|  |  |  |  |
| --- | --- | --- | --- |
| Artifact ID:  DJ-002 | Artifact Title:  Control Computer Justification | |  |
| Revision:  1.0 | Revision Date:  18 OCT 2019 | |
| Prepared by:  Garret Gang | | Checked by:  Checker |
| Purpose:  Clearly describe each of the control systems we considered, looking at benefits vs cost. And explain why we decided to use a Raspberry Pi. | | |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision: | Revised by: | Checked by: | Date: |
| 1.0 | Garret Gang | Checker | 18 OCT 2019 |

# References

|  |  |  |
| --- | --- | --- |
| Artifact ID: | Revision: | Title: |
| N/A | N/A | N/A |

# Breakdown

# The control computer we choose is vital to the success of this project. The key factors we had to take into account while choosing the control computer is response time, functionality, and ease of use.

# The target’s current position is only updated every 500 m, an ideal control computer would start the gimbal moving towards the correct pointing vector the instant the drones position is updated. Due to the minimum range the target will be from the communication link positioning system, we don’t need the new position vector to be calculated instantaneously. As long as our system moves the gimbal to the proper facing within 500ms it will keep the target within the antenna’s field of view. But due to our discussion to have the controller act as a server and be configurable thru a web-browser instead of using specialized control software; the computer needs to be fast enough to handle the load of both the gimbal pointing vector updates and the web server.

We also need to guarantee that our computer will not miss a position update because it is busy with the web server.

The current positioning system we are replacing (which uses an Arduino) had a major memory leak in its initial design. While this leak has been fixed, IMSAR has a positioner out on a roof which they have to reboot once a week because upgrading their control software is more difficult than rebooting the system.

Therefore, a control computer that is makes it easy to update our software/drivers remotely is a huge plus in favor of that design.

# Alternatives Considered

1. Arduino

16 MHz, single core

8 kb SRAM

0 Ethernet Ports

4 USB ports

1. Banana Pi

1.5 GHz 2 core processor

1 GB ddr3 Ram

1 Ethernet Port

2 USB ports

dimensions: 92mmx60mm

1. Orange Pi

1.6 GHz 4 core processor

1 GB ddr3

1 Ethernet port

2 USB ports

dimensions: 85mmx55mm

1. Raspberry Pi B+

1.2 GHz 4 core processor.

1 GB ddr2.

1 Ethernet Port

4 USB ports.

dimensions: 85mmx56mm

# Evaluation Methods and Results

# An acceptable control computer would be fast enough to stay ahead of the drone’s position broadcast rate. It would ideally have a built in Ethernet port. It also should fit inside of the IMSAR’s weatherproof case.

Arduino:

We thought about using an Arduino because they are relatively cheap, easy to program and have really good documentation. But the downsides are that they don’t come with a built in ethernet, so we would need to use a USB to ethernet converter, or gpio-pin to ethernet. It would take a lot of extra work to make any type of Arduino work. And Arduino’s are really slow.

It was the worst option by a very large margin.

Banana Pi, Orange Pi, Raspberry Pi

Are all fast enough, have Ethernet Ports etc. In deciding between the three of these it came down to which would be easier to use.

Raspberry Pis have a large active community of users, who are really friendly and welcoming to newcomers. Orange Pi’s and Banana Pi’s have a much smaller less active user base. Which makes it easier to find solutions /pre-built drivers/compatible hardware for Raspberry Pis.

# Other Considerations

# BYU’s EE shop does not carry Orange Pi’s or Banana Pi’s. It does carry Raspberry Pi’s, which made it much faster to get a Raspberry Pi, then it would have been to get an Orange Pi.