Embedded Systems Development Lab

Projects Ideas

Kyle Mercer & Corrie Russell

# Primary Project

R/F Glove Controller for Hubsan H107 Quadcopter

## Functional Description

In an effort to make a more intuitive user interface with reduced learning times an interactive glove/hand controller will be implemented to interface with a Hubsan X4 (H107L) Quadcopter which implements the standard A7105 RF protocol.

The glove controller will be designed to replace the COTS controller shipped with the Hubsan Quadcopter. A user will wear the glove and be able to easily control the movement of the quadcopter using natural wrist/hand motions and body movement. The new control interface is as follows:

|  |  |
| --- | --- |
| Movement | Quadcopter movement |
| Clenching hand | Throttle control |
| Rotate hand left/right | Yaw control |
| Hand tilt up/down | Pitch control |
| Hand roll (counter)clockwise | Roll control |

## Implementation

The glove controller I/O processing will be driven by an ATmega328 3.3V MCU (8MHz). An MPU-6050 6-axis inertial measurement unit (IMU) will drive motion input into the MCU. Additionally, two buttons will be present on the glove controller. The first will be placed on the top of the glove which commands the devices to calibrate (baseline) the IMU’s steady state input. The second button will be strategically placed on the index finger of the glove and will offer a feature called “scan mode” were the glove controller will transmit zero yaw command to the quadcopter, effectively locking it to the currently visible Y-Plane.

The MCU will be responsible for requesting data from the IMU, listening for user pushbutton events, translating IMU data and button events into SPI controls for the A7105 RF chip to transmit onto the Hubsan Quadcopter. The MCU will be responsible for initializing all peripheral interfaces upon power up, including sending handshake packets to the quadcopter in order to establish a reliable communication link.

The ATmega328 3.3V has a single core 8MHz processor. The majority of this processing bandwidth will be dedicated to basic I2C (MPU-6050) and SPI (A7105) I/O handling as well as the translation code for mapping IMU data into A7105 data packets which the Hubsan Quadcopter can understand. “Off the cuff” calculations revealed that the ATmega328 is incapable of reading raw IMU data from its available 6 degrees of freedom (3-axis accelerometer and 3-axis gyroscope) and subsequently filtering the data through a Kalman filter in order to produce meaningful yaw/pitch/roll values. Implementing Kalman filtering inside the MCU would put significant stress on the timeline of the application for such a processor. As such, this processing will be offloaded on the MPU hardware by activating its built-in Digital Motion Processor feature. This allows the MCU to read processed Quaternion vectors through a FIFO instead of the raw IMU data.

## Resource Requirements

A complete list of parts (including those necessary for development) are as follows:

|  |  |
| --- | --- |
| Part | Quantity |
| Arduino Pro Mini (ATmega328 3.3V 8MHz) | 1-2 |
| FTDI USB OSD Programmer | 1 |
| GY-521 (MPU-6050 breakout board) | 1 |
| Flex Sensor 4.5" | 1 |
| Amphenol FCI Clincher Connector (2 Position, Male) | 1 |
| LiPo battery 3.7V | 1 (at least) |
| Male/Female wires | ~10 |
| A7105 Wireless RF 2.4GHz Transceiver Module | 1-2 |
| Half Breadboard | 1 |

## Development Plan

The following elements listed will be completed serially while sub-elements may be completed concurrently:

1. Module design
   1. A7015 interface design
   2. MPU-6050 interface design
   3. Integration class diagrams
2. Module development
   1. A7105 prototyping: comm link/handshake development
   2. MPU-6050 prototyping: DPM activation, yaw/pitch/roll output
3. Module unit testing
4. Integration
   1. Translation code implementation
5. System testing

## Rough Interface Description

*INSERT BLOCK DIAGRAM HERE*