Writing a simple publisher & Subscriber (C++)

rclcpp {ROS Client Library C++}

class MinimalPublisher : public rclcpp:: Node

→ Creates the node class MinimalPublisher.

```
Public:
  MinimalPublisher()
  : Node("minimal_publisher"), count_ (0)
  {
    Publisher_ = this → create_publisher <std_msgs::msg::String>
                                              ("topic", 10);
    timer_ = this→ create_wall_timer (500ms,
    std:: bind(&MinimalPublisher::timer_callback, this));
  }
Private:
  Void timer_callback()
  {---}

  rclcpp :: TimerBase :: SharedPtr timer_;
  rclcpp:: Publisher <Std_msgs:: ~~MSG~~ msg:: string> :: SharedPtr publisher_;
  ---

int main(int argc, char * argv[])
{
  rclcpp::Init (argc, argv);
  rclcpp :: spin ( std:: make_shared < MinimalPublisher>());
  rclcpp :: Shutdown ();
  return 0;
}
```