**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 | 55.00% | 22 | 22 |
| 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 | 100% | 0 | 0 |
| 8 | Integration of components | 1 | 100 | 100 | 8.00% | 8 | 8 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  | **Planned Total1** | 600 | 420 | **Actual Total** | 137 | 137 |

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.**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **Task** |  |  |  |  |  | **Total Hours** |  |
| **Name** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **Week** | **Cycle** | **Project** |
| **Dillon, Hugh** | **1** | **--** | **--** | **7** | **--** | **1** | **--** | **1** | **10** | **32** | **32** |
| **Hawkins, Zac** | **--** | **1** | **3** | **--** | **--** | **--** | **--** | **4** | **8** | **23** | **23** |
| **Holloway, Rick** | **--** | **3.5** | **3** | **2** | **--** | **--** | **--** | **1** | **9.5** | **29.5** | **29.5** |
| **Smith, Ben** | **--** | **2** | **3** | **--** | **--** | **2.5** | **--** | **1** | **8.5** | **23.5** | **23.5** |
| **Thorington, Hunter** | **--** | **--** | **10** | **--** | **--** | **--** | **--** | **1** | **11** | **29** | **29** |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **TOTALS** | **1** | **6.5** | **19** | **9** | **0** | **3.5** | **0** | **8** | **47** | **137** | **137** |

**Accomplishments since last status report:**

* The control system was updated to allow for cases where the mode is set to IR\_LAND but the camera is not picking up the minimum required data. Additional class methods were added to get and set parameters within the roll/pitch control structure from the camera code.
* We now have an IR camera simulated landing scenario. The simulation is only reliable for preliminary testing as the camera is stationary, but it is downward-facing and approximately 9 feet in the air so we can get a basic idea of how our algorithm will behave without the risk of the quadcopter giving us unbounded output.
* IR Camera pixel coordinates have been translated to real position data (in cm) and can be implemented in the Arduino software setup used for the proposal demonstration. This means that we now know exactly how far we are away from the landing station in real distance units. We also now know the limitations of the IR camera so that we can specify the minimum and maximum distance requirements for use.
* MAVLink was used to successfully set the flight mode of the quadcopter from a Python script. A python library called pymavlink was used in the script and numerous commands can now be sent to the quadcopter.
* An effective group spreadsheet was created to streamline status hour reporting. Each member simply records his hours and task and the tables for weekly status reports are automatically generated.

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

* After creating a working MAVlink script to set the mode to an existing mode, we still can’t set the mode to our custom mode. Additional coding may be required to set flight mode and we are not aware of what is preventing us from setting our new mode. Additional code analysis and test procedures will be designed to provide a definitive answer.
* The I2C interface on the ArduPilot board only has one I2C connector. We have ordered some connectors but some input splicing is required and the I2C code must now be analyzed to determine how to add an I2C item in code (a new hardware address must be added to the I2C bus).
* The user code loop that was to be run in the main ArduPilot file does not seem to work initially. This is the planned entry point for the custom IR camera code. More testing will be done to determine if this is usable, and if not then we will need another entry point for our camera code.

**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.
* 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. We’re looking at purchasing a net to provide a closed test space.
* We’ve been operating on different system components, developing independently. Communication must happen frequently between these system components to ensure connecting them goes smoothly.
* Our camera code has been designed on an Arduino Uno using a serial interface to a computer. We’re attempting to migrate that software to use an I2C interface on the ArduPilot itself. If we decide that will not work, then a lot of time spent to create this algorithm will be wasted. We will approach this problem with the risk in mind, looking for good reasons to abort before continuing.

**Objectives for the next week:**

* Begin to merge the custom IR camera code running on an Arduino Uno and the control code running on the ArduPilot. Identify potential obstacles and start to develop plans to address those obstacles. Testing may be delayed by I2C splicing and addressing.
* Assemble I2C splice and implement software to discriminate between inputs using I2C addressing. Build a working knowledge base while developing to determine if an I2C setup is even feasible.
* Effectively switch to our custom mode using pymavlink.