picoLCD 256x64 SideShow

from Mini-Box.com

- USER MANUAL -

Revision 1.2

CONTENT

1. Hardware installation......................4
2. Windows Vista driver installation....6
3. Linux driver installation................10
3.1 Driver installation..10
3.2 Customizing your picoLCD display with plugins and widgets 12
3.3 Customizing your picoLCD display with external python scripts.....................13
3.4 Controlling the display with the keypad keys.. 13
4. Appendixes.................................14

Welcome!

Thank you for purchasing the picoLCD 256x64 from mini-box.com !

Before you start installing the picoLCD 256x64,
take note of the following precautions:

-

- Unplug the power cord from the wall socket before installing the picoLCD256x64.

- Hold the picoLCD by the edges to avoid touching the display or the integrated circuits on it.

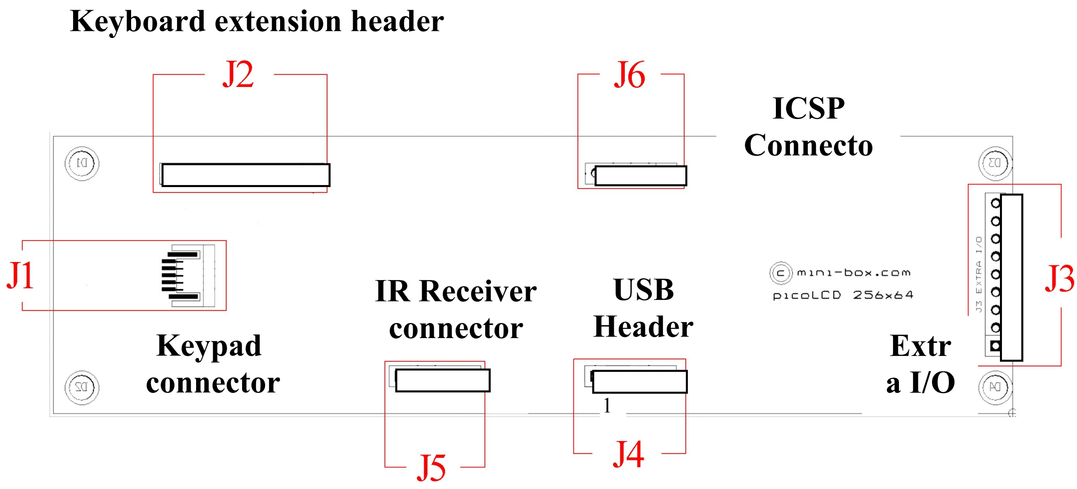
- Before you install or remove any component, ensure that the ATX power supply is switched off or the power cord is detached from the power supply.

- Failure to do so may cause severe damage to the components, peripherals or mainboard.

1. Hardware installation

Depending on the picoLCD 256x64 package that you purchase, picoLCD 256x64 comes with either an external or internal USB connector cable.

Locate the USB header (**J4**) onto the back of the PicoLCD 256x64, see the picture below:



J1: Keypad connector

J2: Keyboard and keyboard extension header connector

Pin 1 – Column 1
Pin 2 – Column 2
Pin 3 – Column 3
Pin 4 – Column 4
Pin 5 – Row 4
Pin 6 – Row 3
Pin 7 – Row 2
Pin 8 – Key (not connected)
Pin 9 – Row 1

J3: Extra I/O

Pin 1 – RB0
Pin 2 – RB1
Pin 3 – RC1
Pin 4 – RC6
Pin 5 – RC7
Pin 6 – RA0
Pin 7 – RA1
Pin 8 – Key (not connected)
Pin 9 – RA2

J4: USB connector

Pin 1 – USB V+
Pin 2 – USB DPin
3 – USB D+
Pin 4 – USB GND
Pin 5 – Shield (USB GND)

J5: IR Receiver

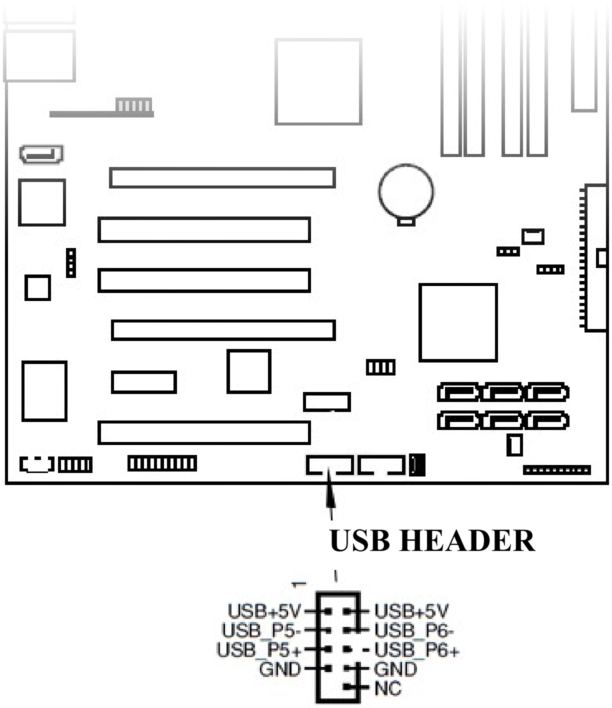
Pin 1 – GND
Pin 2– GND
Pin 3 – VCC (5V)
Pin 4 – Key (not connected)
Pin 5 – IR Signal

J6: ICSP connector

Pin 1 – MCLR
Pin 2 – VCC (5V)
Pin 3 – GND
Pin 4 – Data
Pin 5 – Clock

Please see below the situations that you might encounter:

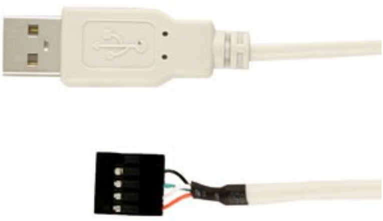
A) The picture below represents a motherboard. Please check the your motherboard's manual and locate a free USB header, like in the picture below:



The connection between the picoLCD256x64 and the motherboard will be done using a USB 5 pin to 5 pin cable.

http://www.mini-box.com/USB-5-pin-to-5-pin-cable

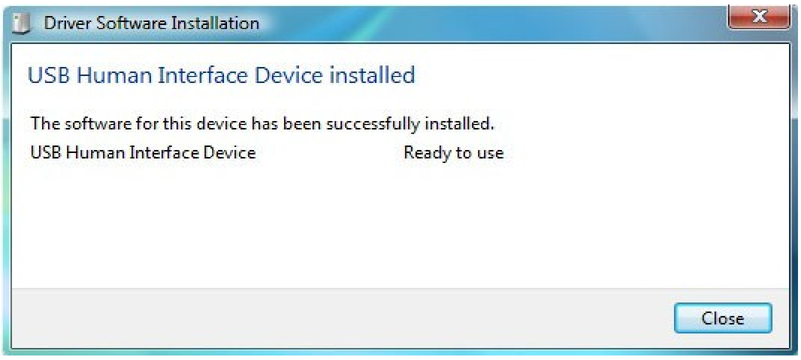
B) Insert the USB cable into a free USB port from your computer.



http://www.mini-box.com/CAB-USB-5pin-typeA

2. Windows Vista driver installation

Windows Vista will automatically detect your device and install the HID drivers.



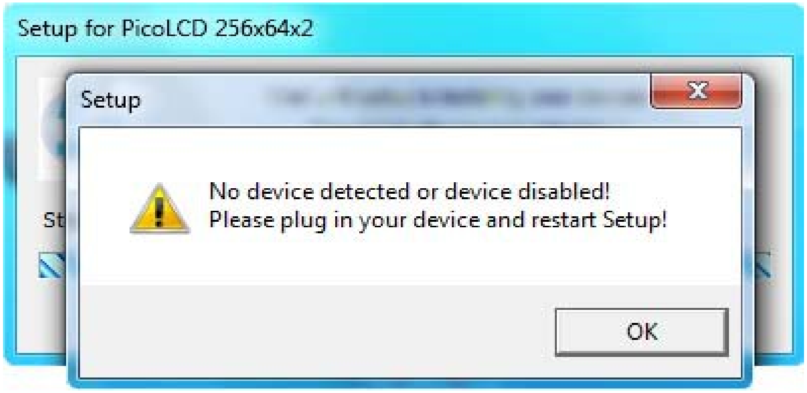
USB Human Interface Device installed

The software for this device has been successfully installed.

USB Human Interface Device   Ready to use

Download the Windows Vista driver from our webpage at:
http://www.picolcd.com/drivers/
Run PicoLCD256x64_1.0.0.4.exe

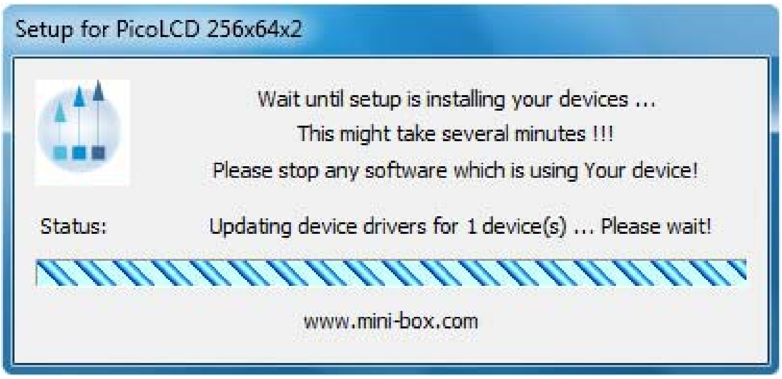
In case that you have forgotten to plug in the USB cable,
an error message will occur, like in the picture below.



No device detected or device disabled!
Please plug in your device and restart Setup!

If the PicoLCD is plugged into an USB slot,
the driver installation will begin.

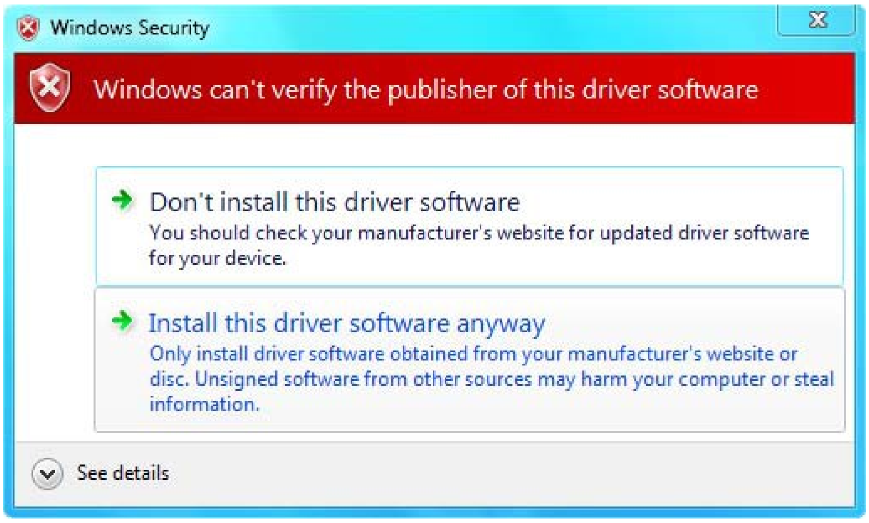
Wait until the setup will install the device driver.



Wait until setup is installing your devices ...
This might take several minutes !!!
Please stop any software which is using Your device!

Status: Updating device drivers for 1 device(s) ... Please wait!

Windows Vista will verify the publisher of the driver software. Accept and install it.



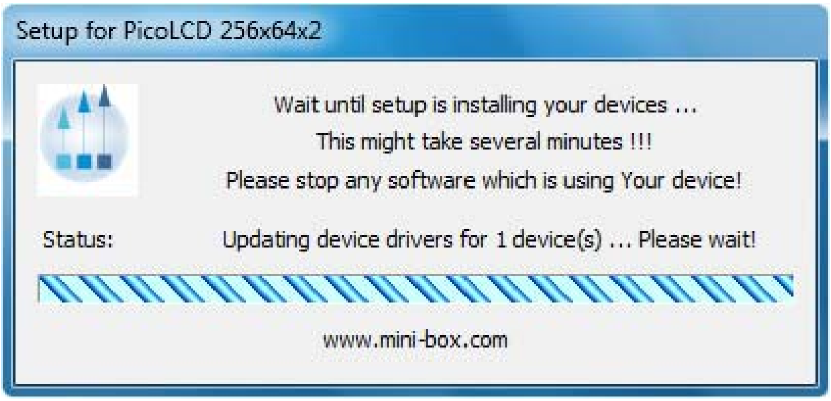
Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Wait until the setup will install the device driver.



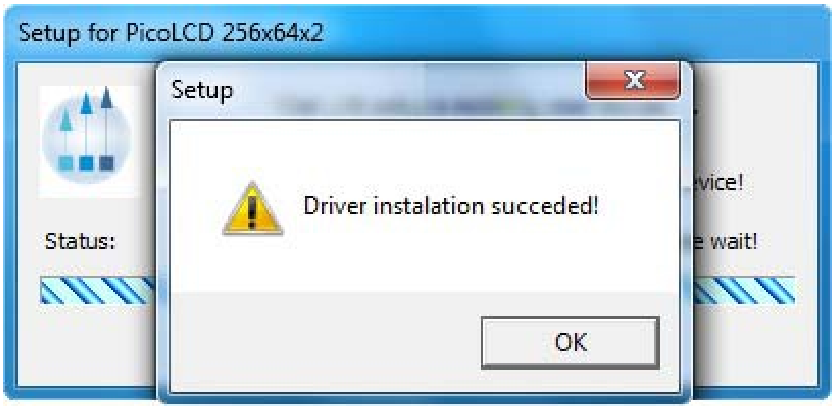
Setup for PicoLCD 256x64x2

Wait until setup is installing your devices ...
This might take several minutes !!!
Please stop any software which is using Your device!

Status: Updating device drivers for 1 device(s) ... Please wait!

www.mini-box.com

Now you have successfully installed the drivers for the PicoLCD 256x64. Press OK.

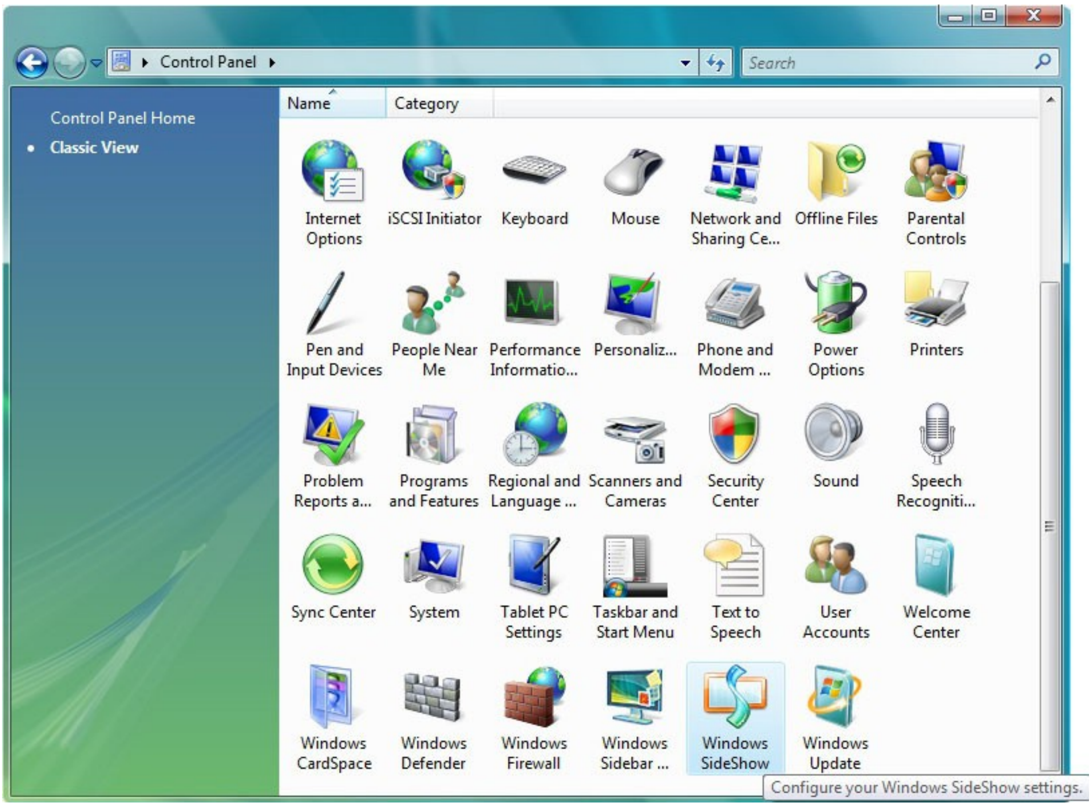


Setup for PicoLCD 256x64x2

Setup

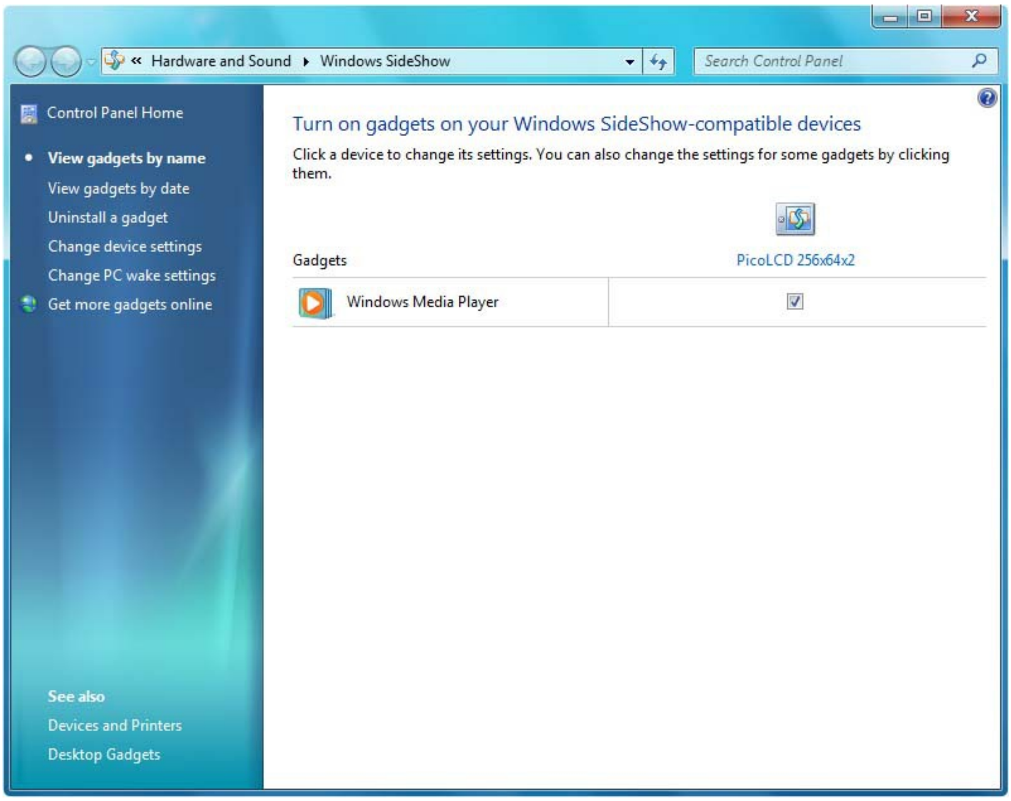
Driver instalation succeded!

Access the Control Panel and open Windows SideShow.
This will let you to configure your Windows SideShow settings.



Configure your Windows SideShow settings.

By default, Windows Vista will display one gadget. If you want more gadgets, please click “Get more gadgets online”

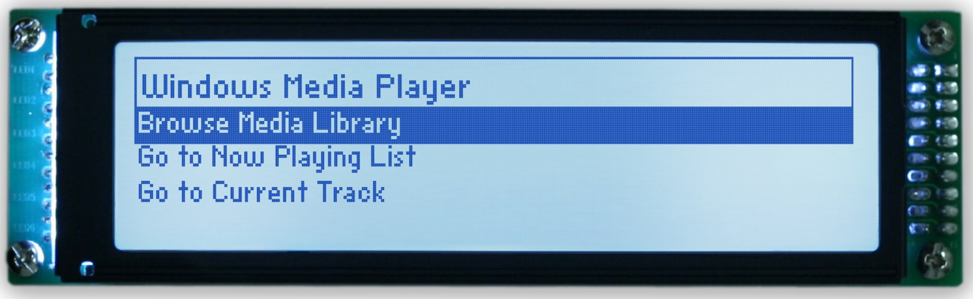


Turn on gadgets on your Windows SideShow-compatible devices

Click a device to change its settings. You can also change the settings for some gadgets by clicking them.

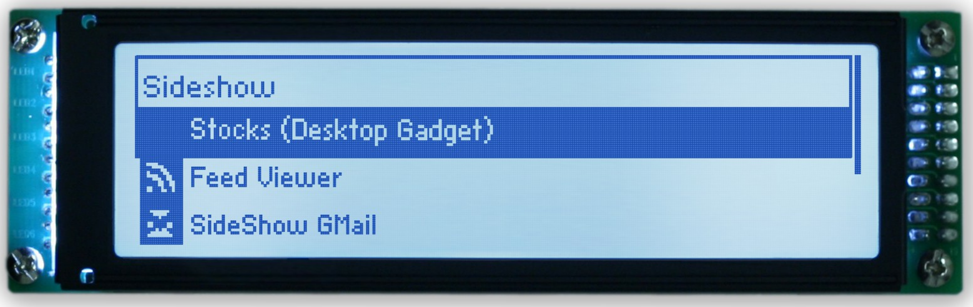
Gadgets	PicoLCD 256x64x2
Windows Media Player	☑

For example select “Windows Media Player” gadget and on your picoLCD256X64 it will show what the status of your Windows Media Player program, like in the pictures below:



Windows Media Player
Browse Media Library
Go to Now Playing List
Go to Current Track

Picture showing Sideshow menu chooser:



Sideshow
Stocks (Desktop Gadget)
Feed Viewer
SideShow GMail

3. Linux driver installation

3.1 Driver installation

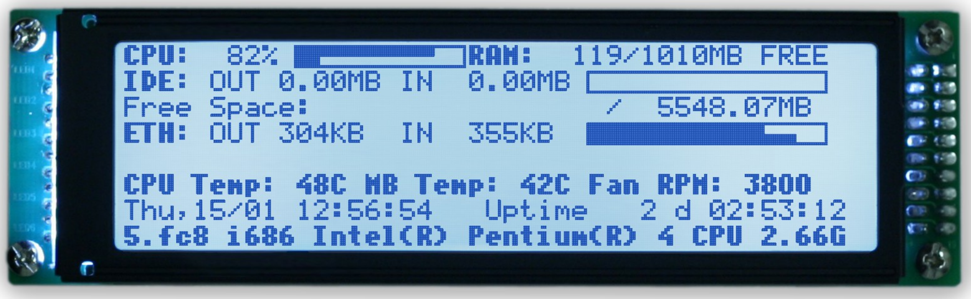
Ubuntu Linux 32/64bit ppa
https://launchpad.net/~mini-box.com/+archive/ppa
follow this instruction to add ppa:mini-box.com/ppa to your Ubuntu distribution,
then use:
  sudo apt-get install picolcdgraphic
command to install the picoLCD driver.
or you can download a precompiled deb package for your Ubuntu distribution from this link.

By default there are 3 examples installed the first one it's a overview of your system internals like memory and cpu utilisation, free hard-drive space, network bandwidth in use, hard-drive io in use, date, uptime and system information.

Look in */etc/picoLCDGraphic/Configs* for the examples.
To change the currently running example copy one of the configs from
/etc/picoLCDGraphic/Configs/ to */etc/lcd4linux.conf*

Example:
  sudo cp /etc/picoLCDGraphic/Configs/lcd4linux.conf.rss /etc/lcd4linux.conf
  sudo chmod 600 /etc/lcd4linux.conf
and restart the running daemon:
  sudo /etc/init.d/picolcdgraphic restart

The program will continue to monitor your system on background and report the information on the LCD.



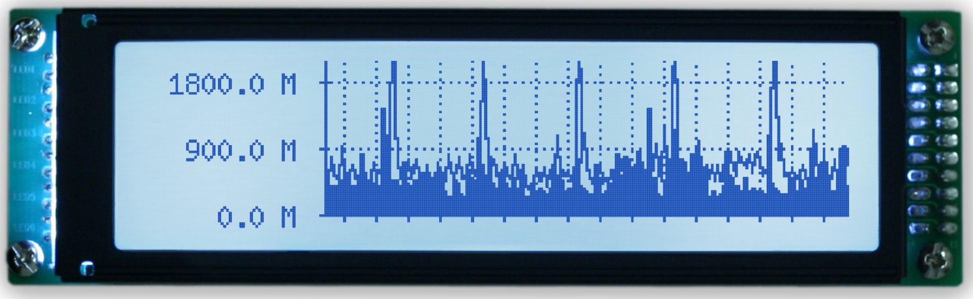
```
CPU:   82%                RAM:   119/1010MB FREE
IDE:  OUT 0.00MB  IN   0.00MB
Free  Space:                   /   5548.07MB
ETH:  OUT 304KB   IN   355KB

CPU Temp: 48C MB Temp: 42C Fan RPM: 3800
Thu,15/01 12:56:54    Uptime   2 d 02:53:12
5.fc8 i686 Intel(R) Pentium(R) 4 CPU 2.66G
```

The picoLCD MRTG monitors allows MRTG graphs to be rendered on the picoLCD display.
To change the location of the MRTG server you will need to edit
/etc/lcd4linux.conf
 (after the example from */etc/picoLCDGraphic/Configs.lcd4linux.conf.mrtg* has been copied to */etc/lcd4linux.conf*)
with a text editor and change the lines:

```
currentImage 'http://192.168.12.113/mrtg/127.0.0.1_2-month.png'
mrtgDayImageURL 'http://192.168.12.113/mrtg/127.0.0.1_2-day.png'
```

```
mrtgWeekImageURL 'http://192.168.12.113/mrtg/127.0.0.1_2-week.png'
```

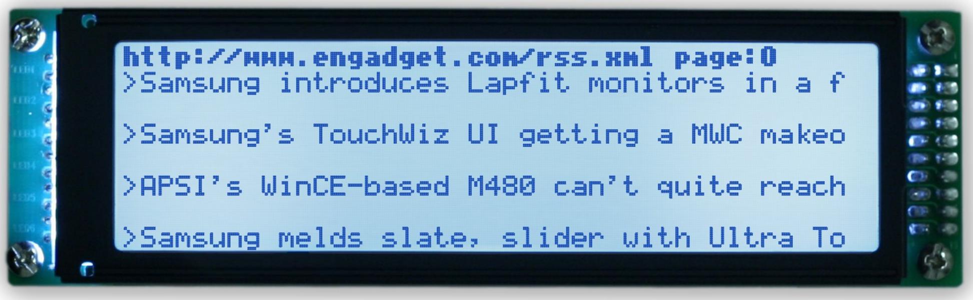
To point to your MRTG server image URLs. You can use the arrows buttons on the LCD to change the current displayed image. A sample MRTG config file is presented in Appendix A.



The picoLCD RSS Viewer allows different RSS feeds to be downloaded and displayed on the LCD. By default there are 3 feeds installed you can change these feeds by editing */etc/lcd4linux.conf* (after the example from */etc/picoLCDGraphic/Configs.lcd4linux.conf.rss* has been copied to */etc/lcd4linux.conf*) and change the lines:

```
rss1   'http://slashdot.org/slashdot.rdf'
rss2   'http://www.engadget.com/rss.xml'
rss3   'http://www.linuxsecurity.com/static-content/debian.rss'
```

To point to your desired RSS feeds. You can use the Home and Back button to change between the feeds and arrows buttons to scroll the feed news up and down.



```
http://www.engadget.com/rss.xml page:0
>Samsung introduces Lapfit monitors in a f

>Samsung's TouchWiz UI getting a MWC makeo

>APSI's WinCE-based M480 can't quite reach

>Samsung melds slate, slider with Ultra To
```

3.2 Customizing your picoLCD display with plugins and widgets

The easiest way to customize the information displayed by the picoLCD is to edit the configuration file */etc/lcd4linux.conf* and add other widgets/plugins, create different behaviors and display data as you see fit. A sample lcd4linux.conf file is available on Appendix B or in /etc/picoLCDGraphic/Configs.

Available plugins are:

- Asterisk plugin
- Config plugin
- /proc/cpuinfo plugin
- /proc/diskstats plugin
- DVB plugin
- exec (external command) plugin
- FIFO plugin
- file reading plugin
- GPS (NMEA) plugin
- hddtemp (hard disk temperature)
- I2C sensors plugin
- iconv (charset converter) plugin
- ISDN Monitor (imon) plugin
- ISDN plugin
- KVV plugin
- /proc/loadavg plugin
- mathematical functions plugin
- /proc/meminfo plugin
- MPd plugin
- MySQL plugin
- /proc/net/dev plugin
- netinfo plugin
- POP plugin
- PPP plugin
- /proc/stat plugin
- Python plugin
- Example plugin
- SETI plugin
- statfs plugin
- string functions plugin
- APM plugin
- Time plugin
- uname plugin
- uptime plugin
- WLAN plugin
- XMMS plugin

Available widgets are:

- Bar Widget
- Icon Widget
- Image Widget
- Timer Widget
- GPO Widget (General Purpose Inputs and Outputs)
- Keypad Widget
- Text Widget

More information on working with widgets, plugins can be found on
lcd4linux page at:
http://ssl.bulix.org/projects/lcd4linux/wiki/Plugins
http://ssl.bulix.org/projects/lcd4linux/wiki/Layout
http://ssl.bulix.org/projects/lcd4linux/wiki/Evaluator

3.3 Customizing your picoLCD display with external python scripts

Another powerful features is customization through user made scripts. The scripts are written in python which is bundled with most linux distributions. Both MRTG and RSS programs are written in python. A sample python script to fetch MRTG images is presented in Appendix C. Python scripts can be added to the display by creating a new widget and display it on a LCD Row as in the example below.

```
Widget MRTGMinute {
    class 'Text'
    expression currentFile=python::exec('mrtg', 'saveimage', currentImage)
    width 42
     update 100
}

Layout picoLCD {
    Row1 {
       Col1 'MRTGMinute'
    }
}
```

The python::exec plugin will load the mrtg.py script and call the saveimage function from the script with the currentImage as paramater. The variable currentImage is a lcd4linux variable defined in the lcd4linux.conf
The python::exec plugin will look for your scripts in default python search paths and in PYTHONPATH environment variable that can be set by user

3.4 Controlling the display with the keypad keys

To make use of the picoLCD keys you first need to define a timer which will read the picoLCD keys at an interval and set a variable to hold the current pressed key:

```
Variables {
   n 0 #the key value taken from picoLCD gpi
   nkey 0 #current pressed key
}
```

```
# A simple timer to get the key values
Widget TimerW {
    class 'Timer'
    expression n=LCD::GPI(1);nkey=n!=0?n:nkey
    active 1
    update 100
}
```

This will read key pressed every 100ms and store the key values in nkey variable. A table for current key / values mapping is pressented in Appendix D.

You will also need to define a widget that does an action when a given key is pressed:

```
#The keypad up key will display text "Arrow Up"

Widget GPO_kup {
   class 'GPO'
   expression nkey==5?'Arrow Up':'Other key';nkey==5?1:0
   update 300
}
```

The last thing you need to do is to declare the key widgets and the timer defined above in the picoLCD layout section:

```
Layout picoLCD {
    Row1 {
       Col1 'GPO_kup'
    }
    Timer1 'TimerW'
    GPO1       'GPO_kup'

}
```

4. Appendixes

Appendix A Sample MRTG file for Linux MRTG display plugin

```
EnableIPv6: no
WorkDir: /var/www/mrtg
XSize[127.0.0.1_2]: 155
YSize[127.0.0.1_2]: 30
Options[127.0.0.1_2]: noborder, nobanner, nolegend, integer
#kMG[127.0.0.1_2]:,,M
YLegend[127.0.0.1_2]: Bytes/s

Target[127.0.0.1_2]: 2:lcsources@127.0.0.1:
SetEnv[127.0.0.1_2]: MRTG_INT_IP="192.168.12.113" MRTG_INT_DESCR="eth0"
MaxBytes[127.0.0.1_2]: 12500000
Title[127.0.0.1_2]: Traffic Analysis for picolcd.com
PageTop[127.0.0.1_2]: <h1>Traffic Analysis for picolcd.com</h1>
               <div id="sysdetails">
                      <table>
                              <tr>
                                     <td>System:</td>
                                     <td>picolcd. com</td>
                              </tr>
                      </table>              </div>
```

Appendix B Sample lcd4linux.conf file for Linux picoLCD software

```
# The display settings
Display picoLCD {
    Driver   'picoLCDGraphic'
    Size     '256x64'
    Contrast  230
    Backlight 1
    Inverted   1
    Icons     1
}
```

```
#Variables needed for script to function
Variables {
   n 0 #the key value taken from picoLCD gpi
   nkey 0 #current pressed key
   # the current URL that will be passed to python script
   currentImage 'http://192.168.12.113/mrtg/127.0.0.1_2-month.png'
   # currentImage will change it's value when arrows keys are pressed
   mrtgDayImageURL 'http://192.168.12.113/mrtg/127.0.0.1_2-day.png'
   mrtgWeekImageURL 'http://192.168.12.113/mrtg/127.0.0.1_2-week.png'
   #currentFile variable holds the location of the saved image by the pyhon script
   currentFile ' '
   tick 500
   tack 100
   minute 60000
}

# A simple timer to get the key values
Widget TimerW {
    class 'Timer'
    expression n=LCD::GPI(1);nkey=n!=0?n:nkey
    active 1
    update 100
}

# This widget will display the URL of the current image on the LCD and
# also save the location of the downloaded file in the currentFile variable
# export PYTHONPATH which should point to the scripts location
Widget MRTGMinute {
    class 'Text'
    expression currentFile=python::exec('mrtg', 'saveimage', currentImage)
    width 42
    #update minute
    update 100
}

# This is a graphic widget that will display the image downloaded by the above
# python script. CurrentFile variable is set by the MRTGMinute widget
Widget BandwidthImage {
    class 'Image'
    file currentFile
    visible 1
    inverted 0
    reload 1
    update minute
}

#The keypad up key which will change the current image to the mrtg week graphs
Widget GPO_kup {
   class 'GPO'
   expression currentImage=nkey==5?mrtgDayImageURL:currentImage;nkey==5?1:0
   update 300
}

Widget GPO_kdown {
  class 'GPO'
  expression currentImage=nkey==7?mrtgWeekImageURL:currentImage;nkey==7?1:0
  update 300
}
Layout picoLCD {
    Row1 {
       Col1 'MRTGMinute'
    }
    Layer 1 {
        X0.Y0 'BandwidthImage'
    }
    Timer1 'TimerW'
    GPO2       'GPO_kup'
    GPO3       'GPO_kdown'

}

Display 'picoLCD'
Layout  'picoLCD'
```

Appendix C Sample python script for Linux picoLCD

```
import urllib
import shutil

download_path = "/tmp/"

def saveimage(imageurl):
        filename = imageurl.split('/')[-1]
        tmpname = filename + ".tmp"
        try:
                urllib.urlretrieve(imageurl, download_path + tmpname)
        except IOError:
                return "Error downloading file"
        else:
                shutil.move(download_path + tmpname, download_path + filename)
                return download_path + filename
```

Appendix D Key and values mapping for Linux software

Key name	Key Value
Back	0x01
Home	0x02
Up Arrow	0x05
Down Arrow	0x07
OK	0x06