

# LogView Studio

**Linux Preparation with Mono**

2014 by Dominik Schmidt  
dominik@logview.info

# Table of Contents

<b>Part I Linux Prepare</b>	<b>1</b>
1 SSH Connect .....	1
2 Raspberry Pi .....	4
System Preparation .....	5
Mono .....	6
Install Mono.....	7
Compile Mono from TAR.....	8
Compile Mono from GIT.....	9
Testing .....	10
3 Cubietruck .....	11
System Preparation .....	12
Mono .....	13
Install Mono.....	14
Compile Mono from TAR.....	15
Compile Mono from GIT.....	16
Testing .....	17
4 Ubuntu .....	18
System Preparation .....	19
Mono .....	19
Install Mono.....	20
Compile Mono from TAR.....	20
Compile Mono from GIT.....	22
5 CentOS .....	23
System Preparation .....	24
Mono .....	24
Compile Mono from TAR.....	25
Compile Mono from GIT.....	27
<b>Index</b>	<b>0</b>

# 1 Linux Prepare



This Topic will explain how to install different Linux Systems in order to get LogView Studio Command running.

At the Moment we can support these Systems:

- Arm6 32Bit - e.g. [Raspberry Pi](#)
- Arm7 32Bit - e.g. [Cubietruck](#)
- Generic x86 / x64 Linux (we will cover (L)Ubuntu and CentOS in this document)

## Procedure

There are typically 4 steps to do:

- Install the Operating System and apply the latest updates
- Install (or Compile) Mono
- Copy the LogView Studio Files to the System
- Run LogView Studio and get the Database Files

## 1.1 SSH Connect

To get a connection to your Linux system it's very handy to have SSH up and running.

For a SSH connection you need some software on your Windows system (SSH client) and some software on your Linux system (SSH server).

This topic will cover Putty as a SSH client.

## Download

Download putty.exe from here

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Or use this direct link for downloading

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

## Installation

Just execute the file and you are done.

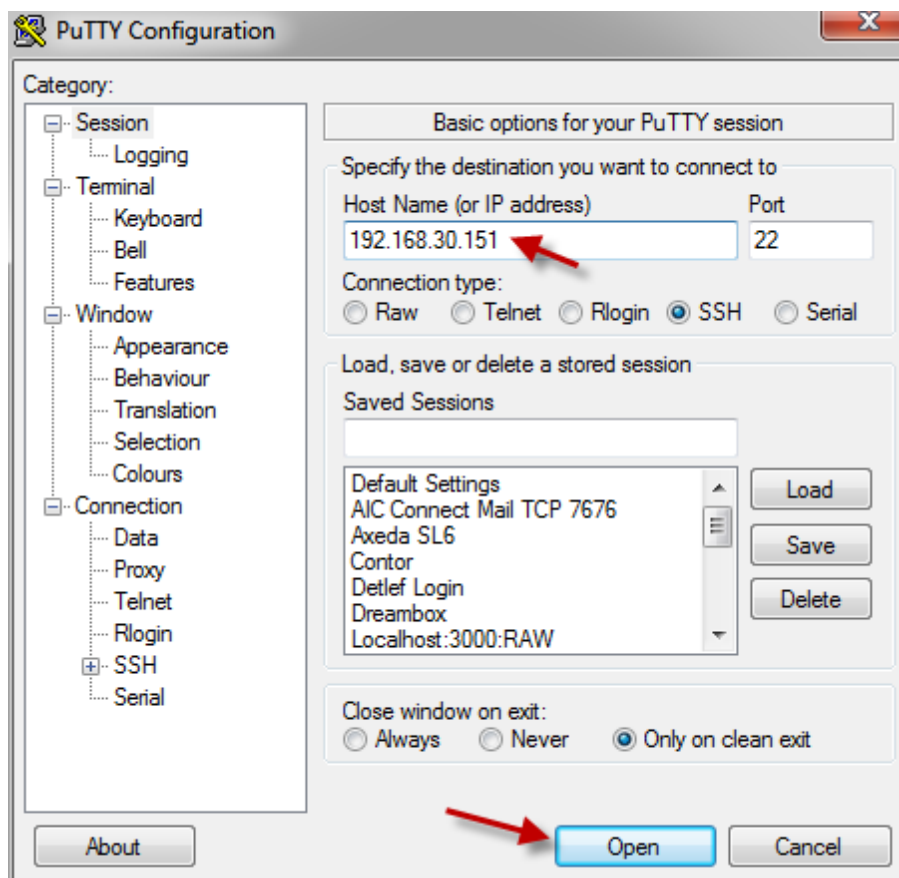
## Get the IP from your Linux system

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:c9:53:f7
          inet addr:192.168.30.151  Bcast:192.168.30.255
Mask:255.255.255.0

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:69364 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10413 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:92340960 (88.0 MiB)  TX bytes:980385 (957.4 KiB)
```

## Run Putty

Start putty.exe, enter the IP-address and click on Open



Accept the key



Enter username and password and you are connected.

```
pi@raspl: ~  
login as: pi  
pi@raspl's password:  
Linux rasp1 3.12.26+ #704 PREEMPT Wed Aug 20 22:35:11 BST 2014 armv6l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Aug 29 10:07:00 2014 from pbdbschn.fritz.box  
pi@raspl ~ $
```

### Additional Information

Mobaxterm as an alternative to putty

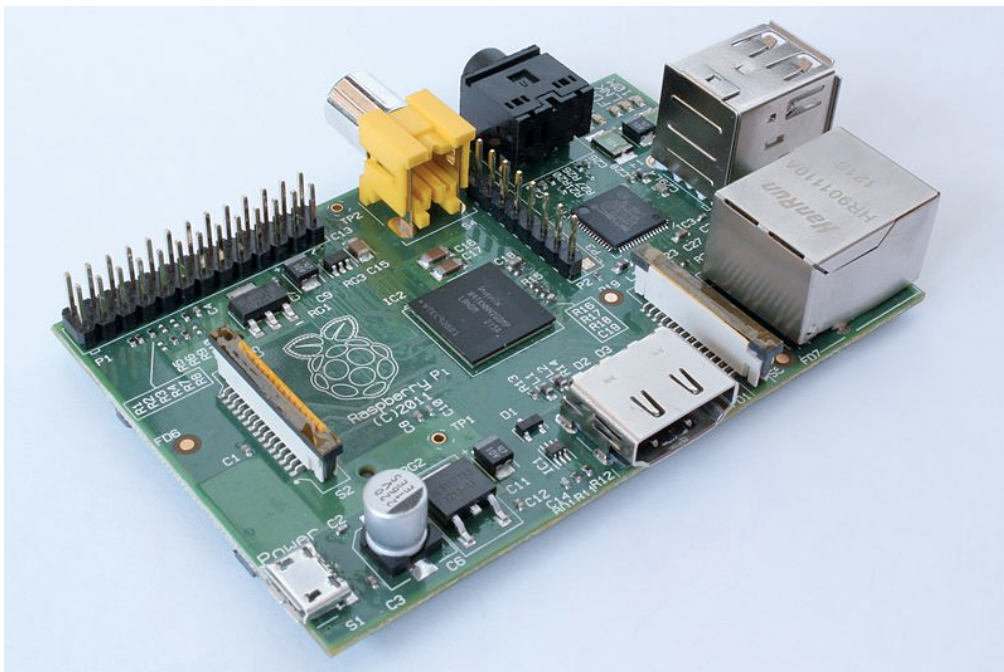
<http://mobaxterm.mobatek.net/>

Xming X Server for Windows

<http://sourceforge.net/projects/xming/>

## 1.2 Raspberry Pi

The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. ([http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi))



### System

700MHz CPU (ARM6)

256MB or 512MB Memory

SD Card Storage

## Webpage

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

## Default Credentials

User pi

Password raspberry

## Additional Information

RPi Low-level peripherals ([http://elinux.org/RPi\\_Low-level\\_peripherals](http://elinux.org/RPi_Low-level_peripherals))

### 1.2.1 System Preparation

The following section will explain how to setup a normal Raspberry Pi. It will not cover every detail but will give a good overview on the important steps.

#### *Hint*

*For the next steps you must connect a display, mouse and a keyboard to your Pi.*

## Download

For the Raspberry Pi we use the normal NOOBS Installer (actually version 1.3.9). You can download it from here:

<http://www.raspberrypi.org/downloads>

## System Installation

Place the extracted files on a clean SD card, put it into the Raspberry and boot it up. Now you can install different Operating Systems with the installer.

We install the recommended Raspbian System.

## System Preparation

After the installation finished and the Raspberry has rebooted it will show you the configuration (raspi-config). Here you can tweak some settings.

- Expand the File system in order to use the whole SD card (may not needed with NOOBS)
- Change the User Password (if you like)
- **Don't change the Local** (Translation of the System). This will cause trouble during compilation.  
You can switch the Local after LogView Studio is running fine.
- To get some more power set the Overclocking (900MHz is safe, 950MHz should work, 1000MHz often results in errors)  
If the system is busy you can check the result with  
`watch -n 1 cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq`
- Ensure that SSH is enabled (Advanced options)  
We will use SSH to connect to the Raspberry after the installation is done.

Reboot the system to activate the changes

From this point we will use [SSH to connect](#) to the Raspberry. You can disconnect any keyboard, mouse and display if you like.

We only need the Raspberry powered up and connected with a network cable.

## Software Install / Update

- Update the software package lists  
`sudo apt-get update`
- Update the installed software packages (this may take a while)  
`sudo apt-get dist-upgrade`
- Install some additional useful packages (maybe some are preinstalled)  
`sudo apt-get install sudoku htop git build-essential autoconf automake screen curl binutils`  
(It's always a good idea to have [Sudoku](#) installed :-)
- Install the latest Raspberry Pi Firmware  
`sudo rpi-update`
- Reboot  
`sudo reboot`

## Additional Information

German Information about Raspberry installation

<http://linuxwelt.blogspot.de/2013/10/raspberry-pi-rasbian-installieren.html>

Raspberry Pi Firmware-Update

<http://rasberrypiguide.de/howtos/raspberry-pi-firmware-update/>

General Information about Debian software installation / updates

<https://help.ubuntu.com/community/AptGet/Howto>

### 1.2.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime.

<http://www.mono-project.com>

## How to get Mono?

There are at least three different ways for getting Mono installed on your Raspberry Pi. All of them have their pros and cons:

### Install it from the Raspbian package repository

- + very easy to handle
- + fast installation
- + works with LogView Studio (LvCmd)
- mostly an old Mono release
- WinForms applications produce errors with some components (like a Textbox)

### Compile it from a TAR package



- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
- o mostly newer than the Raspbian package but older than the GIT version
- takes a long time to compile
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

### Compile it from the GIT repository

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- takes a long time to compile
- needs some extra packages for compiling

### Which method is the best?

For LogView Studio it depends on what you like to do. If you only need the LvCmd version you are just fine with the normal Raspbian package.

If you like to run WinForms applications it's a better choice to compile it from GIT.

#### Note

*If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.*

### AOT

Ahead of Time Compilation or AOT is a feature of the Mono runtime code generator. We don't need it to run LogView Studio and in some cases it results in compilation problems on ARM. So we just disable it.

<http://www.mono-project.com/docs/advanced/aot/>

#### 1.2.2.1 Install Mono

This will install the Mono package from the Raspbian repository.

### Install

Installing Mono on the Raspberry Pi is a simple task. Just use the following command:

```
sudo apt-get install mono-complete libgdiplus
```

If you want to try C# development as well just add `monodevelop`, too.

```
sudo apt-get install mono-complete libgdiplus monodevelop
```

The installation will take some minutes.

### Uninstall

Uninstalling is as simple as installing:

```
sudo apt-get autoremove monodevelope (only if you installed it)
```

```
sudo apt-get autoremove libgdiplus
```

#### Hint

*libgdiplus removes mono-complete as well.*

### 1.2.2.2 Compile Mono from TAR

This will compile Mono from a TAR package. **It takes about 9 hours in sum @ 950MHz.**

#### Procedure

- Install additional needed packages

```
sudo apt-get install libtool libglib2.0-dev libxrender-dev
libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev
libexif-dev gettext libcairo2-dev
```
- Install libgdiplus

```
sudo apt-get install libgdiplus
```

(You can also compile the latest version. See 'Compile Mono from GIT' for details.)
- Create a new folder where we place the files. You can use your home directory for that

```
cd ~
mkdir sources
cd sources
```
- Check for the latest TAR version

```
wget -qO- http://download.mono-project.com/sources/mono/ |grep -o -E
'href="([^\#]+)"' | cut -d'"' -f2 | sort | uniq
```

or just watch at the website  
<http://origin-download.mono-project.com/sources/mono/>  
<http://download.mono-project.com/sources/mono/>
- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)

```
wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2
```
- Extract the TAR file

```
tar -jxvf mono-3.6.0.tar.bz2
```
- Enter the new directory

```
cd mono-3.6.0
```
- run configure (this will take ~10 minutes @ 950MHz)

```
./configure --prefix=/usr/local/
```

/usr/local/ is the installation path of the compiled files
- run make - compile all the files (this will take ~8:30 hours @ 950MHz)

```
sudo SKIP_AOT=true make
```
- run make install - this will install the needed files into your system (this will take ~25 minutes @ 950MHz)

```
sudo SKIP_AOT=true make install
```

#### Uninstall

```
sudo make uninstall
```

#### Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv  
<https://gist.github.com/pjvds/7185693>

### 1.2.2.3 Compile Mono from GIT

This will compile Mono from the GIT repository. **It takes about 10:30 hours in sum @ 950MHz.**

**Hint**

*Since this is updated very often it could include some bugs. But in most cases it works great.*

#### Procedure

- Install additional needed packages  
`sudo apt-get install libtool libglib2.0-dev libxrender-dev libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev libexif-dev gettext libcairo2-dev`
- Create a new folder where we place the files. You can use your home directory for that  
`cd ~`  
`mkdir sources`  
`cd sources`

#### **libgdiplus Compilation** (this will take overall ~16 minutes @ 950MHz)

- Get the libgdiplus sources from GIT  
`git clone git://github.com/mono/libgdiplus`
- Enter the new directory  
`cd libgdiplus`
- Run autogen  
`./autogen.sh --prefix=/usr`
- Compile the sources  
`sudo make`
- Install the files  
`sudo make install`
- Leave the directory  
`cd ..`

#### **Mono Compilation (tak ????)**

- Get the Mono sources from GIT (~45 minutes)  
`git clone git://github.com/mono/mono.git`
- Enter the new directory  
`cd mono`
- Run autogen (~20 minutes)  
`./autogen.sh --prefix=/usr/local`
- Install and configure gmcs  
`sudo make get-monolite-latest`  
`cd mcs/class/lib/monolite`  
`cp basic.exe gmcs.exe`  
`cd /home/pi/sources/mono`

**Hint**

You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.

- Compile the sources (this will take **~9:00 hours** @ 950MHz)  
If you used `sudo make get-monolite-latest` use this:  
`sudo SKIP_AOT=true make EXTERNAL_MCS="/home/pi/sources/mono/mcs/class/lib/monolite/gmcs.exe"`  
If you use `apt-get install mono-gmcs` use this:  
`sudo SKIP_AOT=true make`
- Install the files (this will take ~30 minutes @ 950MHz)  
`sudo SKIP_AOT=true make install`

**Uninstall**

- Run the uninstall  
`sudo make uninstall`
- remove the source files  
`rm -rf ~/sources/mono`

**Additional Information**

Raspberry Pi Mono libgdiplus

<http://www.shataimic.com/raspberry-pi-mono-libgdiplus>

Mono Compile from Git

<http://stackoverflow.com/questions/13365158/installing-mono-3-x-3-0-x-and-or-3-2-x>

<https://gist.github.com/pjvds/7185693>

**1.2.2.4 Testing**

These tests work for all types of installation.

**First Test**

If the installation is done just type the following command:

`mono --version`

```
Mono JIT compiler version 3.2.8 (Debian 3.2.8+dfsg-4+rpil)
Copyright (C) 2002-2014 Novell, Inc, Xamarin Inc and Contributors. www.mono-project.com
  TLS:             __thread
  SIGSEGV:         normal
  Notifications:   epoll
  Architecture:    armel,vfp+hard
  Disabled:        none
  Misc:            softdebug
  LLVM:            supported, not enabled.
  GC:              sgen
```

Ensure that you find the verb "hard" in the Architecture line.

If you get the hard info in the output you are done with the mono installation.

The version should match your installed / compiled version.

**Check Hard Float Unit**

There were some hard float issues in the past releases of Mono. You can simply check if it works with the csharp shell.

```
pi@raspberrypi ~ $ date
Sun Mar 16 13:12:07 UTC 2014
pi@raspberrypi ~ $ csharp
Mono C# Shell, type "help;" for help

Enter statements below.
csharp> DateTime.Now;
03/16/2014 13:12:39
csharp>
```

You must see the correct datetime. Otherwise your mono is damaged.

### Check WinForms

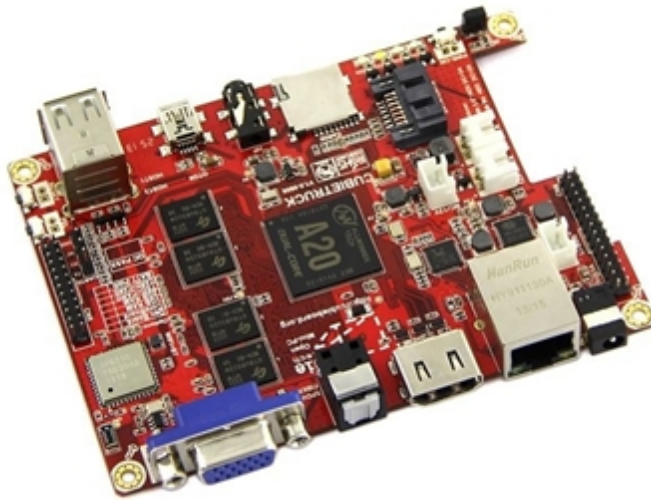
If you need to run WinForms applications just compile a simple demo with a TextBox and ComboBox. You will receive errors when you start that application under Mono. If the application starts fine your Mono is ok.

#### **Hint**

*It could be important to run the application with sudo. Just check that.*

## 1.3 Cubietruck

Cubietruck is the 3rd board of Cubieteam, so we also name it Cubieboard3. It's a new PCB model adopted with Allwinner A20 main chip, just like Cubieboard2.



### System

1000MHz ARM® Cortex™-A7 Dual-Core

1GB or 2GB Memory

HDD and SD Card Storage

### Webpage

<http://cubieboard.org/>

Image

<http://cubian.org/>

## Default Credentials

User cubian

Password cubian

### 1.3.1 System Preparation

The following section will explain how to setup a Cubietruck. It will not cover every detail but will give a good overview on the important steps.

#### **Hint**

*Display, mouse and keyboard are not needed.*

## Download

For the Cubietruck we use Cubian SD Image (actually version **Cubian-desktop-r1-a20-ct**). You can download it from here:

<http://cubian.org/downloads/>

## System Installation

- Unpack the 7z file to get the img file
- Use Win32DiskImager (<http://sourceforge.net/projects/win32diskimager/>) to bring the image to the SD card. You need at least a 4GB SD card.
- Put the SD card in the Cubietruck und power it on

## System Preparation

- Connect to 'cubian' vie [SSH](#) (Port 36000, User cubian, Password cubian)
- Bring Cubian to the NAND-Flash ...
- `sudo bash` (Password cubian)
- Get a PGP Key for debian:  
`wget -O - http://packages.cubian.org/cubian.gpg.key | apt-key add -`
- Update the software package lists  
`apt-get update`
- Update the installed software packages (this may take a while)  
`apt-get dist-upgrade`
- Get the Script for NAND Install  
`apt-get install cubian-nandinstall`
- Run the Script for NAND Install (takes about 10 minutes)  
`cubian-nandinstall`  
If the script finished it will shut down your Cubietruck (after you enter Y).
- Remove the SD card and power on the Cubietruck again. It should boot from NAND now.

- Set Timezone  
`sudo dpkg-reconfigure tzdata`
- Locales in English (en\_US.UTF-8)  
`sudo dpkg-reconfigure locales`
- Kernel update  
`sudo apt-get update && sudo apt-get install cubian-update`

If you get messages like this:

```
locale: Cannot set LC_CTYPE to default locale: No such file or directory
locale: Cannot set LC_MESSAGES to default locale: No such file or directory
locale: Cannot set LC_ALL to default locale: No such file or directory
```

try to reinstall the locales:

```
sudo apt-get install --reinstall locales
```

## Software Install / Update

- Install some additional useful packages (maybe some are preinstalled)  
`sudo apt-get install sudoku htop git build-essential autoconf automake screen curl binutils`  
(It's always a good idea to have [Sudoku](#) installed :-)

## Hints

General Information about Debian software installation / updates

<https://help.ubuntu.com/community/AptGet/Howto>

Cubian Webpage

<http://cubian.org/>

Very good Tutorial how to install Cubian (German)

<http://www.kriwanek.de/cubieboard.html>

<http://www.kriwanek.de/cubieboard/393-cubian-auf-sd-card-installieren.html>

<http://www.kriwanek.de/cubieboard/402-cubian-auf-nand-flash-uebertragen.html>

<http://www.kriwanek.de/cubieboard/396-cubian-auf-sata-festplatte-verlagern.html>

## 1.3.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime.

<http://www.mono-project.com>

## How to get Mono?

There are at least three different ways for getting Mono installed on your Cubietruck. All of them have their pros and cons:

### Install it from the debian package repository

- + very easy to handle
- + fast installation
- + works with LogView Studio (LvCmd)

- mostly an old Mono release
- WinForms applications produce errors with some components (like a Textbox)

### **Compile it from a TAR package**

- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
- o mostly newer than the Raspbian package but older than the GIT version
- takes a long time to compile
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

### **Compile it from the GIT repository**

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- takes a long time to compile
- needs some extra packages for compiling

### **Which method is the best?**

For the Cubietruck the best choice is to compile it from GIT.

#### **Note**

*If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.*

### **AOT**

Ahead of Time Compilation or AOT is a feature of the Mono runtime code generator. We don't need it to run LogView Studio and in some cases it results in compilation problems on ARM. So we just disable it.

<http://www.mono-project.com/docs/advanced/aot/>

Mono Bug

[https://bugzilla.xamarin.com/show\\_bug.cgi?id=17017](https://bugzilla.xamarin.com/show_bug.cgi?id=17017)

#### **1.3.2.1 Install Mono**

This will install the Mono package from the Debian armhf repository.

### **Install**

Installing Mono on the Raspberry Pi is a simple task. Just use the following command:

```
sudo echo "deb http://ftp.debian.org/debian sid main" >> /etc/apt/sources.list
sudo apt-get update
sudo apt-get install mono-complete libgdiplus -t sid
```

If you want to try C# development as well just add `monodevelop`, too.

```
sudo apt-get install mono-complete libgdiplus monodevelop -t sid
```

The installation will take some minutes.



## Uninstall

Uninstalling is as simple as installing:

```
sudo apt-get autoremove monodevelope (only if you installed it)
sudo apt-get autoremove libgdiplus
```

### Hint

*libgdiplus removes mono-complete as well.*

### 1.3.2.2 Compile Mono from TAR

This will compile Mono from a TAR package. **It takes about 4 hours in sum.**

## Procedure

- Install additional needed packages

```
sudo apt-get install libtool libglib2.0-dev libxrender-dev
libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev
libexif-dev gettext libcairo2-dev
```
- Install libgdiplus

```
sudo apt-get install libgdiplus
```

(You can also compile the latest version. See 'Compile Mono from GIT' for details.)
- Create a new folder where we place the files. You can use your home directory for that

```
cd ~
mkdir sources
cd sources
```
- Check for the latest TAR version

```
wget -qO- http://download.mono-project.com/sources/mono/ |grep -o -E
'href="([^\"]#)+"' | cut -d'"' -f2 | sort | uniq
```

or just watch at the website  
<http://origin-download.mono-project.com/sources/mono/>  
<http://download.mono-project.com/sources/mono/>
- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)

```
wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2
```
- Extract the TAR file

```
tar -jxvf mono-3.6.0.tar.bz2
```
- Enter the new directory

```
cd mono-3.6.0
```
- run configure (this will takes ~5 minutes)

```
./configure --prefix=/usr/local/
```

/usr/local/ is the installation path of the compiled files
- run make - compile all the files (this will take ~3:30 hours)

```
sudo SKIP_AOT=true make
```
- run make install - this will install the needed files into your system (this will take ~10 minutes)

```
sudo SKIP_AOT=true make install
```

## Uninstall

```
sudo make uninstall
```

## Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv

<https://gist.github.com/pjvds/7185693>

### 1.3.2.3 Compile Mono from GIT

This will compile Mono from the GIT repository. **It takes about 4 hours in sum.**

#### *Hint*

*Since this is updated very often it could include some bugs. But in most cases it works great.*

## Procedure

- Install additional needed packages

```
sudo apt-get install libtool libglib2.0-dev libxrender-dev  
libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev  
libexif-dev gettext libcairo2-dev
```
- Create a new folder where we place the files. You can use your home directory for that

```
cd ~  
mkdir sources  
cd sources
```

### **libgdiplus Compilation** (this will take overall ~10 minutes)

- Get the libgdiplus sources from GIT

```
git clone git://github.com/mono/libgdiplus
```
- Enter the new directory

```
cd libgdiplus
```
- Run autogen

```
./autogen.sh --prefix=/usr
```
- Compile the sources

```
sudo make
```
- Install the files

```
sudo make install
```
- Leave the directory

```
cd ..
```

### **Mono Compilation**

- Get the Mono sources from GIT

```
git clone git://github.com/mono/mono.git
```
- Enter the new directory

```
cd mono
```
- Run autogen (~11 minutes)

```
./autogen.sh --prefix=/usr/local
```

- Install and configure gmcs

```
sudo make get-monolite-latest
cd mcs/class/lib/monolite
cp basic.exe gmcs.exe
cd ~/sources/mono
```

#### Hint

You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.

- Compile the sources (this will take ~3:30 hours)  
If you used `sudo make get-monolite-latest` use this:

```
sudo SKIP_AOT=true make EXTERNAL_MCS="/home/pi/sources/mono/mcs/class/lib/monolite/gmcs.exe"
```

  
If you use `apt-get install mono-gmcs` use this:

```
sudo SKIP_AOT=true make
```
- Install the files

```
sudo SKIP_AOT=true make install
```

## Uninstall

- Run the uninstall

```
sudo make uninstall
```
- remove the source files

```
rm -rf ~/sources/mono
```

## Additional Information

Mono Compile from Git

<http://stackoverflow.com/questions/13365158/installing-mono-3-x-3-0-x-and-or-3-2-x>

<https://gist.github.com/pjvds/7185693>

### 1.3.2.4 Testing

These tests work for all types of installation.

## First Test

If the installation is done just type the following command:

```
mono --version
```

```
cubie@Cubian:~$ mono --version
Mono JIT compiler version 3.2.8 (Debian 3.2.8+dfsg-7)
Copyright (C) 2002-2014 Novell, Inc, Xamarin Inc and Contributors. www.mono-
project.com
      TLS:              __thread
      SIGSEGV:          normal
      Notifications:    epoll
      Architecture:    armel,vfp+hard
      Disabled:         none
      Misc:             softdebug
      LLVM:             supported, not enabled.
      GC:               sgen
```

Ensure that you find the verb "hard" in the Architecture line.

If you get the hard info in the output you are done with the mono installation.

The version should match your installed / compiled version.

### Check Hard Float Unit

There were some hard float issues in the past releases of Mono. You can simply check if it works with the csharp shell.

```
cubie@Cubian ~ $ date
Sun Mar 16 13:12:07 UTC 2014
cubie@Cubian ~ $ csharp
Mono C# Shell, type "help;" for help
```

```
Enter statements below.
```

```
csharp> DateTime.Now;
```

```
03/16/2014 13:12:39
```

```
csharp>
```

You must see the correct datetime. Otherwise your mono is damaged.

### Check WinForms

If you need to run WinForms applications just compile a simple demo with a TextBox and ComboBox. You will receive errors when you start that application under Mono. If the application starts fine your Mono is ok.

#### **Hint**

*It could be important to run the application with sudo. Just check that.*

## 1.4 Lubuntu

Lubuntu is a fast and lightweight operating system developed by a community of Free and Open Source enthusiasts. The core of the system is based on Linux and Ubuntu.



### System

x86 / X64

### Webpage

<http://lubuntu.net/>

### 1.4.1 System Preparation

For this documentation we use Lubuntu (<http://lubuntu.net/>). It should be possible to use this topic for other Debian based systems as well like Ubuntu, Kubuntu, ...

#### Download

You can download 32 or 64 Bit images from the Lubuntu website:

<https://help.ubuntu.com/community/Lubuntu/GetLubuntu>

#### System Installation

There are tons of information in the internet how to install Lubuntu. We won't cover the details here.

<https://help.ubuntu.com/community/Lubuntu/InstallingLubuntu>

#### System Preparation

- Install SSH  
`sudo apt-get install openssh-server`
- Ensure that you use English locals until LogView Studio runs!

From this point it would be ok to use [SSH to connect to Lubuntu](#) only.

Working on the machine directly would be fine, too.

#### Software Install / Update

- Update the software package lists  
`sudo apt-get update`
- Update the installed software packages (this may take a while)  
`sudo apt-get dist-upgrade`
- Install some additional useful packages (maybe some are preinstalled)  
`sudo apt-get install sudoku htop git-core build-essential autoconf  
automake screen curl binutils`  
(It's always a good idea to have [Sudoku](#) installed :-)

#### Hints

General Information about Debian software installation / updates

<https://help.ubuntu.com/community/AptGet/Howto>

### 1.4.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime.

<http://www.mono-project.com>

#### How to get Mono?

There are at least two different ways for getting Mono installed on your Cubietruck. All

of them have their pros and cons:

#### **Compile it from a TAR package**

- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
- o mostly newer than the Raspbian package but older than the GIT version
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

#### **Compile it from the GIT repository**

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- needs some extra packages for compiling

### **Which method is the best?**

For x86/x64 the best choice is to compile it from GIT because it takes not really much time.

#### **Note**

*If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.*

#### **1.4.2.1 Install Mono**

This will install the Mono package from the (L)Ubuntu repository.

#### **Install**

Installing Mono is a simple task. Just use the following command:

```
sudo apt-get install mono-complete libgdiplus
```

If you want to try C# development as well just add `monodevelop`, too.

```
sudo apt-get install mono-complete libgdiplus monodevelop
```

The installation will take some minutes.

#### **Uninstall**

Uninstalling is as simple as installing:

```
sudo apt-get autoremove monodevelope (only if you installed it)
```

```
sudo apt-get autoremove libgdiplus
```

#### **Hint**

*libgdiplus removes mono-complete as well.*

#### **1.4.2.2 Compile Mono from TAR**

This will compile Mono from a TAR package.

#### **Procedure**

- Install additional needed packages  
`sudo apt-get install libtool libglib2.0-dev libxrender-dev libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev libexif-dev gettext libcairo2-dev`
- Install libgdiplus  
`sudo apt-get install libgdiplus`  
(You can also compile the latest version. See 'Compile Mono from GIT' for details.)
- Create a new folder where we place the files. You can use your home directory for that  
`cd ~`  
`mkdir sources`  
`cd sources`
- Check for the latest TAR version  
`wget -qO- http://download.mono-project.com/sources/mono/ |grep -o -E 'href="([^\"]+)"' | cut -d'"' -f2 | sort | uniq`  
or just watch at the website  
<http://origin-download.mono-project.com/sources/mono/>  
<http://download.mono-project.com/sources/mono/>
- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)  
`wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2`
- Extract the TAR file  
`tar -jxvf mono-3.6.0.tar.bz2`
- Enter the new directory  
`cd mono-3.6.0`
- run configure  
`./configure --prefix=/usr/local/`  
/usr/local/ is the installation path of the compiled files
- run make - compile all the files  
`sudo make`
- run make install - this will install the needed files into your system (this will take ~10 minutes)  
`sudo make install`

## Uninstall

- Run the uninstall  
`sudo make uninstall`
- remove the source files  
`rm -rf ~/sources/mono`

## Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv  
<https://gist.github.com/pjvds/7185693>

### 1.4.2.3 Compile Mono from GIT

This will compile Mono from the GIT repository.

**Hint**

*Since this is updated very often it could include some bugs. But in most cases it works great.*

**Procedure**

- Install additional needed packages  
`sudo apt-get install libtool libglib2.0-dev libxrender-dev libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev libexif-dev gettext libcairo2-dev`
- Create a new folder where we place the files. You can use your home directory for that  
`cd ~`  
`mkdir sources`  
`cd sources`

**libgdiplus Compilation**

- Get the libgdiplus sources from GIT  
`git clone git://github.com/mono/libgdiplus`
- Enter the new directory  
`cd libgdiplus`
- Run autogen  
`./autogen.sh --prefix=/usr`
- Compile the sources  
`sudo make`
- Install the files  
`sudo make install`
- Leave the directory  
`cd ..`

**Mono Compilation**

- Get the Mono sources from GIT  
`git clone git://github.com/mono/mono.git`
- Enter the new directory  
`cd mono`
- Run autogen  
`./autogen.sh --prefix=/usr/local`
- Install and configure gmcs  
`sudo make get-monolite-latest`  
`cd mcs/class/lib/monolite`  
`cp basic.exe gmcs.exe`  
`cd ~/sources/mono`

**Hint**



*You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.*

- Compile the sources (~15 minutes)  
If you used `sudo make get-monolite-latest` use this:  
`sudo make EXTERNAL_MCS=~/.sources/mono/mcs/class/lib/monolite/gmcs.exe"`  
If you use `apt-get install mono-gmcs` use this:  
`sudo make`
- Install the files  
`sudo make install`

## Uninstall

- Run the uninstall  
`sudo make uninstall`
- remove the source files  
`rm -rf ~/.sources/mono`

## Additional Information

Mono Compile from Git

<http://stackoverflow.com/questions/13365158/installing-mono-3-x-3-0-x-and-or-3-2-x>  
<https://gist.github.com/pjvds/7185693>

## 1.5 CentOS

The CentOS Linux distribution is a stable, predictable, manageable and reproduceable platform derived from the sources of Red Hat Enterprise Linux (RHEL)



### System

x86 / X64

### Webpage

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

### 1.5.1 System Preparation

For this documentation we use CentOS 6.5 (<http://www.centos.org/>). It should be possible to use this topic for other Redhat based systems as well.

#### Download

You can download 32/64 Bit images from the CentOS website:  
<http://www.centos.org/download/>

#### System Installation

There are tons of information in the internet how to install CentOS. We won't cover the details here.

<http://wiki.centos.org/>

<http://www.centos.org/docs/>

#### System Preparation

Ensure that you use English locals until LogView Studio runs!

From this point it would be Ok to use [SSH to connect to Ubuntu](#) only.

Working on the machine directly would be fine, too.

#### Software Install / Update

Command may need `su -` for admin rights.

if the network is down use

`ip addr`

`dhclient ethXX` (XX for your interface)

- Install nano (give vi a kick ...)  
`yum install nano`
- Update the installed software packages (this may take a while)  
`yum update`
- Install some additional useful packages (maybe some are preinstalled)  
`yum install gnome-sudoku git-core gcc* gcc-c++ kernel-devel make  
autoconf automake screen curl binutils wget unzip bzip2`  
(It's always a good idea to have [Sudoku](#) installed :-)

#### Hints

General Information about software installation / updates with yum

[https://www.centos.org/docs/5/html/5.1/Deployment\\_Guide/s1-yum-useful-commands.html](https://www.centos.org/docs/5/html/5.1/Deployment_Guide/s1-yum-useful-commands.html)

### 1.5.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common

Language Runtime.

<http://www.mono-project.com>

## How to get Mono?

### **Hint**

*CentOS has no package for Mono. You must compile it by yourself.*

There are at least two different ways for getting Mono installed on your Cubietruck. All of them have their pros and cons:

### **Compile it from a TAR package**

- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
- o mostly newer than the Raspbian package but older than the GIT version
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

### **Compile it from the GIT repository**

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- needs some extra packages for compiling

## Which method is the best?

For x86/x64 the best choice is to compile it from GIT because it takes not really much time.

### **Note**

*If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.*

### 1.5.2.1 Compile Mono from TAR

This will compile Mono from a TAR package. **It takes about 4 hours in sum.**

### **Procedure**

- Get admin rights  
`su -`
- Preparation for installation ...  
`yum install glib2-devel libpng-devel libjpeg-devel giflib-devel libtiff-devel libX11-devel fontconfig-devel bison gettext libtool cairo-devel`
- Create a new folder where we place the files. You can use the tmp directory for that  
`mkdir /tmp/sources`  
`cd /tmp/sources`

### **libgdiplus Compilation**

- Get the libgdiplus sources from GIT

```
git clone git://github.com/mono/libgdiplus
```

- Enter the new directory  

```
cd libgdiplus
```
- Run autogen  

```
./autogen.sh --prefix=/usr
```
- Compile the sources  

```
make
```

Solving "libtool: Version mismatch error"

```
autoreconf -ivf
```
- Install the files  

```
make install
```
- Leave the directory  

```
cd ..
```

### Mono Compilation

- Check for the latest TAR version  

```
wget -qO- http://download.mono-project.com/sources/mono/ |grep -o -E 'href="([^\"]+)"' | cut -d'"' -f2 | sort | uniq
```

or just watch at the website  
<http://origin-download.mono-project.com/sources/mono/>  
<http://download.mono-project.com/sources/mono/>
- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)  

```
wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2
```
- Extract the TAR file  

```
tar -jxvf mono-3.6.0.tar.bz2
```
- Enter the new directory  

```
cd mono-3.6.0
```
- run configure  

```
./configure --prefix=/usr/local/
```

/usr/local/ is the installation path of the compiled files
- run make - compile all the files  

```
sudo make
```
- run make install - this will install the needed files into your system (this will take ~10 minutes)  

```
sudo make install
```

### Uninstall

- Run the uninstall  

```
sudo make uninstall
```
- remove the source files  

```
rm -rf ~/sources/mono
```

### Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv  
<https://gist.github.com/pjvds/7185693>

### 1.5.2.2 Compile Mono from GIT

CentOS has no package for Mono. This will compile Mono from the GIT repository.

**Hint**

*Since this is updated very often it could include some bugs. But in most cases it works great.*

#### Procedure

- Get admin rights  
`su -`
- Preparation for installation ...  
`yum install glib2-devel libpng-devel libjpeg-devel giflib-devel libtiff-devel libX11-devel fontconfig-devel bison gettext libtool cairo-devel`
- Create a new folder where we place the files. You can use the tmp directory for that  
`mkdir /tmp/sources`  
`cd /tmp/sources`

#### libgdiplus Compilation

- Get the libgdiplus sources from GIT  
`git clone git://github.com/mono/libgdiplus`
- Enter the new directory  
`cd libgdiplus`
- Run autogen  
`./autogen.sh --prefix=/usr`
- Compile the sources  
`make`  
Solving "libtool: Version mismatch error"  
`autoreconf -ivf`
- Install the files  
`make install`
- Leave the directory  
`cd ..`

#### Mono Compilation

- Get the Mono sources from GIT  
`git clone git://github.com/mono/mono.git`
- Enter the new directory  
`cd mono`
- Run autogen  
`./autogen.sh --prefix=/usr/local`
- Install and configure gmcs  
`sudo make get-monolite-latest`  
`cd mcs/class/lib/monolite`  
`cp basic.exe gmcs.exe`  
`cd ../../../../..`

**Hint**

*You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.*

- Compile the sources (15 minutes)  
If you used `sudo make get-monolite-latest` use this:  
`sudo make EXTERNAL_MCS="/tmp/sources/mono/mcs/class/lib/monolite/gmcs.exe"`  
If you use `apt-get install mono-gmcs` use this:  
`sudo make`
- Install the files  
`sudo make install`

**Uninstall**

- Run the uninstall  
`sudo make uninstall`
- remove the source files  
`rm -rf ~/sources/mono`

**Additional Information**

Installing Mod\_Mono and Xsp4 on CentOS 6.3

<http://stackoverflow.com/questions/14901271/installing-mod-mono-and-xsp4-on-centos-6-3>

<http://wiki.phonicuk.com/Installing-Mono-in-CentOS-5-x.ashx>