

# Sigyn Custom Micro ROS Library for Teensy 4.1

From: <https://micro-xrce-dds.docs.eprosima.com/en/latest/installation.html#install-agent-client>

```
$ git clone https://github.com/eProsima/Micro-XRCE-DDS.git
$ cd Micro-XRCE-DDS
$ mkdir build && cd build
```

On Linux, inside of the `build` folder, execute the following commands:

```
$ cmake ..
$ make
$ sudo make install
```

```
export RMW_IMPLEMENTATION=rmw_microxrcedds
```

```
sudo apt install python3-vcstool

mkdir -p sigyn_microros_ws/src

cd sigyn_microros_ws/

git clone -b $ROS_DISTRO https://github.com/micro-ROS/micro_ros_setup.git src/micro_ros_setup

ros2 run micro_ros_setup create_firmware_ws.sh generate_lib

tee sigyn_colcon.meta <<EOF
{
  "names": {
    "tracetools": {
      "cmake-args": [
        "-DTRACETOOLS_DISABLED=ON",
        "-DTRACETOOLS_STATUS_CHECKING_TOOL=OFF"
      ]
    },
    "rosidl_typesupport": {
```

```

"cmake-args": [
    "-DROSIDL_TYPESUPPORT_SINGLE_TYPESUPPORT=ON"
]
},
"rcl": {
    "cmake-args": [
        "-DBUILD_TESTING=OFF",
        "-DRCL_COMMAND_LINE_ENABLED=OFF",
        "-DRCL_LOGGING_ENABLED=OFF"
    ]
},
"rcutils": {
    "cmake-args": [
        "-DENABLE_TESTING=OFF",
        "-DRCUTILS_NO_FILESYSTEM=ON",
        "-DRCUTILS_NO_THREAD_SUPPORT=ON",
        "-DRCUTILS_NO_64_ATOMIC=ON",
        "-DRCUTILS_AVOID_DYNAMIC_ALLOCATION=ON"
    ]
},
"microxrcedds_client": {
    "cmake-args": [
        "-DUCLIENT_PIC=OFF",
        "-DUCLIENT_PROFILE_UDP=OFF",
        "-DUCLIENT_PROFILE_TCP=OFF",
        "-DUCLIENT_PROFILE_DISCOVERY=OFF",
        "-DUCLIENT_PROFILE_SERIAL=OFF",
        "-UCLIENT_PROFILE_STREAM_FRAMING=ON",
        "-DUCLIENT_PROFILE_CUSTOM_TRANSPORT=ON"
    ]
},
"rmw_microxrcedds": {
    "cmake-args": [
        "-DRMW_UXRCE_MAX_NODES=1",
        "-DRMW_UXRCE_MAX_PUBLISHERS=30",
        "-DRMW_UXRCE_MAX_SUBSCRIPTIONS=10",
        "-DRMW_UXRCE_MAX_SERVICES=10",
        "-DRMW_UXRCE_MAX_CLIENTS=1",
        "-DRMW_UXRCE_MAX_HISTORY=4",
        "-DRMW_UXRCE_TRANSPORT=custom"
    ]
}
}
}

```

```

}
EOF

tee sigyn_toolchain.cmake <<EOF
SET(CMAKE_SYSTEM_NAME Generic)
set(CMAKE_CROSSCOMPILING 1)
set(CMAKE_TRY_COMPILE_TARGET_TYPE STATIC_LIBRARY)

set(CMAKE_C_COMPILER $ENV{TOOLCHAIN_PREFIX}gcc)
set(CMAKE_CXX_COMPILER $ENV{TOOLCHAIN_PREFIX}g++)

SET(CMAKE_C_COMPILER_WORKS 1 CACHE INTERNAL "")
SET(CMAKE_CXX_COMPILER_WORKS 1 CACHE INTERNAL "")

set(FLAGS "-O2 -mfloat-abi=hard -mfpv5-d16 -ffunction-sections -fdata-sections -fno-exceptions -nostdlib -mcpu=cortex-m7
-mthumb -D'RCUTILS_LOG_MIN_SEVERITY=RCUTILS_LOG_MIN_SEVERITY_NONE'" CACHE STRING "" FORCE)

set(CMAKE_C_FLAGS_INIT "-std=c11 ${FLAGS} -DCLOCK_MONOTONIC=0 -D'__attribute__(x)='" CACHE STRING "" FORCE)
set(CMAKE_CXX_FLAGS_INIT "-std=c++14 ${FLAGS} -fno-rtti -DCLOCK_MONOTONIC=0 -D'__attribute__(x)='" CACHE STRING
"" FORCE)

set(__BIG_ENDIAN__ 0)
EOF

#export TOOLCHAIN_PREFIX=/home/ros/aa-arduino-1.8.19/hardware/tools/arm/bin/arm-none-eabi-
export TOOLCHAIN_PREFIX=/home/ros/.arduino15/packages/teensy/tools/teensy-compile/11.3.1/arm/bin/arm-none-eabi-
#export RMW_IMPLEMENTATION=rmw_microros
ros2 run micro_ros_setup build_firmware.sh /home/ros/sigyn_microros_ws/sigyn_toolchain.cmake
/home/ros/sigyn_microros_ws/sigyn_colcon.meta

cp firmware/build/libmicroros.a ~/Arduino/libraries/micro_ros_arduino/src/imxrt1062/fpv5-d16-hard/

```

For the next, see: [https://micro.ros.org/docs/tutorials/core/teensy\\_with\\_arduino/](https://micro.ros.org/docs/tutorials/core/teensy_with_arduino/)

```

cd ~/sigyn_microros_ws
colcon build --symlink-install
source install/setup.bash
ros2 run micro_ros_setup create_agent_ws.sh
ros2 run micro_ros_setup build_agent.sh

```

```
ros@raven:~/sigyn_microros_ws$ ros2 topic list
/cmd_vel
/main_battery
/parameter_events
/roboclaw_status
/rosout
/sonar0Sensor
/sonar1Sensor
/sonar2Sensor
/sonar3Sensor
/teensy_diagnostics
/teensy_stats
/wheel_odom
```

# Creating custom Micro Ros library for Teensy 4

```
mkdir -p microros_humble_ws/src
```

```
cd microros_humble_ws/
```

```
git clone -b $ROS_DISTRO https://github.com/micro-ROS/micro_ros_setup.git
src/micro_ros_setup
```

```
rm -rf firmware/
```

```
ros2 run micro_ros_setup create_firmware_ws.sh generate_lib
```

nano ~/Arduino/libraries/micro\_ros\_arduino-2.0.7-humble/extras/library\_generation/colcon.meta  
ends with:

```
"microxrcedds_client": {
  "cmake-args": [
    "-DUCLIENT_PIC=OFF",
    "-DUCLIENT_PROFILE_UDP=OFF",
    "-DUCLIENT_PROFILE_TCP=OFF",
    "-DUCLIENT_PROFILE_DISCOVERY=OFF",
    "-DUCLIENT_PROFILE_SERIAL=OFF",
    "-UCLIENT_PROFILE_STREAM_FRAMING=ON",
    "-DUCLIENT_PROFILE_CUSTOM_TRANSPORT=ON",
    "-gg=768"
```

```

    ]
  },
  "rmw_microxrcedds": {
    "cmake-args": [
      "-DRMW_UXRCE_MAX_NODES=1",
      "-DRMW_UXRCE_MAX_PUBLISHERS=20",
      "-DRMW_UXRCE_MAX_SUBSCRIPTIONS=5",
      "-DRMW_UXRCE_MAX_SERVICES=1",
      "-DRMW_UXRCE_MAX_CLIENTS=1",
      "-DRMW_UXRCE_MAX_HISTORY=4",
      "-DRMW_UXRCE_TRANSPORT=custom"
    ]
  }
}
}
}

```

```

export TOOLCHAIN_PREFIX=/home/ros/arduino-1.8.19/hardware/tools/arm/bin/arm-none-eabi-
sudo apt install ros-humble-rmw-fastrtps-cpp

```

```

ros2 daemon stop

```

```

export RMW_IMPLEMENTATION=rmw_microxrcedds

```

```

ros2 run micro_ros_setup build_firmware.sh
~/Arduino/libraries/micro_ros_arduino-2.0.7-humble/extras/library_generation/teensy4_toolchain
.cmake
~/Arduino/libraries/micro_ros_arduino-2.0.7-humble/extras/library_generation/colcon.meta

```

```

ros2 run micro_ros_setup build_agent.sh

```

```

mv
/home/ros/Arduino/libraries/micro_ros_arduino-2.0.7-humble/src/imxrt1062/fpv5-d16-hard/libmic
roros.a
/home/ros/Arduino/libraries/micro_ros_arduino-2.0.7-humble/src/imxrt1062/fpv5-d16-hard/libmic
roros.a-ORIG

```

```

cp firmware/build/libmicroros.a
/home/ros/Arduino/libraries/micro_ros_arduino-2.0.7-humble/src/imxrt1062/fpv5-d16-hard/libmic
roros.a

```

```

source install/local_setup.bash

```

```

clear; teensy

```

---

```
clear;ros2 topic echo -f /teensy_diagnostics
```

```
ros2 topic echo /wheel_odom
```

```
in /microros_humble_ws  
source install/local_setup.bash  
clear;teensy
```

```
cd Arduino/libraries/micro_ros_arduino/  
docker run -it --rm -v $(pwd):/project --env MICROROS_LIBRARY_FOLDER=extras  
microros/micro_ros_static_library_builder:humble
```

```
in /Arduino/libraries/micro_ros_arduino/extras/library_generation/extra_packages/gripper
```

```
action/MoveElevator.action  
# Requested movement  
float32 goal_position  
---  
# Result  
float32 final_position  
---  
# Feedback  
float32 current_position
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 3.8)  
project(gripper)  
  
if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")  
  add_compile_options(-Wall -Wextra -Wpedantic)  
endif()  
  
# find dependencies  
find_package(ament_cmake REQUIRED)  
find_package(rosidl_default_generators REQUIRED)  
# uncomment the following section in order to fill in  
# further dependencies manually.  
# find_package(<dependency> REQUIRED)  
  
rosidl_generate_interfaces(${PROJECT_NAME}  
  "action/MoveElevator.action"
```

```

)

if(BUILD_TESTING)
  find_package(ament_lint_auto REQUIRED)
  # the following line skips the linter which checks for copyrights
  # comment the line when a copyright and license is added to all source
  files
  set(ament_cmake_copyright_FOUND TRUE)
  # the following line skips cpplint (only works in a git repo)
  # comment the line when this package is in a git repo and when
  # a copyright and license is added to all source files
  set(ament_cmake_cpplint_FOUND TRUE)
  ament_lint_auto_find_test_dependencies()
endif()

ament_export_dependencies(rosidl_default_runtime)
ament_package()

```

#### package.xml

```

<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd"
schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>gripper</name>
  <version>0.0.0</version>
  <description>TODO: Package description</description>
  <maintainer email="mike@wimblerobotics.com">ros</maintainer>
  <license>TODO: License declaration</license>

  <buildtool_depend>ament_cmake</buildtool_depend>

  <test_depend>ament_lint_auto</test_depend>
  <test_depend>ament_lint_common</test_depend>

  <build_depend>rosidl_default_generators</build_depend>
  <exec_depend>rosidl_default_runtime</exec_depend>
  <member_of_group>rosidl_interface_packages</member_of_group>
  <depend>action_msgs</depend>

```

```
<export>  
  <build_type>ament_cmake</build_type>  
</export>  
</package>
```

also create dirs  
include/tripper  
src