

The background features a dynamic splash of paint in shades of orange, yellow, and green against a white backdrop. The paint splatters are concentrated in the upper half, with some smaller droplets scattered below. A dark grey horizontal band spans the width of the image, serving as a background for the text.

COGA-OJT 4th Week(2)

박성준 인턴

INDEX

추가사항

- 1번 Using Turtlesim
- 2번 Using Servinggo
- 3번 Using Text File

추가기능

- Auto Start with System Booting
- param으로 최대 속도 지정받기
- Bag file 만들기



1 번 모드 : Using Turtlesim

- Turtlesim으로 조이스틱의 publishing 값을 subscribe함
- Publishing된 값을 .txt 파일로 저장



1 번 모드 : Using Turtlesim

```
// 조이스틱 값을 Subscribe 하여 txt 파일에 일정 양식에 맞게 저장함
joysub = nh.subscribe("/joy", 1, &Recorder::recorderCallback, this);
// 들어온 조이스틱 값을 시각화하기 위해 turtlesim을 활용함
turtlepub = nh.advertise<geometry_msgs::Twist>("turtle1/cmd_vel", 1);
```

```
void Recorder::recorderCallback(const sensor_msgs::Joy &msg) {
    geometry_msgs::Twist getCmdVel;

    getCmdVel.linear.x = msg.axes[1] * (this->max_linear_x);
    getCmdVel.angular.z = msg.axes[0] * (this->max_angular_z);

    this->linear_x = msg.axes[1] * (this->max_linear_x);
    this->angular_z = msg.axes[0] * (this->max_angular_z);

    // linear_x와 angular_z를 인자로 넘겨 txt 파일로 저장되도록 함
    Recorder::cmdVelRecorder(this->linear_x, this->angular_z);

    // turtle1/cmd_vel로 getCmdVel을 써서 turtlesim_node로 볼 수 있도록 함
    this->turtlepub.publish(getCmdVel);
};
```



1번 모드

Using Turtlesim

Sub -> /joy

Pub -> /turtle1/cmd_vel



```
// 조이스틱 값을 Subscribe 하여 txt 파일에 일정 양식에 맞게 저장함
joysub = nh.subscribe("/joy", 1, &Recorder::recorderCallback, this);
// 들어온 조이스틱 값을 시각화하기 위해 turtlesim을 활용함
turtlepub = nh.advertise<geometry_msgs::Twist>("turtle1/cmd_vel", 1);
```

```
void Recorder::recorderCallback(const sensor_msgs::Joy &msg) {
    geometry_msgs::Twist getCmdVel;

    getCmdVel.linear.x = msg.axes[1] * (this->max_linear_x);
    getCmdVel.angular.z = msg.axes[0] * (this->max_angular_z);

    this->linear_x = msg.axes[1] * (this->max_linear_x);
    this->angular_z = msg.axes[0] * (this->max_angular_z);

    // linear_x와 angular_z를 인자로 넘겨 txt 파일로 저장되도록 함
    Recorder::cmdVelRecorder(this->linear_x, this->angular_z);

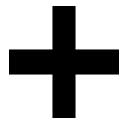
    // turtle1/cmd_Vel로 getCmdVel을 싸서 turtlesim_node로 볼 수 있도록 함
    this->turtlepub.publish(getCmdVel);
};
```

1번 모드

Using Turtlesim

Recorder -> string

Saver -> text file



```
void Recorder::cmdVelRecorder(double linear_x, double angular_z) {  
    std::string str_linear_x = std::to_string(linear_x);  
    std::string str_angular_z = std::to_string(angular_z);  
    std::string record_line = str_linear_x + " " + str_angular_z + "\n";  
    this->record = record.append(record_line);  
};
```

```
void Recorder::recordSaver() {  
    std::ofstream output(this->filename);  
    output << record;  
    // output.close(); // 스코프를 벗어나는 순간 알아서 close() 진행됨  
    std::cout << "Successfully Saved!" << std::endl;  
};[EOF]
```



2번 모드 : Using Servinggo

- Servinggo가 조이스틱의 publishing 값을 subscribe함
- Publishing된 값을 .txt 파일로 저장



2번 모드 Using Turtlesim

Sub -> /joy

Pub -> /cmd_vel



```
joysub = nh.subscribe("/joy", 1, &Player::playerCallback, this);  
servinggopub = nh.advertise<geometry_msgs::Twist>("/cmd_vel", 1);
```

```
void Player::playerCallback(const sensor_msgs::Joy &msg) {  
    geometry_msgs::Twist getCmdVel;  
  
    getCmdVel.linear.x = msg.axes[1] * (this->max_linear_x);  
    getCmdVel.angular.z = msg.axes[0] * (this->max_angular_z);  
  
    this->linear_x = msg.axes[1] * (this->max_linear_x);  
    this->angular_z = msg.axes[0] * (this->max_angular_z);  
  
    Player::cmdVelRecorder(this->linear_x, this->angular_z);  
    this->servinggopub.publish(getCmdVel);  
};
```




3번 모드 : Using Text File

- 저장된 text file을 개행문자 단위로 끊어 읽어옴
- 각 행마다 저장된 정보를 linear.x, angular.z로 파싱한 뒤
- geometry_msgs::Twist로 /cmd_vel에 publishing 함



text file의 형식

"[linear_x]" +

" " +

"[angular_z]"

+

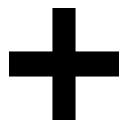
```
Open [v] [f]
1 -0.000000 0.000000
2 0.045910 0.000000
3 0.113193 0.000000
4 0.154428 0.000000
5 0.158771 0.000000
6 0.163114 0.000000
7 0.176138 0.000000
8 0.200000 0.000000
9 0.200000 -0.000325
10 0.200000 -0.035052
11 0.200000 -0.037217
12 0.200000 -0.035052
13 0.200000 -0.030709
14 0.200000 -0.022029
15 0.200000 -0.013342
16 0.200000 -0.002491
17 0.200000 -0.000000
18 0.167451 -0.000000
19 0.137068 -0.000000
20 0.124044 -0.000000
21 0.104512 -0.000000
22 0.091489 -0.000000
23 0.089317 -0.000000
24 0.102341 -0.000000
25 0.111021 -0.000000
26 0.117536 -0.000000
27 0.121873 -0.000000
28 0.126216 -0.000000
29 0.134896 -0.000000
30 0.152263 -0.000000
31 0.147919 -0.000000
32 0.156599 -0.000000
33 0.200000 -0.000000
34 0.200000 -0.000000
35 0.200000 -0.000000
36 0.200000 -0.000000
37 0.200000 -0.004662
38 0.200000 -0.011177
39 0.200000 -0.009006
40 0.200000 -0.004662
41 0.200000 -0.009006
42 0.200000 -0.004662
43 0.200000 -0.009006
```

3번 모드

Using Textfile

Sub -> None

Pub -> /cmd_vel



```
Replayer::Replayer(std::string filename) {
    this->filename = filename;

    std::ifstream file(this->filename);
    std::string tmp;

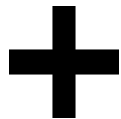
    if (file.is_open()) {
        while (file.good()) {
            getline(file, tmp);
            this->record.push_back(tmp);
        }
    }

    pub = nh.advertise<geometry_msgs::Twist>("/cmd_vel", 1);
};
```

3번 모드 Using Textfile

Sub -> None

Pub -> /cmd_vel



```
void Replayer::replayByServinggo() {
    geometry_msgs::Twist read_cmd_vel;
    std::string dummy;

    std::cout << "START By Entering '1' value>>>" << std::endl;
    std::cin >> dummy;

    std::vector<std::string>::iterator it = std::begin(this->record);

    ros::Rate loop_rate(20);

    while (it != std::end(this->record)) {
        std::string line_info = (*it);

        std::vector<std::string> x_and_z = Replayer::splitInfo(line_info);

        std::string x_val = x_and_z.front();
        std::string z_val = x_and_z.back();

        if (x_val == " " || x_val == "" || z_val == " " || z_val == "") {
            read_cmd_vel.linear.x = 0;
            read_cmd_vel.angular.z = 0;
        }
        else {
            read_cmd_vel.linear.x = std::stod(x_val);
            read_cmd_vel.angular.z = std::stod(z_val);
        }

        std::cout << "(" << read_cmd_vel.linear.x << ", ";
        std::cout << read_cmd_vel.angular.z << ")" << std::endl;

        this->pub.publish(read_cmd_vel);

        ros::spinOnce();

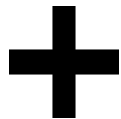
        loop_rate.sleep();

        ++it;
    }
}
```

3번 모드 Using Textfile

Sub -> None

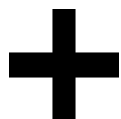
Pub -> /cmd_vel



```
// "0.200000 0.000332" -> ["0.200000", "0.000332"]
std::vector<std::string> Replayer::splitInfo(std::string &line_info) {
    std::vector<std::string> x_and_z;
    boost::split(x_and_z, line_info, boost::is_any_of(" "));
    return x_and_z;
}[EOF]
```

Auto Start With System Booting

`systemd`를 활용해
시스템 부팅 시
특정 launch 파일 실행시키기



5

나는 systemd 를 사용하고 있습니다. 다음은 저에게 도움이 된 몇 가지 링크입니다.

18년 5월 8일 답변May 8 '18

BhanuKiran.Chaluvadi

231 • 44 • 59 • 66

18년 5월 8일 업데이트됨May 8 '18

1. 시스템화된 서비스 생성

2. 부팅 시 라이다를 시작하려고 하므로 포트 권한/사용 가능한 시퀀스로 필요하여 여기에 도움이 될 수 있는 링크 가 있습니다.

다음은 내 서비스와 .sh 파일의 모습입니다.

서비스

```
[Unit]
Description=start roscore
After=remote-fs.target
After=syslog.target
[Service]
ExecStart=/usr/local/bin/rosCoreService.sh
Restart=on-abort
[Install]
WantedBy=multi-user.target
```

.sh 파일

```
#!/bin/bash
source /home/bc/.bashrc
source /opt/ros/kinetic/setup.bash
source /home/bc/catkin_ws/devel/setup.bash
roslaunch multitower_rplidar rplidar.launch
```

그들 중 일부는 필요하지 않을 수 있지만 그 중 어떤 것이 있는지 모르겠습니다.

코멘트를 추가하다

링크

1

단계는 다음과 같습니다.

18년 5월 8일 답변May 8 '18

심박

226 • 4 • 11 • 18

https://www.github.com...

18년 5월 8일 업데이트됨May 8 '18

1. 새 스크립트 파일을 만들 수 있습니다 `startup.sh`.

2. `source /opt/ros/<distro>/setup.sh` 시작 파일에서 roscore 및 roslaunch를 호출하기 전에 ros 환경을 소싱해야 하므로 다음을 추가하여 환경을 소싱하십시오. `/opt/ros/<distro>/setup.sh`

3. `roscore` 이제 또는 와 같은 시작 파일에 ros 명령을 추가할 수 있습니다 `roslaunch`. 직접 roscore 또는 rolaunch가 작동하지 않으면 행을 추가하여 `ros/<distro>/bin`에서 이 명령을 실행하십시오 `/opt/ros/<distro>/bin/roscore`.

4. 파일에 실행 권한 부여 `sudo chmod 755 /etc/init.d/startup.sh`

참고: 새 파일을 만드는 동안 `#!/bin/sh` 첫 번째 줄에 추가했는지 확인하십시오.

Use case: auto start MAVROS node

- Create a shell script and type the commands that you would execute in a normal terminal. For example,

```
mkdir ~/scripts
cd ~/scripts
touch startup_launch.sh
chmod +x startup_launch.sh
```

Type the following in the `startup_launch.sh` file (you can use the `nano startup_launch.sh` command). It is assumed that the username is `odroid`

```
#!/bin/bash
source /opt/ros/kinetic/setup.bash
source /home/odroid/catkin_ws/devel/setup.bash
roslaunch mavros px4.launch
```

- Create `mavros.service` file in `/lib/systemd/system`

```
cd /lib/systemd/system
sudo nano mavros.service
```

- Add the following contents:

```
[Unit]
Description=mavros
[Service]
Type=Forking
ExecStart=/home/odroid/scripts/startup_launch.sh
Restart=on-failure
[Install]
WantedBy=multi-user.target
```

Save and exit by hitting `CTRL+x`, then `y`, then `[ENTER]`

Then run:

```
sudo systemctl daemon-reload
```

And enable it on boot:

```
sudo systemctl enable mavros.service
```

Then, reboot and `px4.launch` should be executed after boot.

To disable a service,

```
sudo systemctl disable mavros.service
```

robot_upstart

melodic noetic [Show EOL distros:](#) ☐[Documentation Status](#)

Package Summary

[✓ Released](#) [✓ Continuous Integration: 6 / 6](#) [✓ Documented](#)

The robot_upstart package provides scripts which may be used to install and uninstall Ubuntu Linux upstart jobs which launch groups of roslaunch files.

- Maintainer status: maintained
- Maintainer: Chris Iverach-Brereton <civerachb AT clearpathrobotics DOT com>, Tony Baltovski <tbaltovski AT clearpathrobotics DOT com>
- Author: Mike Purvis <mpurvis AT clearpathrobotics DOT com>
- License: BSD
- Source: git https://github.com/clearpathrobotics/robot_upstart.git (branch: noetic-devel)

Contents

1. Overview
2. Used by

1. Overview

This package allows robot administrators to quickly and easily generate one or more background jobs to run roslaunch invocations on system startup. You can use it directly from the command-line, or via a simple Python API.

Please see usage and docs here: http://docs.ros.org/jade/api/robot_upstart/html/

2. Used by

- [Robots/Care-O-bot](#)
- [Robots/Grizzly](#)
- [Robots/Husky](#)
- [Robots/Jackal](#)

Package Links

[Code API](#)[FAQ](#)[Changelog](#)[Change List](#)[Reviews](#)[Dependencies \(3\)](#)[Jenkins jobs \(10\)](#)

ROS 2 Documentation

The ROS Wiki is for ROS 1. Are using ROS 2 (Dashing/Foxy/Rolling)? [Check out the ROS 2 Documentation](#)

Wiki

[Distributions](#)[ROS/Installation](#)[ROS/Tutorials](#)[RecentChanges](#)[robot_upstart](#)

Page

[Immutable Page](#)[Info](#)[Attachments](#)

More Actions:

[Raw Text](#)[Do](#)

User

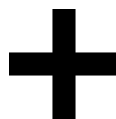
[Login](#)

Auto Start With System Booting

`systemd`를 활용해

시스템 부팅 시

특정 launch 파일 실행시키기



```
<node name="turtle_to_servinggo_node" pkg="turtle_to_servinggo" type="turtle_to_servinggo_node" output="screen">
  <param name="max_linear" type="double" value="0.2" />
  <param name="max_angular" type="double" value="0.2" />
</node>
```

```
double max_linear, max_angular;
max_linear = 0;
max_angular = 0;

ros::NodeHandle nh_param("~");
nh_param.getParam("max_linear", max_linear);
nh_param.getParam("max_angular", max_angular);

Player player(max_linear, max_angular, filename);
```

```
getCmdVel.linear.x = msg.axes[1] * (this->max_linear_x);
getCmdVel.angular.z = msg.axes[0] * (this->max_angular_z);
```

Param으로 최대 속도 지정받기

Roslaunch 의 node 에 param 값을
입력하여 최대 속도를 제한함



Bag file 만들기

Text file 대신 bag file 로 저장하는 방법

```
class DevJoyController {  
private:  
    ros::NodeHandle nh;  
  
    ros::Subscriber sub;    // 조이스틱이 보내는 publish 를 받아서  
    ros::Publisher pub;    // /cmd_vel로 publish 함  
  
    double linear;  
    double angular;  
  
    double max_linear;  
    double max_angular;  
  
    rosbag::Bag cmd_vel_bag;
```

선언

```
DevJoyController::DevJoyController(double max_linear, double max_angular) {  
    std::cout << "=== DevJoyController를 시작합니다. ===" << std::endl;  
  
    sub = nh.subscribe("/joy", 1, &DevJoyController::joyCallback, this);  
    pub = nh.advertise<geometry_msgs::Twist>("/cmd_vel", 1);  
    if (max_linear > 0.5 || max_angular > 0.5) {  
        this->max_linear = 0.33;  
        this->max_angular = 0.33;  
        return;  
    }  
    this->max_linear = max_linear;  
    this->max_angular = max_angular;  
  
    this->cmd_vel_bag.open("cmd_vel_tester.bag", rosbag::bagmode::Write);  
};
```

생성

```
pub.publish(toCmdVel);  
cmd_vel_bag.write("/cmd_vel", ros::Time::now(), toCmdVel);  
.
```

작성

```
DevJoyController::~DevJoyController() {  
    std::cout << "=== DevJoyController를 종료합니다. ===" << std::endl;  
  
    this->cmd_vel_bag.close();  
};
```

소멸

