Table of Contents

1. Operation system of Willy

1.1. Configuration

Currently, Willy is operating using 'Linux Ubuntu' as operation system. The ROS-framework is used for centralized communication between nodes.

For more information about nodes and the working of it, see our ROS Introduction wiki page.

Nodes are referred as different software/hardware components of Willy: think of; GPS, sensors, compass and software functions. Some nodes may require

```
dependent-ROS-packages
```

To execute and compile software nodes, these packages are required. Both ROS and dependencies require the same version. The current version of ROS is

```
'ROS-kinetic'
```

1.2. Current installed packages

The first step in updating the OS is to determine which packages and dependencies are installed. Because previous teams made a lot of changes to Willy, not all Linux packages may still be required.

To list the manually installed packages, the following command was used.

```
_comm -23 <(apt-mark showmanual | sort -u) <(gzip -dc
/var/log/installer/initial-status.gz | sed -n 's/^Package: //p' | sort
-u)_
```

The following packages where manually installed;

```
brightness-controller
dhcpcd5
dotnet-sdk-2.0.0
git
google-chrome-stable
htop
nmap
openssh-server
pgadmin3
postgresql
python-pip
ros-kinetic-desktop-full
ros-kinetic-joystick-drivers
ros-kinetic-rosbridge-server
ros-kinetic-rosserial-python
ros-kinetic-move-base
ros-kinetic-hector-mapping
ros-kinetic-sicktim
ros-kinetic-opencv
ros-kinetic-openni
ros-kinetic-rosserial-server
ros-kinetic-teleop-twist-joy
ros-kinetic-teleop-twist-keyboard
ros-lunar-catkin
screen
vsftpd
x11vnc
xfce4
xrdp
```

Some of the above packages are required to compile and execute the WTGD code that is available from GIT. These packages are listed bold. Other packages may be required for the web platform or may have another goal than compiling and executing ROS code. In this project there will be major changes in the web platform, because some of the code will be changed. Dependencies will be determined during the development of the code. Other packages are explained in documentation that will be available with the final delivery.

1.3. Determined packages for Ubuntu 16.04 and ROS-kinetic

To create a clear view about the current 'WTGD' code that Willy contains, and how this works on Ubuntu 16.04 with ROS-Kinetic, a test environment was created. None of the previous listed packages where installed, only Ubuntu 16.04 was installed and the ROS-framework on top of this. The code was not able to build successful in this test environment, however based on the error messages, dependencies where determined. Every time a dependency was missing, the error massage was inspected. We concluded that the following packages are required to execute the 'WTGD' code on Willy.

```
ROS-kinetic-desktop-full
Screen
ROS-kinetic-rosserial
ROS-kinetic-rosserial-arduino
ROS-kinetic-rosapi
```

1.4. Install dependencies

To install above dependencies, the following commands are required.

Install ROS:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release
-sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80
--recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
```

```
sudo apt-get update
sudo apt-get install -y ros-kinetic-desktop-full
sudo rosdep init
rosdep update
```

Link the ROS framework to the ubuntu bash:

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Start a new bash prompt and run (test):

```
roscore
```

Install ROS-dependencies:

```
sudo apt-get install Screen
sudo apt-get install ros-kinetic-rosserial
sudo apt-get install ros-kinetic-rosserial-arduino
sudo apt-get install ros-kinetic-rosapi
sudo apt-get install ros-kinetic-rosbridge-server
```

Give user permission to access USB ports:

```
sudo usermod -a -G dialout willy
```

With the above installation changes made to Ubuntu, the current WTGD code was able to run.

1.5. Devices network

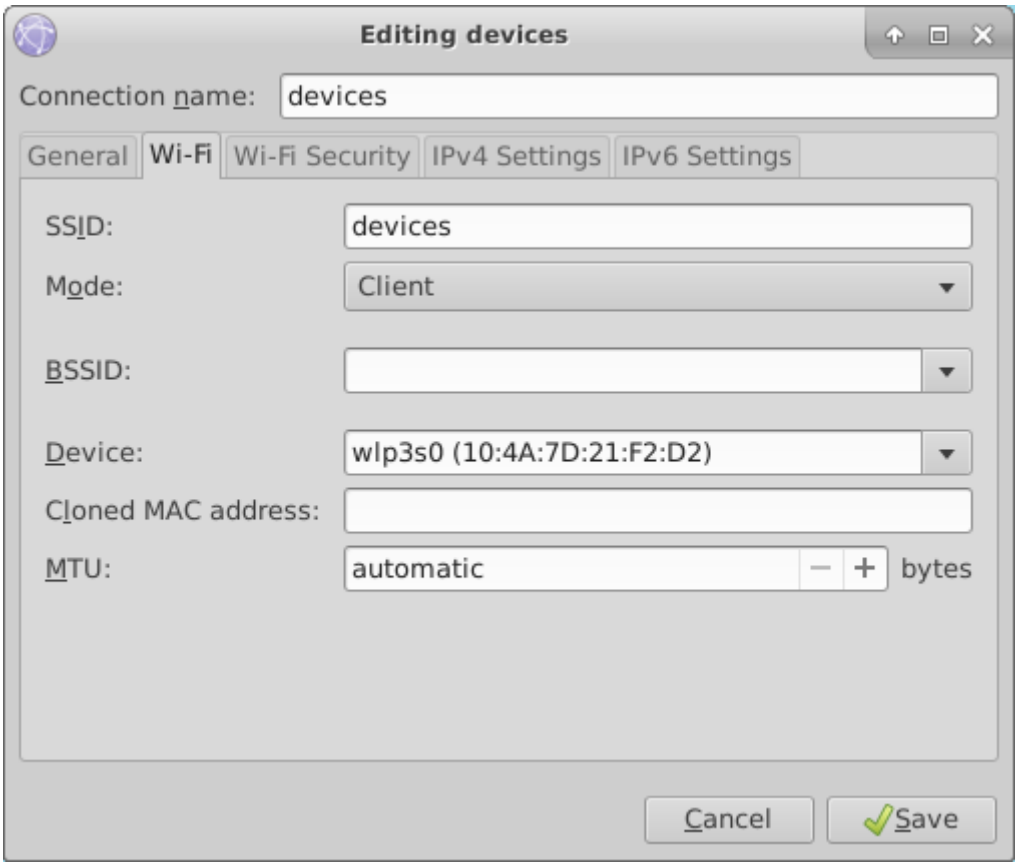
The following information can be used to connect with the 'devices' wifi-network at Windesheim.

```
IP-adres: 145.44.211.32
Subnet mask: 255.255.255.192
Default gateway: 145.44.211.1
DNS: 8.8.8.8
```

An connection can only be made with the following MAC address. If this is changed, please ask the support-desk of Windesheim to change the MAC-address.

MAC: 10:4a:7d:21:f2:d2

The configuration on Willy is as shown below



1.6. Leap

Download the Leap package from https://www.leapmotion.com/setup/desktop/linux

Save the package and unzip the 64-bit version

Go to the folder the .deb was unzipped to and run `sudo dpkg --install Leap-*-x64.deb`

2. Remote access

2.1. RDP

The following configuration was used when RDP was deployed by using XRDP and XFCE4

```
sudo apt-get install xrdp
sudo apt-get install xfce4
```

Edit .Xsesion file in home directory

```
echo xfce4-session >~/.xsession
```

Edit XRDP configfile: “/etc/xrdp/starwm.sh” for using XFCE4

```
#!/bin/sh
if [ -r /etc/default/locale ]; then
  . /etc/default/locale
export LANG LANGUAGE
fi
startxfce4
```

Fix the Tab button by editing:

```
~/.config/xfce4/xfconf/xfce-perchannel-xml/xfce4-keyboard-shortcuts.xml
```

Replacing

```
<property name="&lt;Super&gt;Tab" type="string" value="switch_window_key"/>
```

By

```
<property name="&lt;Super&gt;Tab" type="string" value="empty"/>
```

2.2. Openssh server

Install openssh

```
sudo apt-get install openssh-server
```

Enable ssh on boot

```
sudo systemctl enable ssh
```

References

- link: https://artofrobotics.github.io/WillyWiki/Archive/Research/Alternative-interaction.adoc