Summary

To enable your ROS 2 Humble setup on a Raspberry Pi 5 to communicate with a Pixhawk 6C running PX4 via the uXRCE-DDS (formerly microRTPS) bridge, you will:

1. **Install system prerequisites** (Ubuntu 22.04 packages, ROS 2 Humble, PX4 dev tools)
2. **Install and launch the Micro XRCE-DDS Agent** on the companion computer
3. **Configure and start the uXRCE-DDS client** on the Pixhawk (via PX4)
4. **Create and build a ROS 2 workspace** with `px4_msgs` and `px4_ros_com`
5. **Launch** the agent, simulator/client, and ROS 2 bridge nodes

Below is a detailed, step-by-step guide with commands and explanations.

---

1. System Prerequisites

1.1 Update and Install Basic Tools

Ensure your Raspberry Pi is running Ubuntu 22.04 or a compatible Debian-based OS. Then install required utilities:

```
sudo apt update && sudo apt upgrade -y
sudo apt install locales curl gnupg2 lsb-release software-properties-common -y
```

These packages provide locale support, HTTPS transport, and repository management. (ROS 2 User Guide | PX4 Guide (main))

1.2 Install ROS 2 Humble

Set up the ROS 2 repository and install Humble desktop and development tools:

```
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key \
  -o /usr/share/keyrings/ros-archive-keyring.gpg
echo "deb [arch=
```

```
\((dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg]
\
  http://packages.ros.org/ros2/ubuntu\)
(lsb_release -sc) main" \
  | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
sudo apt update && sudo apt install ros-humble-desktop ros-dev-tools -y
echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc
source /opt/ros/humble/setup.bash
```

This installs ROS 2 Humble LTS and sets up your environment. (ROS 2 User Guide | PX4 Guide (main))

1.3 Install PX4 Development Environment (for SITL)

If you plan to use the PX4 simulator or build PX4 yourself:

```
cd ~
git clone https://github.com/PX4/PX4-Autopilot.git --recursive
bash ./PX4-Autopilot/Tools/setup/ubuntu.sh --no-sim-tools
cd PX4-Autopilot
make px4_sitl
```

This fetches PX4 source, installs dependencies, and builds the SITL binary. (ROS 2 User Guide | PX4 Guide (main))

---

2. Install & Launch the Micro XRCE-DDS Agent

PX4 v1.14+ uses uXRCE-DDS middleware, requiring an agent on your Pi to proxy uORB messages over DDS. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

2.1 Clone and Build Agent from Source

```
git clone -b v2.4.2 https://github.com/eProsima/Micro-XRCE-DDS-Agent.git
cd Micro-XRCE-DDS-Agent
mkdir build && cd build
cmake ..
make
sudo make install
sudo ldconfig /usr/local/lib/
```

This installs the Agent binary and dependencies (Fast CDR, Micro XRCE-DDS libraries). (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

2.2 (Alternative) Install via Snap

On Ubuntu systems:

```
sudo snap install micro-xrce-dds-agent --edge
```

Note: Snap's stable channel had known topic-creation errors; edge works reliably. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

2.3 Start the Agent

For PX4 SITL (UDP on port 8888):

```
MicroXRCEAgent udp4 -p 8888
```

For a UART link (e.g., Raspberry Pi's `/dev/ttyAMA0` at 921600 baud):

```
sudo MicroXRCEAgent serial --dev /dev/ttyAMA0 -b 921600
```

Run one agent per connection channel. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

---

3. Configure & Start the uXRCE-DDS Client on PX4

PX4's `uxrce_dds_client` is built into firmware by default but must be enabled/configured at runtime. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

1. **Disable existing MAVLink ports** (e.g., TELEM2) in the QGroundControl or via parameters:

   ```
   param set MAV_1_CONFIG 0
   ```

2. **Set uXRCE-DDS parameters** (if using Ethernet/UDP):

   ```
   param set UXRCE_DDS_PRT 8888
   ```

```
param set UXRCE_DDS_AG_IP <agent_IP_as_int32>
```

3. **Reboot PX4** for parameters to take effect.
4. **Start client** (default in SITL) or manually via:

```
uxrce_dds_client start -t udp -p 8888 -h <agent_IP> -n <namespace>
```

You can convert human-readable IP to the required int32 format with PX4's `convert_ip.py`. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

---

4. Create and Build the ROS 2 Workspace

Your ROS 2 workspace must include the exact message definitions used by PX4's client. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

4.1 Workspace Setup

```
mkdir -p ~/px4_ros_bridge_ws/src
cd ~/px4_ros_bridge_ws/src
```

4.2 Clone Interface Packages

```
git clone https://github.com/PX4/px4_msgs.git
git clone https://github.com/PX4/px4_ros_com.git
```

- `px4_msgs` provides uORB-to-ROS message definitions (ensure branch matches your PX4 version). (ROS 2 User Guide (PX4-ROS 2 Bridge) | PX4 User Guide (v1.12))
- `px4_ros_com` contains example nodes and the bridge logic. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))

4.3 Build with Colcon

```
cd ~/px4_ros_bridge_ws
source /opt/ros/humble/setup.bash
colcon build
```

This generates Fast DDS IDL files, compiles the agent code, and builds ROS 2 nodes. (ROS 2 User Guide (PX4-ROS 2 Bridge) | PX4 User Guide (v1.12))

---

5. Launch the Bridge

Open **three** terminals or use `tmux`:

1. **Agent Terminal**

```
source /opt/ros/humble/setup.bash
MicroXRCEAgent udp4 -p 8888
```

2. **PX4 Simulator/Client Terminal**

```
cd ~/PX4-Autopilot
make px4_sitl gz_x500   # starts SITL + uXRCE-DDS client
# or manually:
# pxh> micrortps_client start -t UDP
```

3. **ROS 2 Bridge Terminal**

```
cd ~/px4_ros_bridge_ws
source /opt/ros/humble/setup.bash
source install/local_setup.bash
ros2 launch px4_ros_com sensor_combined_listener.launch.py
```

You should see `RECEIVED DATA FROM SENSOR COMBINED` printed as uORB data is bridged. (ROS 2 User Guide (PX4-ROS 2 Bridge) | PX4 User Guide (v1.12))

---

6. Verification & Troubleshooting

- **Check topics**:

```
ros2 topic list
ros2 topic hz /SensorCombined_PubSubTopic
```

Verify expected topics and message rates. (ROS 2 User Guide (PX4-ROS 2 Bridge) | PX4 User Guide (v1.12))

- **Agent logs**: should show topic creation messages (e.g., `create_publisher`). (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))
- **Client logs**: PX4 console will log `uxrce_dds_client` synchronization. (uXRCE-DDS (PX4-ROS 2/DDS Bridge) | PX4 Guide (main))
- **Version mismatches**: ensure `px4_msgs` branch and PX4 release match; otherwise run the **ROS 2 Message Translation Node** (for v1.16+). (ROS 2 User Guide | PX4 Guide (main))

With these steps, your Raspberry Pi 5 running ROS 2 Humble will successfully communicate with your Pixhawk 6C via the PX4 uXRCE-DDS bridge.