



## eYRC 2021-22: Agri Bot (AB)

## Catkin Workspace

- **catkin** is the official build system of ROS and the successor to the original ROS build system, `roscpp`.
- **catkin** combines CMake macros and Python scripts to provide some functionality on top of CMake's normal workflow.
- `catkin` was designed to be more conventional than `roscpp`, allowing for better distribution of packages, better cross-compiling support, and better portability.

### src

- The `src` folder contains the source code of catkin packages. This is where you can extract/checkout/clone source code for the packages you want to build.
- Each folder within the `src` folder contains one or more catkin packages. This folder should remain unchanged by configuring, building, or installing.
- The root of the `src` folder contains a symbolic link to catkin's boiler-plate 'toplevel' CMakeLists.txt file. This file is invoked by CMake during the configuration of the catkin projects in the workspace. It can be created by calling `catkin_init_workspace` in the `src` folder directory. When we execute the `catkin_make` command from the workspace folder, it checks inside the `src` folder and builds each package.

### build

- The `build` folder is where CMake is invoked to `build` the catkin packages in the `src` folder.
- CMake and catkin keep their cache information and other intermediate files here.
- The `build` folder does not have to be contained within the workspace nor does it have to be outside of the `src` folder, but this is recommended.

### devel

- The development folder (or `devel` folder) is where built targets are placed before being installed.
- The way targets are organized in the `devel` folder is the same as their layout when they are installed.
- This provides a useful testing and development environment which does not require invoking the installation step.
- The location of the `devel` folder is controlled by a catkin specific CMake variable called `CATKIN_DEVEL_PREFIX`, and it defaults to `build/devel` folder.
- This is the default behavior because it might be confusing to CMake users if they invoked CMake in a `build` folder and that modified things outside of the current directory.
- It is recommended, however, to set the `devel` folder directory to be a peer of the `build` folder directory.

```
source ~/<workspace_name>/devel/setup.bash
```



