PicsimLab_0_8_2

Luis Claudio Gambôa Lopes <lcgamboa@yahoo.com>

Download: github

November 8, 2020

Contents

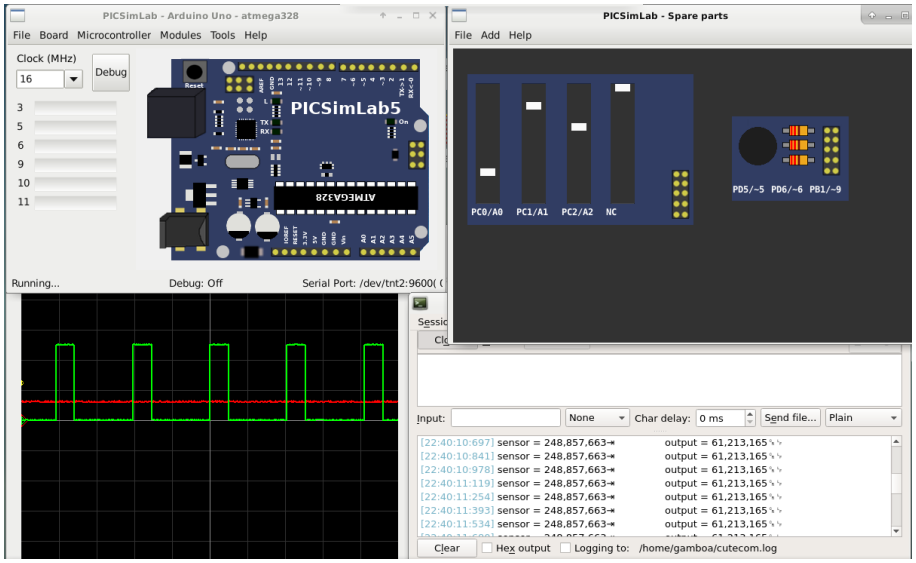
Chapter 1

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.



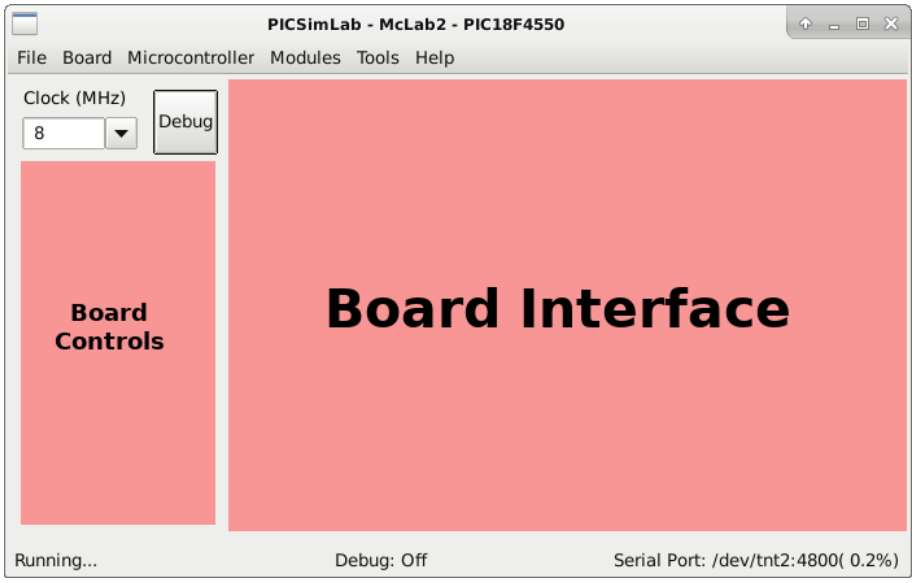
Chapter 2

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

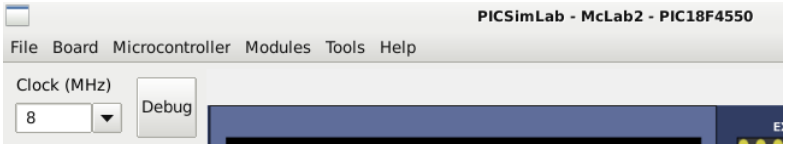
the simulation may present some difference than expected and the CPU load will be increased.

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 Interaction with the Board

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.

2.3 Command Line

PICSimLab supports two command lines format:
One for load a PICSimLab Workspace file (.pzw)

```
picsimlab file.pzw
```

And Other for load .hex files

```
picsimlab boardname microcontroller [file.hex] [file.pcf]
```

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:

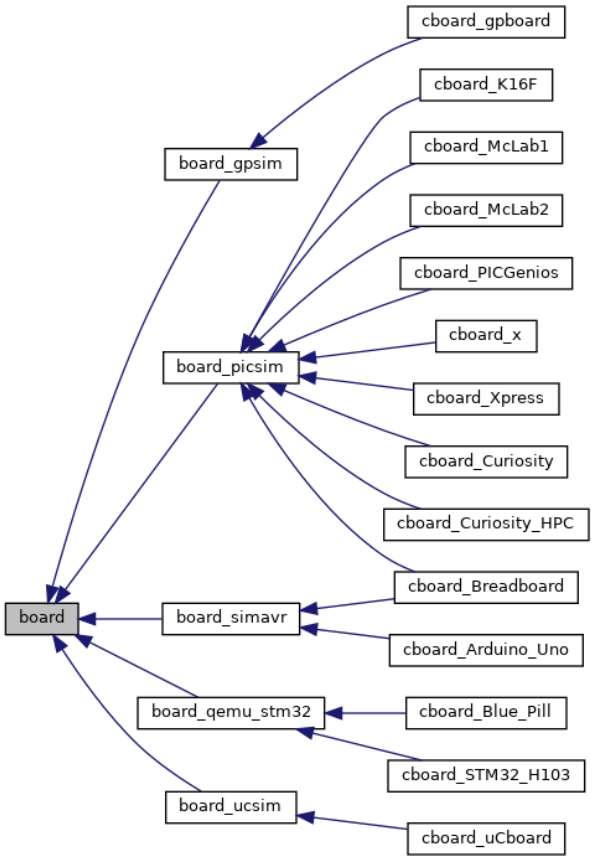


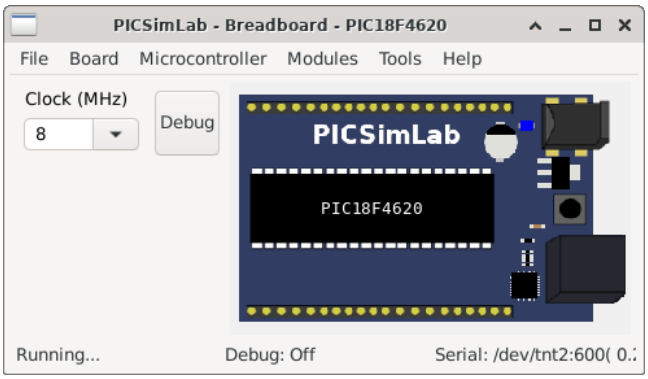
Figure 3.1: Boards backend simulators

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

Backend	Debug Support
picsim	MPLABX Integrated Debug (see section 6.1)
simavr	MPLABX Integrated Debug (see section 6.1) and remote avr-gdb (see section 6.3)
qemu-stm32	remote arm-gdb (see Chapter 6.4)
uCsim	uCsim remote console (telnet) (see section 6.5)
gpsim	none yet

3.1 Features of Board Breadboard

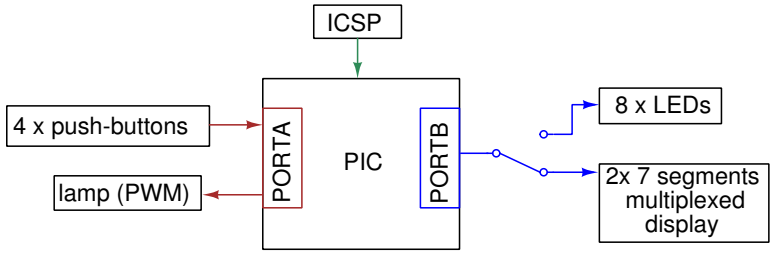
It is a generic board only with reset, serial and crystal circuits and support to multiple microcontrollers of picsim and simavr.

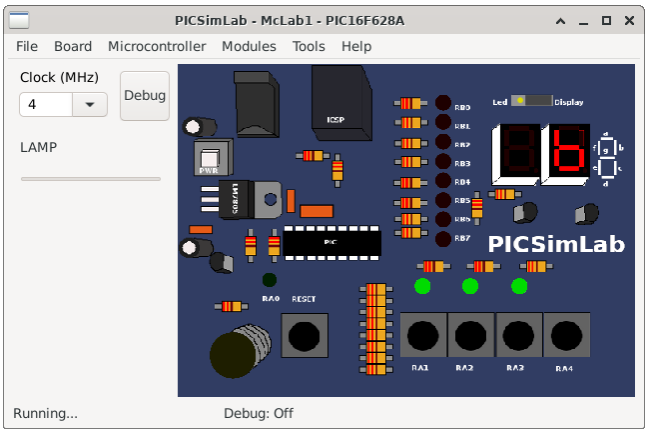


Examples

3.2 Features of Board McLab1

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





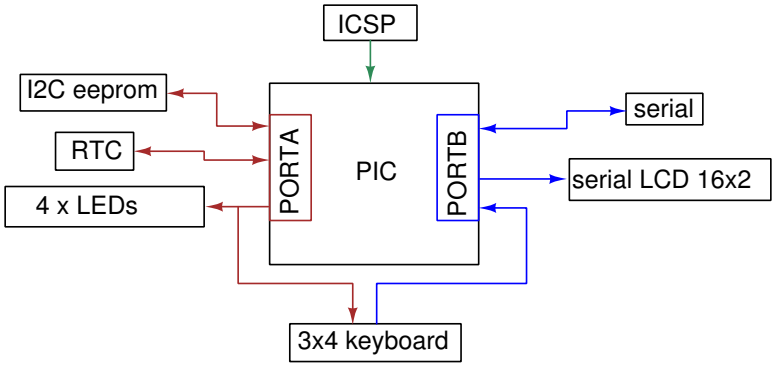
Board McLab1 schematics.

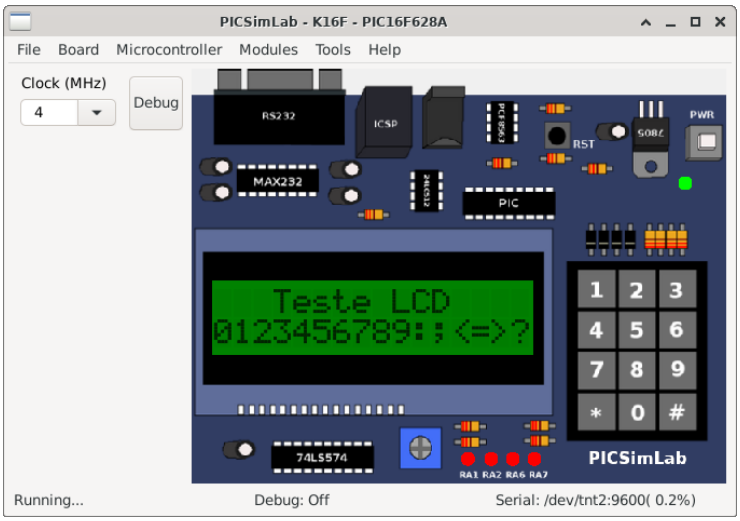
The code examples can be loaded in PICSimLab menu **Help->Examples**.

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 of picsim.





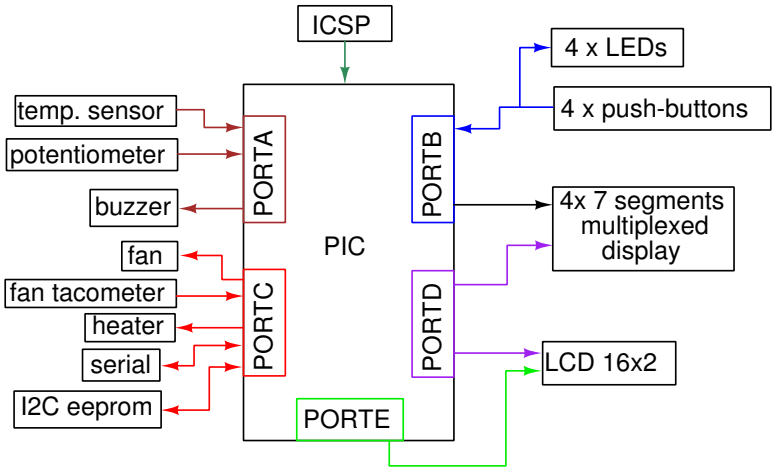
Board K16F schematics.

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 of picsim.



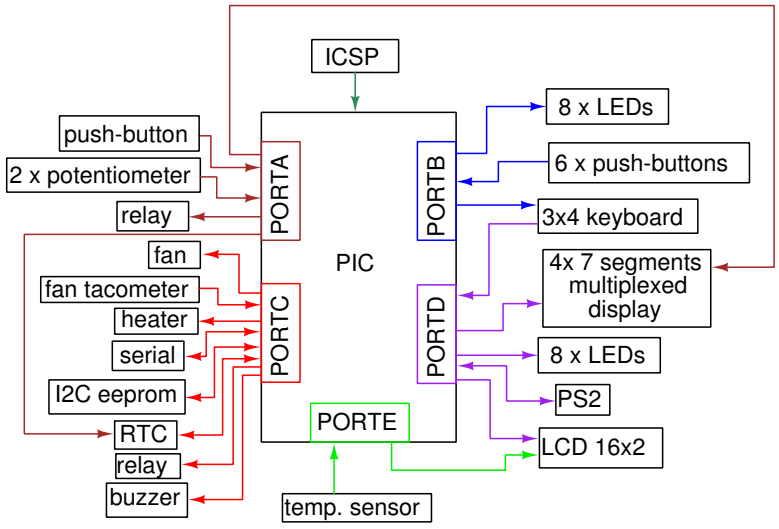
Board McLab2 schematics.

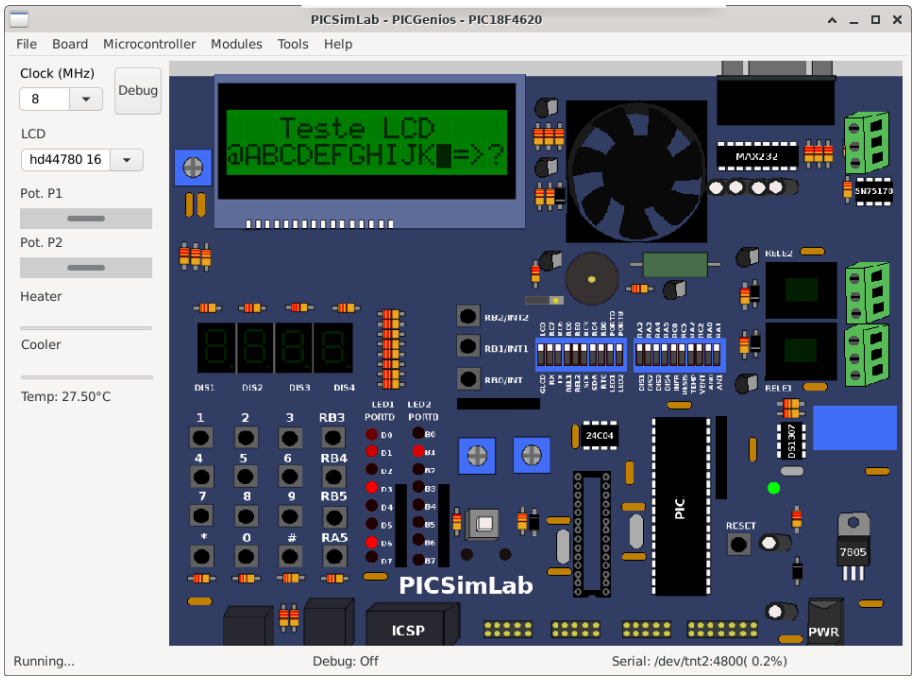
The code examples can be loaded in PICSimLab menu **Help->Examples**.

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 of picsim.





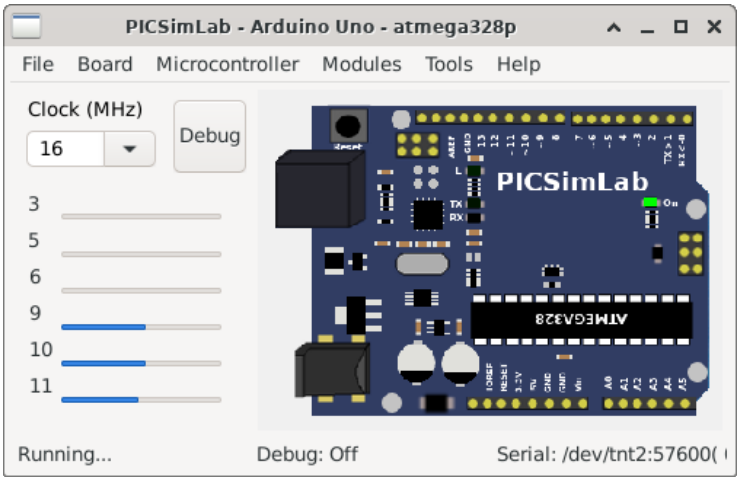
Board PICGenios schematics.

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 ATMEGA328P microcontroller of simavr.



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 information about the Arduino in www.arduino.cc

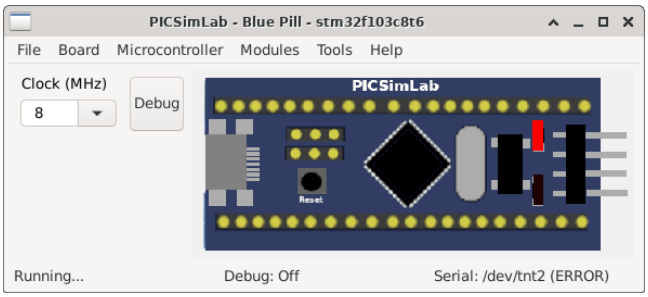
Chapter 4

Experimental Boards

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

4.1 Features of Board Blue Pill

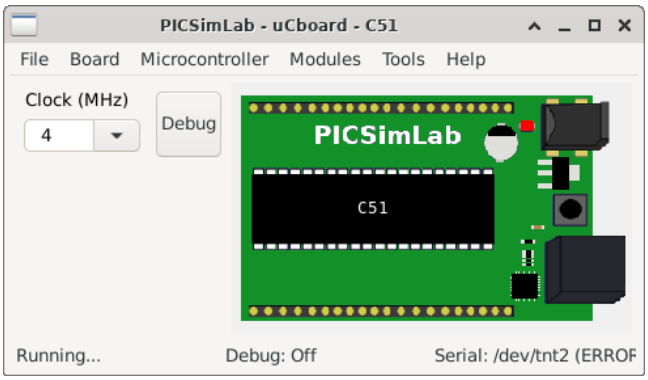
It is a generic board only with reset, serial and crystal circuits and support to stm32f103c8t6 microcontroller of qemu-stm32.



Examples

4.2 Features of Board uCboard

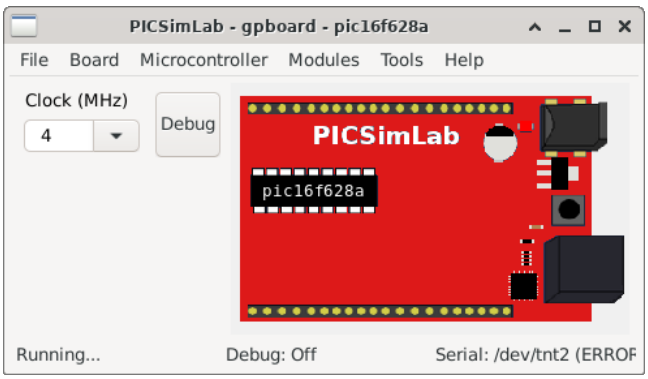
It is a generic board only with reset, serial and crystal circuits and support to multiple microcontrollers (initially C51, Z80 and STM8S103)of uCsim.



Examples

4.3 Features of Board gpboard

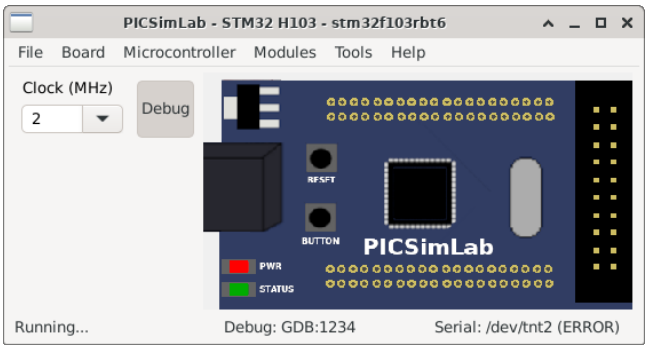
It is a generic board only with reset, serial and crystal circuits and support to multiple microcontrollers of gpsim.



Examples

4.4 Features of Board STM32 H103

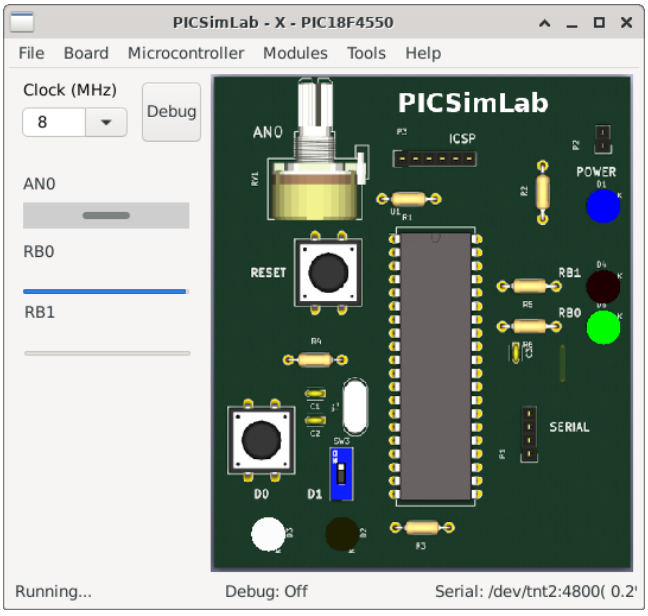
It is a generic board only with reset, one push button, serial and crystal circuits and support to stm32f103rbt6 microcontroller of qemu-stm32.



Examples

4.5 Features of Board X

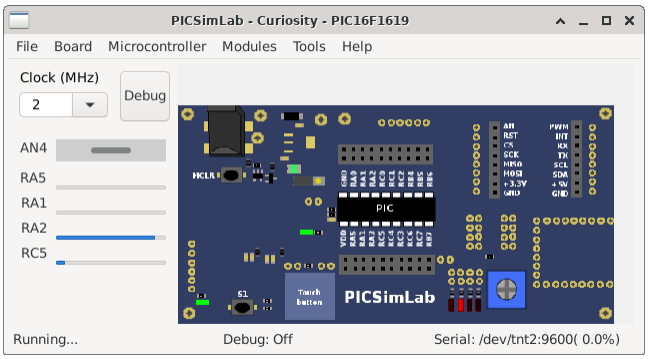
It is a generic board, used as example in How to Compile PICsimLab and Create New Boards.



Examples

4.6 Features of Board Curiosity

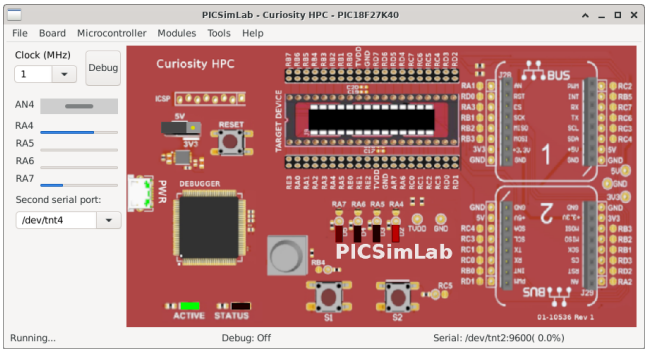
This is a simple PIC microcontroller development board that uses picsim.



Examples

4.7 Features of Board Curiosity HPC

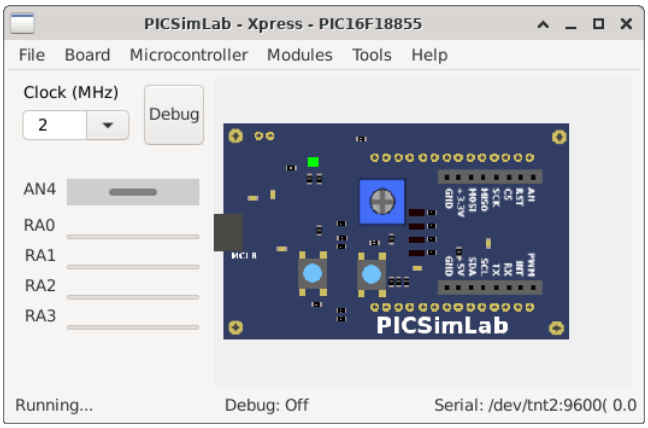
This is a simple PIC microcontroller development board that uses picsim.



Examples

4.8 Features of Board Xpress

This is a simple PIC microcontroller development board that uses picsim.


PICSimLab - Xpress - PIC16F18855
File
Board
Microcontroller
Modules
Tools
Help
Clock (MHz)
2
Debug
AN4
RA0
RA1
RA2
RA3
PICSimLab
Running...
Debug: Off
Serial: /dev/tnt2:9600(0.0

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:

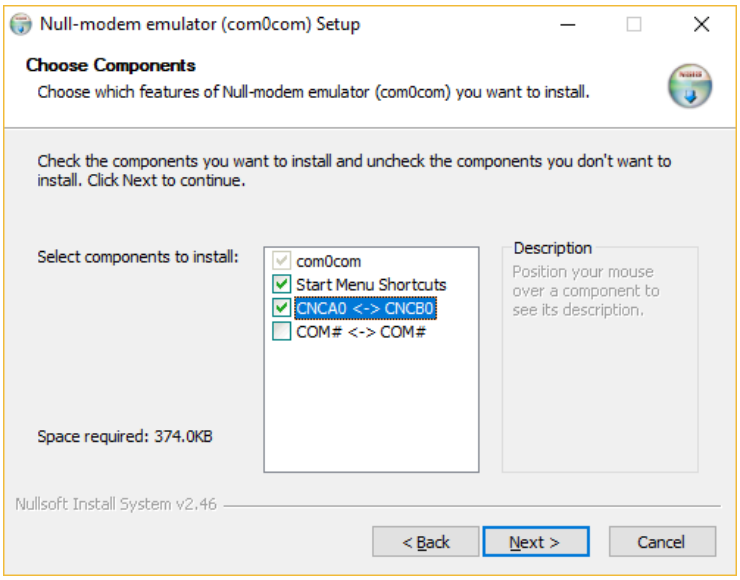
OS	PicsimLab port	Cutecom port	NULL-Modem prog.	Connection
Windows	com1	com2	com0com	com1<=>com2
Linux	/dev/tnt2	/dev/tnt3	tty0tty	/dev/tnt2<=>/dev/tnt3

5.1 Com0com Installation and Configuration(Windows)

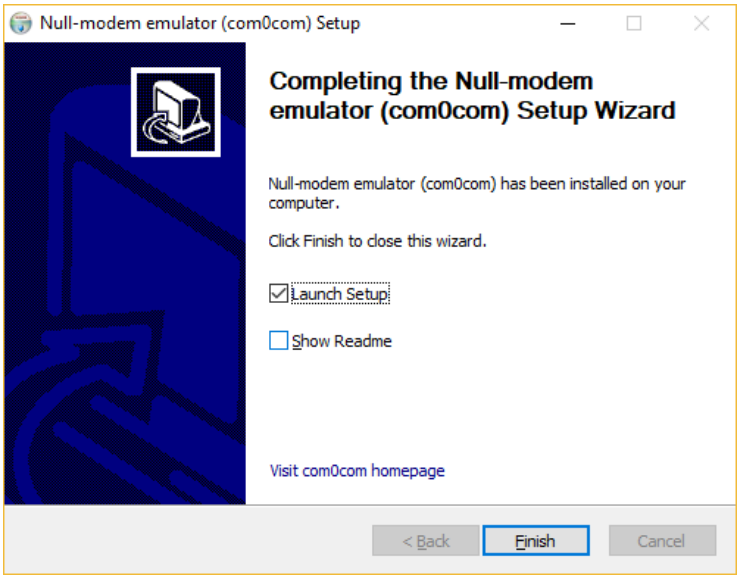
Download the signed version of com0com.

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.

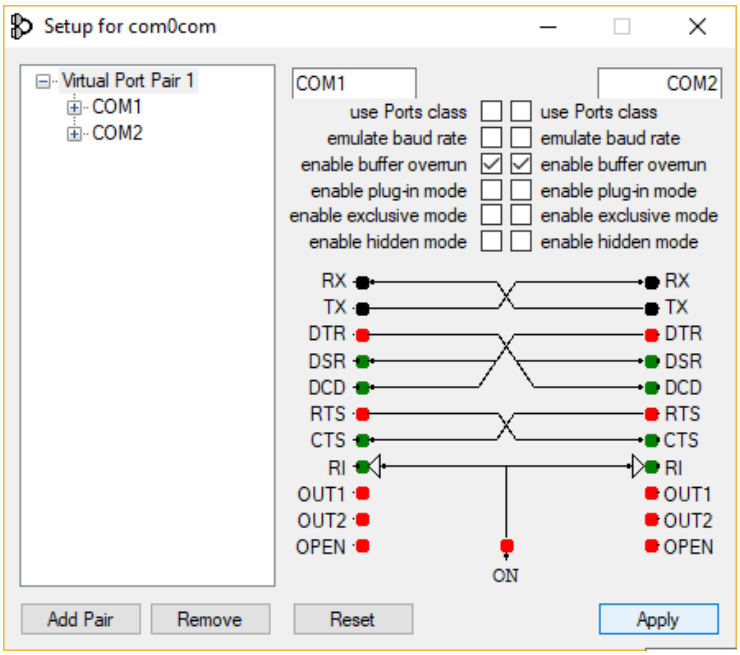
Configure the “choose components” window as the figure below:



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

Programmer and Debugger Support

The type of debug interface depends on the backend simulator utilized.

6.1 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.

6.2 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.

6.3 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.

6.4 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.

6.5 uCsim Debug

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

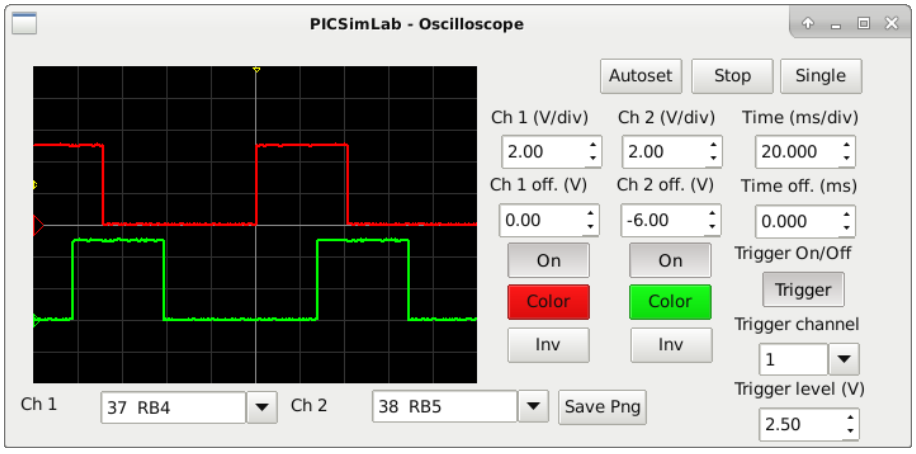
```
telnet localhost 1234
```

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

Chapter 7

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 8

Spare Parts

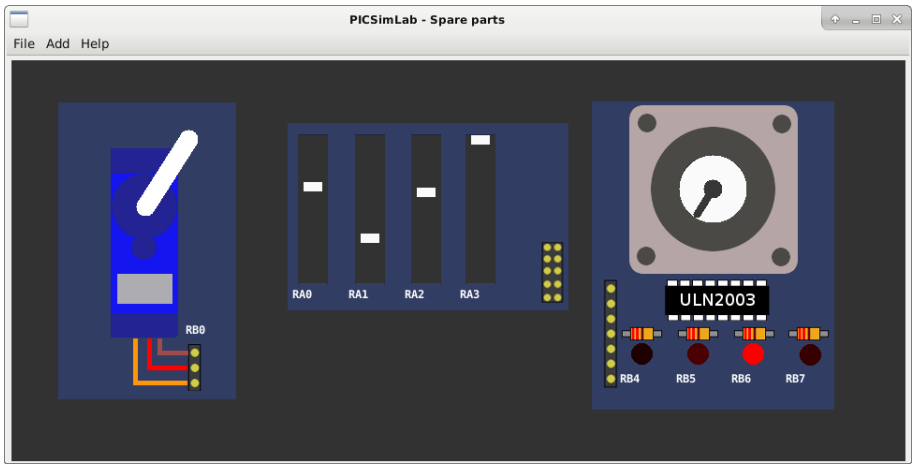
The PICSimLab has a window that allows the connection of spare parts to the microcontroller, it can be accessed through the menu " Modules-> Spare parts ".

The main window has the menu with the following functions:

- File
  - New configuration - Clear the spare parts window
  - Save configuration - Saves the current settings of the spare parts into .pcf file
  - Load configuration - Loads the settings from .pcf file
  - Save pin alias - Saves the current pin alias to .ppa text file
  - Load pin alias - Loads the pin alias from .ppa file
- Edit
  - Clear pin alias - Clear the pin alias
  - Toggle pin alias - Enable/Disable pin alias use
  - Edit pin alias - Open current pin alias .ppa file in text editor
  - Reload pin alias - Reload the current .ppa pin alias file (need after edit .ppa file)
  - Zoom in - Increase draw scale
  - Zoom out - Decrease draw scale
- Add
  - 7 Segments Display - Adds four multiplexed 7 segments displays
  - Buzzer - Adds a active/passive buzzer
  - D. Transfer Function - Adds a discrete transfer function mathematical model
  - ETH w5500 - Adds a ethernet shield w5500

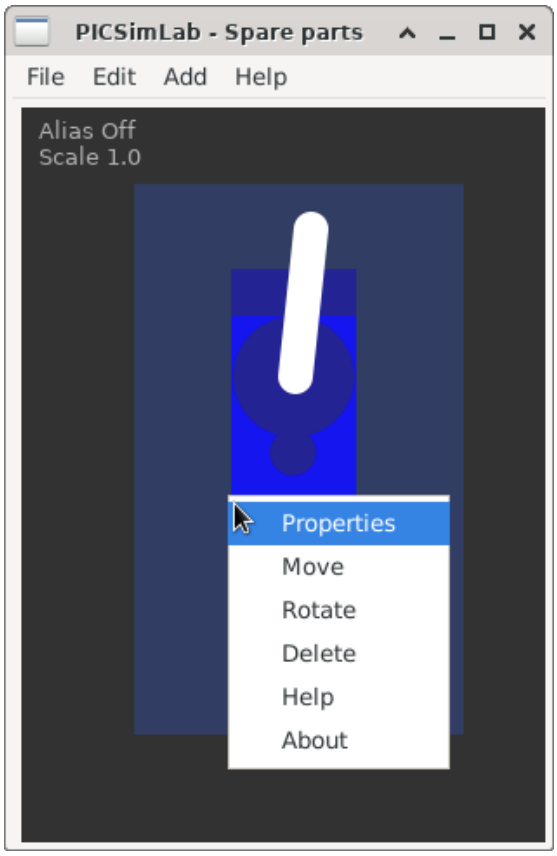
  - Gamepad - Adds a gamepad
  - Gamepad (Analogic) - Adds a gamepad with one analogic output
  - IO 74xx595 - Adds one 74xx595 SIPO 8 bit shift register
  - IO MCP23S17 - Adds one MCP23S17 serial SPI IO expander
  - IO PCF8574 - Adds one PCF8574 serial I2C IO expander
  - IO UART - Adds a UART serial port
  - Jumper Wires - Adds sixteen jumper wires
  - Keypad - Adds one matrix keypad
  - LCD hd44780 - Adds a text display hd44780
  - LCD ili9340 - Adds a color graphic display ili9340
  - LCD pcd8544 - Adds a monochrome graphic display pcd8544 (Nokia 5110)
  - LCD pcf8833 - Adds a color graphic display pcf8833
  - LCD ssd1306 - Adds a monochrome graphic display ssd1306
  - LED Matrix - Adds a 8x8 LED matrix with MAX72xx controller
  - LEDs - Adds 8 red LEDs
  - MEM 24CXXX - Adds a 24CXXX serial I2C EEPROM memory
  - Potentiometers - Adds 4 potentiometers
  - Push Buttons - Adds 8 push buttons
  - Push Buttons (Analogic) - Adds 8 push buttons with analog output
  - RGB LED - Adds one RGB LED
  - RTC ds1307 - Adds a ds1307 real time clock
  - RTC pfc8563 - Adds a pfc8563 real time clock
  - SD Card - Adds a SD card shield
  - Servo Motor - Adds a servo motor
  - Signal Generator - Adds a virtual signal generator
  - Step Motor - Adds a step motor
  - Switchs - Adds eight switchs
  - Temperature System - Adds a temperature control system
  - VCD Dump - Adds a digital value file dump recorder
  - VCD Dump (Analogic) - Adds a analog value file dump recorder
  - VCD Play - Adds a digital value file dump player

- Help
  - Contents - Open Help window
  - About - Show message about author and version



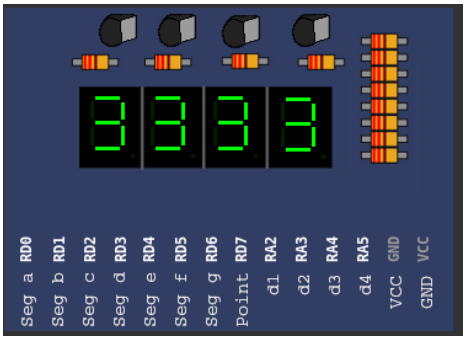
After adding the part, with a right click of the mouse you can access the options menu of the part with the options:

- Properties - Opens the connection settings window
- Move - Unlocks the part to move
- Rotate - Change the orientation of part
- Delete - Remove part
- Help - Open Help window of part
- About - Show message about author and version of part



8.1 7 Segments Display

This is a four multiplexed 7 segments displays.



Examples

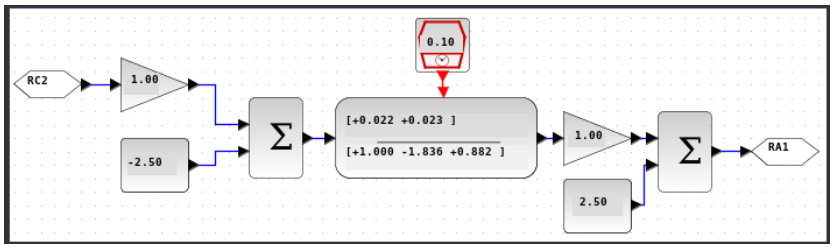
8.2 Buzzer

This is a active/passive buzzer.

Examples

8.3 D. Transfer Function

This is a discrete transfer function mathematical model.



Examples

8.4 ETH w5500

This part is a ethernet shield w5500 with support to 8 sockets simultaneously.

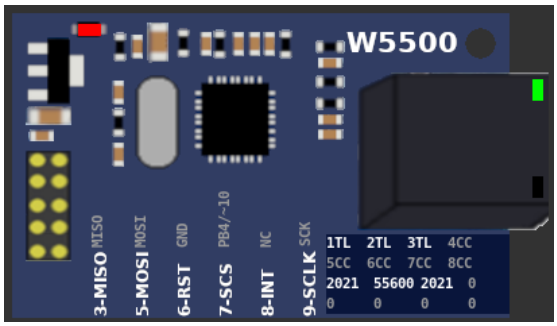
Only TCP/UDP unicast address sockets is supported. DHCP is emulated and return a fake ipv4 address.

All listening ports below 2000 are increased by 2000 to avoid operational system services ports. For example listening on port 80 becomes 2080.

w5500 Status Legend:

1º Letter - Type	2º Letter - Status	3º Letter - Error
C - Closed	C - Closed	B - Bind
T - TCP	I - Initialized	S - Send
U - UDP	L - Listen	R - Receive
M - MACRAW (don't supported)	S - Syn sent	L - Listen
	E - Established	U - Reuse
	W - Close wait	C - Connecting
	U - UDP	D - Shutdown
	M - MACRAW (don't supported)	

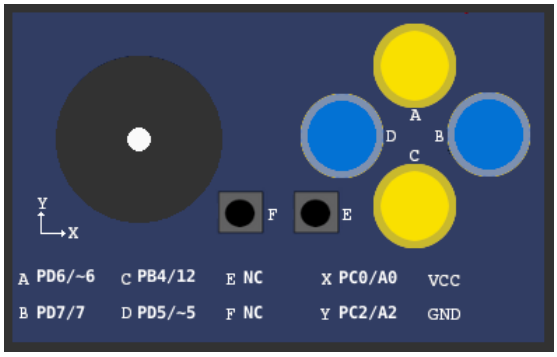
Click on connector to toggle link status.

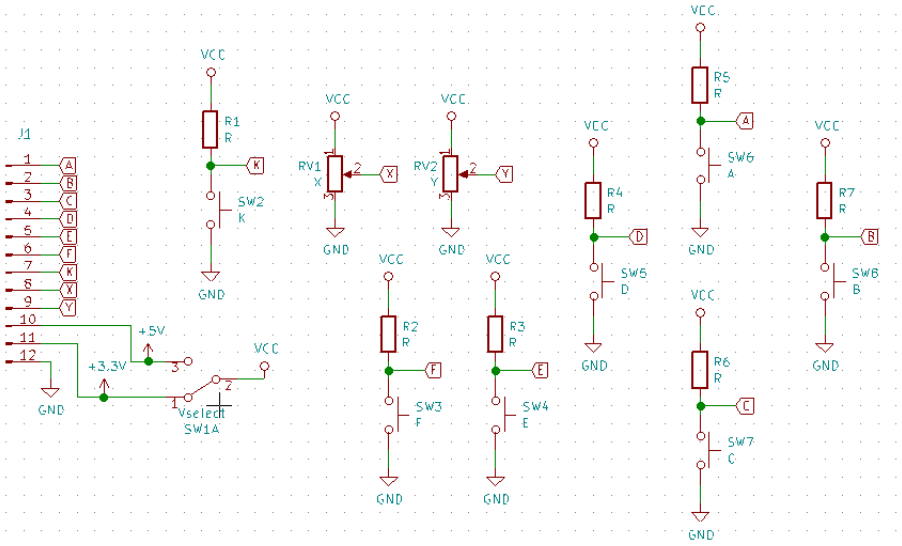


Examples

8.5 Gamepad

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





The gamepad can be controlled by keyboards keys:

- X axis - keys 'A' and 'D'
- Y axis - keys 'W' and 'S'
- Button A - key 'I'
- Button B - key 'L'
- Button C - key 'K'
- Button D - key 'J'
- Button E - key 'E'
- Button F - key 'O'
- Button K - key 'R'

Examples

8.6 Gamepad Analogic

This part is a gamepad with 5 push buttons and one analogic output.

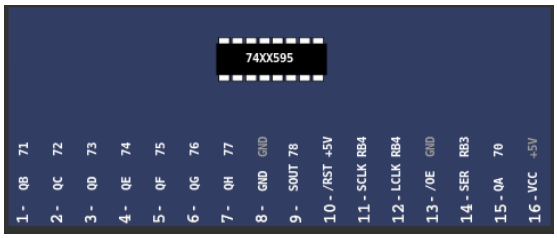
The gamepad can be controlled by keyboards keys:

- Button A - key 'L'
- Button B - key 'I'
- Button C - key 'K'
- Button D - key 'J'
- Button E - key 'O'

Examples

8.7 IO 74xx595

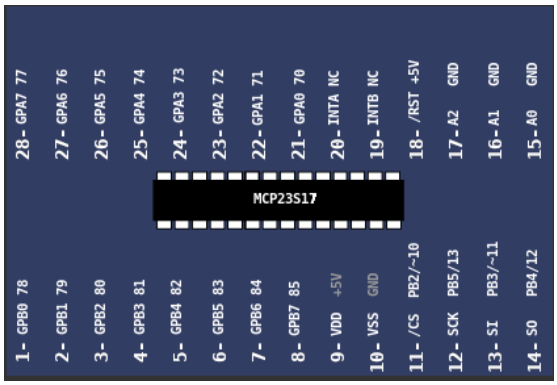
This is one 74xx595 serial input and parallel output 8 bit shift register.



Examples

8.8 IO MCP23S17

It is a MCP23S17 serial SPI IO expander part.



Examples

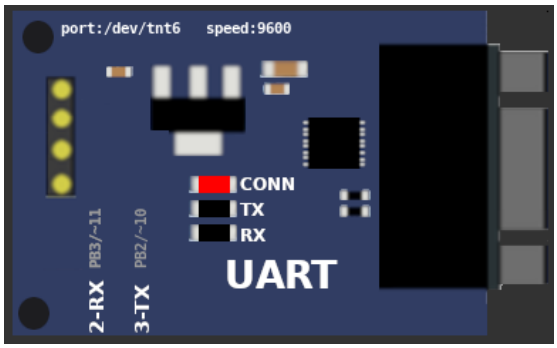
8.9 IO PCF8574

It is a PCF8574 serial I2C IO expander.

Examples

8.10 IO UART

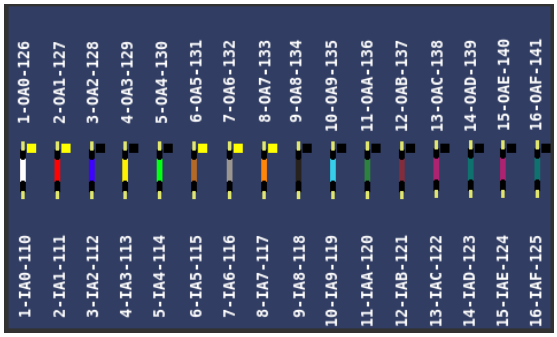
This part is a UART serial port. This part connects the hardware/software UART IO pins of microcontroller to one real/virtual PC serial port.



Examples

8.11 Jumper Wires

This part are formed by sixteen jumper wires. Each jumper has one input and one output. The jumper input must be connected to one pin output, the jumper output can be connected to multiple pin inputs. The jumper can be used to connect microcontroller pins or make connection between spare parts pins.



Examples

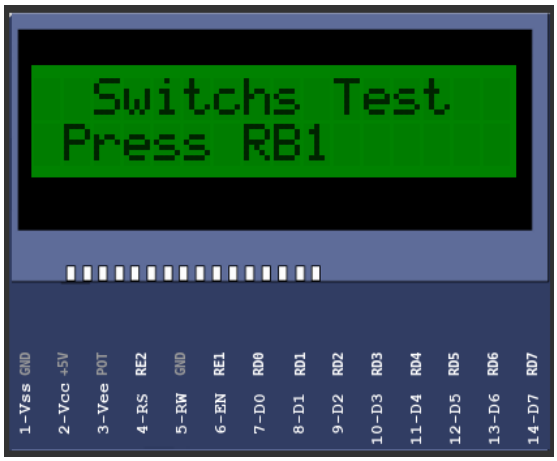
8.12 Keypad

It is a matrix keyboard configurable to 4x3 , 4x4 or 2x5 rows/columns.

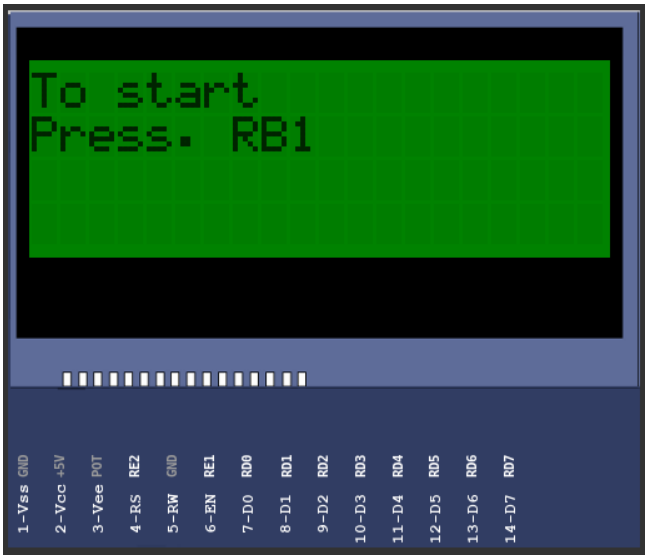
Examples

8.13 LCD hd44780

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


Switchs Test
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

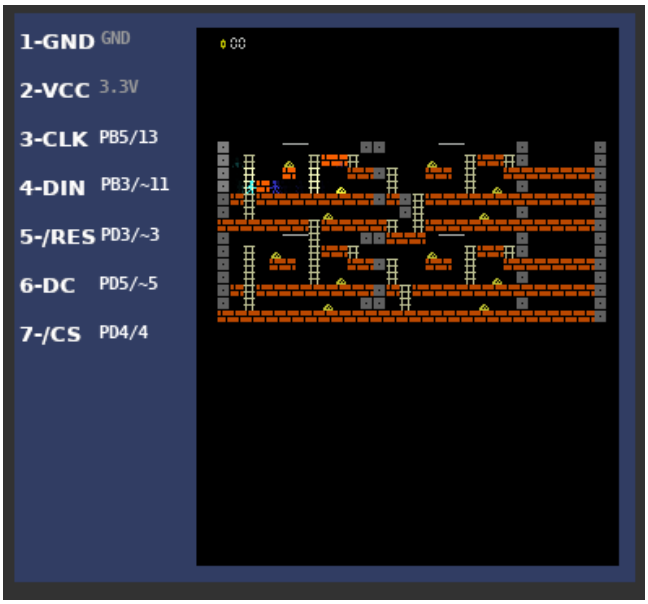

To start
Press. RB1
1-Vss GND
2-Vcc +5V
3-Vee POT
4-RS RE2
5-RW GND
6-EN RE1
7-D0 RD0
8-D1 RD1
9-D2 RD2
10-D3 RD3
11-D4 RD4
12-D5 RD5
13-D6 RD6
14-D7 RD7



Examples

8.14 LCD ili9341

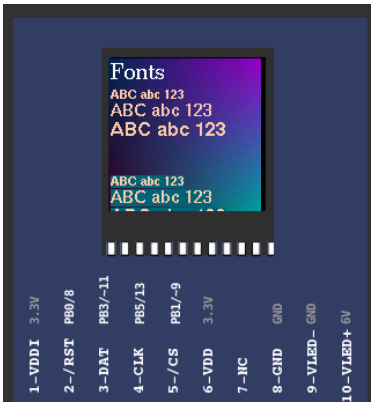
This part is a color graphic display with 240x320 pixels. Only 4 SPI mode is avaliable.



Examples

8.15 LCD pcf8833

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



Examples

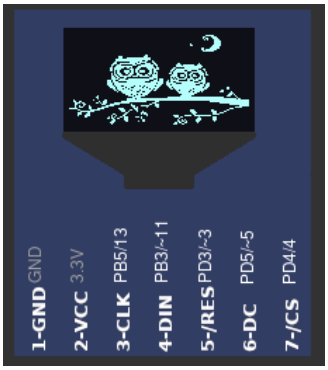
8.16 LCD pcd8544

This part is a monochrome graphic display with 48x84 pixels. (Nokia 5110)

Examples

8.17 LCD ssd1306

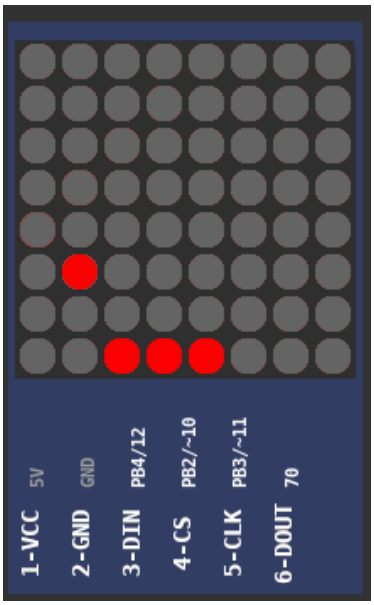
This part is a monochrome oled graphic display with 128x64 pixels. The part suport I2C and 4 SPI serial mode.



Examples

8.18 LED Matrix

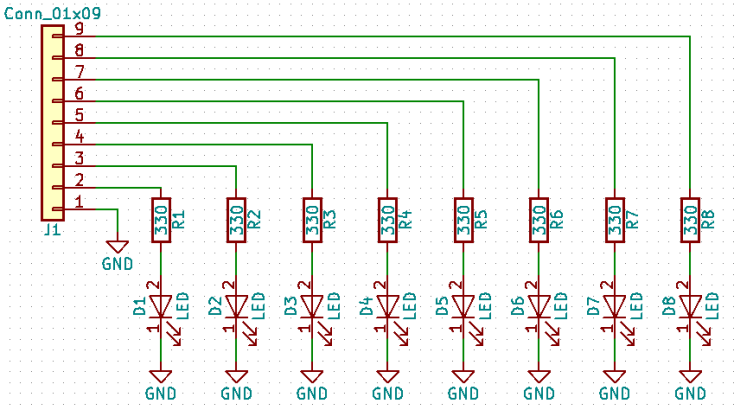
It is a 8x8 LED matrix with MAX72xx controller.



Examples

8.19 LEDs

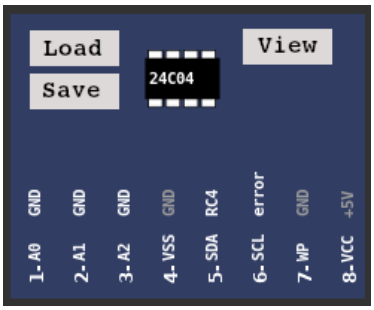
This part is a bar of 8 independent red LEDs.



Examples

8.20 MEM 24CXXX

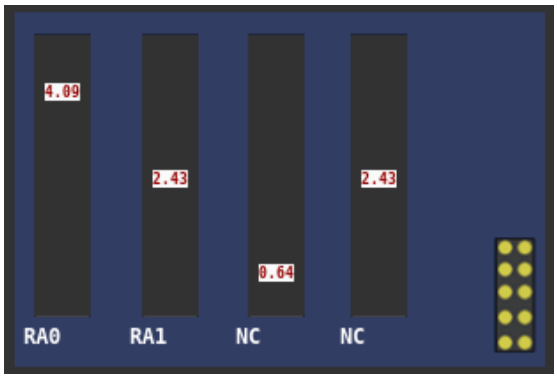
It is a 24CXXX serial I2C EEPROM part. There are support to the models 24C04 and 24C512.

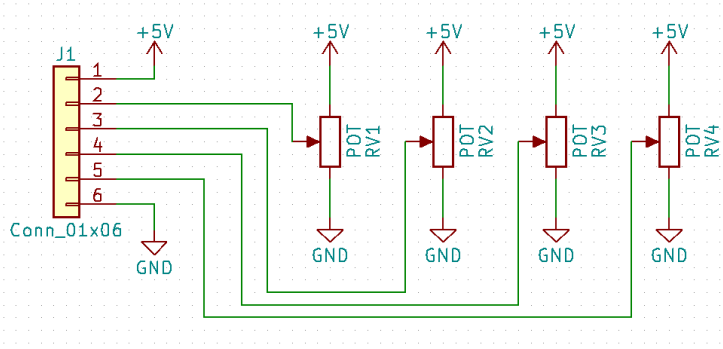


Examples

8.21 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.

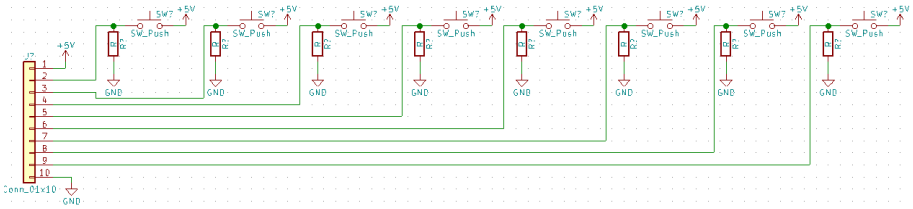




Examples

8.22 Push Buttons

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



Examples

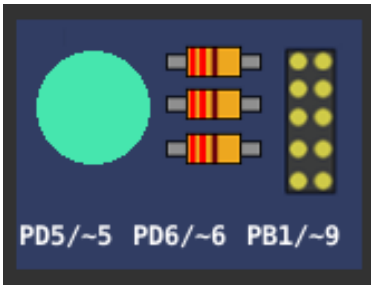
8.23 Push Buttons (Analogic)

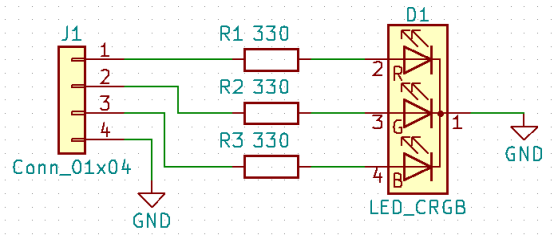
This part consists of 8 push buttons connected in a resistive ladder.

Examples

8.24 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.

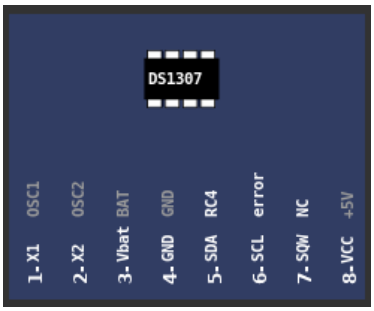




Examples

8.25 RTC ds1307

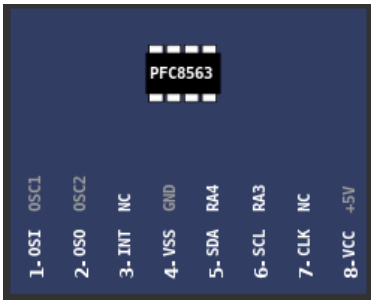
This part is a ds1307 real time clock with serial I2C interface.



Examples

8.26 RTC pfc8563

This part is a pfc8563 real time clock with serial I2C interface.



Examples

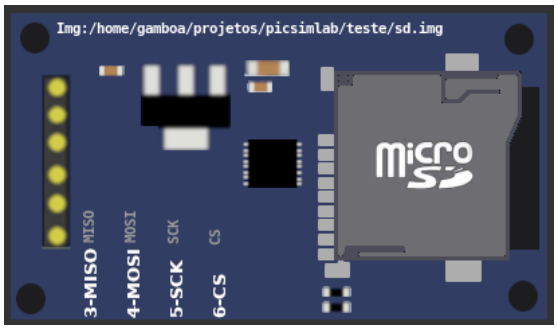
8.27 SD Card

This part is a SD Card shield. It's necessary set one sd card file image before use it. (Click on SD card connector to open file dialog)

On Linux one empty image can be created with this command:

```
dd if=/dev/zero of=sd.img bs=1M count=32
```

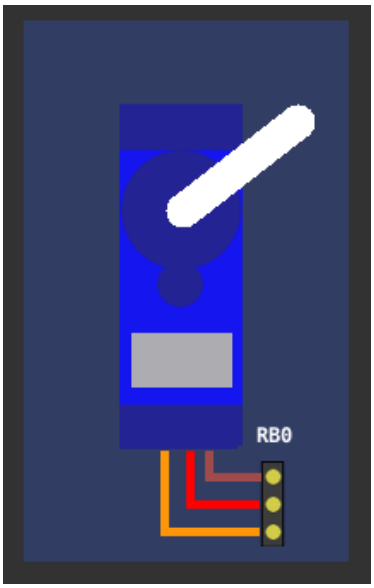
This empty image can be used with raw sd card access, to work with FAT file system the image need to be formatted before the use. (using SdFormatter.ino for example)

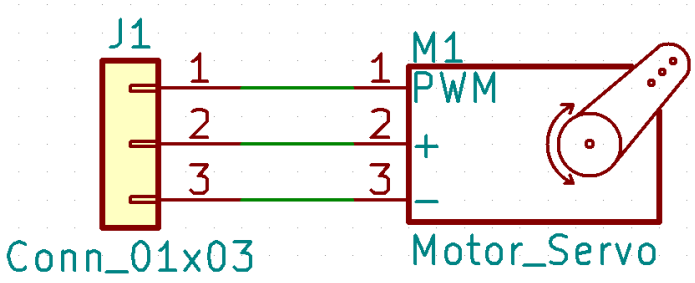


Examples

8.28 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.

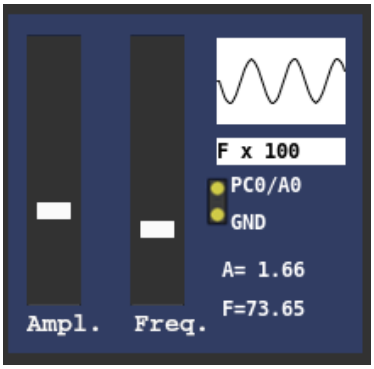




Examples

8.29 Signal Generator

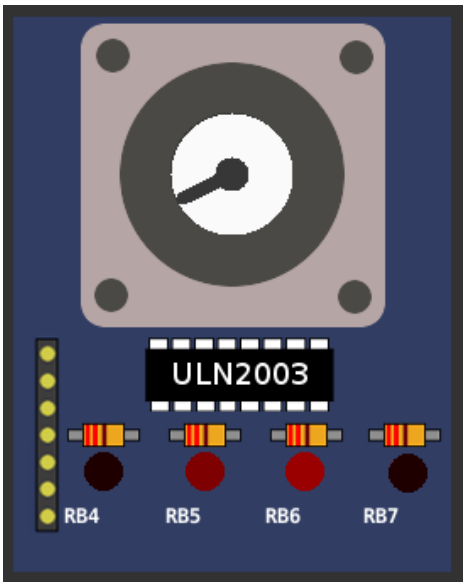
This part is a virtual signal generator with support for sine, square and triangular waves generation with amplitude and frequency adjustment.

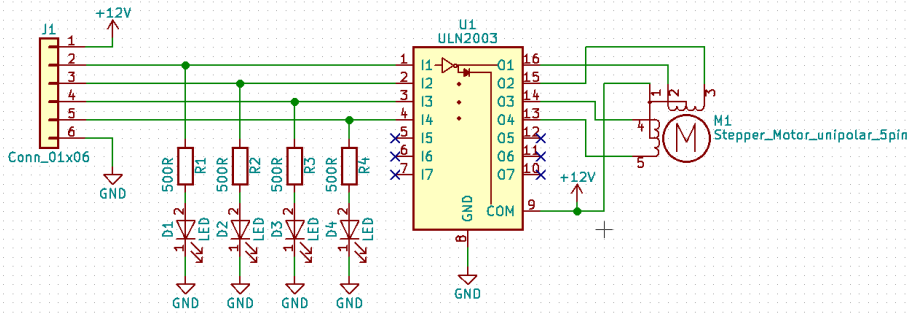


Examples

8.30 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.

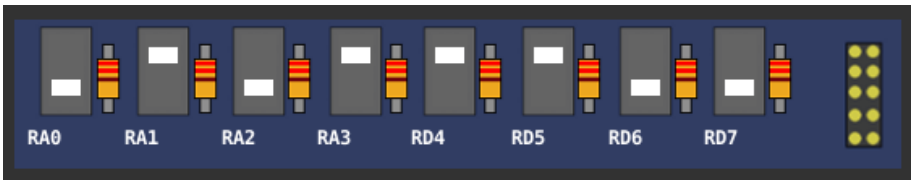


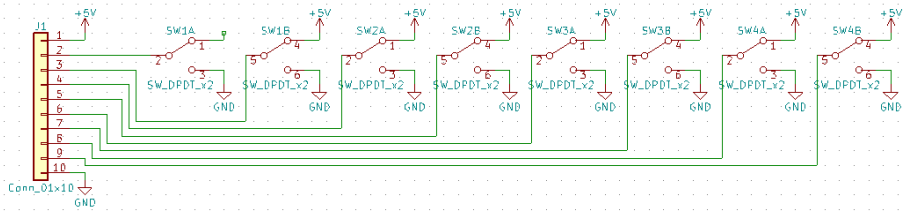


Examples

8.31 Switchs

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

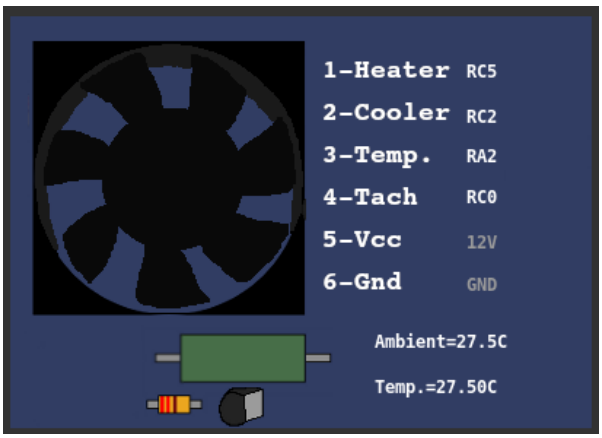




Examples

8.32 Temperature System

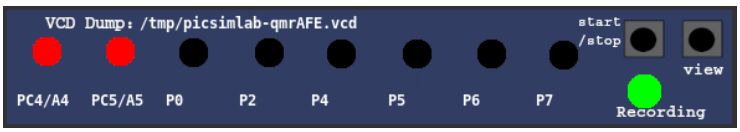
This part is a temperature control system. The temperature control system consists of a heating resistor, an LM35 temperature sensor, a cooler and an infrared tachometer.

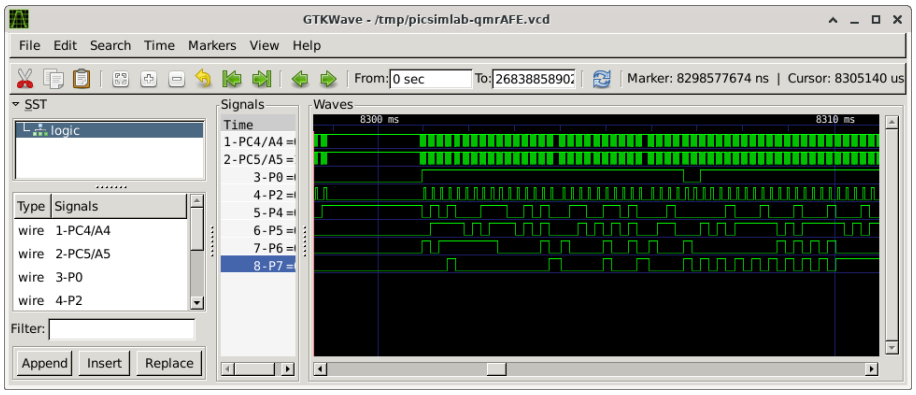


Examples

8.33 VCD dump

This part is a digital value file dump recorder. The file can be visualized with gtkwave.

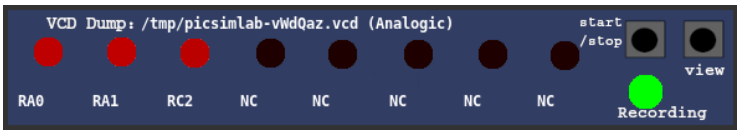


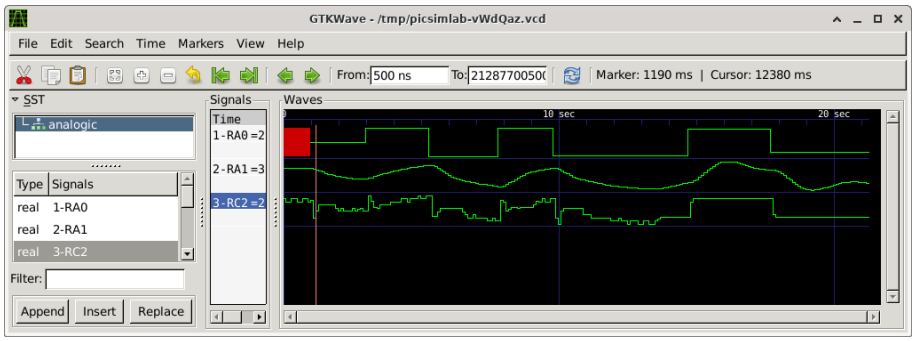


Examples

8.34 VCD dump (Analogic)

This part is a analog value file dump recorder. The file can be visualized with gtkwave.





Examples

8.35 VCD Play

This part play a VCD file saved from VCD Dump part.


VCD Play : /tmp/picsimlab-3x3Wp6.vcd
PC4/A4
PC5/A5
PB7/X2
PD6/~6
PB1/~9
PB2/~10
PB3/~11
PB4/12
Load
View
Play

Examples

Chapter 9

How To's

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

Chapter 10

License