

# Integrating MAVLink into LabVIEW



Hyoki Lee, Zach Zens  
Dr. Kim Pierson Ph.D., Dr. Peter Bui Ph.D.

## Project Overview

This project involves the integration of an open-source robotics command and control communication protocol called MAVLink into a simple and widely used graphical programming language known as LabVIEW. MAVLink-based control boards can be incorporated into robotic systems used in research for geographic, geologic, agricultural, search and rescue, and atmospheric scientific investigations. Our success in developing a basic MAVLink program to send commands to the robot allows future students to take advantage of inexpensive MAVLink-enabled robotics systems that can be programmed with LabVIEW.



## About the Robot

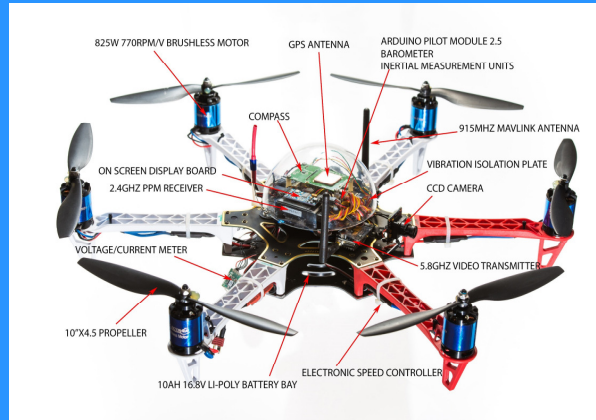
For the construction of our robot, we decided on a hexacopter format (six rotors) for safety and lift capability. Hexacopters can still be controlled even if two motors fail. Each motor draws up to 280W of power, which combined, equals to 1680W (2.25HP). The robot is fitted with a CCD camera for remote piloting up to a mile away, and a HUD for instrument flying assistance. The APM 2.5 flight controller allows for both manual attitude controlled flight through the pulse width modulated transmitter, autonomous flight via its sensors, or autonomous flight with manual guidance. The hardware for the construction of this vehicle is commercially available from various outlet for under \$1000.00. This makes research in this field of aerial robotics very affordable.



Hexacopter construction pictures

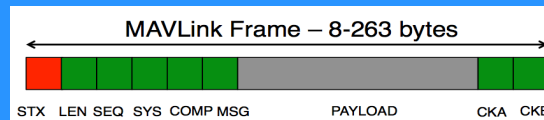
## A little bit about MAVLink

MAVLink is a popular open-source licensed command-and-control communication protocol used for small autonomous vehicles. It is the most widely used UAV/UAS communication protocol in the civilian sector. The ground control system communicates with the aircraft through the transmission of data packets. Communication between the aircraft and ground control is redundant, encoded and tracked via a robust communication strategy that similar to commercial aircraft.



## MAVLink Data Structure

Packets sent between the aircraft and ground control are composed of 8 different parts: a message header, the message length, the sequence number, the vehicle ID, the component ID, the message ID, the message data and the CRC (cyclic redundancy check). The first 6 bytes of the message and the last two ensure that the message is meant for the right system, and the CRC ensures the message is whole and intact.



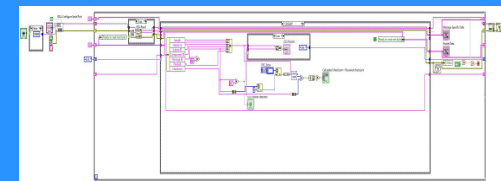
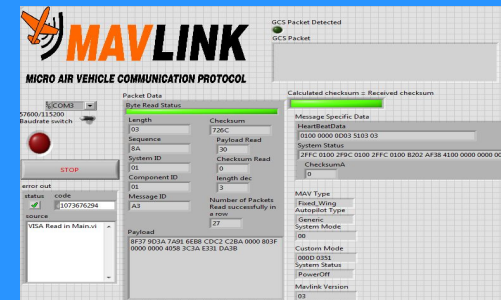
The message header is a static value that acts as a flag indicating that a new message is starting (STX). The message length is a value that depicts the total length of the message being sent (LEN). The sequence number acts as a counter to keep track of message sequence (SEQ). The vehicle ID is the value assigned to each aircraft or ground control station (SYS), whereas the component ID varies based upon which component of the system the message is sent to (COMP). The message ID is a value that dictates the nature of message (MSG). The message data is the actual data/command being sent (PAYLOAD). Finally, the checksum consists of two byte hexadecimal that are calculated through a message ID specific CRC that ensures that there were no transmission errors while communicating (CKA and CKB).

## MAVLink in LabVIEW

MAVLink protocol was reverse engineered and implemented into a LabVIEW program interface for communication with a MAVLink enabled Hexacopter UAS. There were many challenges in integrating MAVLink into LabVIEW due to little or no documentation regarding MAVLink protocol. To get a better understanding of MAVLink protocol, construction of a MAVLink-based module for interception and analysis of data packets and evaluation of MAVLink protocol code written in C and Python was necessary. We received help from various discussion groups within the technical forums such as NI.com and DIYDrones.com.

## Our Program:

We modified our program to produce our own packets and transmit them from LabVIEW to our aircraft. One difficulty that we encountered was the checksum error code. During our analysis of MAVLink packets, we realized that checksum is generated through an algorithm after the rest of the packet is created and the checksum varies based on what the payload message says. We had to create several different algorithms and compare our generated packets to the intercepted packets to ensure that the information was consistent before we were successful in generating a valid packet. For the next step of the project, we plan to reconfigure the program structure to enhance efficiency and eliminate any timing issues with packet interception.



## Progress

At the start of the project we did not understand MAVLink protocol due to its lack of documentation. Persistence paid off and we have made giant step towards the goal of being able to interpret the data and also being able to send data/commands of our own. We have come to a full understanding of the finer details of MAVLink and we have shown that communication between LabVIEW and Arduino robots is possible. Since our packet interception program fully functions we would like to implement more user interface into the program. Additionally, we will be altering the format of the program to take advantage of buffering data to increase the speed and reliability of the program algorithm.

## Conclusion and Future Research

With the help of ORSP and numerous faculty, we are able to demonstrate the ability for LabVIEW to interact with the popular APM flight control boards, which opens doors to create new robotic control programs for Arduino-based boards. Future students will be able to take advantage of our research and develop new software to customize robotics control for a variety of applications and interdisciplinary research projects.

**Acknowledgements:**  
Dr. Kim Pierson, Ph.D. Professor at UW-Eau Claire  
UW-Eau Claire Physics and Astronomy Department  
Dr. Peter Bui, Ph.D. Professor at UW-Eau Claire  
UW-Eau Claire Computer Science Department  
Qgroundcontrol.org, NI.com, DIYDrones.com and 3DRobotics