Installation

This Wiki page describes the system requirements for DomotiGa and all the steps to install DomotiGa on your system, including all 3rd party dependencies.

The installation consists of the following steps:

- System Requirements
- Operating System
- Linux Configuration
- Gambas
- MySQL Server
- DomotiGa

System Requirements

DomotiGa will run on any system which can run Gambas, this includes the most widely used Linux distributions.

Linux Distribution

The following list of distributions have been successfully tested with DomotiGa:

Distribution	Architecture	End Of Support	DomotiGa Supported?
Ubuntu 14.04	x86 / x86_64	April 2019	Yes
Ubuntu 16.04	x86 / x86_64	April 2021	Yes
Ubuntu 17.10	x86 / x86_64	July 2018	Yes
Ubuntu 18.04 TLS	x86 / x86_64	April 2023	Yes *1
Debian 8 (Jessie)	x86 / x86_64		Yes
Debian 9 (Stretch)	x86 / x86_64		Yes
Raspbian (Jessie)	Raspberry (ARMv6, ARMv7, ARMv8)		Yes
Raspbian (Stretch)	Raspberry (ARMv6, ARMv7, ARMv8)		Yes
Armbian (Jessie)	Cubieboard2 (ARMv7)		Yes

x86 is a 32-bits and x86_64 is a 64-bits architecture

For a full list of Linux distributions which should work with Gambas, see the following link: http://gambasdoc.org/help/doc/distro

*1 Requires Gambas 3.11 or higher

Hardware

DomotiGa will run on any hardware which supports Linux and Gambas3. The following hardware has been successfully tested:

Hardware
Any Inte

l / AMD x86 CPU
Any Intel / AMD x64 CPU
Raspberry Pi (Model B, B+)
Raspberry Pi 2 (Model B)
Raspberry Pi 3 (Model B)
Cubieboard2
Cubietruck
Phoenix A20

It is recommended to have at least 512MB of memory for DomotiGa and 256MB of memory for DomotiGaServer. Also a minimum 500MB of diskspace is required for the DomotiGa installation.

Virtualization environments like VMware, VirtualBox and Parallels have been successfully tested with DomotiGa.

Operating System

It is recommended to install the Linux Operating System with the standard packages, including the graphical desktop (GNOME, KDE, LXDE, Unity, etc). The graphical desktop is required to run DomotiGa. The DomotiGaServer will run without the graphical desktop installed, but the customized installation is only recommended for advanced users.

The following table contains links to the installation procedures for the main Linux distributions:

Distribution	Link to Install manuals
Ubuntu (Desktop)	https://help.ubuntu.com/community/GraphicalInstall
	https://help.ubuntu.com/community/Installation
Lubuntu (Desktop)	https://help.ubuntu.com/community/Lubuntu/InstallingLubuntu
Xubuntu (Desktop)	
Debian (Desktop, x86 / x86_64)	http://www.debian.org/releases/stable/installmanual
	http://wiki.debian.org/LennyIllustratedInstall
Debian (Desktop, Raspberry Pi)	http://elinux.org/RPi_Easy_SD_Card_Setup
	http://elinux.org/RPi_raspi-config
Armbian (cubieboard2, cubietruck,...)	https://docs.armbian.com/User-Guide_Getting-Started/
direct download link A20:	https://www.armbian.com/donate/?f=http://image.armbian.com/Armbian_5.20_Cubieboard2_Debian_jessie_4.7.3.7z

Installation of Gambas, MySQL Server and DomotiGa are done via the command line. If you are inexperienced with this, please read the section **Starting a Terminal** in the following link: https://help.ubuntu.com/community/UsingTheTerminal

Linux Configuration

After the Operating System is installed a few options are highly recommended to install/configure on the system. These are NTP, SSH and USB Device permissions.

NTP

It is important that your system has the right time, you can achieve this by running ntpd (Network Time Protocol daemon). The following steps will install, configure and check it.

First install the required software package:

```
$ sudo apt-get install ntp
```

The next step is optional, but still highly recommended. This will replace the generic “ntp.ubuntu.com” server with the nearby NTP servers:

```
$ sudo nano /etc/ntp.conf
```

Replace the line with the following ones, the example is if you live in the Netherlands. For a complete list for your local NTP servers, check the following link (and go to the active servers): http://www.pool.ntp.org

```
server 0.nl.pool.ntp.org
server 1.nl.pool.ntp.org
server 2.nl.pool.ntp.org
server 3.nl.pool.ntp.org
```

Restart the NTP daemon:

```
$ sudo /etc/init.d/ntp restart
* Stopping NTP server ntpd                                            [ OK ]
* Starting NTP server ntpd                                            [ OK ]
```

Now check if you system is in synchronization with the NTP servers:

```
$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 metronoom.dmz.c PPS(0)           2 u   18   64    1   28.042  36694.5   0.001
 ntp1.hro.nl     193.67.79.202    2 u   17   64    1   30.153  36694.5   0.001
 canopus.no-sens 192.87.106.3     2 u   16   64    1   26.090  36694.6   0.001
 ntp1.trans-ix.n 193.79.237.14    2 u   15   64    1   25.615  36696.2   0.001
```

The offset is the difference between your servers clock and the atom clock time. After a few minutes you will see that the difference is corrected.

```
$ ntpq -p
 remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*metronoom.dmz.c PPS(0)           2 u   10   64    1   26.920   -0.251   0.001
 ntp1.hro.nl     193.67.79.202    2 u   25   64    1   30.268    0.648   0.001
 canopus.no-sens 192.87.106.3     2 u   25   64    1   26.651    0.273   0.001
 ntp1.trans-ix.n 193.79.237.14    2 u   49   64    1   24.986    2.846   0.001
```

Star (*) before server address indicates that you are synchronized with this NTP server (metronoom.dmz.c in this example).

SSH

By default most Linux distributions don't install the ssh daemon (server), which is required to remotely access the system. If you want to access your system remotely via command-line install ssh daemon with the following command:

```
$ sudo apt-get install ssh
```

USB Device Permissions

It is highly recommended to run DomotiGa not under the “root” user. By default the standard user account(s) can't read and/or write to the USB serial devices mainly used by DomotiGa. The following command will add the “dialout” group to your user account to give the right permissions, in the example the user account is named “domo”:

```
$ sudo addgroup domo dialout
```

> After you added the “dialout” group to your user account, you need to logout and login again to activate this change.

RRDTool

The Round Robin Database Tool (RRDTool) is the open source industry standard, high performance data logging and graphing system for time series data. With it you can create graphs of the collected data. It is highly recommended to install it, and can be done with the following command:

```
$ sudo apt-get install rrdtool
```

Remote Desktop Connection W7 to (X)Ubuntu

To connect the Domotiga Server with the (X)ubuntu desktop you can use the Remote Desktop Connection application from W7. The only software to activate on the client is the 'xrdp', you can install it by shell prompt or app installation in the Ubuntu GUI

Gambas

> DomotiGa version 1.0.018 and up will only run with Gambas 3.6.x and higher, this Gambas version is backwards compatible, so you are encouraged to upgrade your Gambas installation in advance.

Gambas is a free development and runtime environment based on a Basic interpreter, a bit like Microsoft Visual Basic (but it is not a clone) – http://gambas.sourceforge.net/. DomotiGa is written in Gambas and requires it to run. The latest stable version of Gambas is

3.11.3, this is the recommended version to use with DomotiGa.

There are 2 ways of installing Gambas:

- **From PPA (Personal Package Archives)** – This archive contains pre-compiled Gambas packages for Ubuntu. This is the fastest and easiest method of installing, but isn't available for all Ubuntu releases and/or architecture (then you need to compile it from source). This is described in section “Gambas From PPA”
- **Compile from Source** – Latest regular release. This is described in section “Gambas From Source“

Gambas From PPA (x86, x86_64, Cubieboard, PhoenixA20, etc)

Installing Gambas from an Ubuntu PPA is the fastest and easiest way. This will only work on Ubuntu and still maintained release (e.g. Ubuntu 11.10, 12.10 are End Of Life already). Normally the x86, x86_64 and Cubieboard2/Cubietruck/PhoenixA20 architectures are supported. The following commands will install Gambas3 from the PPA.

```
$ sudo apt-add-repository ppa:gambas-team/gambas3
$ sudo apt-get update
$ sudo apt-get install gambas3
```

> If your setup can't use the Ubuntu PPA, you need to compile Gambas from source, as described in the next section.

Gambas From PPA (Raspberry Pi, Pi 2, Pi 3)

Currently there are 3 versions of Raspbian for the Raspberry Pi, named "Wheezy", "Jessie" and "Stretch". To check which version you're running, check the */etc/debian_version* file (Wheezy=7.x, Jessie=8.x, Stretch=9.x). The following steps will configure the Stretch (latest) version, but if you are running the older *Wheezy* or *Jessie*, just replace *stretch* with *wheezy* or *jessie*.

```
$ wget -O - https://apt.domotiga.nl/public.gpg.key | sudo apt-key add -
$ sudo su -c "echo 'deb https://apt.domotiga.nl stretch main' >> /etc/apt/sources.list"
$ sudo apt-get install -y apt-transport-https
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install gambas3
```

Gambas from Source

NOTE: If you have installed older version of Gambas from source and prepare to compile and install a newer version, uninstall the old version before installing the new one. See "Uninstalling old gambas version" at the bottom of this chapter.

This section describes the compilation from source and installation of Gambas. To compile from source, we need to first install library dependencies. The specific list of dependencies are dependant on Linux distribution and release.

Library Dependencies

The specific list of dependencies are dependant on Linux distribution and release, please select your distribution and release install them. If your release is newer then in the list below, choose the latest one below.

Ubuntu 12.04

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libcurl4-gnutls-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird2.1-dev libqt4-dev libglew1.5-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-base0.10-dev \
  libgmp-dev libopenal-dev libalure-dev libgtk-3-dev
```

Ubuntu 14.04 LTS / Ubuntu 15.04 / Ubuntu 15.10 / Debian 8.1 / Raspbian Jessie (Raspberry Pi / Pi 2 / Pi 3)

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libcurl4-gnutls-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird-dev libqt4-dev libglew-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-base0.10-dev \
  libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgmp-dev \
  libopenal-dev libalure-dev libgtk-3-dev libsdl2-image-dev libsdl2-ttf-dev \
  llvm-dev libpq-dev libsdl2-mixer-dev postgresql-server-dev-all libpoppler-private-dev \
  libqt5webkit5-dev libqt5opengl5-dev libqt5svg5-dev libqt5x11extras5-dev
```

Ubuntu 16.04 LTS / Ubuntu 16.10

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird-dev libqt4-dev libglew-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-base0.10-dev \
  libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgmp-dev \
  libopenal-dev libalure-dev libgtk-3-dev libsdl2-image-dev libsdl2-ttf-dev \
  llvm-dev libpq-dev libsdl2-mixer-dev postgresql-server-dev-all libpoppler-private-dev \
  libqt5webkit5-dev libqt5opengl5-dev libqt5svg5-dev libqt5x11extras5-dev autopoint \
  dh-autoreconf libcurl4-openssl-dev libllvm3.5v5 libmodplug-dev libmpg123-dev \
  libqtwebkit-dev libsdl-gfx1.2-5 libsdl-gfx1.2-dev
```

Ubuntu 17.10

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird-dev libqt4-dev libglew-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-good1.0-dev \
  libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgmp-dev \
  libopenal-dev libalure-dev libgtk-3-dev libsdl2-image-dev libsdl2-ttf-dev \
  llvm-dev libpq-dev libsdl2-mixer-dev postgresql-server-dev-all libpoppler-private-dev \
  libqt5webkit5-dev libqt5opengl5-dev libqt5svg5-dev libqt5x11extras5-dev autopoint \
  dh-autoreconf libcurl4-openssl-dev  libmodplug-dev libmpg123-dev \
  libqtwebkit-dev libsdl-gfx1.2-5 libsdl-gfx1.2-dev
```

Ubuntu 18.04.1 LTS

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird-dev libqt4-dev libglew-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-good1.0-dev \
  libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgmp-dev \
  libopenal-dev libalure-dev libgtk-3-dev libsdl2-image-dev libsdl2-ttf-dev \
  llvm-dev libpq-dev libsdl2-mixer-dev postgresql-server-dev-all libpoppler-private-dev \
  libqt5webkit5-dev libqt5opengl5-dev libqt5svg5-dev libqt5x11extras5-dev autopoint \
  dh-autoreconf libcurl4-openssl-dev  libmodplug-dev libmpg123-dev \
  libqtwebkit-dev libsdl-gfx1.2-5 libsdl-gfx1.2-dev
```

Copy gambas icon and application

```
$ sudo cp app/src/gambas3/img/logo/logo-ide.png /usr/share/pixmaps/gambas3.png
$ sudo cp app/desktop/gambas3.desktop /usr/share/applications
```

Raspbian Wheezy (Raspberry Pi)

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libcurl4-gnutls-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird2.1-dev libqt4-dev libglew1.5-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-base0.10-dev \
  libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \
  libpoppler-cpp-dev libpoppler-private-dev libpoppler-qt4-dev libpangox-1.0-dev \
  libgtk-3-dev libgmp-dev libopenal-dev libalure-dev
```

Raspbian Stretch (Raspberry Pi / Pi 2 / Pi 3)

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 default-libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libcurl4-gnutls-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird-dev libqt4-dev libglew-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-base0.10-dev \
  libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgmp-dev \
  libopenal-dev libalure-dev libgtk-3-dev libsdl2-image-dev libsdl2-ttf-dev \
  llvm-dev libpq-dev libsdl2-mixer-dev postgresql-server-dev-all libpoppler-private-dev \
  libqt5webkit5-dev libqt5opengl5-dev libqt5svg5-dev libqt5x11extras5-dev
```

Debian 7.8

```
$ sudo apt-get install subversion build-essential autoconf mysql-client libbz2-dev \
  libfbclient2 libmysqlclient-dev unixodbc-dev libpq-dev libsqlite0-dev libsqlite3-dev \
  libgtk2.0-dev libldap2-dev libcurl4-gnutls-dev libgtkglext1-dev libpcre3-dev \
  libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-dev libxml2-dev \
  libxslt1-dev libbonobo2-dev libcos4-dev libomniorb4-dev librsvg2-dev libpoppler-dev \
  libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev libaa1-dev \
  libxtst-dev libffi-dev kdelibs5-dev firebird-dev libqt4-dev libglew-dev \
  libimlib2-dev libv4l-dev libsdl-ttf2.0-dev libgnome-keyring-dev libgdk-pixbuf2.0-dev \
  linux-libc-dev libgmime-2.6-dev libgsl0-dev libgstreamer-plugins-base0.10-dev \
  libgmp-dev libopenal-dev libalure-dev libgtk-3-dev llvm-dev libpq-dev \
  postgresql-server-dev-all libpoppler-private-dev
```

Debian 8.6 (armbian)

```
sudo apt-get install build-essential g++ automake autoconf libbz2-dev libmysqlclient-dev unixodbc-dev \
libpq-dev libsqlite0-dev libsqlite3-dev libglib2.0-dev libgtk2.0-dev libcurl4-gnutls-dev libgtkglext1-dev \
libpcre3-dev libsdl-sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-dev libxml2-dev \

libxslt1-dev librsvg2-dev libpoppler-dev libpoppler-glib-dev libasound2-dev libesd0-dev libdirectfb-dev \
libxtst-dev libffi-dev libqt4-dev libglew-dev libimlib2-dev libv4l-dev libsdl-ttf2.0-dev \
libgnome-keyring-dev libgdk-pixbuf2.0-dev linux-libc-dev libgstreamer1.0-dev \
libgstreamer-plugins-base1.0-dev libcairo2-dev libgsl0-dev libncurses5-dev libgmime-2.6-dev \
llvm-dev libalure-dev libgmp-dev libgtk-3-dev libsdl2-dev libsdl2-mixer-dev libsdl2-ttf-dev \
libsdl2-image-dev sane-utils libdumb1-dev libqt5opengl5-dev libqt5svg5-dev libqt5webkit5-dev \
libqt5x11extras5-dev qtbase5-dev libwebkit-dev libqtwebkit-dev libtool
```

CentOS 7

```
# yum groupinstall "Development Tools" "Development Libraries"
# yum install automake gcc make libtool cairo-devel gtk2-devel librsvg2-devel \
   poppler-devel qt-devel libv4l-devel SDL-devel unixODBC-devel postgresql-devel \
   libXtst-devel mysql-devel pcre-devel mesa-libGLU-devel mesa-libGLw-devel \
   mesa-libGL-devel glew-devel dbus-devel libzip libzip-devel bzip2-devel libcurl-devel \
   libxml2-devel libxslt-devel libv4l-devel libjpeg-turbo-devel ncurses-devel gsl-devel \
   libXtst-devel gstreamer-devel gstreamer-plugins-base-devel libtool-ltdl-devel \
   libffi-devel
```

It is suggested to also enable the EPEL repository if using CentOS 7, because this supplies the "gmime" libaries.

The EPEL can be added in the following way:
https://www.tecmint.com/how-to-enable-epel-repository-for-rhel-centos-6-5/

And the "gmime" can be installed as follows:

```
# yum install gmime-devel gmime
```

Download Gambas

After the dependencies are installed, we will download the Gambas source, as latest released version. After the download, the source will be compiled and installed on Linux.

Create temporary directory
We will create a temporary directory for the Gambas installation. The Gambas source will be downloaded and compiled here.

```
$ mkdir -p ~/install
```

Download Gambas Latest Release
After the temporary directory is created, we will download the latest Gambas release. The following commands will do that.

```
$ cd ~/install
$ wget https://gitlab.com/gambas/gambas/-/archive/3.11.3/gambas-3.11.3.tar.bz2
$ tar xvjf gambas-3.11.3*.tar.bz2
$ cd gambas-3.11.3*
```

Or you can download latest source with git

```
$ cd ~/install
$ git clone https://gitlab.com/gambas/gambas.git
$ cd gambas
```

Configure Gambas

After we successfully downloaded the Gambas source, the build environment needs to be configured. Depending on the hardware, this can take some time (on a Cubieboard2 with Gambas 3.6.1 about 80 minutes, on a RaspberryPi 3 hours and 15 minutes and <15 minutes on a i5-4570S @ 2.90GHz), so please be patient.

```
$ ./reconf-all && ./configure -C
```

The configuration should run successfully and the following message can be safely ignored (normally the disabled components aren't used/needed by DomotiGa):

```
||
|| ALL COMPONENTS WILL BE COMPILED
||
```

Compile Gambas

After the build environment is prepared, we can compile the Gambas source to binary format. Depending on the hardware, this can take some time (on a Cubieboard2 with Gambas 3.6.1 about 30 minutes, on a Raspberry Pi... 2 hours 10 minutes), so please be patient.

```
$ make      (single core)
$ make -jX  (with X being the number of cores)
```

Install Gambas

The last step will install the Gambas binaries into the right directories. Depending on the hardware, this can take some time, (On a Raspberry Pi around 10 minutes) please be patient.

```
$ sudo make install
```

Test Gambas

As final step we will test if Gambas can be started successfully. This should launch the Gambas IDE (Integrated Development Environment).

```
$ gambas3
```

When it shows the following screen, Gambas is successfully installed and working:

If things do not work

If you get something like:

```
gbc: error: Component not found: gb.desktop.x11
```

Check the permissions on these components in /usr/share/gambas3/info/ (ls -l or stat -c "%a %n" *). The files should be 644 or -rw-r--r--, watch for files being 640 or -rw-r----- and if so change them to have the right permissions.

Uninstalling old Gambas version

If you want to install a newer version of Gambas from source, it's wise to uninstall the old version first to make sure old components are uninstalled and do not interfere.

```
$ cd gambas3-3.8.3
$ sudo make uninstall
```

MySQL Server

DomotiGa uses the MySQL database to store its data in. The MySQL server hasn't been installed yet, this can be done with the following command.

```
$ sudo apt-get install mysql-client mysql-server
```

At the installation of MySQL the new password for the MySQL “root” user is asked, choose a new none-standard password for optimal security. This password will be used later to setup the DomotiGa tables into MySQL.

Increase MySQL Connection Time-Out

DomotiGa Client and back-end require a persistent connection to the MySQL database. Normally DomotiGa generate enough database queries and updates to the connection open, but it is recommended to increase the connection time-out to a higher value. The MySQL configuration file is usually found in “/etc/mysql/my.cnf”. Execute the following command on the DomotiGa back-end system to edit the MySQL configuration file:

```
$ sudo nano /etc/mysql/my.cnf
```

Check if the following line already exists, else insert it into the [mysqld] section:

```
wait_timeout            = 2147483
```

Make databases available to clients

If you want to connect with DomotiGa clients other than DomotiYii, you need to let the database daemon listen to all network interfaces, instead of just localhost/127.0.0.1

Change the bind-address value to this

```
bind-address            = 0.0.0.0
```

If unfamiliar with the “nano” editor, please check the following link for a cheat sheet: http://www.tuxradar.com/content/text-editing-nano-made-easy

Restart the MySQL Server to use the new configuration with the following command:

```
$ sudo /etc/init.d/mysql restart
```

DomotiGa

Install git before you download DomotiGa, install it with:

```
$ sudo apt-get install git-core
```

DomotiGa can be easily downloaded and installed with the following commands:

```
$ cd ~
$ mkdir install
$ cd install
$ git clone https://github.com/DomotiGa/DomotiGa.git domotiga
$ cp -rp ~/install/domotiga ~
```

> Never run DomotiGa as “root” user.

If you don't want to develop, remove git files like so:

```
$ cd ~/domotiga
$ rm -rf `find . -type d -name .git`
```

If you want to check out a different branch (beta,develop m.m) add a -b switch to git clone

```
$ git clone -b beta https://github.com/DomotiGa/DomotiGa.git domotiga
```

Or if you already did a normal git clone goto this directory and simply type

```
$ git checkout beta
```

With git branch you can check which branch is active (which files do you look at)

```
$ git branch
* beta
  master
  pushbullet
  enocean
```

Error executing the **comment_form** macro (Missing partial wiki_extensions/_comment_form with {:locale=>[:nl, :en], :formats=>[:pdf], :variants=>[], :handlers=>[:erb, :builder, :raw, :ruby, :rsb]}. Searched in: *

"/home/redmine/redmine/plugins/redmine_wiki_extensions/app/views" * "/home/redmine/redmine/plugins/redmine_login_audit/app/views" * "/home/redmine/redmine/plugins/redmine_local_avatars/app/views" * "/home/redmine/redmine/plugins/hooks_manager/app/views" * "/home/redmine/redmine/app/views" * "/home/redmine/data/tmp/bundle/ruby/2.3.0/gems/wice_grid-3.5.0/app/views" * "/home/redmine/data/tmp/bundle/ruby/2.3.0/gems/kaminari-0.17.0/app/views")

Updated by: 4rdnzl, 12 maanden geleden bijgewerkt
Access count: 126181

Attached Files

Bestanden

Gambas3-logo.png	40.4 KB	29-07-2013	Alexie
mysql.png	1.41 KB	30-07-2013	Alexie
gambas3-startup.png	143 KB	02-08-2015	Alexie