Software Overview

Year: 2020 Semester: 1 Team: 4 Project: Sowin Seeds

Creation Date: 09/07/2020 Last Modified: March 3, 2015

Author: Ethan Campbell Email: campb338@purdue.edu

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Software Overview** | 5 | x2 |  |  |
| **Description of Algorithms** | 5 | x2 |  |  |
| **Description of Data Structures** | 4 | x2 |  |  |
| **Program Flowcharts** | 5 | x3 |  |  |
| **State Machine Diagrams** | 5 | x3 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** | 5 | x2 |  |  |
| **Formatting and Citations** | 5 | x1 |  |  |
| **Figures and Graphs** | 5 | x2 |  |  |
| **Technical Writing Style** | 5 | x3 |  |  |
| **Total Score** | 98 | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

*Excellent work. Please check my comments.*

1.0 Software Overview

The pollination drone will be able to autonomously take off, navigate close to apple blossoms, ]identify apple blossoms, and attempt pollination with a pollination appendage. The drone will be able to repeat the pollination process with other flowers and return home when finished. The drone will only be able to operate within a pre-specified geofence. The drone will self monitor for critical errors, safety hazards, and be able to accept manual radio controlled input at any point in time.

The pollination drone will require many software components to all interact. There are three main environments where software will be running. First, the Pixhawk Mini will be running its own software and firmware that we will be able to customize and configure. Most of that can be completed via the tools provided by Pixhawk. The Pixhawk Mini is responsible for maintaining stable flight, and abstracting the flight sensors and computations away from the STM32H745. Second, the STM32H745 will be making the flight decisions for the drone. These decisions will include deciding where to fly, which blossoms to pollinate, and when to return home. These processes can be broken into the firmware interactions of the STM32H745 with components, and with the software analysis of the data from the Pixhawk and the Nano. The last major component is the Jetson Nano. The Jetson Nano will be running very specialized software. Its job is to process the video stream through either a neural network or a color clustering algorithm. Both the neural network and the color clustering algorithm are used to identify potential flowers within the video stream. When the STM32H745 queries the Jetson Nano, the Nano will provide it with data on flowers located nearby.

2.0 Description of Algorithms

STM32H745:

This microcontroller is responsible for making the flight decisions during the autonomous pollination mode. When the mode is enabled, the microcontroller will verify the drone is currently operating within the pre-defined geofence. If the drone is not within the geofence, the STM32H745 will either shut down or fly to the region specified by the geofence. The microcontroller will make the decision based on predicted flight time and remaining battery.

While the drone is in autonomous mode and in the specified geofence, it will enable flower detection in the Jetson Nano. Once the Jetson Nano has flower detection enabled, it will begin processing a video stream through a neural network. The neural network will be trained to detect apple blossoms specifically. If the performance of the neural network is too slow, the STM32H745 will tell the Jetson Nano to switch into color blob detection. The Nano will then process the video stream through a geometric / color blob detector optimized for apple blossom detection. This may be especially useful when the pollination drone is very close to the apple blossoms and high precision is required.

The STM32H745 will be connected to the pollination appendage’s contact sensor. This sensor will indicate if the pollination appendage is currently physically in contact with another physical object, ideally apple blossoms. The STM32H745 will determine if pollination was successful by combining the input from the contact sensor with the camera data from the Jetson Nano. Once the STM32H745 has determined pollination was likely, it will then decide what flower to pollinate next.

The general strategy behind the pollination drone’s path is to touch many flowers on a single tree, and then move on to another tree. Apple blossoms are self-sterile meaning they cannot pollinate within the same tree. For this reason, it is important that the STM32H745 understands how many flowers to pollinate on a particular tree and when to move on to another tree. This will be built into the software as a calibratable setting.

The STM32H745 will return home if any of several conditions occur. If the drone is running low on battery and the projected return home flight duration is approaching the remaining length of the battery’s charge, the drone will return home. If the STM32H745 has determined the drone has adequately pollinated within its geofence, the drone will return home.

Jetson Nano:

The Jetson Nano will be using TensorRT to process a trained Convolutional Neural Network (CNN). We will be training the CNN with a dataset specifically targeting apple blossoms. With the model trained, Nvidia provides the TensorRT framework to optimize running models on the Jetson Nano [2]. If the performance of the neural network is slow or poor in quality, the Nano will be able to switch into a geometrical color-blob detection mode.

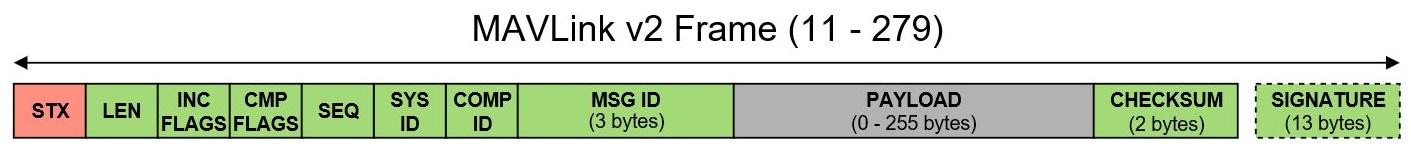
3.0 Description of Data Structures

There are several data structures that are critical to the operation of the pollination drone. The most important is the neural network trained to run on the Jetson Nano. The neural network model on the Jetson Nano will be a Convolutional Neural Network (CNN). The advantage of using a CNN is that feature extraction becomes automated. Extracting features manually for a large apple blossom dataset would be both tedious and sub-optimal. By training CNN on the dataset, the model should be able to detect apple blossoms without human assisted feature classification and extraction. The disadvantage of using this method is a lack of transparency. It is nearly impossible to analyze what the CNN is processing within its convolutional layers.

Images are stored as a video stream piped into OpenCV. OpenCV will then process the stream and we will extract flower data from OpenCV’s processing. That data will be stored in a file on the Jetson Nano that will be available to export to the STM32H745 when the STM32H745 queries the Nano to do so.

Another important data structure is the flower information data. This will be gathered by the STM32H745 via querying the Jetson Nano to send its current flower data and metadata over SPI. The data will consist of at least (x,y,z) coordinates, confidence values, and scores for estimated optimal pollination sources. This data will be stored in a storage buffer that is allocated for approximately 100 flowers.

The Mavlink protocol will be used to communicate between the STM32H745 and the Pixhawk Mini. The Mavlink protocol is a serialization protocol similar to CAN [1].



Source: [1]

Figure 1: The MavLink Message Components

For our purposes, we will be using the C library for Mavlink communication, so we will not have to deal with programming the individual bytes and communication for the protocol itself. However, it is still important to recognize the standards and the limitations of the Mavlink Protocol. The minimum and maximum packet lengths are 12 and 280 bytes respectively. Within the payload section, the protocol does not specify any particular standards. For this reason, it is necessary for both the STM32H745 and the Pixhawk mini to agree on the format of the data in the payload.

4.0 Sources Cited:

[1] “Packet Serialization,” Mavlink, 10-Sep-2020. [Online]. Available: <https://mavlink.io/en/guide/serialization.html>

[2]“Nvidia TensorRT,” Nvidia. [Online]. Available: https://developer.nvidia.com/tensorrt

Appendix 1: Program Flowcharts

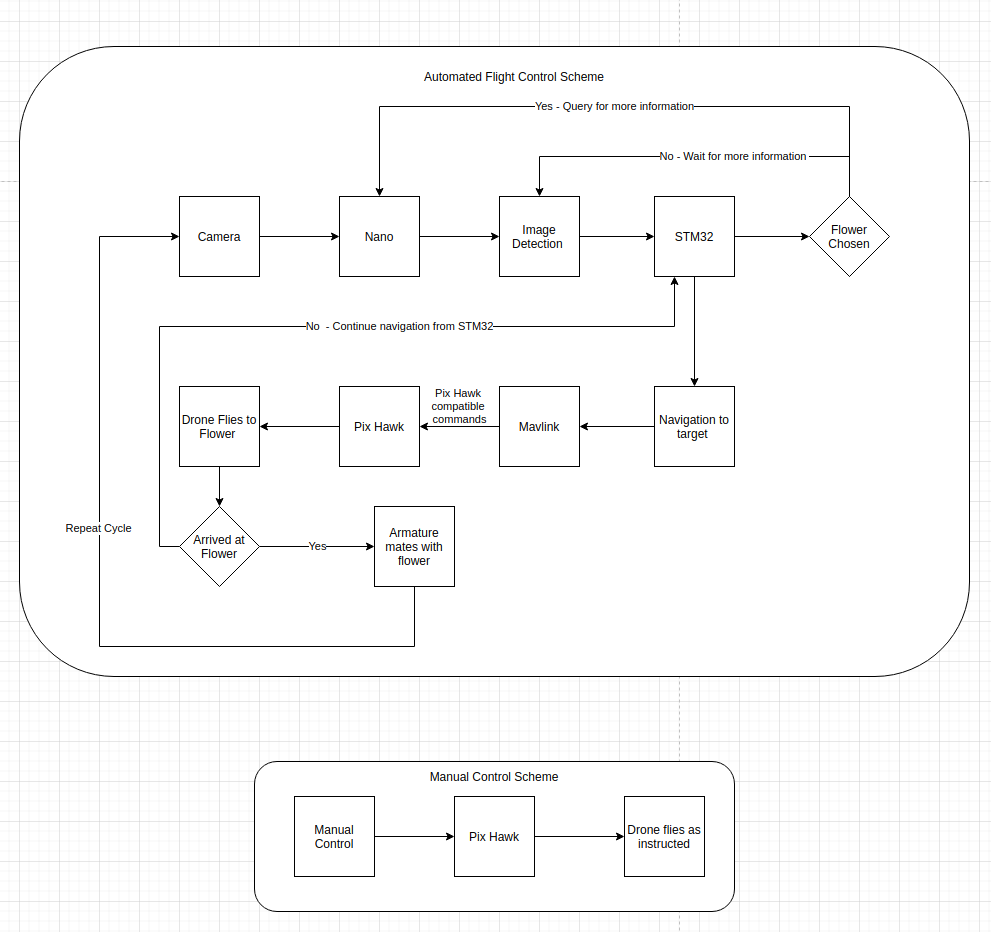


Figure 2: Flight Control Schemes

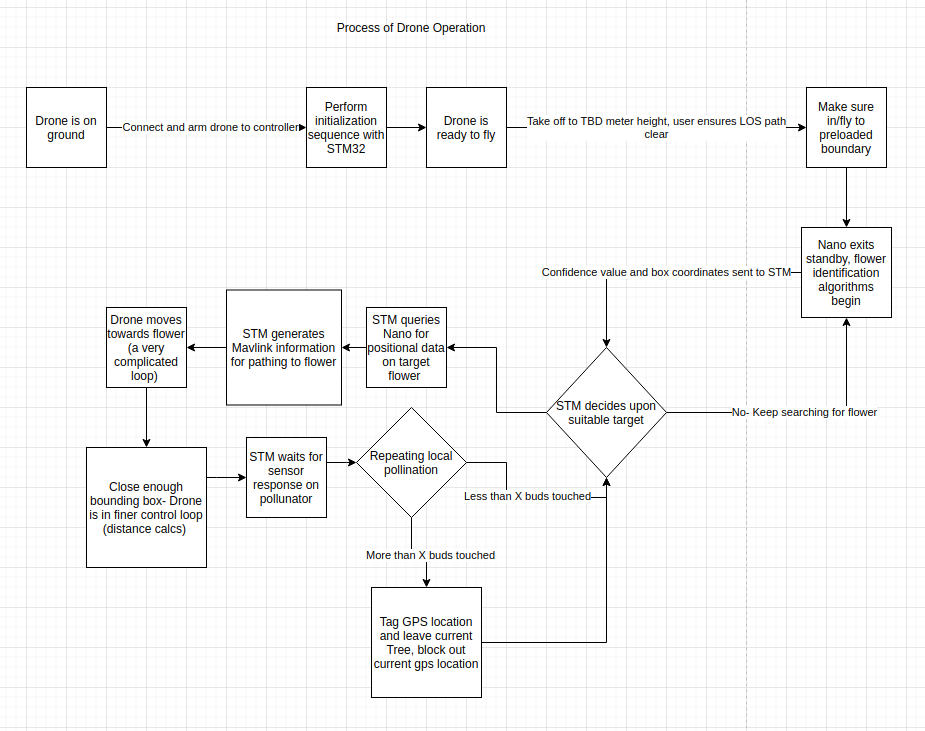


Figure 3: Process of Drone Operation

Appendix 2: State Machine Diagrams

