**Mavlink Tutorial**

Tristan Sherman

2/28/18

**Introduction:**

This tutorial gives the basic instructions for downloading and compiling your own Mavlink code. To do this tutorial you must have Linux (Ubuntu) installed. You will also need an FTDI cable to connect from USB on the computer to Serial 1 on the Pixhawk. If you are running servos you need to make sure they are powered with a battery.

**Setup**  
  
1. Download example code c\_uart\_interface\_example

In your terminal write:  
    git clone <https://github.com/mavlink/c_uart_interface_example.git>  
  
2. Set up computer to downlaod the mavlink files via this website: <https://mavlink.io/en/getting_started/#install>  
  
3. Run the mavgenerate program  
 python -m mavgenerate

4. In the Mavgenerate GUI

i. Select mavlink/ardupilotmega.xml as the input

5. select the folder with your source code in it as output. Append /Mavlink to the path

6. Choose c++11 as output

7. In the folder with the source code, delete the folder named “makefile” and create a new file named “CMakeLists.txt”

8. In the cmake folder, copy the code from the appendix of this tutorial into the document and save.

9. In the terminal run cmake file with: cmake ./

10. In the terminal type: make

11. You should now get four errors for the next four times you type make. For the first three, change the <common/mavlink.h> to “Mavlink/common/mavlink.h” in the three files that need it. For the second error, just comment out the line that says “MAV\_CMD…..” not defined.

12. Run make one last time and it should compile.

**Adding functions**

1. Find needed function from online function list: mavlink.org/messages/common
2. Locate corresponding mavlink header file in mavlink/common
3. Locate the command or message class. The object defined at the end of this class is what you will use as a definition in your function.



1. Locate the encoder function. This will encode the message.



1. Begin a function in the autopilot\_interface.cpp file. A function with basic commands will consist of the following parts.
   1. Writing\_status=true;
   2. Setup of command class
      1. Copy command boxed in step 3 into your function
      2. Directly after, write your desired object name
      3. Use the object you just created, and setup all the variables in the class to the desired numbers. The explanations are in the header file shown in step 3.
   3. Set up a message to be sent
      1. mavlink\_message\_t message;
   4. encode your message using the encode function from step 4
   5. write the message
   6. writing\_status=false;
2. Add function name to header file autopilot\_interface.h
3. You can then use this function in the main code

Useful links

Portal to mavlink: <http://ardupilot.org/dev/docs/mavlink-commands.html>

List of commands: <http://mavlink.org/messages/common>

Step by step mavlink: <https://discuss.ardupilot.org/t/mavlink-step-by-step/9629>

Mavlink common folder: https://github.com/mavlink/c\_library\_v2/tree/master/common