**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: | February 5, 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 management | 2 | 60 | 30 | 20.00% | 6 | 6 |
| 2 | IR land control method | 1 | 120 | 120 | 47.08% | 56.5 | 56.5 |
| 3 | IR camera implementation | 1 | 40 | 40 | 87.50% | 35 | 35 |
| 4 | Ground Station control method | 1 | 40 | 40 | 65.00% | 26 | 26 |
| 5 | Landing station | 2 | 20 | 10 | 10.00% | 1 | 1 |
| 6 | Reports | 2 | 180 | 80 | 10.63% | 8.5 | 8.5 |
| 7 | Marketing display | 2 | 40 | 0 | 0.00% | 0 | 0 |
| 8 | Integration of components | 1 | 100 | 100 | 60.00% | 60 | 60 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  | **Planned Total1** | 600 | 420 | **Actual Total** | 193 | 193 |

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** | **--** | **--** | **--** | **--** | **--** | **--** | **--** | **10.5** | **10.5** | **42.5** | **42.5** |
| **Hawkins, Zac** | **--** | **--** | **--** | **3** | **--** | **--** | **--** | **3** | **6** | **29** | **29** |
| **Holloway, Rick** | **--** | **--** | **--** | **1** | **--** | **--** | **--** | **13.5** | **14.5** | **44** | **44** |
| **Smith, Ben** | **--** | **--** | **--** | **--** | **--** | **--** | **--** | **9** | **9** | **32.5** | **32.5** |
| **Thorington, Hunter** | **--** | **--** | **--** | **--** | **--** | **--** | **--** | **16** | **16** | **45** | **45** |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **TOTALS** | **0** | **0** | **0** | **4** | **0** | **0** | **0** | **52** | **56** | **193** | **193** |

**Accomplishments since last status report:**

* The IR\_LAND mode can be set remotely with the MAVLink command from a python script, meaning that any computer using a MAVLink-based front end can be used to trigger the IR switch mode once it’s integrated with the camera hardware (see obstacles). This differs from the previous weeks efforts in that we can now set our custom mode.
* Significant amounts of the IR\_LAND hardware system have been integrated; they don’t all currently work, but we have moved from system parts into a system as a whole.
* The user code will run, and most of the existing code from the demonstration has been integrated into the ArduPilot.

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

* The I2C splice is not as easy to work with as it appears. We are running into a large amount of code that we are learning about quickly (the ArduCopter hardware abstraction layer (HAL), I2C scheduling within the chip), but as of right now it doesn’t work. As it stands, there is no documentation whatsoever, so we are testing and editing at our own risk. The plan is to set up a modularized test system to localize the communication problem and identify any further obstacles.
* The ArduPilot arduino is does not used standard Arduino libraries, and the wire communication code integration did not work as expected. Thus, we’ve had to rewrite the I2C code we originally used for communicating with our camera. Rewriting and testing has been the bulk of the work this week, and it has been hard to integrate 5 people into a task that is software related. 2 team members are usually working on that (we’ve been alternating who), and the other 3 are usually not fully occupied by what’s leftover to do. Some testing has been done on our camera algorithm, and enhancements have been designed, etc, but the I2C wire code not integrating has severely bottlenecked our work.

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

* As we said last week, we’re doing a significant code addition. 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.
* We’re attempting a large component integration, meaning that there are a large number of places where we could have problems. We’re designing smaller test environments using the Arduino Unos in the lab, so that we can mimic the Ardupilot setup and test with code we know, working around the software changes that have to be made for the I2C integration. These modular tests are helping us determine what parts of the system are causing problems.
* Designing good tests that actually prove something is difficult. We’re writing down exactly what data we’re taking and how we plan to prove functionality as we add these tests.
* Since we’ve been rather successful up to this point, some of the team morale could drop off. We’ve been taking breaks and switching places to see new perspectives on a problem, but lack of success or visible progress is sort of hard to deal with. We’re hoping that the test scenarios we’re using will boost our confidence that our system still (mostly) works.
* Due to switching places on the I2C integration, there’s potential for miscommunication and having to re-verify functionality for a lot of things at once. We’re mitigating that with documentation, making notes so that we should be able to “assume” certain system modules work, after running through a documented checklist. This should remove some variables and testing time (and boost our spirits).

**Objectives for the next week:**

* Develop a test setup to determine the inner workings of I2C and give feedback on what could be going wrong in the I2C interface.
* Determine source of communication errors between camera module and ArduPilot software. We’re assuming we’ve got the I2C interface configured incorrectly, but that may not ultimately be the case.
* Build appropriate documentation database to record what we did for future debugging.