**ELEC 4000 Senior Design Status Report – Page 1 of 2**

|  |  |
| --- | --- |
| Project Name: | MailBird: An Autonomous Delivery System |
| Team #, Members: | Team 1, Ben Smith, Hugh Dillon, Hunter Thorington, Rick Holloway, Zac Hawkins |
| Report Date: | January 31, 2014 |
| Project Description: | A landing system that can guide a vehicle using IR LEDs within 1 inch of a target. |
| Cycle (1, or 2): | Cycle 1 |
| Cycle Intent: | Build a working prototype of IR module and use to mimic loiter behavior over LED station |

**TASKS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Planned |  |  | Actual |  |
| Task # | Task Description (Add rows as needed) | Cycle planned for completion | Total planned hours | Planned hours this cycle | Status (% complete) | Actual hours this cycle | Total hours |
| 1 | Team management2 | 2 | 60 | 30 | 8.33% | 5 | 5 |
| 2 | IR land control method | 1 | 120 | 120 | 20% | 50 | 50 |
| 3 | IR camera implementation | 1 | 40 | 40 | 40% | 16 | 16 |
| 4 | Ground Station control method | 1 | 40 | 40 | 30% | 13 | 13 |
| 5 | Landing station | 2 | 20 | 10 | 2% | 1 | 1 |
| 6 | Reports | 2 | 180 | 80 | 3% | 5 | 5 |
| 7 | Marketing display | 2 | 40 | 0 | 0% | 0 | 0 |
| 8 | Integration of components | 1 | 100 | 100 | 0% | 0 | 0 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  | **Planned Total1** | 600 | 420 | **Actual Total** | 90 | 90 |

1Planned Total should equal (# of team members) x (10 hrs. per week) x (Cycle 1 weeks 6) + Cycle 2 weeks (6) = 12 weeks).

2Assumes 5 hours per week for 12 weeks. Should be mainly team leader(s).

**ELEC 4000 Senior Design Status Report – Page 2 of 2**

**TEAM MEMBER HOURS**

**Record # of hours each person spent on each task this week, then total by week, cycle, and project.**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **Task3** |  |  |  |  |  | **Total Hours** |  |
| **Name** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **Week** | **Cycle** | **Project** |
| **Dillon, Hugh** | **5** | **3** | **--** | **12** | **--** | **2** | **--** | **--** | **22** | **22** | **22** |
| **Hawkins, Zac** | **--** | **14** | **--** | **1** | **--** | **--** | **--** | **--** | **15** | **15** | **15** |
| **Holloway, Rick** | **--** | **17** | **--** | **--** | **1** | **2** | **--** | **--** | **20** | **20** | **20** |
| **Smith, Ben** | **--** | **14** | **--** | **--** | **--** | **1** | **--** | **--** | **15** | **15** | **15** |
| **Thorington, Hunter** | **--** | **2** | **16** | **--** | **--** | **--** | **--** | **--** | **18** | **18** | **18** |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **TOTALS** | **5** | **50** | **16** | **13** | **1** | **5** | **0** | **0** | **90** | **90** | **90** |

**Accomplishments since last status report:**

* Custom ArduPilot code can be written and compiles without error, but remains untested. The control structure has been updated to reflect a potential input switch to move the position error source from the GPS module to a custom-built IR module.
* The IR camera that will be used for guidance can now plot 3-dimensional locations. A script interfaces the camera to MATLAB to translate the input data to a real-time plot in 3D.
* The quadcopter communicates with its flight planner using a command protocol called MAVlink. A python library has been installed to send and receive MAVlink commands from a PC/Mac. Current “heartbeat” information is the only thing that has been read from the quadcopter to date.
* An autopilot simulator has been installed on a Linux VM for testing purposes. This will not help substantially in our actual landing testing, but may be of some use for simulating missions and verifying that the vehicle is switching flight modes correctly once the code is in place to do so.

**Obstacles encountered since last status report and actions to deal with same:**

* Ardupilot codebase is difficult to read and develop. Numerous resources have been located to assist in the development process, but the documentation is not comprehensive by any stretch of the word.
* Testing the quadcopter without it breaking something (itself or something around it) is going to be very difficult. Preliminary APM mission planner testing had some unexpected results. A testing procedure will need to be developed in-depth but as the implementation of the landing software is still not finalized, tests will be restricted to MAVlink parameter queries via USB cable.

**Risks facing the project and actions to deal with same:**

* We’re doing a significant code addition so we’re using the best software tools we know to maintain the modified Arducopter codebase. Code checkout and change lists are a good way to mitigate the risk of a lot of wasted time editing code and fixing code that used to work.
* Testing the quadcopter is dangerous, particularly since we’re adjusting the input of the stabilization controller. We’re using as much simulation as we can, and then our test procedures are being designed to take advantage of the MAVlink protocol and mimicking flight to see the values that controller has updated without the spinning blades. When we do fly, we’re going to fly in an open area with a tether to keep the quadcopter from flying away from us.

**Objectives for the next week:**

* Establish a good test procedure using MAVlink commands so that we can test the software we’ve created without breaking our quadcopter. Python will be used for direct debug communication via USB.
* Physically integrate the IR camera into the ArduCopter GPIO pins
* Using the test setup in MAVlink, be able to switch the quadcopter into our custom flight mode and see the results via a MAVlink query.