Table of Contents

1. Lessons Learned 2
    1.1. Driving 2
    1.2. Design 3
    1.3. Software 3
    1.4. Hardware 3
2. Navigation 4
    2.1. Social interaction 4

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Development Guide
- Driving Willy
- Documentation

Build of Willy

- Design history
- Requirements
- Design reference
- Physical build
- Hardware

Robotic Operating System

- Introduction to ROS
- ROS Tutorials
- Multi master

Architecture

- Software Architecture
- Hardware Architecture
- Skylab Architecture
- ROS topic design

Hardware nodes

- sensor node
- si node
- power node
- WillyWRT

Components

- ROS master
- New ROS master on Lubuntu
- Brain

- Sonar
- Lidar
- Localization and navigation
- Motor controller
- Joystick
- Social interaction
- Speech
- Speech recognition

Skylab

- Setup Skylab
- Python scripts
- Webserver
- Functions of the webserver
- Skylab servers
- ROS installation on Ubuntu VMs in Skylab
- DNS,DHCP, pfSense & Ubuntu

Radeffect App

- Radeffect App

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive

This pages needs to be up to date. 9-1-2019 latest updated.

1. Lessons Learned

This page contains a list of lessons we learned during the development of Willy during 2018 semester.

1.1. Driving

- It is quite hard to deploy and undeploy the brakes.
- When the battery voltage drows below a certain level, Willy will stop driving at irregular

intervals. This is probably an internal security system. The brakes are also deployed causing an abrupt stop.

- When driving manually, Willy uses characters pressed on the keyboard to move. When a lot of characters are pressed in a short time, Willy will try to process the complete list of characters, seeming to be uncontrollable.
- Willy has difficulty driving through doors when driving autonomously. This is caused by Willy being quite wide and limitations in the software for autonomous driving.
- The ultrasonic sensors have a lot of blind spots. Since the LIDAR can't scan the back of Willy, the ultrasonic sensors are the only sensors controlling the back of Willy. When Willy is driving backwards, this can cause problems.

1.2. Design

- Because of the width of Willy with garbage bins attached to the side, it is impossible to go through a door without detaching the bins, including the detachable buffers with ultrasonic sensors in it.

1.3. Software

- The motor controller is a device that is used to control the cmd_vel topic created by the ros navigation stack. The ros navigation stack published geometry::Twist messages
- ROS is complicated and difficult to learn. But if everything is setup just right, ROS makes it really easy to communicate between your hardware.
- If you want to create files to start multiple projects? Do not use bash files. ROS can't handle bash files properly. use ros launch files instead.
- The making of the DrivingController was a complex situation. That is because of the fact every ROS node has only one ROS nodehandler. To make it possible to subscribe and advertise everywhere in the DrivingWilly code, we must send the nodehandler to every subcontroller by using pointers. This request a detailed knowledge of C++ and ROS.
- The setup of the ROS navigation stack is difficult because of the fact that every robot is different. A lot of components needs to be setup on your own. As example the move_base, the transformations and rotations and the cmd_vel topic.

1.4. Hardware

- The current batteries should be powerfull enough to power the current Willy.
- The 230 volt touchscreen in combination with the power converter is replaced by a 19v display.
- The brakes are not easy to deploy. Thats because the levers on the side of willy are to loose to deploy the brake. We can't tight them because the screw is malformed.
- The GPS sensor and compass only work outside and are controlled using the GPSController.
- The kinect cannot be used outside. The IR camera can't handle the bright sunlight.

2. Navigation

- Move_base as of this moment is not aware of the shape of the robot. This results in issues where the robot thinks it can pass between objects or rotate while in reality it can't. It advisable to make move_base aware of the shape of the robot.
- The global planner has not been tuned yet. As of this moment it tries to create a route which is very close to the walls because it wants the fastest route. It would be better for the robot to be driving in the middle of a hallway then hug the wall. The global planner als makes very sharp corners which the robot can't handle very well. A custom planner or tweaking the standard planner can help a lot.
- The the default default planner which is currently in optimized for a robot with powered rear wheels and a steering front wheels like a car. However willy has two powered wheels which steer using differential power and two swivel wheels.

2.1. Social interaction

- The ambient noise can make it hard to speak with Willy. We used the mute function on the microphone provided with Willy.
- The speech of Willy can be rather robotic.