Artifact ID: REQ-004	Artifact Title: Software Bill of Materials	XPRIZE®
Revision: 03	Revision Date: 2025-02-11	
Prepared by: Joshua Crookston	Checked by: Jacob Wilkins	
Purpose: A list identifying all software components required to construct two autonomous drones according to our requirements. It is important to note that daily changes and updates are made apart from the major revisions listed below.		

Revision History			
Revision	Revised by	Checked by	Date
01	Joshua Crookston	Jacob Wilkins	2024-11-18
02	Joshua Crookston	Israel Zenteno	2025-02-10
03	Israel Zenteno	Joshua Crookston	2025-02-11

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
- 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
- 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
- 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
- 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
- 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
- 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>
}
```