Table of Contents

1. Development Guide

1.1. Prerequisite

To be able to develop Willy there are a few Prerequisites:

- Integrated Development Environment
- Basic knowledge about C++ and C
- Basic knowledge about ROS
- Basic knowledge about GIT

Integrated Development Environment
Visual Studio Code is a Integrated Development Environment. This program is used to write code and use git to send the documents to Willy. Visual studio can be extended with a lot of extensions. These extensions make it possible to align the code and see if the syntax of your code is correct or incorrect.

Basic knowledge about GIT
GIT is used as file transfer and version manager of the current Willy project. Everything starts with setting up the GIT repository and be able to push and pull parts of code. That makes GIT essential in the Willy development.
A tutorial about GIT and the setup of GIT can be found here:

https://www.atlassian.com/git/tutorials/setting-up-a-repository

Basic knowledge about C++ and C
The Willy code is written in C++. To add features and understand the code, some basic knowledge of C++ is required. If you want to learn the basics of C++? Follow a semester programming or see the following tutorial

https://www.studytonight.com/cpp/introduction-to-cpp.php

Basic knowledge about ROS
Because Willy is based on the Robot Operating System (ROS), a basic knowledge of ROS is required to communicate with the different modulair parts of Willy. ROS learing may be difficult, but you can follow the following tutorial to learn the basics:

http://wiki.ros.org/ROS/Tutorials

1.2. Development

To start development, make sure you complies to the requirements above.

1. Make a folder on your local computer where you want the Willy project to take place.
2. Install git.

Git can be downloaded from the following website: https://git-scm.com/downloads

1. Clone the WTGD git repository.
   This can be done with the following command:
   `git clone https://github.com/ArtOfRobotics/WTGD`
2. Checkout at the `test` branch.
   This can be done with the following command:
   `git checkout test`
3. Open Visual Studio Code and open the cloned WTGD folder.
   This can be done by following this steps:
   File → Open Folder → Navigate to WTGD folder → Select Folder
4. In the navigation window on the left, navigate to:
   willy → src

In this folder all the Willy ROS packages are listed.

1.3. Compilation

To compile your changes you have made to the Willy project, a few steps are required.

1. Commit and push your changes to the GIT WTGD repository. This can be done using git bash or Visual Studio Code integrated source control.
2. Startup the computer of Willy, login using the default login credenticals and open a terminal

window.

3. Enter the following command: `cd /home/willy/Documents/WTGD/willy`
4. Pull the git changes (There may be no changes because of the automatic git sync) `sudo git pull`
5. Build the package using catkin_make `catkin_make`
6. Make the terminal point to the source code of driving_willy `source devel/setup.bash`
7. Now you can launch the project as described in startup wiki page.

For more information about starting up willy? See our startup guide: https://artofrobotics.github.io/WillyWiki/Startup/Driving-Willy.html

1.4. Testing/Debugging

To test the new builded code, you need to launch the project you builded. Make sure you followed the steps above and leave the terminal open.

Make sure to press the emergency button or the switch on the motor controller that makes the LED turn off. Otherwise the motors cannot start.

To launch the willy_navigation package you can use the roslaunch command. `roslaunch willy_navigation willy_navigation.launch`

Do not forget to depress the emergency button or switch the lever on the motor controller after starting the project. Otherwise Willy will not drive.

Step if Willy will not drive

Willy can sometimes be a little unreliable and stops driving. This can have multiple reasons. Here are some of them and the way to fix the problems:

1. Forgot to press the emergency button before starting up the project.
   → Use the command `pkill ros` in a new terminal to stop ROS.
   → Follow the steps above or the startup guide to restart Willy.
2. Forgot to depress the emergency button after starting the project.
   → Depress the emergency button and willy will start driving
3. Forgot to set a goal to navigate to.
   → Use your mouse to send a new goal in Rviz. This button can be found in the top navigation bar in rviz.
4. Rviz hangs on startup
   → Use the command `pkill ros` in a new terminal to stop ROS.
   → Use the command `pkill rviz` in a new terminal to stop Rviz.
   → Reboot the project as described in the startup guide.
5. Mini pc will not power on.
   → Use the switch on the top of willy to power on Willy.