

ROS2 C++ NOTLARI

mkdir -p ros2_ws/src : workspace oluşturmak için.

ros2 pkg create my_cpp_pkg --build-type ament_cmake --dependencies rclcpp : my_cpp_pkg adında bir paket oluşturduk ve rclcpp bağımlılığını ekledik.

colcon build : workspacei build eder.

Eğer vscode rclcpp kütüphanesini tanımazsa CTRL+ Shift + P yapıp C/C++ Edit Conf (JSON)'u açıp **includePath** kısmına **/opt/ros/galactic/include/** bu pathi ekliyoruz.

İlk C++ Node

```

1  #include "rclcpp/rclcpp.hpp"
2
3  int main(int argc, char const *argv[])
4  {
5      rclcpp::init(argc, argv); // initialize ROS communication
6      auto node = std::make_shared<rclcpp::Node>("cpp_test"); // create shared pointer to node
7      RCLCPP_INFO(node->get_logger(), "Hello Cpp Node");
8
9      rclcpp::spin(node);
10     rclcpp::shutdown();
11     return 0;
12 }
13

```

Kodda kullanılan **make_shared** (akıllı işaretçi) kullanılmasının nedeni: shared pointer, rclcpp::Node nesnesinin yaşam döngüsünü otomatik olarak yönetir ve nesne, artık kullanılmadığında otomatik olarak bellekten serbest bırakılır. Ayrıca, make_shared kullanımı, nesne ve shared pointerın ayrı ayrı oluşturulmasını ve daha sonra shared pointerın nesneye atanmasını önler, dolayısıyla bellek kullanımını azaltır ve performansı artırır.

Kodumuzu yazdıktan sonra CmakeLists.txt üzerinde değişiklik yapmamız gereklidir:

```

add_executable(cpp_node src/my_first_node.cpp)
ament_target_dependencies(cpp_node rclcpp)

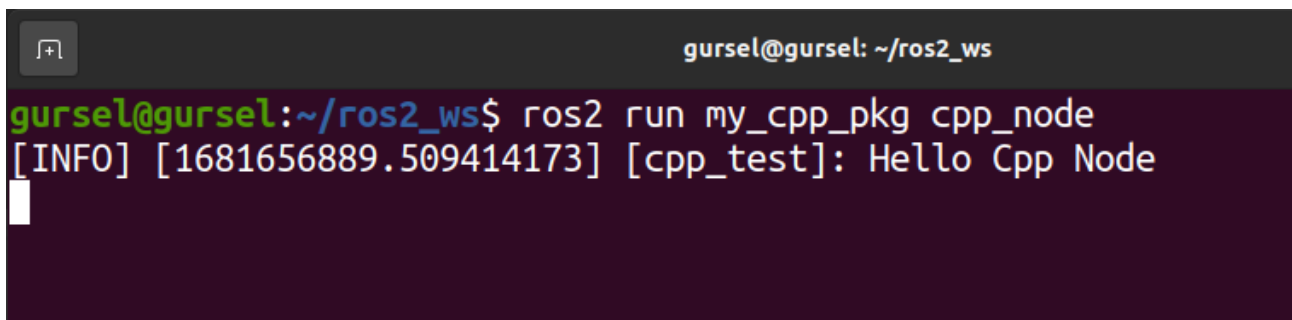
```

```

install(TARGETS
  cpp_node
  DESTINATION lib/${PROJECT_NAME}
)

```

Şimdi kodumuzu run edebiliriz.



```

gursel@gursel: ~/ros2_ws
gursel@gursel:~/ros2_ws$ ros2 run my_cpp_pkg cpp_node
[INFO] [1681656889.509414173] [cpp_test]: Hello Cpp Node

```

İlk C++ Node (OOP)

```
1  #include "rclcpp/rclcpp.hpp"
2
3  class MyNode: public rclcpp::Node
4
5  {
6  public:
7      MyNode(): Node("cpp_test"), counter_(0)
8      {
9          RCLCPP_INFO(this->get_logger(),"Hello Cpp Node");
10         timer_ = this->create_wall_timer(std::chrono::seconds(1),           // her 1 saniyede data gönderiyor
11                                         std::bind(&MyNode::timerCallback, this) // timer ile fonksiyonu bağladık
12         );
13     }
14
15 private:
16
17 void timerCallback() // call function in a given rate
18 {
19     counter++;
20     RCLCPP_INFO(this->get_logger(),"Gursel %d", counter_); // %d bir tamsayı değişkeni içeren bir yer tutucusu olarak işlev görür.
21 }
22
23 rclcpp::TimerBase::SharedPtr timer_;
24 int counter_;
25
26 };
27
28 int main(int argc, char const *argv[])
29 {
30     rclcpp::init(argc, argv); // initilaze ROS communication
31     auto node = std::make_shared<MyNode>(); // dinamik bellekte bir nesne oluşturur ve onunla ilişkili bir std::shared_ptr nesnesi döndürür.
32     rclcpp::spin(node);
33     rclcpp::shutdown();
34     return 0;
35 }
```