ILLINOIS INSTITUTE OF TECHNOLOGY
SCHOOL OF APPLIED TECHNOLOGY
ITMT 495 - TOPICS IN ITM

PRECISION LANDING DEVICE FOR MAV UNITS USING THE ARDUINO PLATFORM

Fabiana Sandes Soares
Illinois Institute of Technology
3201 S State Street
Box #2668
Chicago, IL - 60616
fsoares@hawk.iit.edu

Igor de Sousa Dantas
Illinois Institute of Technology
3201 S State Street
Box #2287
Chicago, IL - 60616
idesousa@hawk.iit.edu

Pedro Henrique de Brito Souza
Illinois Institute of Technology
3201 S State Street
Box #2218
Chicago, IL - 60616
pdebrito@hawk.iit.edu

ABSTRACT

This paper presents the development of a precision landing device for MAV (micro air vehicle) units. This device should be capable of precisely guiding the MAV to a landing spot placed within a 4 square meters area, using the sound that is emitted by another device placed on the this landing spot. The Arduino platform and several common electronic components, such as microphones and speakers, were used to put the idea into practice, and simulate the operation of the precision landing device.

Keywords

Arduino, precision, landing, UAV.

1. INTRODUCTION

With the advances in robotics and automation technology, as well as the increase on the use of MAV (micro air vehicle) units in the e-commerce industry, companies have been putting even more efforts to adapt their operations to this technology tendency and bring a more efficient, less expensive way to deliver their products to their customers. Although this technique can be of extremely benefit to the companies, as well as to the environment, reducing the toxic substances emission rates and the amounts of residue caused by the use of terrestrial vehicles in their delivery system, there are several issues that need to be solved to provide a reliable service to the customers, such as safety and precision, and thinking about the latter, the idea presented on this paper came to mind. The precision provided by the Global Positioning System (GPS) is not enough precision to guide a MAV to a specific location, like a balcony or a front door, and for this reason a product delivery could be totally compromised if the package ended up in a pool or in the wrong side of a fence. Trying to solve this problem, a *Precision Landing Device* prototype was developed, using a simple mechanism that guides the MAV to the correct spot within the it's final destination area, allowing it to perform the delivery task with more reliability and precision.

2. HARDWARE AND CIRCUIT

For this first phase of the project, as it being a prototype, it is built on top of the Arduino platform. Its circuit is built mainly of four pre-amplified microphones, connected with two controller units (simulated by two Arduino boards: UNO and MEGA), responsible for processing the digital signal coming from the microphones, translating it into movement simulations.

2.1. CONTROLLER UNITS

An Arduino UNO, the first of the two controllers, is the unit responsible for processing the signal coming from the microphones using a set of interrupt mechanisms involving the Arduino's registers and global variables. After processing the signal, the Arduino's digital ports are triggered and this signal is sent to the second controller unit. This processing stage is based on the amplitude of the sound wave that gets to the microphones, matching a pre-established threshold in order to trigger the interrupt; This can be easily redesigned to work with specific frequency ranges, using frequency filters to isolate and discard any noise that can interfere on the device's proper operation.

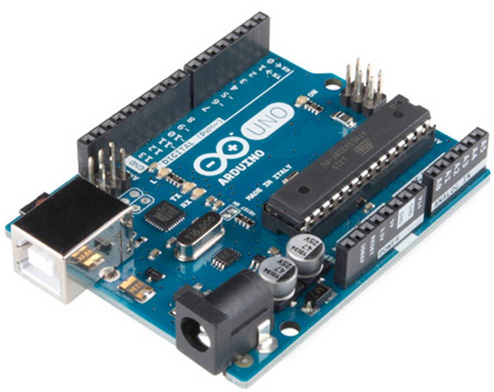


Image 2 – Arduino Nano

The second controller unit, an Arduino Mega board, is used to receive the signal that comes from the first controller unit on its digital ports. This signal is handled by another set of interrupts, which changes timing variables used on the course adjustment calculations.

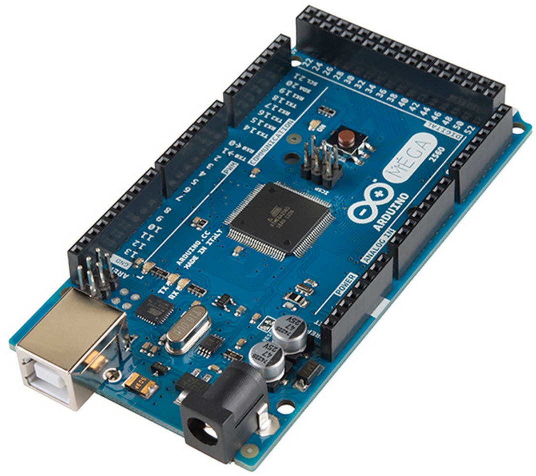


Image 1 - Arduino MEGA

2.2. DIGITAL MICROPHONES

Four digital amplified microphones were used to catch the sonorous signal, sending it to the first controller unit (the Arduino UNO).

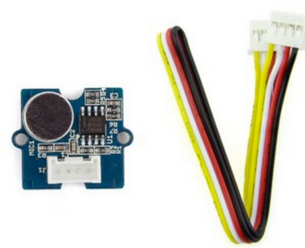


Image 3– Arduino Nano

It is important to notice that working with two Arduino boards became necessary due to several obstacles that were faced in order to achieve the desired results. Some of these obstacles are related to the limitations on the processing power of the Arduino boards, what makes each millisecond taken to processing a set of instructions crucial to obtain acceptable results for this device's proper operation.

3. THE PROTOTYPE

The prototype was built following the general idea of the project; The four microphones are placed on the MAV unit's body while a sonorous signal is emitted. Timing calculations are being processed by the controller units, indicating which direction the MAV should move to based on the results o these calculations. This simulation was run using LEDs to illustrate the MAV's propellers, lighting up two LEDs at every course change, indicating which course the MAV should take.

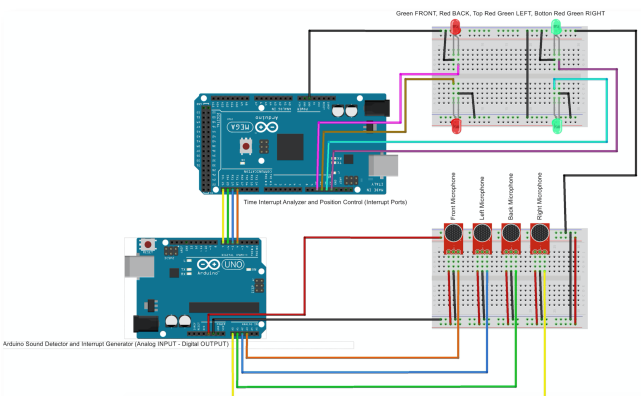


Image 4 - Controlling units schematics

The timing calculations are part of the course adjustment process; they are based on the time that the sound – which travels at 343m/s at 68°F – takes to reach each one of the four microphones, what creates different patterns based on the MAV's positions relative to the sound emitter device. Although the microphones are placed side by side in a breadboard in the illustration above, the tests were reproduced with the microphones placed in a cross-shaped scheme.

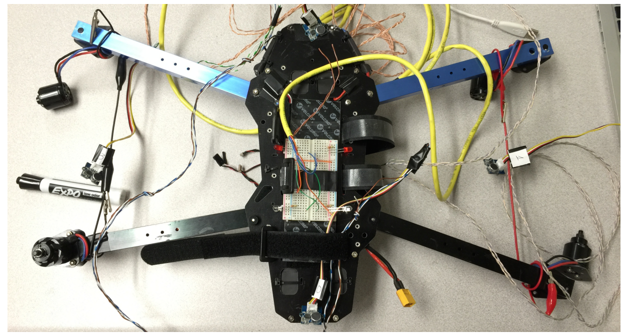
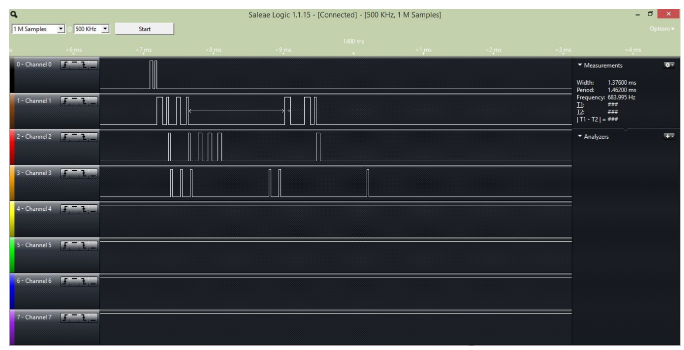


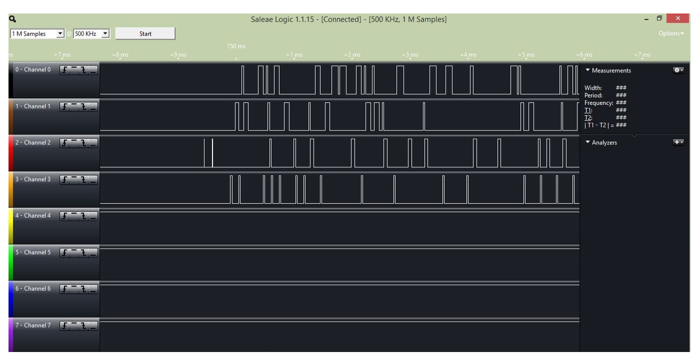
Image 5 - MAV unit's body

The course changes are better illustrated by the following digital outputs that were obtained by using a logic analyzer. Each sample shows the square wave patterns corresponding to the

Move FRONT sample:



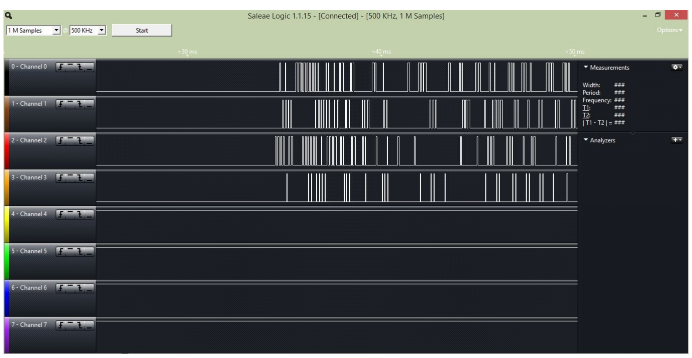
Move BACK sample:



Move LEFT sample:

Move RIGHT sample:

LANDING sample:



As the samples above show, each channel represents a single microphone positioned in the MAV's body, each one representing a direction. The sound wave reaches each microphone on different moments; these timing differences are the key to this prototype's operation. Once all four microphones have caught the sound wave, the controlling units check for which one got it first - or if all of them are in the same timing threshold - and display a LED pattern according to the sound emitter position relative to the MAV unit where the device is placed.

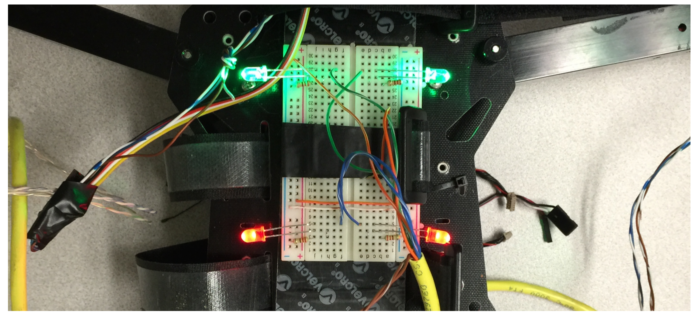


Image 11 - LED simulation

The mechanisms for this method gives us a precision within 2 - 5 centimeters of error margin, relative to the center of the landing point. For the last case, the LANDING sample, different thresholds mechanisms were defined in order to filter the little time differences, what could result in the MAV moving to the wrong direction. It's important to say that this landing method still needs to be refined.

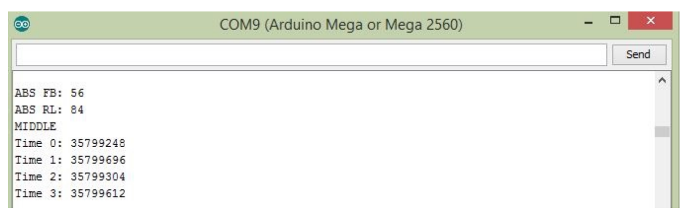


Image 12 - Timing calculations for triggered interrupts

As shown in the example above, Time 0 represents the frontal microphone, while time 2 represents the back microphone. *ABS FB* represents the result of the time difference between these two microphones, which is 56 microseconds. The same rule applies on the *ABS RL* - the difference between times 1 and 3, which represent the left and right microphones. If these two measurements match the minimum threshold, the controller units understand that the MAV unit is in the middle of the landing point, flashing the four LEDs - which represents the landing position.

4. TECHNICAL DIFFICULTIES

Some technical limitations and difficulties appeared during the development process. As mentioned before, the Arduino board's processing power - 16 Mhz - came out to be a considerable obstacle when dealing with timing calculations. In the scope of this project, the time necessary for the Arduino board to perform the operations will directly interfere in the results. As this prototype is designed to work in a real time situation, the slightest delay in providing the MAV unit with the necessary information could result in a considerable inaccurate movement.

This sort of problem is consequence of the high level abstraction functions and instructions that are usually used to build the vast majority of applications using the Arduino platform. The way to solve this problem was to deal with low level code, modifying global variables and changing pin values directly on the registers. Also, the interrupt mechanism, implemented at the closest level to the Arduino processor, played an essential role in reducing the time taken to complete all necessary calculation, making the results appear almost instantly as the system runs its code.

Furthermore, the quality of the components available for use in this experiment was not an obstacle, but a limitation to the final results. The microphones could not reach higher inaudible frequencies, what could give us a better way to isolate noise and the possibility of making a more elegant sound emitter device.

6. FUTURE WORK AND FINAL CONSIDERATIONS

For future work, as this project reached to a simulation-only level, the objectives are already established: use the results to put the idea in practice in the real world. It is entirely viable to build this prototype in a small and energy-efficient circuit board. In a different platform, and using better components, as well as refining the logic responsible for the course adjustment mechanisms, a higher level of precision could be achieved, making a final version even more reliable.

5. CONCLUSION

To conclude, it is important to say that, although this project is not at its final state, it leaves a complete background for any further work with this specific technology. The simplicity of the idea, together with the currently available technology is what gives it even more freedom to be refined - or even redesigned on top of another more efficient platform.

6. ACKNOWLEDGEMENTS

A warm thanks from the team to Prof. Jeremy Hajek, Daniel Tomal Ph.D. and the Department of Information Technology and Management, and our best regards to the mentor of this project, Mark Milhouse.

7. REFERENCES

http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf