

Asuro Robot - Linux

Introduction:

This manual describes how you can install a development environment for the Asuro Robot on a Linux system. This installation is tested on Ubuntu 16.04 LTS (64 bit) and Linux Mint 17.3 (64 bit).

1. Geany

- Install Geany via the Software centre.

Geany is an IDE (Integrated Development Environment), which can be used to write programs for the Asuro Robot. The programming language for the Asuro Robot is the language C.

2. Wine

- Install Wine via the Software centre.
- Start Wine configuration. This action will install the necessary configuration folder.

Wine is a program that makes it possible to use a Windows program on a Linux system. For the flashing of the Asuro Robot we use the Windows version of the flash program, this because the Linux version will not work on a modern Linux system.

3. Cutecom

- Install Cutecom via the Software centre.

Cutecom is a program for communications via the serial port. The program Cutecom makes it possible to receive data from or send data to the Asuro Robot.

Remark: If the program Cutecom is active the flashing of the Asuro Robot will not work.

4. gcc-avr

- Install gcc-avr via the Software centre.

Gcc-avr is a supplement to the gcc compiler that is being used to convert a program that is written in the programming language C to machine-code for the AVR micro-controller on the Asuro Robot. The gcc compiler itself is already installed on the most Linux systems.

5. Asuro Robot source files

- Copy from the CD the folder ASURO_src and the program ASUROFlash155.exe to a folder on the PC.
- Download the AsuroLib version 2.6.1 and unpack the file:
<https://sourceforge.net/projects/asuro/files/AsuroLib/>
- Copy the file /lib/inc/asuro.h to the folder ASURO_src/SelfTest/
- Open the file ASURO_src/SelfTest/asuro.h with Geany.
Find the line: `#include` and alter the settings.
`#include </usr/lib/avr/include/avr/io.h>`
`#include </usr/lib/avr/include/avr/interrupt.h>`
Save the file.

- Copy the file ASURO_src/SelfTest/asuro.h to the folder ASURO_src/FirstTry/.
- Copy the file /lib/src/asuro.c to the folder ASURO_src/SelfTest/
- Open the file ASURO_src/SelfTest/asuro.c with Geany.
Replace SIGNAL (SIG_OVERFLOW2) with ISR (TIMER2_OVF_vect).
Replace SIGNAL (SIG_INTERRUPT1) with ISR (INT1_vect).
Replace SIGNAL (SIG_ADC) with ISR (ADC_vect).
Save the file.
- Copy the file ASURO_src/SelfTest/asuro.c to the folder ASURO_src/FirstTry/.
- Open the file ASURO_src/SelfTest/makefile with Geany.
Find the line: # **Define directories, if needed** and alter the settings.
DIRAVR = /usr/lib/avr
DIRAVRBIN = \$(DIRAVR)/bin
DIRAVRUTILS = \$(DIRAVR)/include/utils/bin
DIRINC = .
DIRLIB = \$(DIRAVR)/lib
Save the file.
- Copy the file ASURO_src/SelfTest/makefile to the folder ASURO_src/FirstTry/.

6. USB infra-red transceiver

- Connect the USB Infra-red transceiver to a USB port.
- Enter the following in a Terminal.
dmesg | grep tty
The result will be something like this:
[0.000000] console [tty0] enabled
[519.009508] usb 3-3: FTDI USB Serial Device converter now attached to ttyUSB0
The USB infra-red transceiver is recognized by Linux on port: ttyUSB0.
- Link the USB infra-red transceiver to a Windows com port. This will be used later by Wine.
Enter the following in a Terminal.
ln -s /dev/ttyUSB0 ~/.wine/dosdevices/com3
- Make the tty port accessible to the user. In this example the user is: richard.
Enter the following in a Terminal.
sudo adduser richard dialout
- Now sign off and on as an user to update all the changes.

7. Flashing of the Asuro Robot

- Open the program ASUROFlash155.exe with the Wine program-loader.
- Select the com port, in this example number 3.
- Select the program ASURO_src/SelfTest/SelfTest.hex
- Put the USB infra-red transceiver next to the Asuro Robot.
- Click on OK to start the flashing.
- Switch on the Asuro Robot directly after that. And wait until the flashing has been completed successfully.
- Switch the Asuro Robot off and on to start the “new” program.