## Software Bill of Materials (SBOM)

## Project Information

Date Created: November 18, 2024  
Last Updated: February 10, 2025

## 1. Component: Raspberry Pi Imager (Deb Package)

* Name: Raspberry Pi Imager
* Version: Latest (as of download date)
* Format: .deb
* Source URL: <https://downloads.raspberrypi.org/imager/imager_latest_amd64.deb>
* Dependencies:
  + libc6
  + libstdc++6
  + libgcc1
  + Other standard libraries required for Debian-based systems.

## 2. Component: Raspberry Pi Imager (Windows Executable)

* Name: Raspberry Pi Imager
* Version: Latest (as of download date)
* Format: .exe
* Source URL: <https://downloads.raspberrypi.org/imager/imager_latest.exe>
* Dependencies:
  + Windows system libraries (e.g., .NET Framework, Visual C++ Redistributables, if required).

## 3. Component: ROS 2 Jazzy Desktop

* Name: ROS 2 Jazzy Desktop
* Version: Jazzy Release
* Format: Debian Packages (.deb)
* Source URL: [ROS Jazzy Installation](https://docs.ros.org/en/jazzy/Installation/Ubuntu-Install-Debs.html)
* Dependencies:
  + software-properties-common
  + curl
  + Locale setup (en\_US.UTF-8)
  + ROS 2 GPG key and apt repository setup
  + System libraries:
    - libpython3-dev
    - build-essential

## 4. Component: QGroundControl

* Name: QGroundControl
* Version: Latest (as of download date)
* Formats: Executable for Windows, Linux binaries
* Source URL: [QGroundControl](https://docs.qgroundcontrol.com/)
* Dependencies:
  + For Linux:
    - Qt libraries (qt5-default, qtdeclarative5-dev)
    - GStreamer (gstreamer1.0\*)
    - Multimedia libraries.
  + For Windows:
    - Bundled dependencies in the installer.

## 5. Component: uXRCE-DDS Agent and Client (PX4-ROS2/DDS Bridge)

* Name: uXRCE-DDS Agent and Client
* Version: Latest (as of download date)
* Source URL: [PX4 uXRCE-DDS Documentation](https://docs.px4.io/main/en/middleware/uxrce_dds.html)
* Dependencies:
  + eProsima Micro XRCE-DDS library
  + Build tools:
    - cmake
    - make
  + System libraries:
    - libfastcdr-dev
    - libfastrtps-dev
  + ROS2 integration dependencies:
    - ROS2 workspace with px4\_msgs cloned and built using colcon.

## 6. Component: Mission Planner

* Name: Mission Planner
* Version: Latest (as of download date)
* Formats: Windows executable, Linux via MONO runtime
* Source URL: [Mission Planner Installation](https://ardupilot.org/planner/docs/mission-planner-installation.html)
* Dependencies:
  + For Windows:
    - DirectX
    - .NET Framework
    - Visual C++ Redistributables
  + For Linux:
    - MONO runtime.

## 7. Component: Prepware Remote Pilot by ASA

* Name: Prepware Remote Pilot by ASA
* Version: Latest (as of download date)
* Platform: Android application
* Source URL: Available on Google Play Store or official ASA website.
* Dependencies:
  + Android OS compatibility based on app-defined API levels.

## 8. Component: PX4\_msgs

* Name: PX4\_msgs (ROS2 Message Definitions for PX4)
* Version: Latest (as of download date)
* Source URL: [PX4\_ros\_com GitHub Repository](https://github.com/PX4/px4_ros_com)
* Dependencies:
  + ROS2 workspace setup.
  + Synchronization with PX4 firmware versions for compatibility.

## 9. Component: PX4\_ros\_com

* Name: PX4\_ros\_com (PX4 to ROS2 Bridge)
* Version: Latest (as of download date)
* Source URL: [PX4\_ros\_com GitHub Repository](https://github.com/PX4/px4_ros_com)
* Dependencies:
  + Direct dependency on px4\_msgs.
  + ROS2 workspace setup.
  + Build tools like colcon.

## 10. Python Libraries

The following Python libraries are required:

|  |  |  |
| --- | --- | --- |
| Library Name | Version | Source/Installation Command |
| numpy | Latest | pip install numpy |
| pandas | Latest | pip install pandas |
| scipy | Latest | pip install scipy |
| rclpy | ROS2-specific | Installed via ROS2 setup instructions |

Additional standard Python libraries used include:  
os, math, json, time, threading, collections.deque, and others.

### Libraries to include in Python Files:

import os

import math

import json

import time

import threading

import tkinter as tk

from collections import deque

from datetime import datetime

import numpy as np

import pandas as pd

import matplotlib

matplotlib.use("TkAgg")

from matplotlib.figure import Figure

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from scipy.ndimage import gaussian\_filter

import rclpy

from rclpy.node import Node

from rclpy.qos import QoSProfile, ReliabilityPolicy

from sensor\_msgs.msg import Image

from std\_msgs.msg import String

from px4\_msgs.msg import VehicleCommand, VehicleAttitude, SensorGps

from builtin\_interfaces.msg import Time

## 11. C++ Libraries

The following C++ libraries are used in conjunction with ROS2 and PX4:

|  |  |
| --- | --- |
| Library Name | Source/Description |
| <chrono> | Standard C++ library |
| <cstdio>, <cstring> | Standard C++ library |
| <rclcpp/rclcpp.hpp> | ROS2 C++ client library |
| <sensor\_msgs/msg/image.hpp> | ROS2 message type for sensor data |
| Seek Thermal SDK | Includes headers like <seekcamera.h> |

### Libraries to include for C++ Files:

#include <chrono>

#include <cstdio>

#include <cstring>

#include <rclcpp/rclcpp.hpp>

#include <sensor\_msgs/msg/image.hpp>

// Seek Thermal SDK (C includes)

extern "C" {

#include <seekcamera/seekcamera.h>

#include <seekcamera/seekcamera\_manager.h>

#include <seekcamera/seekcamera\_error.h>

#include <seekcamera/seekcamera\_frame.h>

#include <seekcamera/seekcamera\_version.h>

#include <seekframe/seekframe.h>

}