Simplified Guidelines for Getting Sensor Data in PX4

Given the build challenges on Raspberry Pi, here are some simplified approaches:

Option 1: Use Existing PX4 Tools

Instead of building custom apps, use the built-in tools that already exist in your firmware:

1. **Use the listener command**:

   Copy

   ```
   listener sensor_accel -n 60
   ```

   This will display 60 sensor readings from the accelerometer. Similarly:

   Copy

   ```
   listener sensor_baro -n 60
   listener sensor_gyro -n 60
   listener sensor_mag -n 60
   ```

2. **Use the topic echo command**:

   Copy

   ```
   uorb top echo sensor_accel 60
   ```

These commands don't require building any custom code.

Option 2: Modified Simple Sensor App

If you want to create a custom app but avoid the build issues:

1. Create a simple version (as provided) that only targets one sensor at a time to minimize complexity.
2. Make sure it's added to a build location that's already being compiled.

Installation Steps:

1. Create the directory:

bash Copy

```
mkdir -p ~/PX4-Autopilot/src/examples/simple_sensor_app
```

2. Copy the three files:
   - `simple_sensor_app.c`
   - `CMakeLists.txt`
   - `Kconfig`
3. If you're still having build issues, pick an existing example app that's already working in your firmware (like the `px4_simple_app`) and modify it directly rather than creating a new app.

Option 3: Modify Existing App

The simplest approach may be to modify an existing app that's already working:

1. Edit the existing `px4_simple_app.c` file:

bash Copy

```
nano ~/PX4-Autopilot/src/examples/px4_simple_app/px4_simple_app.c
```

2. Replace the counter in the for loop:

c Copy

```
for (int i = 0; i < 60; i++) {
```

3. If that still doesn't work, consider adding a simple printf after each reading to ensure you're seeing them:

c Copy

```
printf("Reading %d: %f %f %f\n", i, (double)accel.xyz[0], (double)accel.xyz[1], (do
```

Option 4: Use MAVLink

If you need to capture and process the sensor data outside of PX4:

1. Set up MAVLink streaming of the raw sensor data using QGroundControl
2. Capture the data using a MAVLink library in Python or another language on your computer

This avoids the need to build custom PX4 apps entirely.