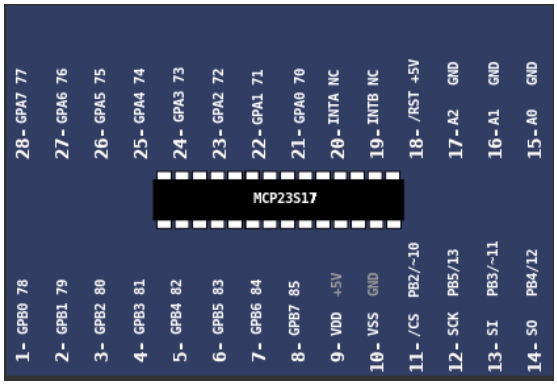
TODO



Examples

12.6 IO MCP23S17

TODO



Examples

12.7 IO PCF8574

TODO

Examples

12.8 Keypad

TODO

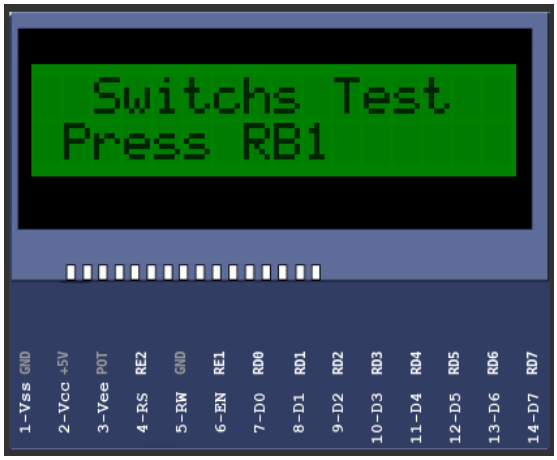

1 2 3
4 5 6
7 8 9
* 0 #
1-L1 RD3
2-L2 RD2
3-L3 RD1
4-L4 RD0
5-C1 RB0
6-C2 RB1
7-C3 RB2


1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D
1-L1 RD3
2-L2 RD2
3-L3 RD1
4-L4 RD0
5-C1 RB0
6-C2 RB1
7-C3 RB2
8-C4 RB3

Examples

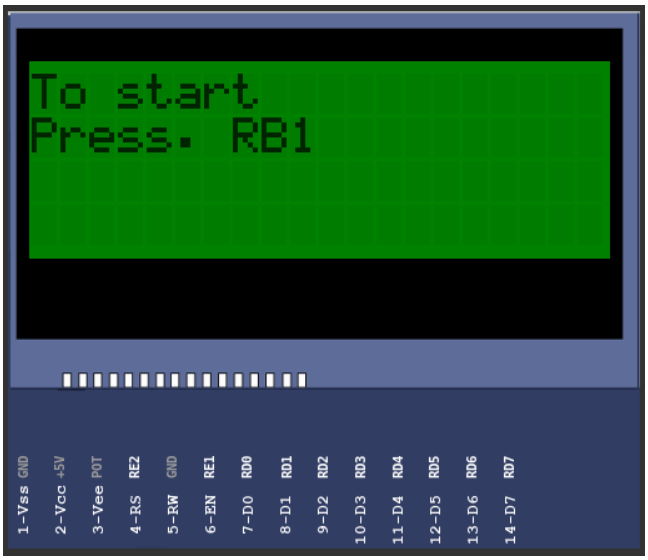
12.9 LCD hd44780

This part is a text display with 2 (or 4) lines by 16 (or 20) columns.




To start
Press. RB1
1-Vss GND
2-Vcc +5V
3-Vee POT
4-RS RE2
5-RW GND
6-EN RE1
7-D0 RD0
8-D1 RD1
9-D2 RD2
10-D3 RD3
11-D4 RD4
12-D5 RD5
13-D6 RD6
14-D7 RD7

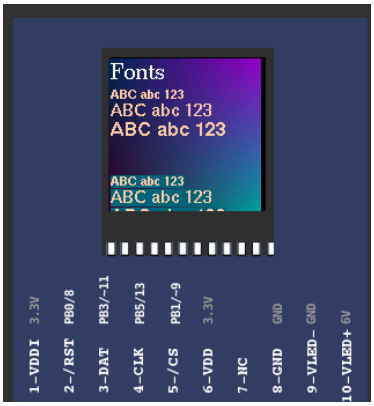

To start
Press. RB1
1-Vss GND
2-Vcc +5V
3-Vee POT
4-RS RE2
5-RW GND
6-EN RE1
7-D0 RD0
8-D1 RD1
9-D2 RD2
10-D3 RD3
11-D4 RD4
12-D5 RD5
13-D6 RD6
14-D7 RD7



Examples

12.10 LCD pcf8833

This part is a color graphic display with 132x132 pixels.



Examples

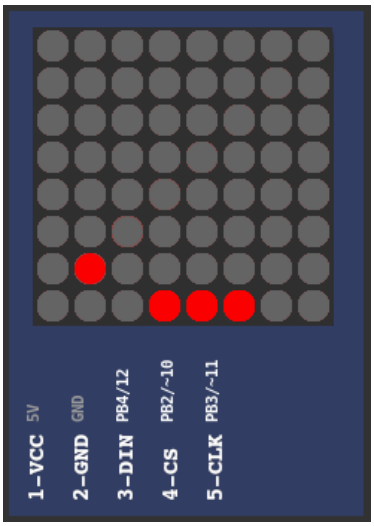
12.11 LCD pcd8544

This part is a monochrome graphic display with 48x84 pixels.

Examples

12.12 LED Matrix

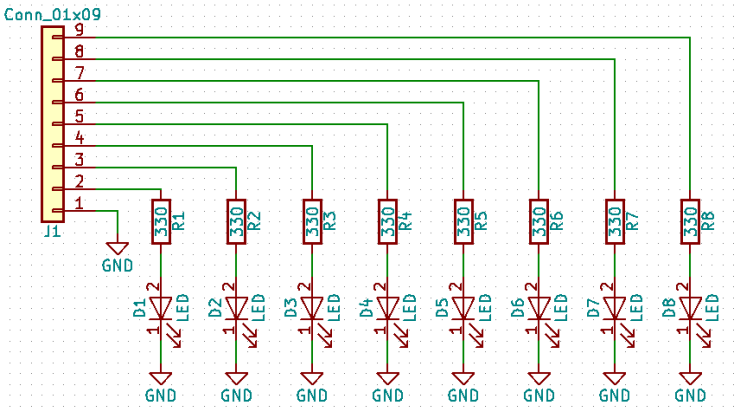
TODO



Examples

12.13 LEDs

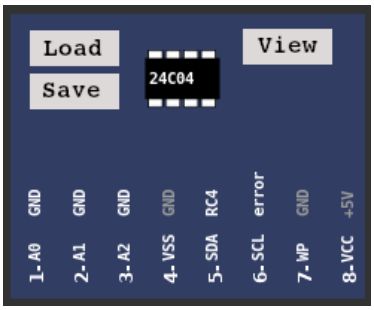
This part is a bar of 8 independent red LEDs.



Examples

12.14 MEM 24CXXX

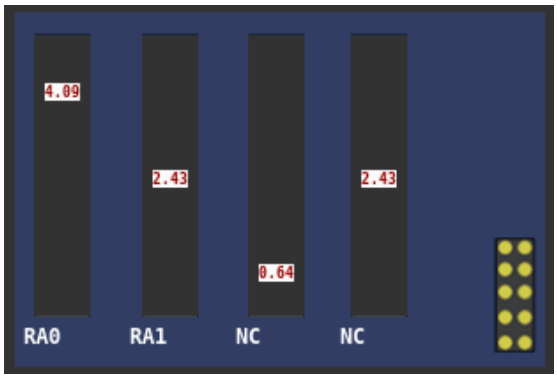
TODO

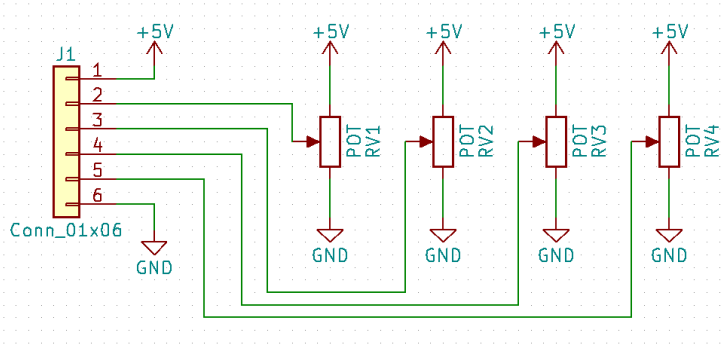


Examples

12.15 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.

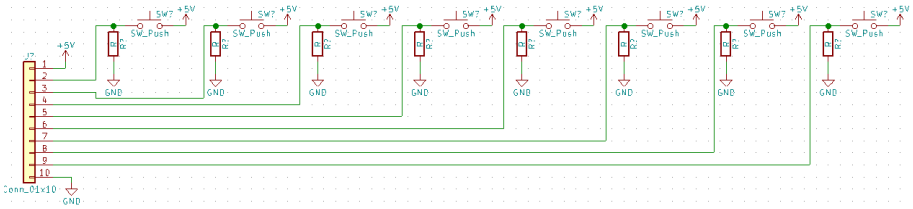




Examples

12.16 Push Buttons

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



Examples

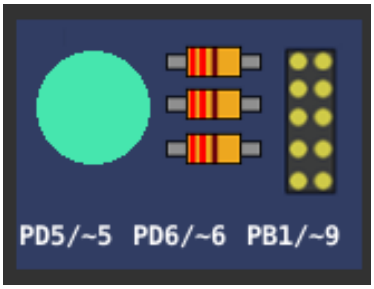
12.17 Push Buttons (Analogic)

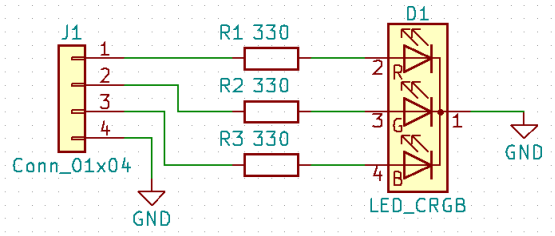
TODO

Examples

12.18 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.

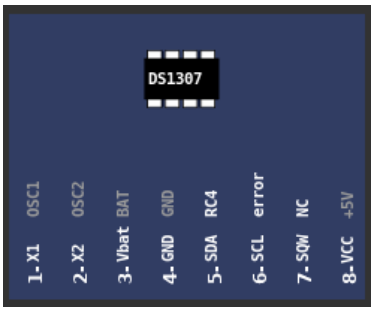




Examples

12.19 RTC ds1307

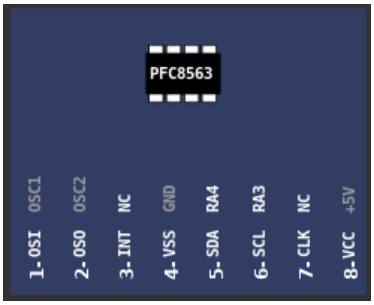
TODO



Examples

12.20 RTC pfc8563

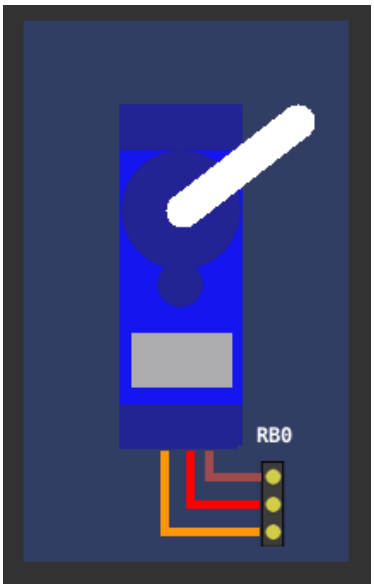
TODO

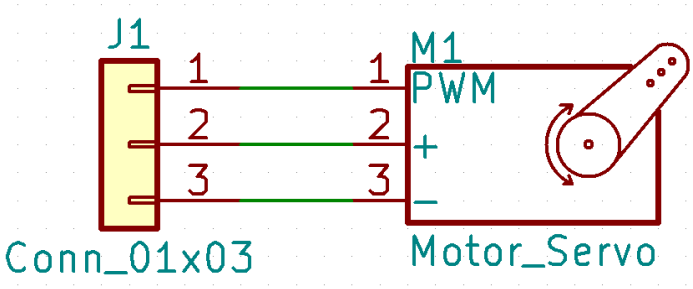


Examples

12.21 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 -90º, one from 1.5ms to 0º and one from 2ms to 90º.

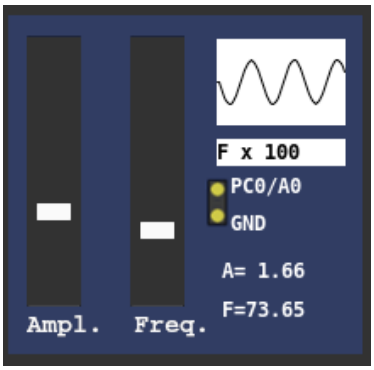




Examples

12.22 Signal Generator

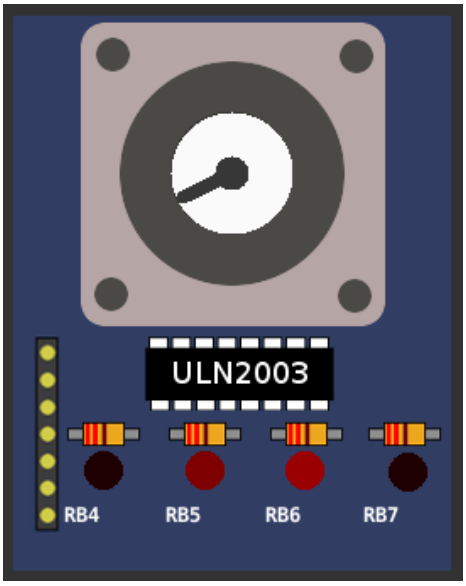
TODO

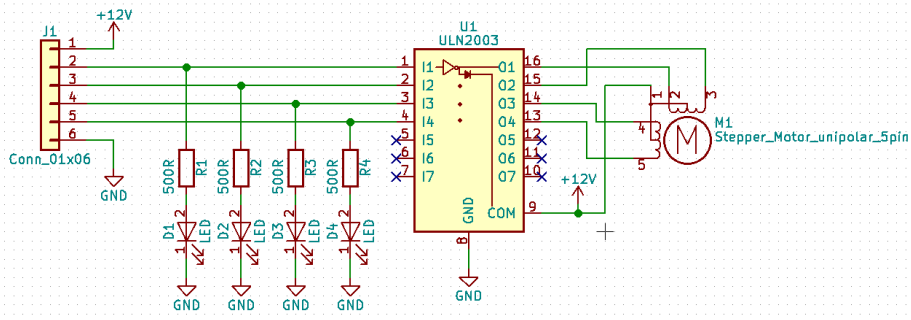


Examples

12.23 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 1.8º.

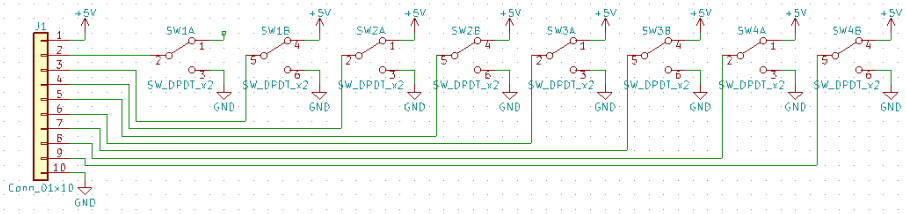




Examples

12.24 Switchs

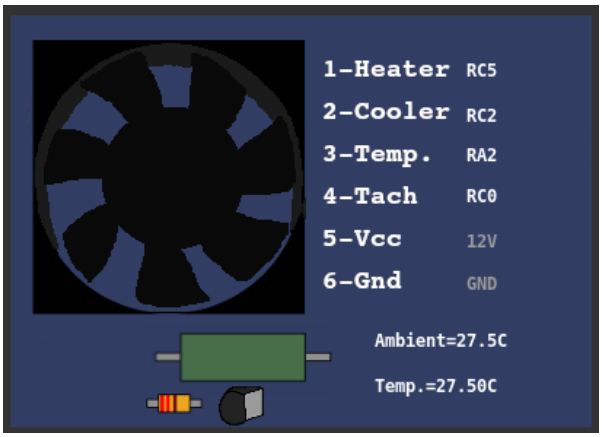
This part consists of 8 keys with on or off position (0 or 1).



Examples

12.25 Temperature System

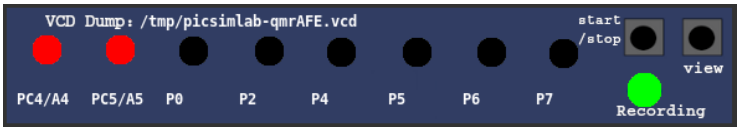
TODO

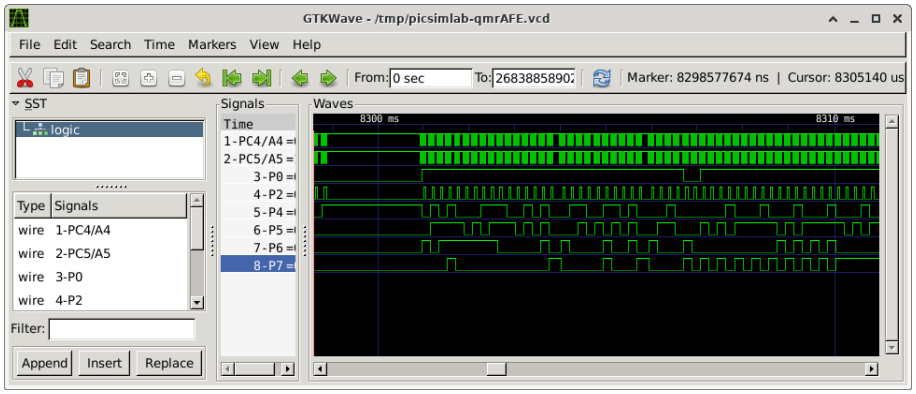


Examples

12.26 VCD dump

TODO

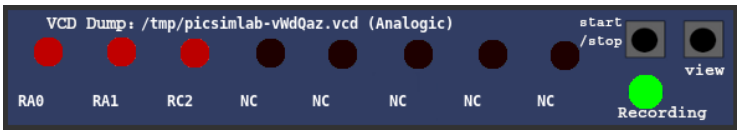


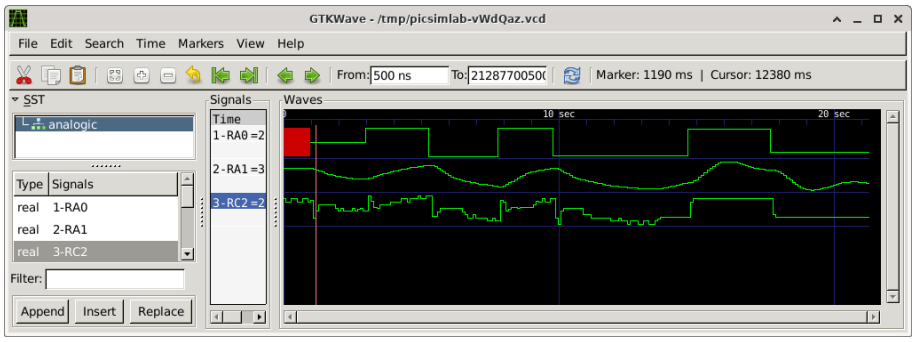


Examples

12.27 VCD dump (Analogic)

TODO


VCD Dump: /tmp/picsimlab-vWdQaz.vcd (Analogic)
start
/stop
view
RA0
RA1
RC2
NC
NC
NC
NC
NC
Recording


GTKWave - /tmp/picsimlab-vWdQaz.vcd
File Edit Search Time Markers View Help
From: 500 ns
To: 2128770050(
Marker: 1190 ms | Cursor: 12380 ms
SST
analogic
Type Signals
real 1-RA0
real 2-RA1
real 3-RC2
Filter:
Append Insert Replace
Signals
Time
1-RA0 =2
2-RA1 =3
3-RC2 =2
Waves
10 sec
20 sec

Examples

Chapter 13

How To's

- How to use MPLABX to program and debug PICsimLab.
- How to Compile PICsimLab and Create New Boards.

Parte II

Manual em Português

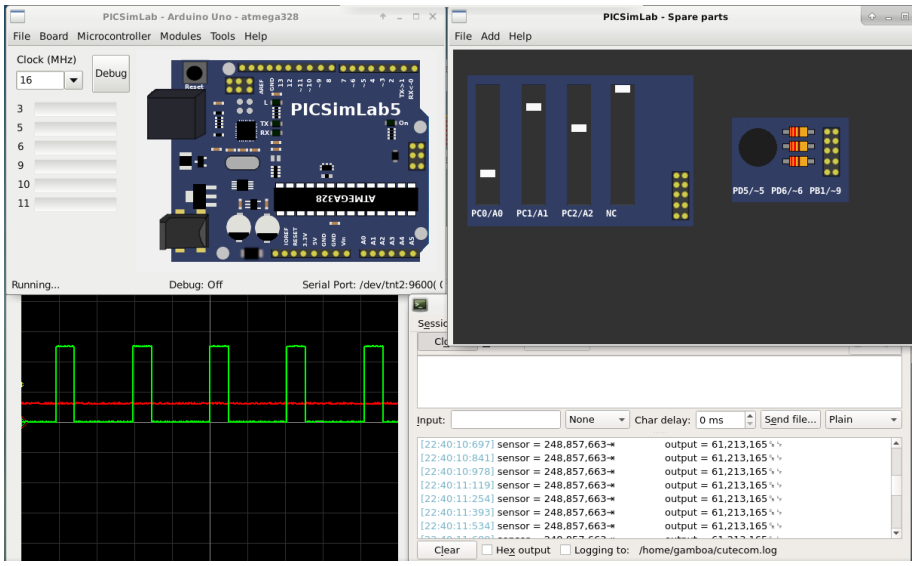
Table of Contents

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 simavr (ATMEGA328). O PICSimLab tem integração com as IDE MPLABX/Arduino para programação dos microcontroladores das placas.



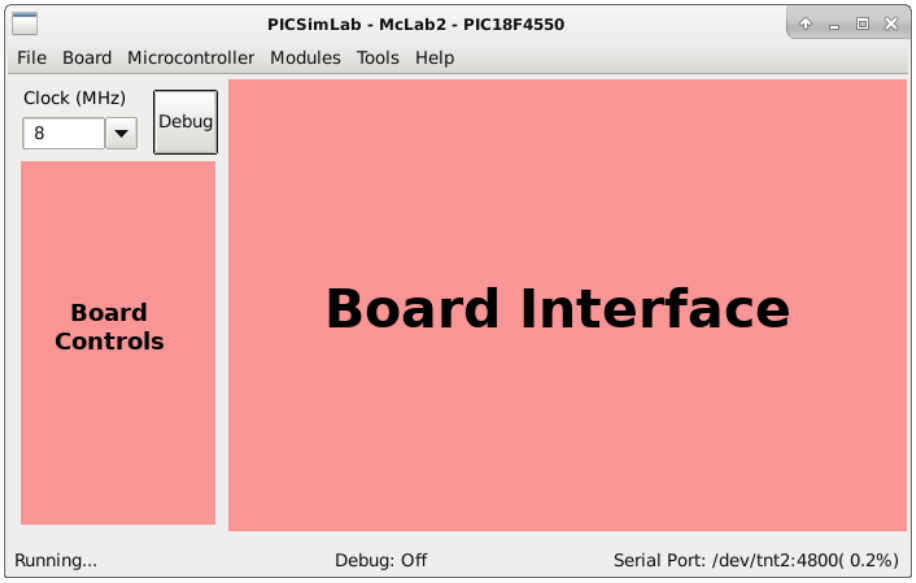
Capítulo 2

Interface do Simulador

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

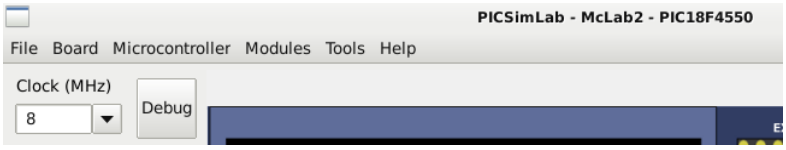
selecionado. Neste caso a simulação pode apresentar alguma diferença do esperado e a carga da CPU ser aumentada.

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
  - Save Hex - Salva a memória em um arquivo .hex
  - 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
  - Breadboard - Escolhe a placa Breadboard
  - McLab1 - Escolhe a placa McLab1
  - K16F - Escolhe a placa K16F
  - McLab2 - Escolhe a placa McLab2
  - PICGenios - Escolhe a placa PICGenios
  - Arduino Uno - Escolhe a placa Arduino Uno
- 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 Terminal - Abre o terminal serial Cutecom
  - Serial Remote Tank - Abre o simulador de tank remoto
  - Esp8266 Modem Simulator - Abre o simulador de modem Esp8266
  - Arduino bootloader - Carrega o microcontrolador com o bootloader serial do Arduino
- Help

- Contents - Abre a janela de Ajuda
- Board - Abre a janela de Ajuda da placa
- Examples - Carrega exemplos
- About Board - Mostra mensagem de versão e autor da placa
- About PICSimLab - Mostra mensagem de versão e autor do PICSimLab



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 ...

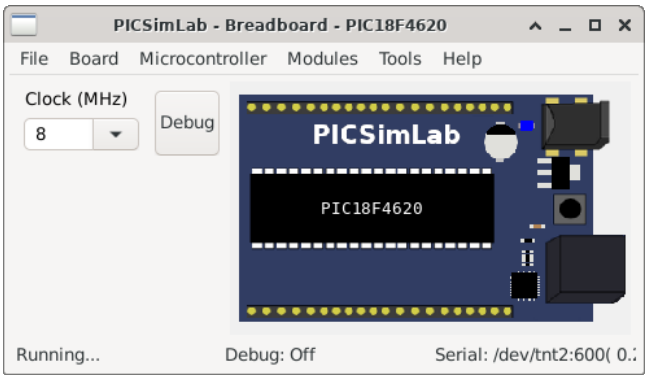
Capítulo 3

Placas

3.1 Características da Placa Breadboard

TODO

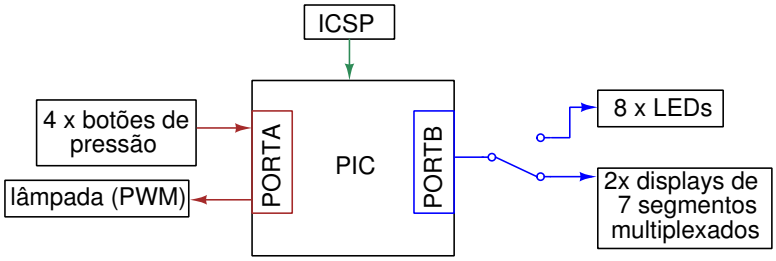
É uma placa genérica só com os circuitos de reset, serial e cristal com suporte a múltiplos microcontroladores.

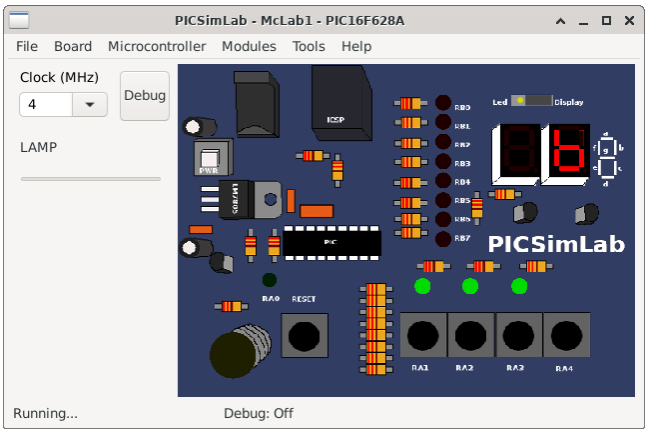


Exemplos

3.2 Características da Placa McLab1

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





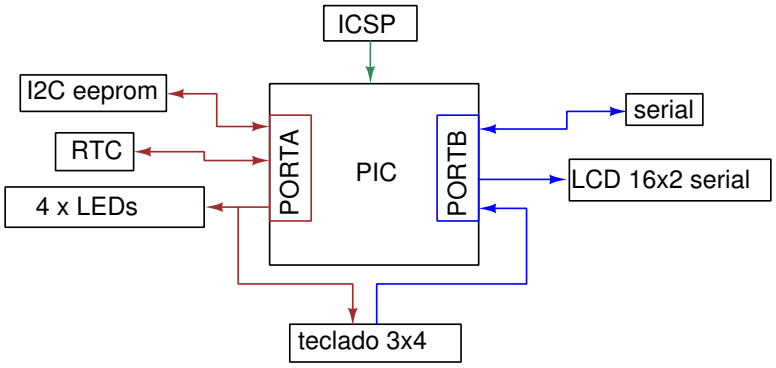
Esquemático da placa McLab1.

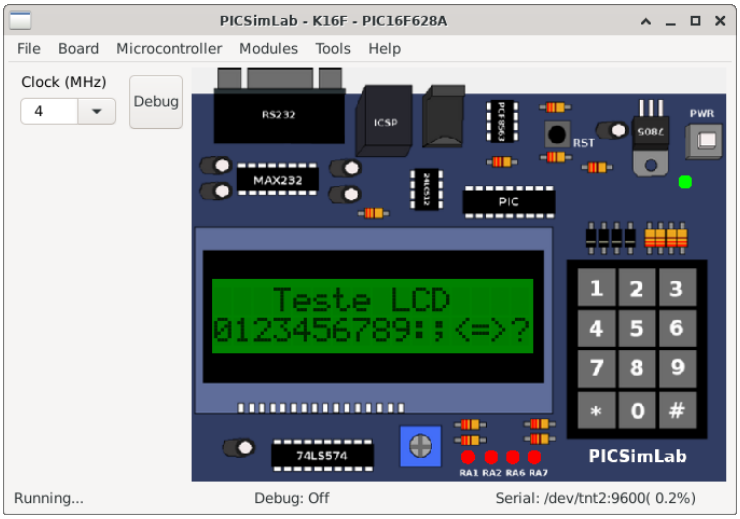
Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSimLab.

O código fonte dos exemplos da placa McLab1 utilizando o MPLABX e o compilador XC8 está no link: board_McLab1.

3.3 Características da Placa K16F

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





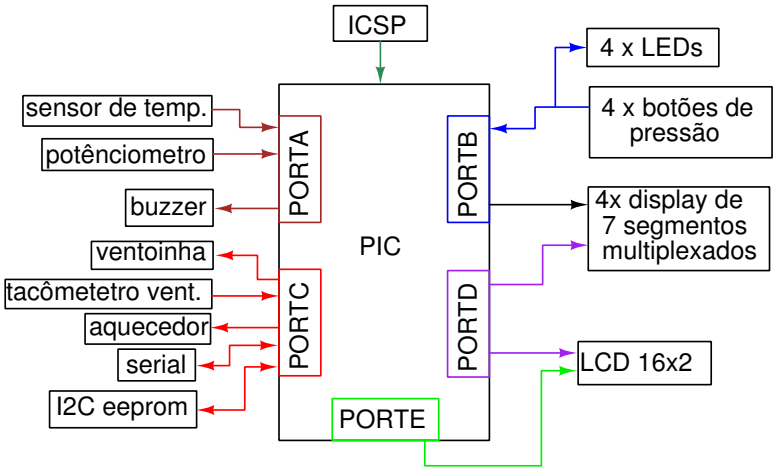
Esquemático da placa K16F.

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

O código fonte dos exemplos da placa K16F utilizando o MPLABX e o compilador XC8 está no link: board_K16F.

3.4 Características da Placa McLab2

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



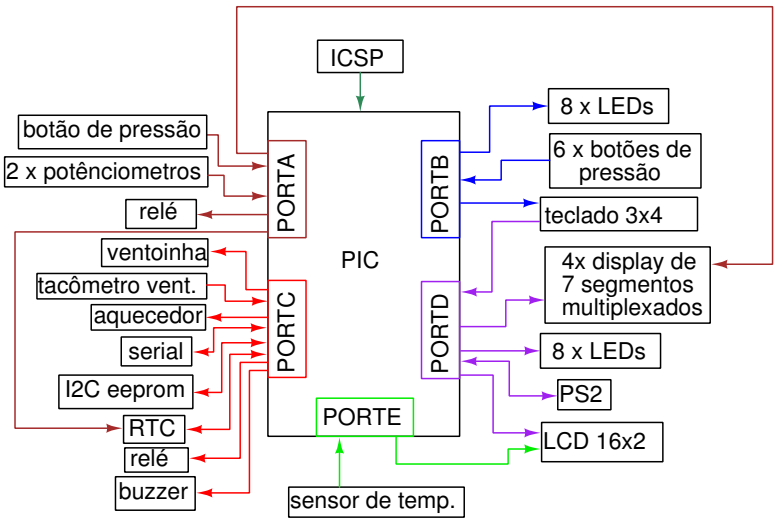
Esquemático da placa McLab2.

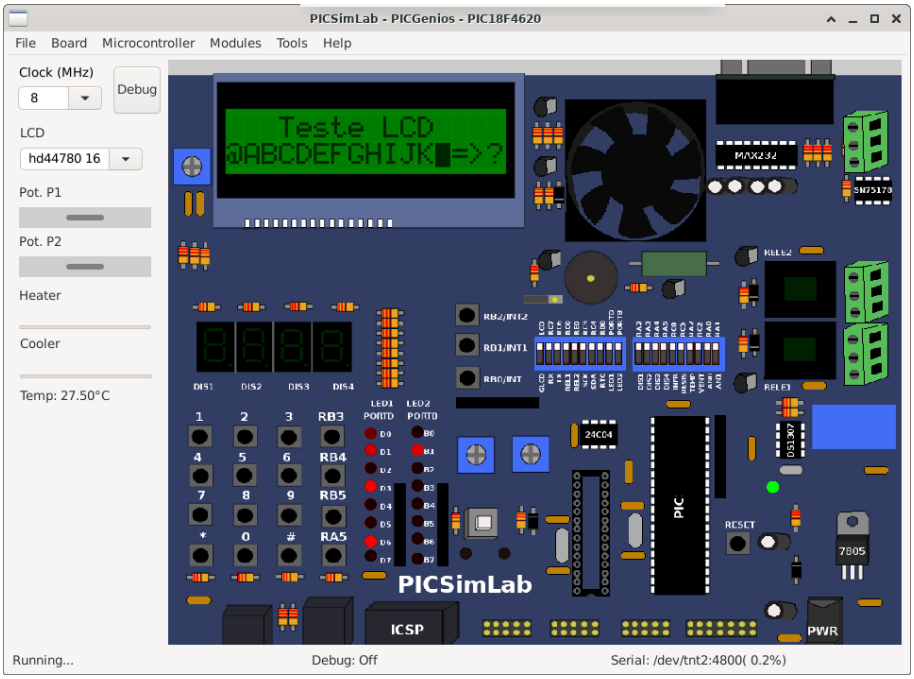
Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSimLab.

O código fonte dos exemplos da placa McLab2 utilizando o MPLABX e o compilador XC8 está no link: board_McLab2.

3.5 Características da Placa PICGenios

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 PICGenios.

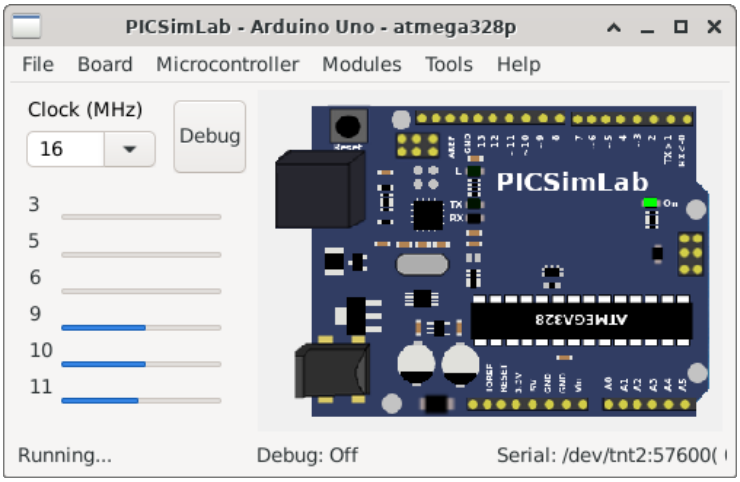
Os códigos fontes de exemplo podem ser carregados através do menu **Help->Examples** do PICSimLab.

O código fonte dos exemplos da placa PICGenios utilizando o MPLABX e o com-

pilador XC8 está no link: board_PICGenios.

3.6 Características da Placa Arduino Uno

Emula a placa de desenvolvimento Arduino Uno que utiliza um microcontrolador ATMEGA328.



Esquemático da placa Arduino Uno.

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

O código fonte dos exemplos da placa Arduino Uno utilizando a IDE Arduino com o avr-gcc está no link: board_Arduino_Uno.

Mais informações sobre o Arduino em 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:

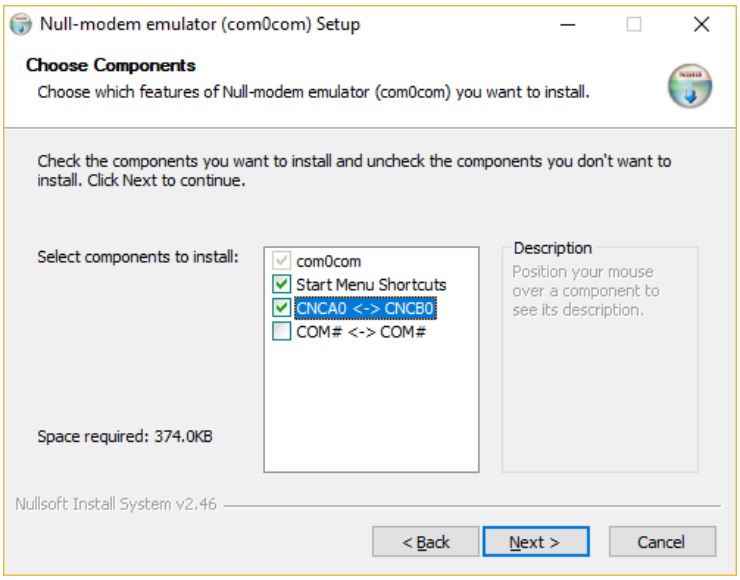
OS	porta PICSimLab	porta Cutecom	NULL-Modem prog.	Conexão
Windows	com1	com2	com0com	com1<=>com2
Linux	/dev/tnt2	/dev/tnt3	tty0tty	/dev/tnt2<=>/dev/tnt3

4.1 Instalação e Configuração do com0com (Windows)

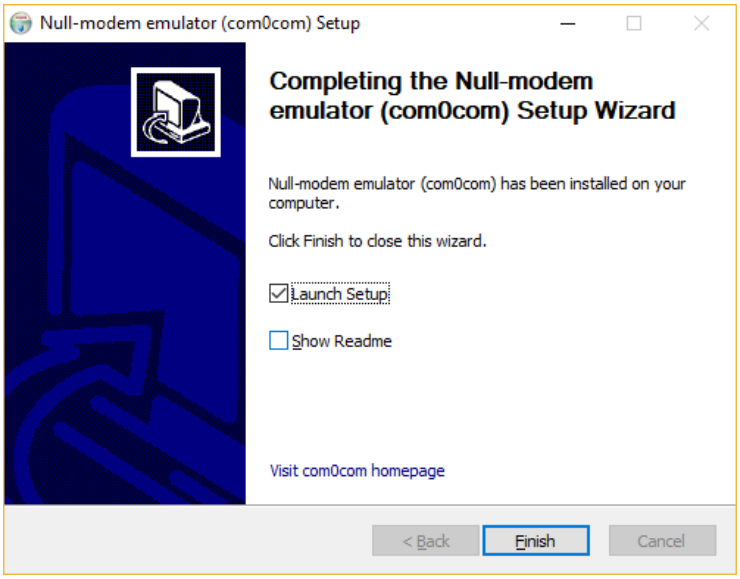
Faça o download da versão assinada do com0com.

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.

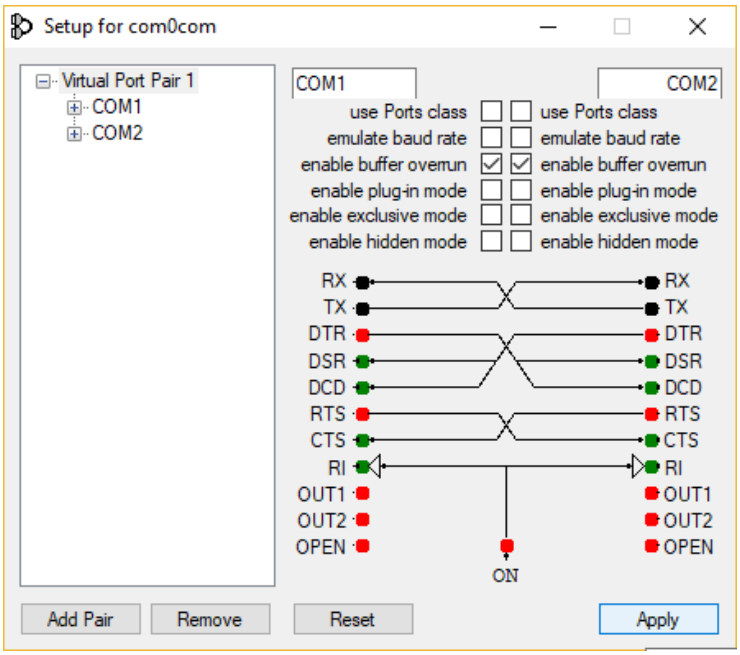
Configure a janela “choose components” como a figura abaixo:



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 tty0tyy. 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 ncessário 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 (picsim e simavr)

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 (ou outra definida na janela de configuração), permita o acesso no firewall.

Tutorial: Como usar o MPLABX para programar e depurar o PICsimLab (Inglês)

É possível importar e depurar um sketch do Arduino no MPLABX utilizando o Arduino import plugin.

Capítulo 6

Integração com a IDE do Arduino (simavr)

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 "Tools->Arduino Bootloader".

No windows, considerando o com0com fazendo uma conexão NULL-MODEM entre a porta COM1 e COM2, basta conectar o PICSimLab na porta COM1 (definido na janela de configuração) 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/liblistSerialsj.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 (simavr)

Com o suporte a depuração habilitado é possível utilizar o avr-gdb para depurar o código utilizado no simulador. Use a janela de configurações para escolher entre o MDB (MPLABX) e o GDB para depuração dos microcontroladores AVR.

Utilize o avr-gdb com o arquivo .elf como parâmetro:

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

e o comando abaixo para se conectar:

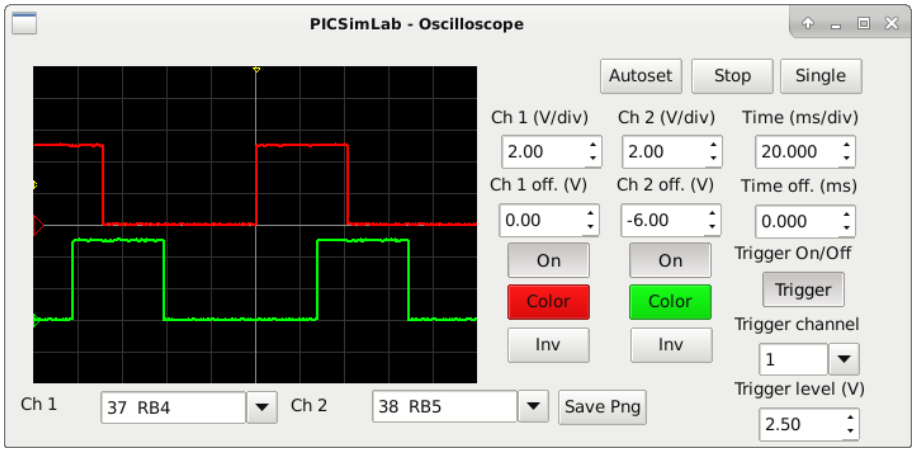
```
target remote localhost:1234
```

O debug no modo gráfico pode ser feito utilizando a IDE eclipse junto com o Sloeber Arduino plugin.

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

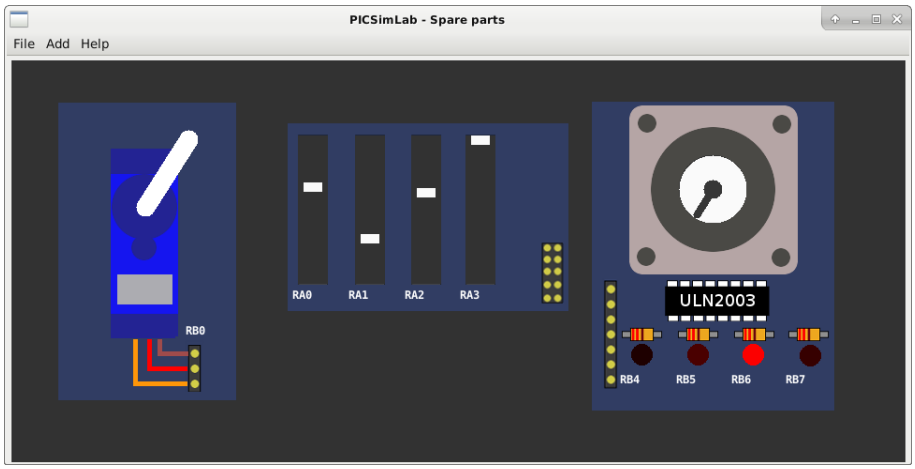
Partes Avulsas

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
  - 7 Segments Display - TODO
  - Buzzer - TODO
  - D. Transfer Function - TODO
  - Gamepad - Adicionae um gamepad
  - IO 74xx595 - TODO
  - IO MCP23S17 - TODO
  - IO PCF8574 - TODO
  - Keypad - TODO
  - LCD hd44780 - Adiciona um display de texto hd44780
  - LCD pcf8833 - Adiciona um display gráfico colorido pcf8833
  - LCD pcd8544 - Adiciona um display gráfico monocromático pcd8544
  - LED Matrix - TODO
  - LEDs - Adiciona 8 LEDs vermelhos
  - MEM 24CXXX - TODO
  - Potentiometers - Adiciona 4 potenciômetros

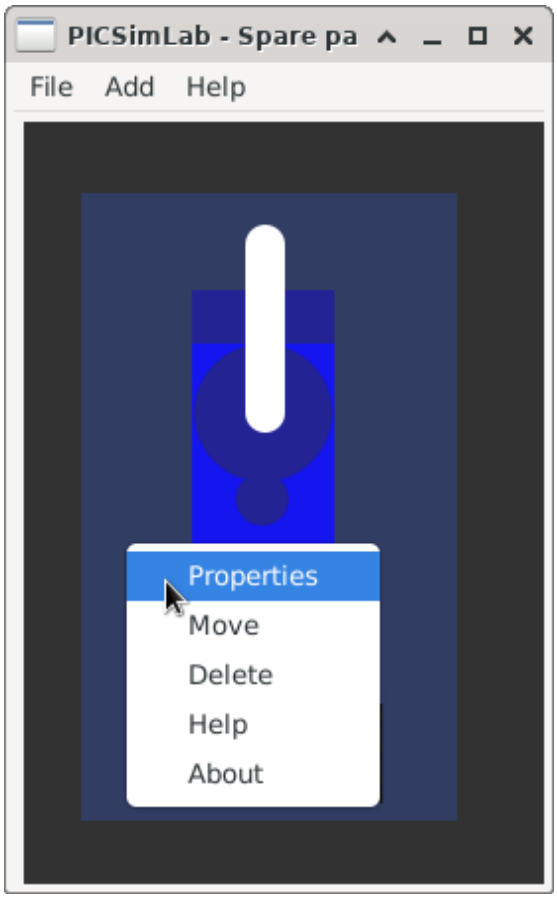
  - Push Buttons - Adiciona 8 botões de pressionar
  - Push Buttons (Analogic) - TODO
  - RGB LED - Adiciona 1 LED RGB
  - RTC ds1307 - TODO
  - RTC pfc8563 - TODO
  - Servo Motor - Adiciona um servo motor
  - Signal Generator - TODO
  - Step Motor - Adiciona um motor de passo
  - Switchs - Adiciona 8 chaves
  - Temperature System - TODO
  - VCD Dump - TODO
  - VCD Dump (Analogic) - TODO
- 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
- Help - Abre a janela de Ajuda da parte

- About - Mostra mensagem de versão e autor da parte



9.1 Parte 7 Segments Display

TODO

9.2 Parte Buzzer

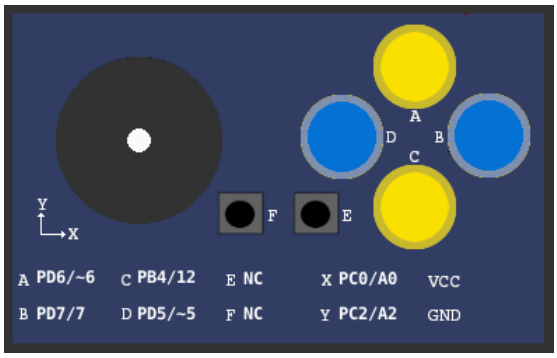
TODO

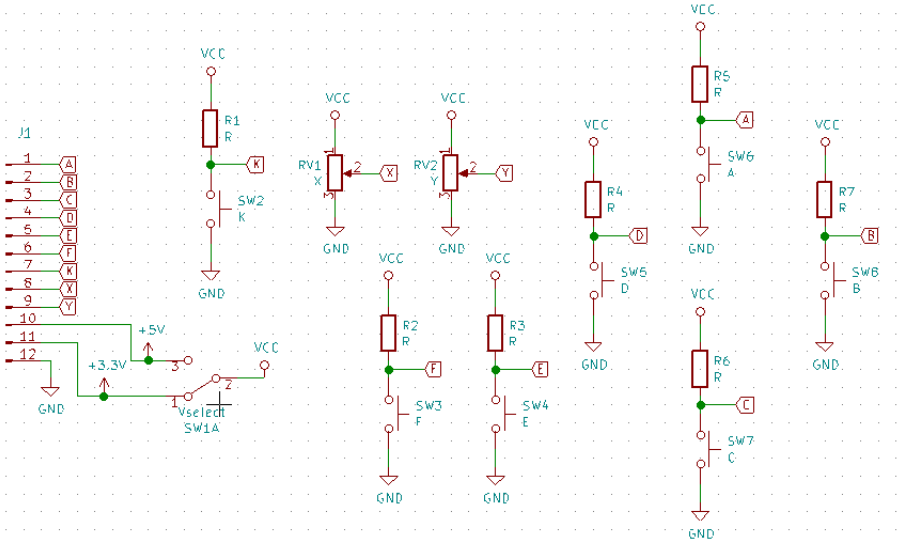
9.3 Parte D. Transfer Function

TODO

9.4 Parte Gamepad

Esta parte é um gamepad com dois eixos analógicos e 7 botões de pressão.



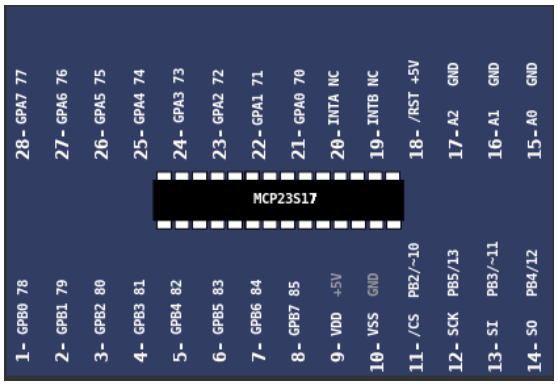


9.5 Parte IO 74xx595

TODO

9.6 Parte IO MCP23S17

TODO



Examples

9.7 Parte IO PCF8574

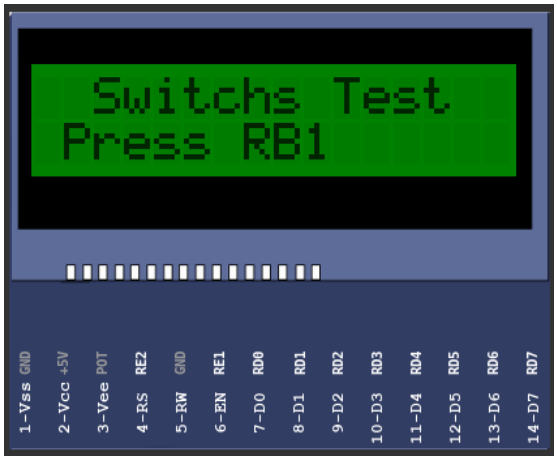
TODO

9.8 Parte Keypad

TODO

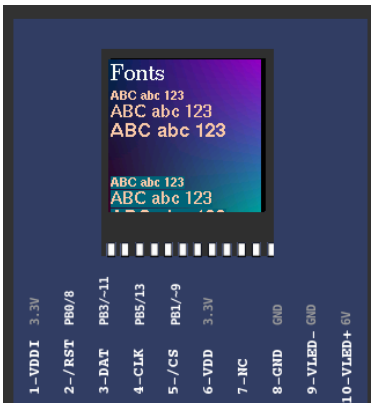
9.9 Parte LCD hd44780

Essa parte é um display de texto de 2 (ou 4 linhas) por 16 colunas.



9.10 Parte LCD pcf8833

Essa parte é um display de gráfico colorido de 132x132 pontos.



9.11 Parte LCD pcd8544

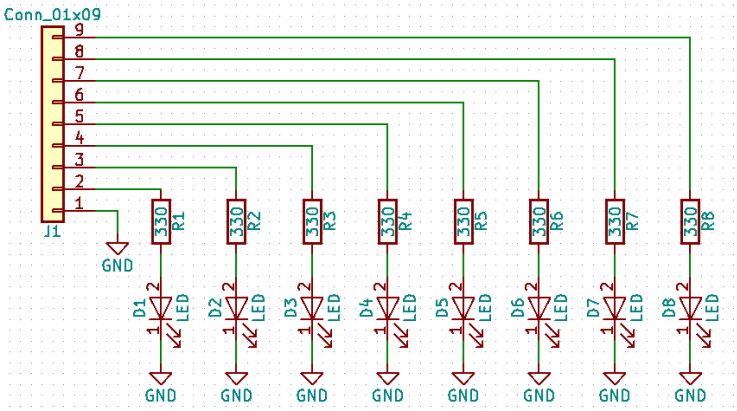
Essa parte é um display de gráfico monocromático de 48x84 pontos.

9.12 Parte LED Matrix

TODO

9.13 Parte LEDs

Essa parte é uma barra de 8 LEDs vermelhos independentes.

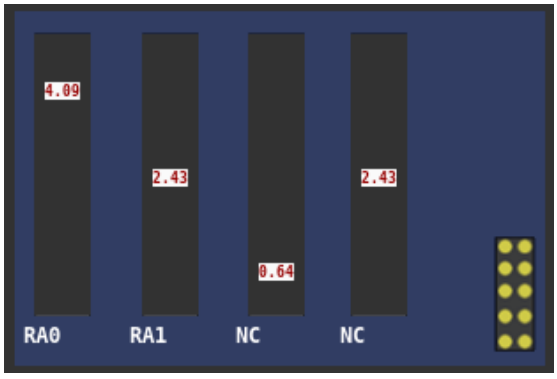


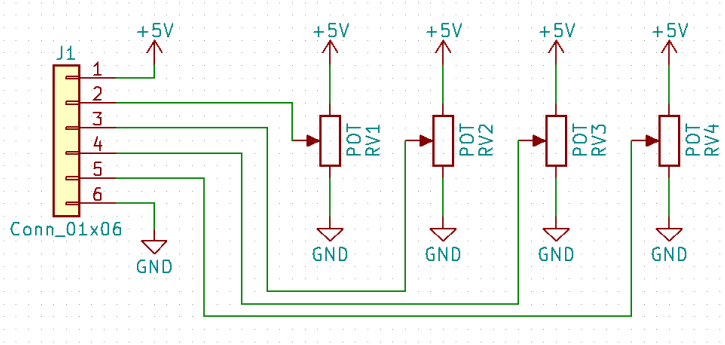
9.14 Parte MEM 24CXXX

TODO

9.15 Parte Potentiometers

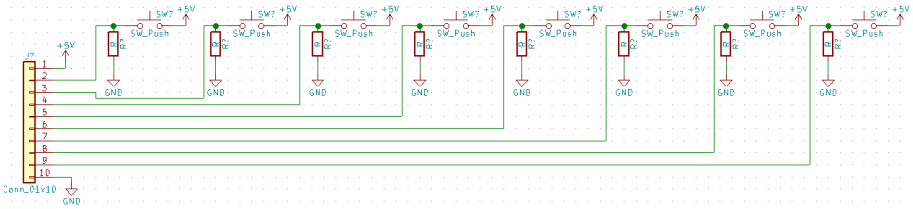
Essa parte é formada por 4 potenciômetros ligados entre 0 e 5 Volts, a saída está ligada ao cursor e varia dentro dessa faixa de tensão.





9.16 Parte Push Buttons

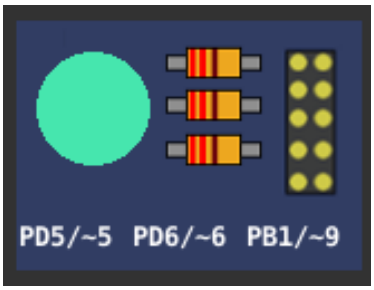
Esta parte é formada por 8 botões de pressão. Quando pressionado a saída vai para nível lógico “1”.

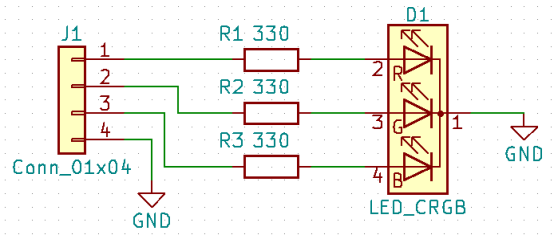


9.17 Parte Push Buttons (Analogic)

9.18 Parte 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.





9.19 Parte RTC ds1307

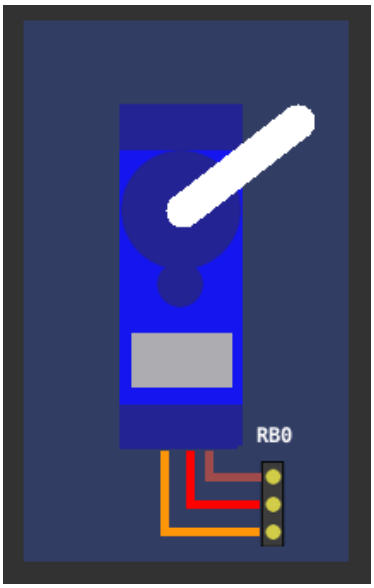
TODO

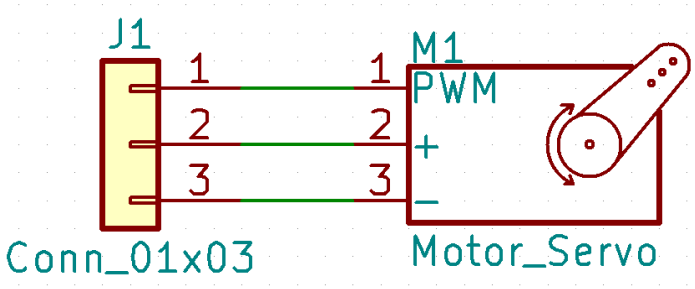
9.20 Parte RTC pfc8563

TODO

9.21 Parte 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 -90º, um de 1,5ms a 0º e um de 2ms a 90º.



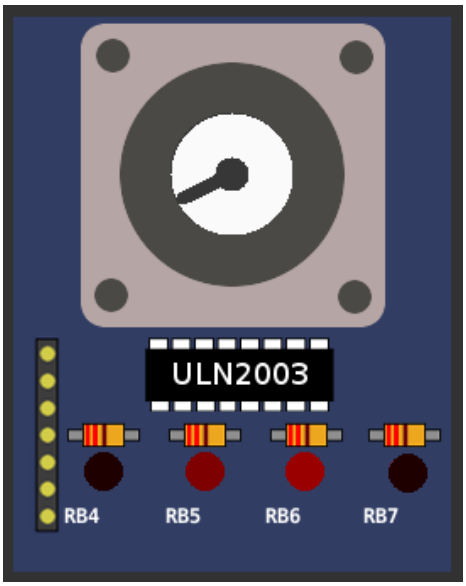


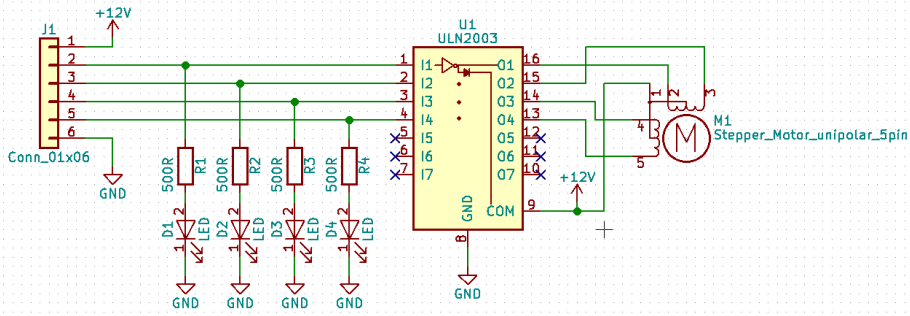
9.22 Parte Signal Generator

TODO

9.23 Parte Step Motor

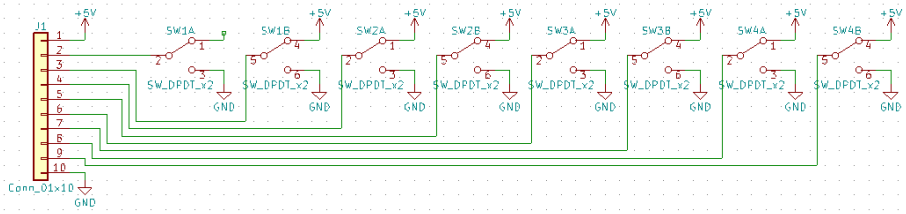
O motor de passo e um componente com 4 bobinas que devem ser acionadas na ordem correta para fazer o deslocamento do rotor. Cada passo do motor é de 1.8º.





9.24 Parte Switchs

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



9.25 Parte Temperature System

TODO

9.26 Parte VCD dump

TODO

9.27 Parte VCD dump (Analogic)

TODO

Capítulo 10

Como Fazer (How To's)

- How to use MPLABX to program and debug PICsimLab.
- How to Compile PICsimLab and Create New Boards.

Parte III

License