Table of Contents

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

Previous groups

1. Group of 2019 Q3 & Q4

1.1. Members

Table 1. Student list of 2019 Q3-Q4

Student	Study Major
Lars Assen	SE
Dion Koster	ESA
Bas Frericks	IDS
Thijmen Bos	IDS
Dennis ter Haar	IDS

1.2. Main tasks

Our goals for this semester were to fix the autonomous driving, create a manual for future groups and to clean out any files that are not used. A common occurring problem is that project groups have a hard time starting this project. A lot of vague documentation and a lot of outdated features create a lot of confusion. To fix this problem we have decided to create a manual to help future groups get started.

1.3. Archive

https://drive.google.com/drive/folders/1iXSppo7ixLBBuq7eIAkdq87K1_OjAdsh

2. Group of 2019 Q1 & Q2

2.1. Members

Table 2. Student list of 2019 Q1-Q2

Student	Study Major
Tom van den Noort	ESA
Jeroen van 't Hul	ESA
Thomas Zwaanswijk	ESA

2.2. Main tasks

The goals of this group was to improve the autonomous driving capabilities of Willy.

2.3. Archive

https://drive.google.com/drive/folders/19r20QrsDuQnvz_Ml-B0VzvG879wsa93G

3. Group of 2018 Q3 & Q4

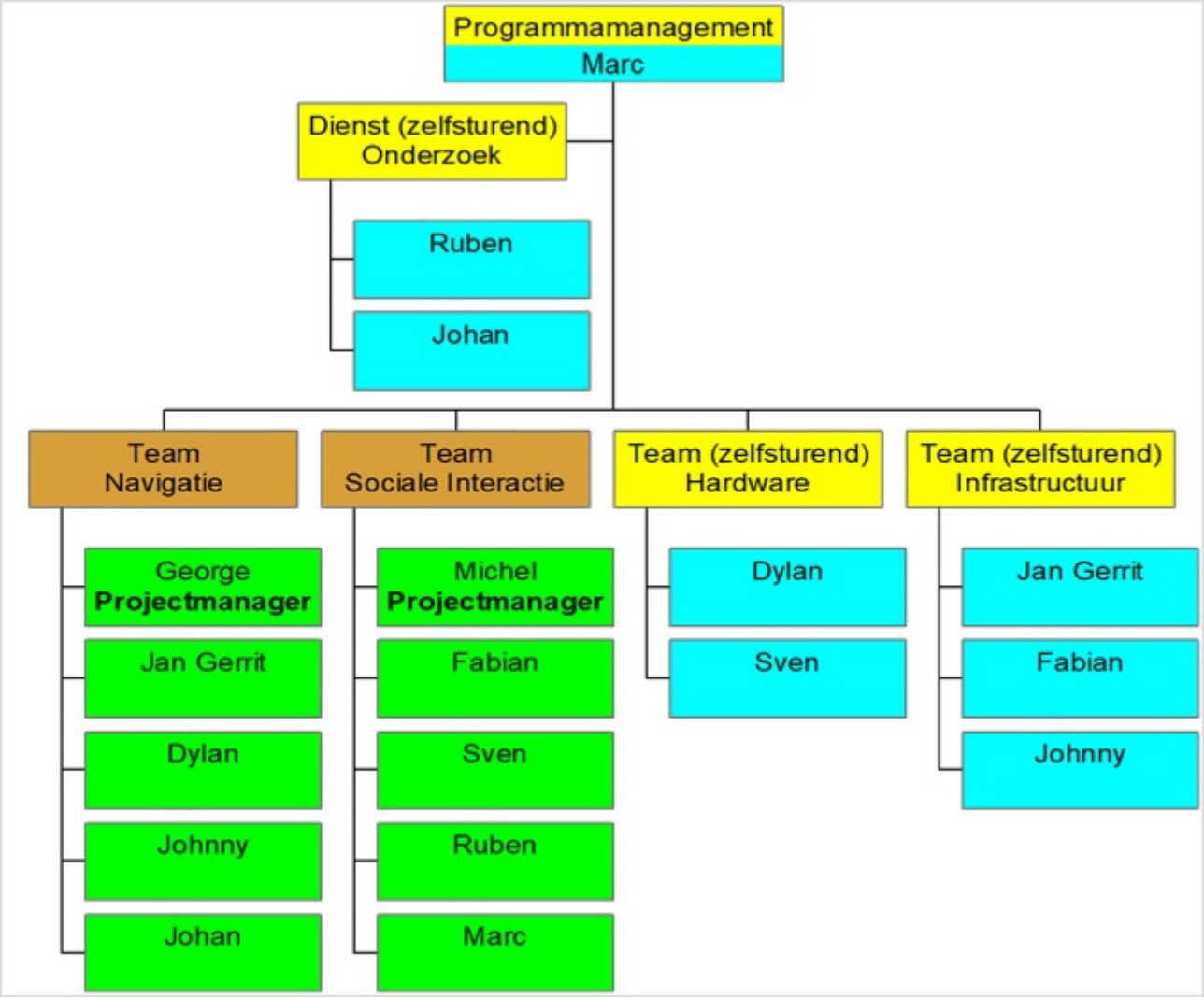
3.1. Members

Table 3. Student list of 2018 Q3-Q4

Student	Study Major
Dylan Reimerink	SE
Fabian van de Bor	SE
George Wassink	BIM
Jan Gerrit Elzinga	SE
Johan in t Hout	BIM
Johnny Borg	SE
Marc van Walt Meijer	BIM
Michel Stompe	BIM
Ruben Stuut	BIM
Sven Pook	SE

3.2. Main tasks

Our goals for this semester consist of creating an autonomous driving robot on T5 as well as integrating Social Interaction in Willy. We started off with two teams of five, but we came to realise very soon that its not possible to achieve one thing, without the other. Therefore, weve had numerous different virtual teams, for physical hardware, software architecture, hardware architecture and so on.



We've also set some ground rules for transference in the process, as we've learned that many groups before us, including ourselves, struggled at getting the system up and running in the first weeks. Besides the two main goals, and transference, weve also redesigned a lot of functionality. Weve created a logical and modular design on both hard- and software. If used as described, the following project group can use our rework and fastrack towards their own new project goals.

3.3. Archive

All our documents will be transferred to the new student group through SharePoint. Everything including templates and images for virtual machines and other hardware will be included in the process.

2018 Q3 & Q4 SharePoint archive

4. Group of 2018 Q1 & Q2

4.1. Members

Table 4. Student list of 2018 Q1-Q2

Student	Study Major
Jesse Bouwman	BIM
Vincent van Dijk	IDS
Jonathan ten Hove	SE
Martijn van Olst	ESA
Gerard Zeeman	ESA

4.2. Research archive

2018 Q1 & Q2 SharePoint archive

5. Group of 2017 Q3 & Q4

5.1. Archive

2017 Q3 & Q4 SharePoint archive

6. Group of 2017 Q1 & Q2

6.1. Archive

2017 Q1 & Q2 SharePoint archive

7. Group of 2016 Q3 & Q4

7.1. Archive

2016 Q3 & Q4 SharePoint archive

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy

- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node
- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain

- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

8. Past reseach

- 2018 Q3 & Q4 SharePoint archive
- 2018 Q1 & Q2 SharePoint archive
- 2017 Q3 & Q4 SharePoint archive
- 2017 Q1 & Q2 SharePoint archive
- 2016 Q3 & Q4 SharePoint archive

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node

- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

9. Skylab Architecture

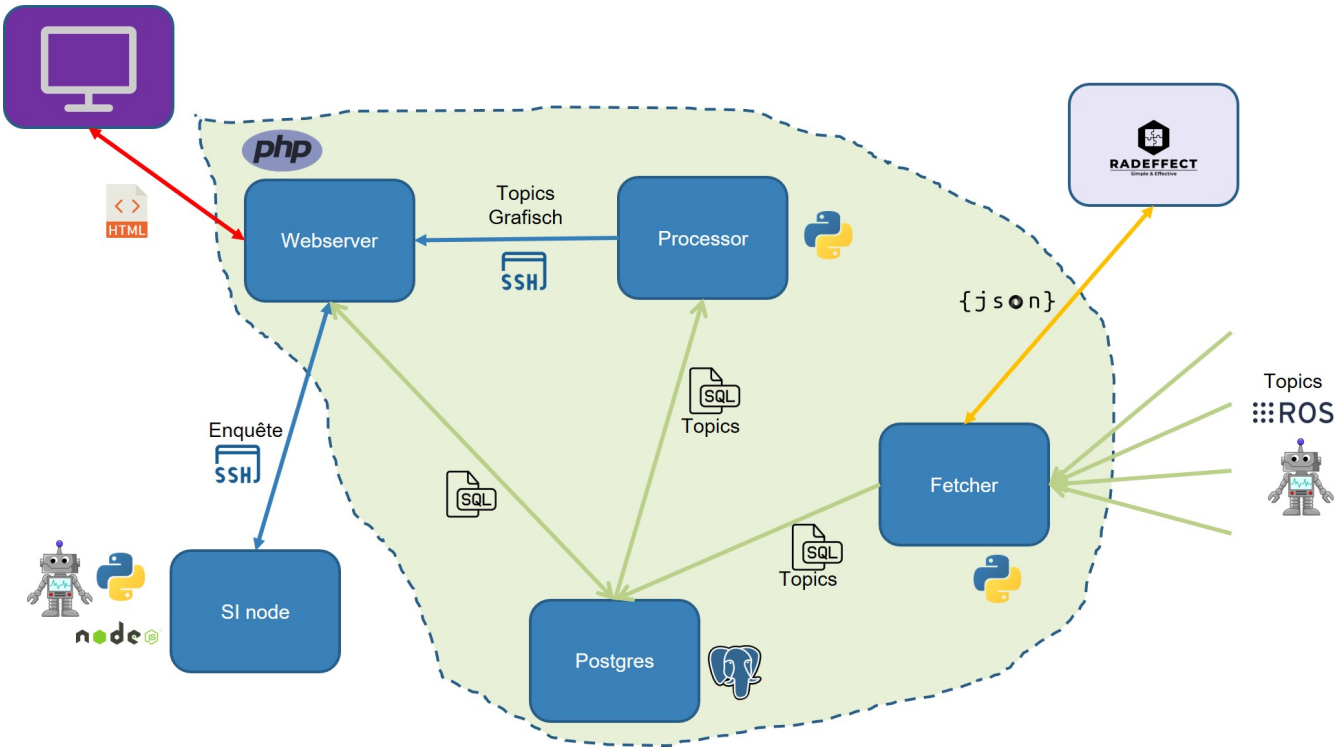
9.1. Principles

The Skylab architecture is design with a set of principles just as the hardware and software architecture.

- Skylab can give limited resources to a node, therefore modularity is essential. All solutions in Skylab should be built with the possibility of upscaling in mind.
- Skylab is the place for all non-vital functions of Willy.
- Skylab is seen as a redundant environment with high enough up-time. There is no need for a dual datacenter architecture.
- Variations of OS software is possible as long as topic communication is possible.
- Variations of ROS software is possible as long as topic communication is possible.
- Simultaneous development is possible.
- Skylab communicates safely with Willy.
- Skylab has limited port connection to the Internet, fi mail anf FTP is is not possible.
- Skylab supplies for ROS Topic fetcher, PostgreSQL and Webserver services. Additional nodes are used for all sorts of processing like Data Sciense and RADeffect connectivity.

9.2. Design

These standpoints resulted in the following hardware design.



9.3. Usability

By having multiple VPN instances, its possible - if all hardware is powered on - to connect through Skylabs to the individual hardware nodes on Willy even if not physically near him. Another option

is to physically connect to WillyLAN and access either Skylab or Willy hardware nodes.

Designing this architecture and keeping it up to date can be done by editing the pptx file in Microsoft PowerPoint on the SharePoint site of Willy.

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history

- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node
- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition

- IMU
- Human Detection
- Radeffect App

10. Setup Skylab Enviroment

Skylab consist of 3 major components:

- Firewall (for connectivity)
- ESXhost or Virtualbox on Windowshost
- Ubuntu Servers

This page explains howto setup the basic Skylab environment.

10.1. First step request a skylab environment

- Request project lab use: “Project Willy” as a reference, this is a known project for Skylab managers. Via https://skylab.windesheim.nl (Login with: studentnr only! not with @student.windesheim.nl)
- After approval you can request machines.

10.2. Next request machines

- First request a PFsense firewall.

[requestpfsense] | *media/requestpfsense.png*

- Next request a Windows 10 or Server machine to configure the PFsense firewall.
- Next configure correct networks to adapters. Goto items > select Machines on the left > click the item which blueprint name is “pfsense”. Select “opnieuw configureren”.

[pfnwadapter] | *media/pfnwadapter.png*

- Next select network > Adaptor ID 1 and click: “bewerken” and select <studentnr>studentnet0, this is your default lan. Next do the same for Adaptor ID 0 but here select the Transit01-dhcp network this is network gives you internetaccess, also known as WAN port.

10.3. Configure Pfsense

- Connect to the pfsense firewall through the console option in Skylab.

[skylabconsole] | *media/skylabconsole.png*

- Login with admin and default password: pfsense and check interfaces Lan should be 192.168.1.1 and Wan> DHCP 145.44.234.* . If not press option 1 and select correct interfaces.

- Next perform step 4 on the Windows machine and login to the console of this machine: with administrator and default password: Welkom01!
- Browse to the pfsense firewall default ip-adress: http://192.168.1.1
- Login to the pfsense with default admin/pfsense
- Make sure internet is working correctly on firewall.
- Go to Diagnostics > Backup/restore and select the config file (sharepoint\skylab): pfsenseskylab.xml and click restore.
- Wait 10 minutes, because it will install nessecary packages, which takes a while, then restart the Windows Machine.
- Now the firewall can be managed at: http://10.10.1.1
- To create a new vpn user goto > system > user manager
- Next click: add user and fillout the username and password and make the user a member of the VPN group.

10.4. Connect Willy to Skylab

- First make sure the router node (rpi) and laptop are turned on. and verify you have an internet connection.
- Next verify vpn connection: On the laptop try to ping to 10.10.1.1 if replies are succesfull it all works!
- If not please check DNSName: ping skylabwilly.dynu.net (this should point to currenct Skylab ip) If not please change manually: https://www.dynu.com/en-US/ControlPanel/DDNS login with Willy google account.
- If this does not work, reimage the Pi version model 3b+ with the version found on sharepoint. (routernode.img)

This could be an out-of-date image, since OpenWRT was recently (as time of writing at 12/02/19) ported to the 3b+.

- A more detailed description can be found on sharepoint in the skylab folder on how to set up openwrt.

10.5. Connect to Skylab over VPN Client

- Now install the openvpn client, the installer can be found on sharepoint: \skylab\install
- After installation you can connect to skylab by starting the openvpngui, then rightclick on the icon in the taskbar and select connect to skylabwilly.dynu.net and enter username password from before.
- To verify connection type ping 10.10.1.1 (which is the pfsense firewall in skylab).

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

11. Python scripts

11.1. Overview

Several tasks around Willy are done through Python scripts. In this chapter the scripts with their purposes are shown. The Python scripts are self-documented.

11.2. Scripts

Table 5. Scripts for Willy

Name	Location	Purpose	Remarks
Willy_Speech_Regocnition_0.py	Willy	Fetch speech from microphone. If text is “hallo willy” put topic /interaction/is_active on 1. Publish recognized on topic /interaction/clear_text	Uses packages rospy, Speechrecognition and subprocess. Uses Google speech recognition

Name	Location	Purpose	Remarks
Skylab_topic_listen_0.py	Skylab	Receive many topics from ROS environment and store them in several tables in Postgres	Uses packages rospy, pandas, datetime, configparser, sqlalchemy, psycopg2. Retrieves Postgres password from willy.ini. Postgres server on 10.10.1.34
Skylab_app_get_status_0.py	Skylab	Fetch every 10 seconds the location that was called by a user from the App site	Coop with Radeffect
Skylab_app_post_status_0.py	Skylab	Post status from Willy activity topic and classroom location to App website every 10 seconds	Coop with Radeffect

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

12. Skylab Webserver

In Skylab a webserver is installed on 10.10.1.45. The webserver can be used for

- Seeing the content of the fetched topics
- Upload and download survey files

- Maintenance on the webserver environment

The installation is on Ubuntu 16.04 and has a LAMP installation.

12.1. The installation of LAMP is done by:

```
sudo apt-get update
sudo apt-get install apache2

sudo nano /etc/apache2/apache2.conf
```

Add at the bottom:

```
ServerName 10.10.1.45
```

Restart Apache with:

```
sudo systemctl restart apache2
```

Allow incoming traffic:

```
sudo ufw allow in "Apache Full"
```

12.2. Install MySQL

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

use standard password

Secure MySQL environment

```
mysql_secure_installation
```

12.3. Install PHP

```
sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql
```

edit dir.conf and move index.php higher up in the list

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

Activate PostgreSQL connectivity: Open the php.ini file (/etc/php/7.0/cli/php.ini) and check if the following line is un-commented. If it is not, you can remove the semicolon (;) in front of the entry.

```
extension=php_pdo_pgsql.dll
```

Install Postgres driver

```
sudo apt-get install php-pgsql
```

12.4. Finalizing

Restart Apache

```
sudo systemctl restart apache2
```

Test server:

```
sudo nano /var/www/html/info.php
```

```
<?php
phpinfo();
?>
```

Check with:

```
http://10.10.1.45/info.php
```

12.5. Install additional packages & commands

```
sudo apt-get install libssl-dev
sudo apt-get install php7.0-cli -y
sudo apt-get install libssh2-1 php-ssh2 -y
```

add to php.ini

```
extension = ssh2.so
```

Give website rights in home directory of Willy

```
sudo chmod -R 777 /home/willy
```

Enable CGI

```
sudo a2enmod cgi
```

Restart Apache

```
sudo systemctl restart apache2
```

12.6. reference:

https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node

- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

13. Functions of the webserver

The webserver is the central information and management point of Willy.

The webserver is situated in Skylab and is connected to Willy by the VPN connection between Skylab and Willy. The Webserver is installed on Ubunto 16.04 LTS and additionally installed with LAMP (https://windesheim-willy.github.io/WillyWiki/skylab/Webserver.html).

The main functions of the webserver are divided in:

- Topics
- Enquete
- Onderhoud
- Willy management
- Willy Wiki

13.1. Topics

The first item Topics gives access to the saved content of several Topics. The script make_web_oage.php (https://github.com/Windesheim-Willy/Skylab/blob/master/webserver/make_web_page.php) is the central used script. Calling this script with the wanted table constructs the complete page. The second function on the Topics page is the display of graphical representation of Topic contents. The webserver works together with the Processor node which constantly updates the graphical representation so all graphics are near-realtime up-to-date.

13.2. Survey

The Enquete (Dutch for survey) item provides all functions to supply and retrieve the survey function on the SI brain node. The items on the page are:

For upload to the SI brain:

- Upload enquete vragen (survey.csv & survey.json) van lokale PC naar Webserver
- Bekijk en edit enquete informatie op Webserver
- Bekijk enquete vragen op Webserver
- Voeg enquete vragen en enquete informatie toe aan Postgres database
- Upload enquete vragen van Webserver naar Willy

For download from the SI brain:

- Download enquete antwoorden van Willy naar Webserver
- Bekijk enquete antwoorden op Webserver
- Voeg enquete antwoorden toe aan Postgres database
- Download enquete antwoorden van Webserver naar lokale PC

The first item gives the possibility to upload the prepared questions and general survey information from a PC of an operator to the website. The PC must have a direct VPN connection to Skylab. The format of both files are described in the file "Willy enquete.docx" on the Sharepoint site

of Willy.

The second item gives the possibility to look at and if wanted change the information of the general survey information.

The third item gives the possibility to look at the questions of the survey.

The fouth items gives the possibility to add the general survey information and questions to the Postgres database server in Skylab.

The fifth item gives the possibility to upload the general survey information and questions from the webserver to the SI brain. When the general information is uploaded from the webserver to the SI brain the data is converted from the TXT format to the required JSON format.

The sixth item gives the possibility to download the given answers from the SI brain to the webserver.

The seventh item gives the possibility to look at the answers of the survey.

The eigth item gives the possibility to add the survey answers to the Postgres database server in Skylab.

The last item gives the possibility to download the answers from the website to a PC of an operator. The PC must have a direct VPN connection to Skylab.

13.3. Onderhoud

The Onderhoud (Dutch for maintenance) item give the possibility to edit the configuration and password file used by several scripts in the Webserver.

13.4. Willy management

The Willy management item gives the possibility to do RDP sessions to the two Hyper-V hosts in Skylab and take over the screen of Willy (the SI node).

13.5. Willy Wiki

The items on the page are:

- Wiki van Willy in nieuw venster
- Maak Wordcloud (duurt een paar seconden)
- Beheer Wordcloud websites
- Beheer Wordcloud stopwoorden
- Beheer Wordcloud cloudset

The first option links to the Wiki of Willy.

The second option makes a Wordcloud of the Wiki of Willy. The processing takes several seconds.

The third option gives the possibility to look at and if wanted change the websites that are included in the Wordcloud.

The fourth option gives the possibility to look at and if wanted change the stopwords will be exluded when making the Wordcloud.

The last options gives the possibility to look at and if wanted change configuration settings on how to make the Wordcloud.

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node
- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect

- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

14. Servers in Skylab

Skylab is the private cloud environment of Windesheim. The management portal of Skylab can be found here: skylab.windesheim.nl

In Skylab there are several servers active. Two servers are Windows 2016 Hyper-V servers. Both have Virtual box installed. In VirtualBox there are several Ubuntu 16.04 VM's active.

All the servers with their functions are listed in the next table.

Server name	IP number	OS	Function
pfSense.localdomain	10.10.1.1		Firewall, DHCP, DNS, Gateway, VPN to Willy

dbsvr.willy.local	10.10.1.5	Windows 2016	Hyper-V host of 10.10.1.3x VM's
appsvr.willy.local	10.10.1.10	Windows 2016	Hyper-V host of 10.10.1.4x - 5.x VM's
u1633.willy.local	10.10.1.33	Ubuntu 16.04 LTS	Fetcher of topics to PostgreSQL, Interface to and from RADeffect app
u1634.willy.local	10.10.1.34	Ubuntu 16.04 LTS	ROS master for Skylab, not used anymore, only for test purposes
u1635.willy.local	10.10.1.35	Ubuntu 16.04 LTS	PostgreSQL database
u1644.willy.local	10.10.1.44	Ubuntu 16.04 LTS	Server for several Data Science and other scripts
u1645.willy.local	10.10.1.45	Ubuntu 16.04 LTS	Webserver of Willy, based on LAMP
zabbix.willy.local	10.10.1.55	Ubuntu 16.04 LTS	Zabbix management server

15. Ubuntu servers in Skylab

The servers in Skylab are all Ubuntu 16.04 LTS servers and installed with default settings. The used user and password can befound in the file "inloggegevens.docx" at SharePoint.

The images of the Ubuntu servers in OVA format can be found in the map "Winnie Skylab\OVA" at SharePoint. The images are the status of December 21, 2018. Some scripts are updated at a later date which can be found at Github.

On both the Fetcher (10.10.1.33) and the ROS master (10.10.1.34) ROS Kinetic is installed as described at http://wiki.ros.org/kinetic/Installation/Ubuntu. For the ROS master Multimaster as the "second" master (port 11312) is installed as described at https://windesheim-willy.github.io/WillyWiki/ROS/Multi_master.html

On the Postgres server (10.10.1.35) PostgreSQL is installed with default settings. The used user and password can be found in the file "inloggegevens.docx" at SharePoint.

On the Fetcher and the Processor (10.10.1.45) the following Python packages are additionally installed:

```
* Pandas
* Configparser
* Time
* Random
* urllib3
* json
* ssl
* datetime
* sqlalchemy
* psycopg2
```

On the Zabbix server (10.10.1.55) Zabbix is installed with default settings. The used user and password can be found in the file "inloggegevens.docx" at SharePoint. Some implementation of servers is done in Zabbix, but is stopped as not being part of the MVP.

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

16. Installation of Ubuntu on Skylab VM;s

The VM's Fetcher and Sylab ROS master have ROS installed. Run these commands on Ubuntu 16.04 to install ROS Kinetic ready to run.

```
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-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-key
421C365BD9FF1F717815A3895523BAEEB01FA116

sudo apt-get update

sudo apt-get install ros-kinetic-desktop-full

apt-cache search ros-kinetic

sudo rosdep init

rosdep update

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

source ~/.bashrc

sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool
build-essential

mkdir -p ~/catkin_ws/src

cd ~/catkin_ws/

catkin_make

source devel/setup.bash
```

reference: http://wiki.ros.org/kinetic/Installation/Ubuntu

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy

- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node
- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain

- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

17. DNS, DHCP, pfSense and Ubuntu

When the Ubuntu systems are configured with static DHCP from pfSense, they get the correct IP adress. But they don't get a external nameserver, so they can't connect to the outside World. Therefore all the Ubuntu systems have an manually added nameserver of 8.8.8.8.

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node
- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

18. MPU9250

The MPU9250 is a 9-axis Inertia Measurement Unit. This means that it can detect rotation in 9 directions. It does this by utilizing an accelerometer, gyroscope and magnetometer. The IMU is located on the top of the screen, in a 3-d printed box with the words "IMU TOP" printed on the lid.

The magnetometer (AK9863) is separate from the accelerometer and gyroscope, but can be read using the same library. The data sheet is referenced in the technical design document, which also contains a more detailed explanation.

18.1. Arduino

The MPU9250 is connected using I2C to an arduino, which does a large part of the processing. It calculates the quaternion, as well as transforming some of the data into useful units. These are then changed into separate bytes, and sent via Serial communication to the Sensor Node Raspberry Pi.

18.2. Raspberry Pi

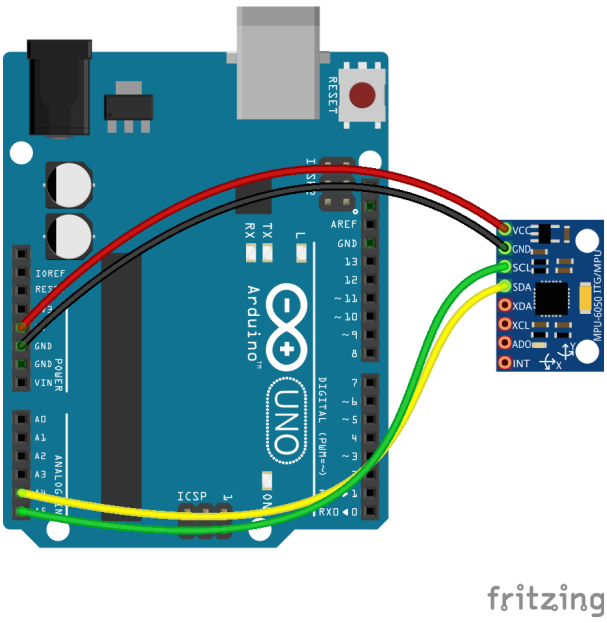
The raspberry pi stores all the data it receives into a string, which it then proceeds to check for a start byte. After this, it uses Unions, an old C technique, to process the separated bytes into the desired data types. These are built into a message, and sent to the master.

If there is an error in the data from the arduino, such as missed data or un-needed data, the entire message is binned and the input is cleared.

18.2.1. Quaternions

The MPU9250 uses a complex mathematical system of tracking orientation called "quaternions". These are a method of storing current rotation in a 3-d space, while avoiding the issue of Gimbal lock found in Euler angles. While these are complex, they are the preferred method of tracking rotation according to ROS rep 103. A comprehensive tutorial on the mathematics behind quaternions can be found here.

18.3. Wire



18.4. set setpoint to 0

Sometimes it is needed to set the "set point" to 0. The command to do this is as follows:

```
rosservice call /imu/set_zero_orientation
```

19. IMU

The IMU sensor (pose tracking) is used to get a higher accuracy of Willy the Robot's orientation for navigation and localization purposes.

19.1. Repository

Windesheim-Willy/pose_tracking

19.2. Prerequisites

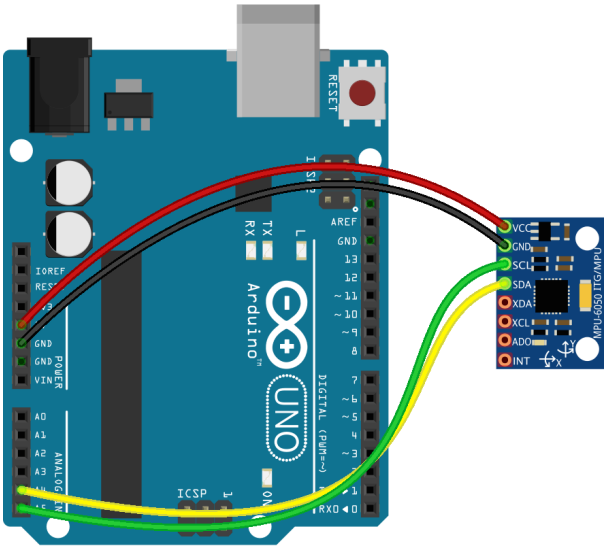
The IMU sensor (Arduino) must be connected to the Sensor Pi via USB.

19.3. How to run?

The pose tracking will start automatic when the PI is running. If it doesn't, the pose tracking node can be started manually by going into the root of the pose tracking node's source directory:

```
./START start
```

19.4. Wiring schematic



19.5. Calibration

Sometimes it is needed to set the sensor's origin to 0. The command to do this is as follows:

```
rosservice call /imu/set_zero_orientation
```

It would be wise to do this each time WTR is started, and after its position/orientation has been set in RViz.

19.6. Background

The IMU data that is published on the topic `/imu/data`, is used by move_base, amcl and/or the laser_scan_matcher node.

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node
- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master
- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction

- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

20. Multi master setup on Willy

The Willy environment is built on two locations (Willy and Skylab) which are coupled with an Wi-fi connection. By using an multi master set up there are two advantages:

- Through the set up it can be decided which topics will be transferred between the two environments, thereby potentially saving bandwidth (configured on master_sync node)
- Skylab will have a dedicated master even when Willy is not operational

To have the sync module always up-and-running, it seems that this module will be running on the master in Skylab. It has to be tested if the master on Willy will be automatically be recognized when it is started or some actions has to be taken. Seems to be a task of the master_discovery node.

It has to be tested in the Willy environment if this set up works. Actions are:

- ☑ Make snapshot of test master in Skylab
- ☑ Install multimaster on the (now) test master in Skylab
- ☑ Install test master with multimaster on a RPi
- ☑ Confirm that it works
- ☑ Evaluate the stop and restart of master on RPI and document possible restart actions

Go/no go decision

Go:

- ☐ Install multimaster on production RPi on Willy
- ☐ Depreciate test master on RPi
- ☐ Document and make instructions

No go:

- ☑ Revert snapshot on master in Skylab
- ☑ Make decision on further action

Decision is that the nodes in Skylab will be using the ROS master of Willy.

20.1. Installation and using multimaster

Install multimaster on every master Kinetic node with:

- sudo apt-get install ros-kinetic-multimaster-fkie
- Check DNS, localhost and host name
- Enable multicast feature on both hosts and for all ROS nodes:
  - edit the /etc/sysctl.conf file and add the following line:
    - net.ipv4.icmp_echo_ignore_broadcasts=0

Start multi master:

- first master:
  - export ROS_MASTER_URI=http://brainnode:11311
  - roscore >/dev/null 2>&1 &
  - rosrun master_discovery_fkie master_discovery >/dev/null 2>&1 &
  - rosrun master_sync_fkie master_sync >/dev/null 2>&1 &
- second master:
  - export ROS_MASTER_URI=http://u1634:11312
  - roscore --port 11312 >/dev/null 2>&1 &
  - rosrun master_discovery_fkie master_discovery >/dev/null 2>&1 &
  - rosrun master_sync_fkie master_sync >/dev/null 2>&1 &

20.2. For info:

- start node manager with: node_manager
- rosservice call /master_discovery/list_masters

Take note: there must be a binding between the IP address and the host name for all the computers in all the networks in order for the ROS framework to work properly.

References:

http://wiki.ros.org/multimaster_fkie/Tutorials/Setup%20a%20ROS%20master%20synchronization

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi's

- Sensor node
- Social Interaction node
- Power node

Components

- ROS master
- New ROS master on Lubuntu
- Sonar
- Lidar
- Kinect
- Localization and navigation
- Motor controller
- Joystick

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive
- Skylab Architecture
- Skylab
- Multi master

- WillyWRT
- Realisation
- Hardware
- Brain
- Design Guild
- Social interaction
- Speech
- Speech recognition
- IMU
- Human Detection
- Radeffect App

21. Hardware Architecture

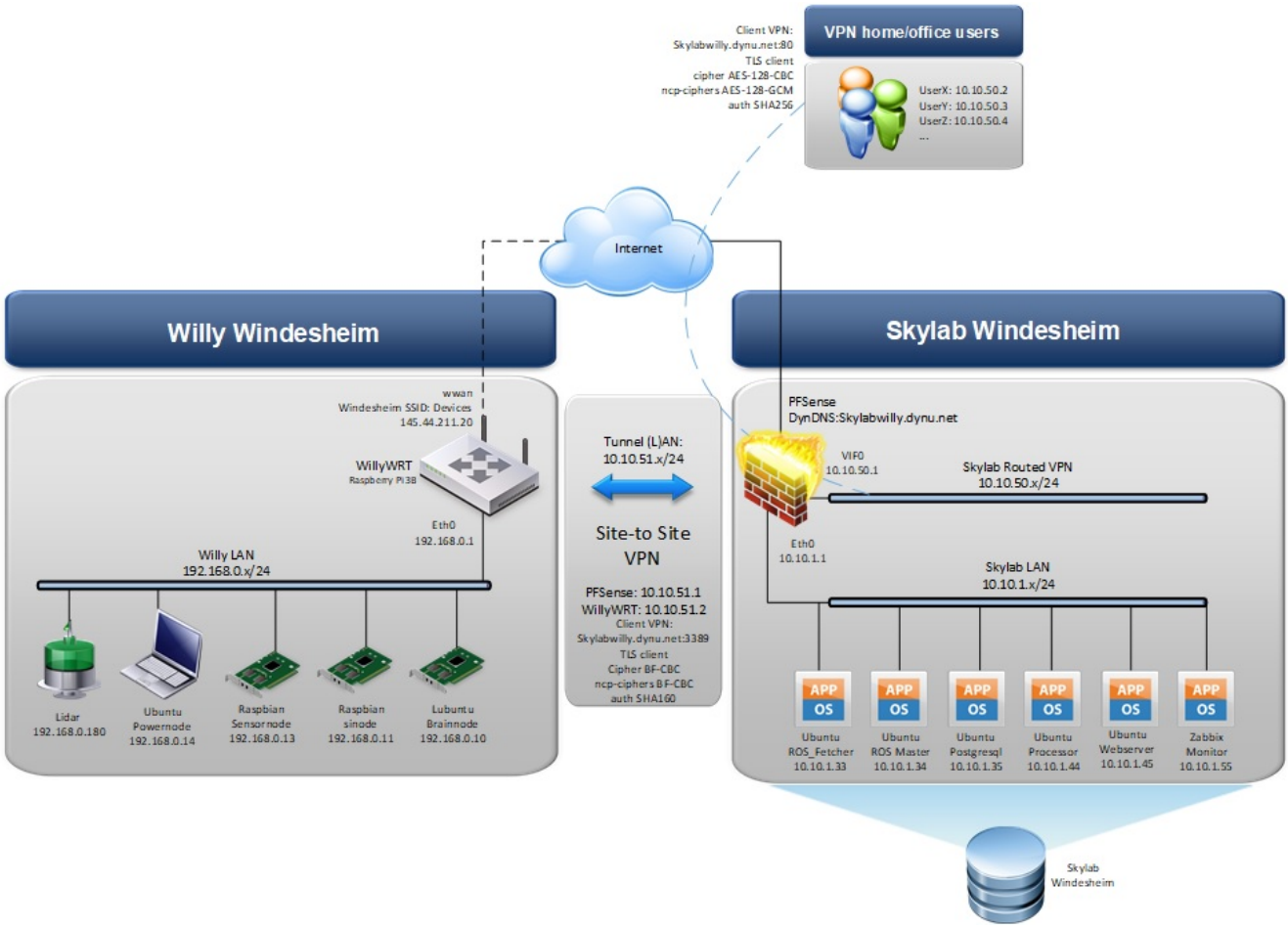
21.1. Standpoints

The hardware architecture is designed with a set of standpoints just as the software architecture.

- Use low-power hardware if possible to reduce power consumption
- Clustering similar functions on a separate hardware node
- Using multiple hardware nodes to ensure non-disruptive node failure
- Functions that can be run offsite - Skylab - should
- Variations of OS software is possible as long as topic communication is possible
- Variations of ROS software is possible as long as topic communication is possible
- Simultaneous development is possible

21.2. Design

These standpoints resulted in the following hardware design.



21.3. Usability

By having multiple VPN instances, it's possible - if all hardware is powered on - to connect through Skylabs to the individual hardware nodes on Willy even if not physically near him. Another option is to physically connect to WillyLAN and access either Skylab or Willy hardware nodes.

Designing this architecture and keeping it up to date can be done by editing the vsdx files in Microsoft Visio.