

Week_2

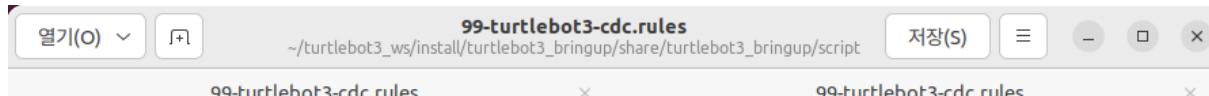
14. USB Port Setting for OpenCR

```
$ sudo cp `ros2 pkg prefix turtlebot3_bringup`/share/turtlebot3_bringup/script/99-turtlebot3-cdc.rules /etc/udev/rules.d/  
$ sudo udevadm control --reload-rules  
$ sudo udevadm trigger
```

```
$ sudo cp `ros2 pkg prefix turtlebot3_bringup`/share/turtlebot3_bringup/script/99-turtlebot3-cdc.rules /etc/udev/rules.d/  
$ sudo udevadm control --reload-rules  
$ sudo udevadm trigger
```

- sudo cp `ros2 pkg prefix turtlebot3_bringup`/share/turtlebot3_bringup/script/99-turtlebot3-cdc.rules /etc/udev/rules.d/ 명령어는 패키지의 설치 경로를 반환한다.
- ros2 pkg prefix 명령어는 '/share/turtlebot3_bringup/script/' 을 덧붙이고, 99-turtlebot3-cdc.rules 파일을 복사한다.
- 99-turtlebot3-cdc.rules 파일을 [패키지의 설치 경로]/share/turtlebot3_bringup/script/로부터 복사해서

/etc/udev/rules.d/ 디렉토리에 붙혀넣는 명령이다.



```
1 #http://linux-tips.org/t/prevent-modem-manager-to-capture-usb-serial-devices/284/2.  
2  
3 #cp rules /etc/udev/rules.d/  
4 #sudo udevadm control --reload-rules  
5 #sudo udevadm trigger  
6  
7 ATTRS{idVendor}=="0483" ATTRS{idProduct}=="5740", ENV{ID_MM_DEVICE_IGNORE}="1", MODE=="0666"  
8 ATTRS{idVendor}=="0483" ATTRS{idProduct}=="df11", MODE=="0666"  
9 ATTRS{idVendor}=="fff1" ATTRS{idProduct}=="ff48", ENV{ID_MM_DEVICE_IGNORE}="1", MODE=="0666"  
10 ATTRS{idVendor}=="10c4" ATTRS{idProduct}=="ea60", ENV{ID_MM_DEVICE_IGNORE}="1", MODE=="0666"
```



```
1 #http://linux-tips.org/t/prevent-modem-manager-to-capture-usb-serial-devices/284/2.  
2  
3 #cp rules /etc/udev/rules.d/  
4 #sudo udevadm control --reload-rules  
5 #sudo udevadm trigger  
6  
7 ATTRS{idVendor}=="0483" ATTRS{idProduct}=="5740", ENV{ID_MM_DEVICE_IGNORE}="1", MODE=="0666"  
8 ATTRS{idVendor}=="0483" ATTRS{idProduct}=="df11", MODE=="0666"  
9 ATTRS{idVendor}=="fff1" ATTRS{idProduct}=="ff48", ENV{ID_MM_DEVICE_IGNORE}="1", MODE=="0666"  
10 ATTRS{idVendor}=="10c4" ATTRS{idProduct}=="ea60", ENV{ID_MM_DEVICE_IGNORE}="1", MODE=="0666"
```

1. 즉, 이 명령어는 Udev 룰을 터틀봇 패키지에서 리눅스 시스템으로 복사하고
2. \$ sudo udevadm control - -reload-rules로 장치 관리자의 규칙을 불러온다.
3. \$ sudo udevadm trigger 으로 규칙을 적용한다.

OpenCR Setup

3. 3. OpenCR Setup

1. Connect the [OpenCR](#) to the Raspberry Pi using the micro USB cable.
2. Install required packages on the Raspberry Pi to upload the [OpenCR](#) firmware.

```
$ sudo dpkg --add-architecture armhf
$ sudo apt update
$ sudo apt install libc6:armhf
```

- \$ sudo dpkg - -add-architecture armhf
 - 패키지 관리자(dpkg)의 목록에 ARM hard-float(armhf) 의 지원을 추가하는 명령어이다.
ARM프로세서 기반의 장치의 패키지로 ‘libc6:armhf’ 패키지를 설치한다
- 라즈베리파이4는 arm architecture 64bit를 사용한다.

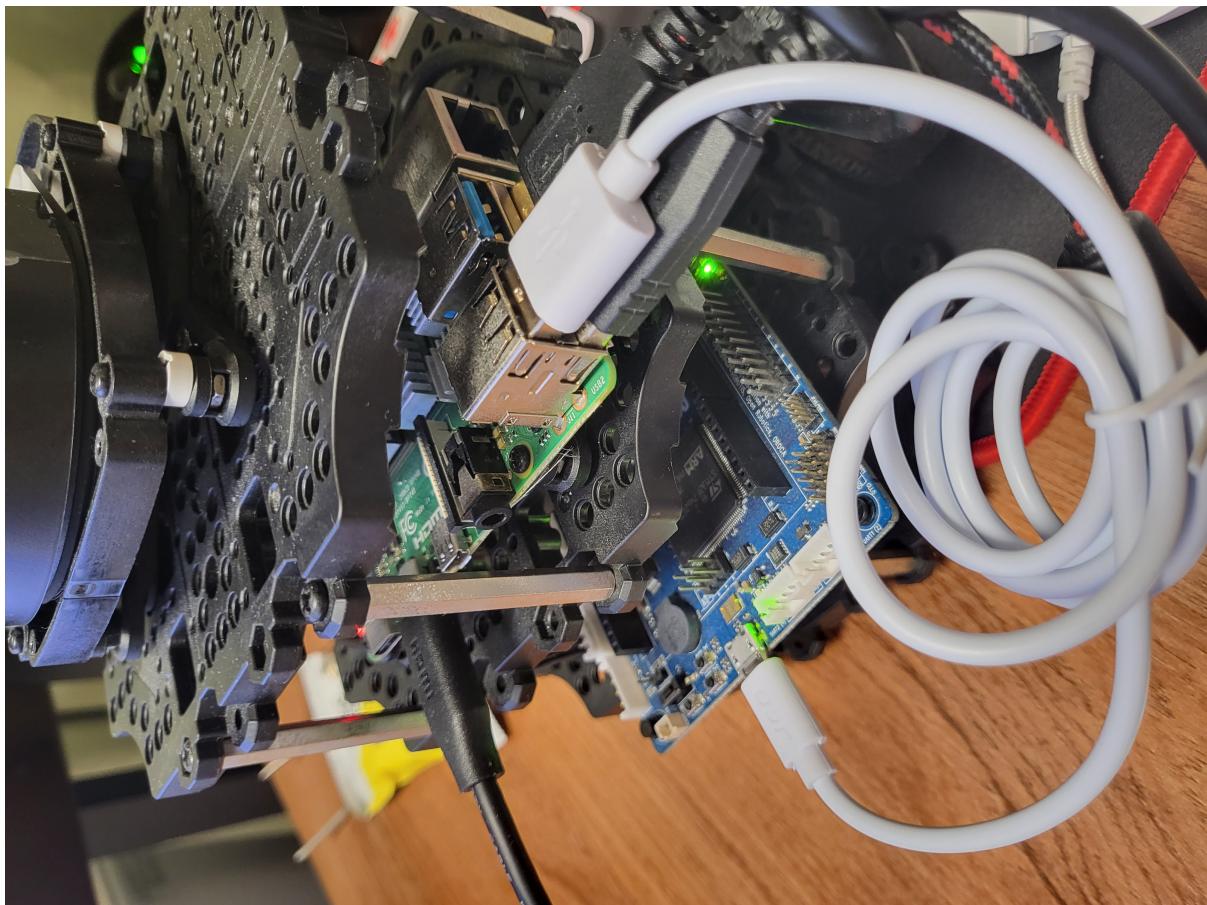
```
turtle@turtle:~$ uname -m
aarch64
```

- OpenCR은 ARM Cortex-M7 32bit를 사용한다.

-
3. Depending on the platform, use either [burger](#) or [waffle](#) for the **OPENCR_MODEL** name.

```
$ export OPENCR_PORT=/dev/ttyACMO
$ export OPENCR_MODEL=burger
$ rm -rf ./opencr_update.tar.bz2
```

- \$ export OPENCR_PORT=/dev/ttyACMO
 - OPENCR_PORT 환경 변수로 /dev(device) 디렉토리의 ttyACMO를 설정한다.
- device 디렉토리는 소프트웨어가 하드웨어와 통신하는 인터페이스 포인트이다.
USB 등으로 라즈베리파이4에 OpenCR이 연결되면
OS는 device file을 만들어 연결을 표현한다.
- 라즈베리파이와 OpenCR을 USB 포트로 연결



- /dev 디렉토리에 ttyACM0 파일이 생성된 것을 확인할 수 있다.

```
ls /dev/ttyACM*
ls: '/dev/ttyACM*'에 접근할 수 없음:
/ttyACM0
```

4. Download the firmware and loader, then extract the file.

```
$ wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS2/latest/opencr_update.tar
$ tar -xvf ./opencr_update.tar.bz2
```

- 펌웨어와 loader를 다운받고, 압축을 풀다.

- wget은 웹에서 파일을 다운받을 수 있게 해주는 커멘드라인 유틸리티이다.

5. Upload firmware to the OpenCR.

```
$ cd ~/opencr_update
$ ./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
```

- update.sh파일에
 - 환경변수(\$) OPENCR_PORT
 - \$ OPENCR_MODEL .opencr
 - 즉,
 - 인자1로 /dev/ttyACM0
 - 인자2로 burger.opencr

을 전달해서 실행한다.

```
turtle@turtle:~/opencr_update$ ./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
search64
arm
OpenCR Update Start..
opencr_ld_shell ver 1.0.0
opencr_ld_main
  ] file name      : burger.opencr
  ] file size       : 178 KB
  ] fw_name        : burger
  ] fw_ver         : 1.2.2
[OK] Open port      : /dev/ttyACM0
  ]
  ] Board Name     : OpenCR R1.0
  ] Board Ver       : 0x17020800
  ] Board Rev       : 0x00000000
[OK] flash_erase    : 0.99s
[OK] flash_write    : 1.58s
[OK] CRC Check      : 12206B2 12206B2 , 0.006000 sec
[OK] Download
[OK] jump_to_fw
```

- Model_Scan 예제를 이용해서 OpenCR에 연결된 Dynamixel을 찾음

a_Model_Scan | Arduino IDE 2.1.1

File Edit Sketch Tools Help

OpenCR Board

```

a_Model_Scan.ino
 6 * You may obtain a copy of the License at
 7 *
 8 *   http://www.apache.org/licenses/LICENSE-2.0
 9 *
10 * Unless required by applicable law or agreed to in writing, software
11 * distributed under the License is distributed on an "AS IS" BASIS,
12 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 * See the License for the specific language governing permissions and
14 * limitations under the License.
15 ****
16 /* Authors: Taehun Lim (Darby) */
17
18 #include <DynamixelWorkbench.h>
19
20 #if defined(__OPENCM904__)
21     #define DEVICE_NAME "3" //Dynamixel on Serial3(USART3) <-OpenCM 485EXP
22 #elif defined(__OPENCN__)
23     #define DEVICE_NAME ""
24 #endif
25
26 #define BAUDRATE 1000000
27
28 DynamixelWorkbench dxl_wb;
29
30 void setup()
31 {
32     Serial.begin(57600);
33     while(!Serial); // Wait for Opening Serial Monitor
34
35     const char *log = NULL;
36     bool result = false;
37
38     uint8_t scanned_id[16];
39     uint8_t dxl_cnt = 0;
40     uint8_t range = 100;
41
42     result = dxl_wb.init(DEVICE_NAME, BAUDRATE, &log);
43     if (result == false)
44     {
45
        Serial Monitor x Output
Not connected. Select a board and a port to connect automatically.
Succeeded to init : 1000000
Wait for scan...
Find 0 Dynamixels
Succeeded to init : 1000000
Wait for scan...
Succeeded to init : 1000000
Wait for scan...
Succeeded to init : 1000000
Wait for scan...
Find 0 Dynamixels
Succeeded to init : 1000000
Wait for scan...
Find 0 Dynamixels
Succeeded to init : 1000000
Wait for scan...
Find 0 Dynamixels
Succeeded to init : 1000000
Wait for scan...
Find 2 Dynamixels
id : 1 model name : XL430-W250
id : 2 model name : XL430-W250

```

```

Succeeded to init : 1000000
Wait for scan...
Find 2 Dynamixels
id : 1 model name : XL430-W250
id : 2 model name : XL430-W250

```

- 터틀봇 구동 예제 1,2번
 - 1번 스위치(12인치 전방으로 이동)

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b6fcc898-6f24-45c0-b22b-dc3040c31370/20230708_175619.mp4

- 2번 스위치(180도 회전)

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/560f0787-2ebf-419d-a9df-810c3e3b9664/20230708_175555.mp4

- ssh로, PC에서 라즈베리파이4 원격 접속 성공

```
kody@desktop:~$ ssh turtle@192.168.0.86
The authenticity of host '192.168.0.86 (192.168.0.86)' can't be established.
ED25519 key fingerprint is SHA256:1C/zmflE1nFc+mpllh+OKMqzLNPNvhs5n2YbdbQ4AsJw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.86' (ED25519) to the list of known hosts.
turtle@192.168.0.86's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1033-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
추가 업데이트를 확인하려면 apt list --upgradable 을 실행하세요.

19 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

turtle@turtle:~$ ]
```

- \$ ros2 launch turtlebot3_bringup robot.launch.py 실행시 프로세스가 죽는 오류 발생

→ 커뮤니티에서 비슷한 오류를 발견 ,

[https://www.robotis.com/service/forum_view.php?
bbs_no=2591656&page=1&save_sca=&sca=&save_stx=&stx=&sfl=](https://www.robotis.com/service/forum_view.php?bbs_no=2591656&page=1&save_sca=&sca=&save_stx=&stx=&sfl=)

```
[robot_state_publisher-1] [INFO] [1688808550.360057228] [robot_state_publisher]: got segment wheel_right_link
[ERROR] [turtlebot3_ros-3]: process has died [pid 4447, exit code -6, cmd '/home/turtle/turtlebot3_ws/install/turtlebot3_node/lib/turtlebot3_node/turtlebot3_ros -i /dev/ttyACM0 --ros-args --params-file /home/turtle/turtlebot3_ws/install/turtlebot3_bringup/share/turtlebot3_bringup/param/burger.yaml'].
```

- 화면이 안떠서, 리셋하고 다시

- UBUNTU SERVER 22.04.2 LTS 64bit 이미지로 설치
- (메뉴얼에는 22.04.5 버전으로 설치이지만 선택창에 없어서 비슷한 것으로 설치)

```
$ sudo nano /writable/etc/netplan/50-cloud-init.yaml

network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: yes
      dhcp6: yes
      optional: true
  wifis:
    wlan0:
      dhcp4: yes
      dhcp6: yes
      access-points:
        WIFI_SSID:
          password: WIFI_PASSWORD
```

- ifconfig 실행시

- inet (IPv4)는
 - 네트워크에서 고유한 호스트를 식별 가능하게 해주는 32비트 주소이다.
 - ex, 192.168.0.87
- netmask (subnet mask)는
 - 네트워크 영역과 호스트 영역을 분할하기 위한 32비트 정보이다.
 - 1은 네트워크 영역으로 사용하겠다는 의미이고
 - 0은 호스트 IP로 사용하겠다는 의미이다

- ex, 255.255.255.0
- (1111 1111 . 1111 1111 . 1111 1111 . 0000 0000)
- broadcast는
 - 네트워크 상의 모든 호스트를 대상으로 데이터를 전송하기 위한 IP 주소이다.
 - ex. 192.168.0.255

```

source /opt/ros/numode/setup.bash
source ~/turtlebot3_ws/install/setup.bash
export ROS_DOMAIN_ID=30 #TURTLEBOT3
export LDS_MODEL=LDS-01
ubuntu@ubuntu:~$ export OPENCR_PORT=/dev/ttyACM0
ubuntu@ubuntu:~$ export OPENCR_MODEL=burger
ubuntu@ubuntu:~$ export TURTLEBOT3_MODEL=burger
ubuntu@ubuntu:~$ ros2 launch turtlebot3_bringup robot.launch.py
[INFO] [launch]: All log files can be found below /home/ubuntu/.ros/log/2023-07-12-19-42-13-558702-ubuntu-15117
[INFO] [launch]: Default logging verbosity is set to INFO
urdf_file_name : turtlebot3_burger.urdf
[INFO] [robot_state_publisher-1]: process started with pid [15118]
[INFO] [hlds_laser_publisher-2]: process started with pid [15120]
[INFO] [turtlebot3_ros-3]: process started with pid [15122]
[turtlebot3_ros-3] [INFO] [1689158534.355653662] [turtlebot3_node]: Init TurtleBot3 Node Main
[turtlebot3_ros-3] [INFO] [1689158534.359340144] [turtlebot3_node]: Init DynamixelSDKWrapper
[turtlebot3_ros-3] [INFO] [1689158534.363103810] [DynamixelSDKWrapper]: Succeeded to open the port(/dev/ttyACM0)!
[turtlebot3_ros-3] [INFO] [1689158534.374949392] [DynamixelSDKWrapper]: Succeeded to change the baud rate!
[hlds_laser_publisher-2] [INFO] [1689158534.382671722] [hlds_laser_publisher]: Init hlds_laser_publisher Node Main
[hlds_laser_publisher-2] [INFO] [1689158534.385194073] [hlds_laser_publisher]: port : /dev/ttyUSB0 frame_id : base_scan
[robot_state_publisher-1] [INFO] [1689158534.389793097] [robot_state_publisher]: got segment base_footprint
[robot_state_publisher-1] [INFO] [1689158534.390908525] [robot_state_publisher]: got segment base_link
[robot_state_publisher-1] [INFO] [1689158534.391148521] [robot_state_publisher]: got segment base_scan
[robot_state_publisher-1] [INFO] [1689158534.391197483] [robot_state_publisher]: got segment caster_base_link
[robot_state_publisher-1] [INFO] [1689158534.391228557] [robot_state_publisher]: got segment imu_link
[robot_state_publisher-1] [INFO] [1689158534.391254242] [robot_state_publisher]: got segment wheel_left_link
[robot_state_publisher-1] [INFO] [1689158534.391280352] [robot_state_publisher]: got segment wheel_right_link
[turtlebot3_ros-3] [ERROR] [1689158534.409986740] [turtlebot3_node]: Failed connection with Devices
[turtlebot3_ros-3] [INFO] [1689158534.418790219] [turtlebot3_node]: Add Motors
[turtlebot3_ros-3] [INFO] [1689158534.419319397] [turtlebot3_node]: Add Wheels
[turtlebot3_ros-3] [INFO] [1689158534.419490783] [turtlebot3_node]: Add Sensors
[turtlebot3_ros-3] terminate called after throwing an instance of 'rclcpp::exceptions::RCLError'
[turtlebot3_ros-3] what(): could not create publisher: rcl node's context is invalid, at ./src/rcl/node.c:428
[ERROR] [turtlebot3_ros-3]: process has died [pid 15122, exit code -6, cmd '/home/ubuntu/turtlebot3_ws/install/turtlebot3_node/lib/turtlebot3_node/turtlebot3_ros -i /dev/ttyACM0 --ros-args --params-file /home/ubuntu/turtlebot3_ws/install/turtlebot3_bringup/share/turtlebot3_bringup/param/burger.yaml']. ^C[WARNING] [launch]: user interrupted with ctrl-c (SIGINT)
[robot_state_publisher-1] [INFO] [1689158549.991573479] [rclcpp]: signal_handler(signum=2)
[hlds_laser_publisher-2] [INFO] [1689158549.992020380] [rclcpp]: signal_handler(signum=2)
[ERROR] [hlds_laser_publisher-2]: process has died [pid 15120, exit code 255, cmd '/opt/ros/humble/lib/hlds_lfcd_lds_driver/hlds_laser_publisher --ros-args -r __node:=hlds_laser_publisher --params-file /tmp/launch_params_0wkmcp9']. [INFO] [robot_state_publisher-1]: process has finished cleanly [pid 15118]

```

```

source /opt/ros/humble/setup.bash
source ~/turtlebot3_ws/install/setup.bash
export ROS_DOMAIN_ID=30 #TURTLEBOT3
export LDS_MODEL=LDS-01

```

```

ubuntu@ubuntu:~$ export OPENCR_PORT=/dev/ttyACM0
ubuntu@ubuntu:~$ export OPENCR_MODEL=burger
ubuntu@ubuntu:~$ export TURTLEBOT3_MODEL=burger
ubuntu@ubuntu:~$ ros2 launch turtlebot3_bringup robot.launch.py
[INFO] [launch]: All log files can be found below /home/ubuntu/.ros/log/2023-07-12-19-42-13-558702-
ubuntu-15117
[INFO] [launch]: Default logging verbosity is set to INFO
urdf_file_name : turtlebot3_burger.urdf
[INFO] [robot_state_publisher-1]: process started with pid [15118]
[INFO] [hlds_laser_publisher-2]: process started with pid [15120]
[INFO] [turtlebot3_ros-3]: process started with pid [15122]
[turtlebot3_ros-3] [INFO] [1689158534.355653662] [turtlebot3_node]: Init TurtleBot3 Node Main
[turtlebot3_ros-3] [INFO] [1689158534.359340144] [turtlebot3_node]: Init DynamixelSDKWrapper
[turtlebot3_ros-3] [INFO] [1689158534.363103810] [DynamixelSDKWrapper]: Succeeded to open the
port(/dev/ttyACM0)!
[turtlebot3_ros-3] [INFO] [1689158534.374949392] [DynamixelSDKWrapper]: Succeeded to change
the baudrate!
[hlds_laser_publisher-2] [INFO] [1689158534.382671722] [hlds_laser_publisher]: Init
hlds_laser_publisher Node Main
[hlds_laser_publisher-2] [INFO] [1689158534.385194073] [hlds_laser_publisher]: port : /dev/ttyUSB0
frame_id : base_scan
[robot_state_publisher-1] [INFO] [1689158534.389793097] [robot_state_publisher]: got segment
base_footprint
[robot_state_publisher-1] [INFO] [1689158534.390908525] [robot_state_publisher]: got segment
base_link
[robot_state_publisher-1] [INFO] [1689158534.391148521] [robot_state_publisher]: got segment
base_scan
[robot_state_publisher-1] [INFO] [1689158534.391197483] [robot_state_publisher]: got segment
caster_back_link
[robot_state_publisher-1] [INFO] [1689158534.391228557] [robot_state_publisher]: got segment
imu_link
[robot_state_publisher-1] [INFO] [1689158534.391254242] [robot_state_publisher]: got segment
wheel_left_link
[robot_state_publisher-1] [INFO] [1689158534.391280352] [robot_state_publisher]: got segment
wheel_right_link
[turtlebot3_ros-3] [ERROR] [1689158534.409986740] [turtlebot3_node]: Failed connection with
Devices
[turtlebot3_ros-3] [INFO] [1689158534.418790219] [turtlebot3_node]: Add Motors
[turtlebot3_ros-3] [INFO] [1689158534.419319397] [turtlebot3_node]: Add Wheels
[turtlebot3_ros-3] [INFO] [1689158534.419490783] [turtlebot3_node]: Add Sensors
[turtlebot3_ros-3] terminate called after throwing an instance of 'rclcpp::exceptions::RCLError'
[turtlebot3_ros-3] what(): could not create publisher: rcl node's context is invalid, at
./src/rcl/node.c:428
[ERROR] [turtlebot3_ros-3]: process has died [pid 15122, exit code -6, cmd
'/home/ubuntu/turtlebot3_ws/install/turtlebot3_node/lib/turtlebot3_node/turtlebot3_ros -i /dev/ttyACM0 --
ros-args --params-file
/home/ubuntu/turtlebot3_ws/install/turtlebot3_bringup/share/turtlebot3_bringup/param/burger.yaml'].
^C[WARNING] [launch]: user interrupted with ctrl-c (SIGINT)
[robot_state_publisher-1] [INFO] [1689158549.991573479] [rclcpp]: signal_handler(signum=2)
[hlds_laser_publisher-2] [INFO] [1689158549.992020380] [rclcpp]: signal_handler(signum=2)
[ERROR] [hlds_laser_publisher-2]: process has died [pid 15120, exit code 255, cmd
'opt/ros/humble/lib/hls_lfcד_ids_driver/hlds_laser_publisher --ros-args -r __node:=hlds_laser_publisher

```

```
--params-file /tmp/launch_params_0wkmcp9'].  
[INFO] [robot_state_publisher-1]: process has finished cleanly [pid 15118]
```