Self-configuration이 가능한 모듈러 드론의 개발

Development of a Modular Drone with Self-Configuration

Dong Ho Lee, Department of Aerospace Engineering (leedh0124@kaist.ac.kr)

Ho Hyeong Lee, Department of Aerospace Engineering (gms9095@kaist.ac.kr)

Prof. David Hyun Chul Shim, Department of Aerospace Engineering

TA. Han Seob Lee, Department of Aerospace Engineering

Abstract

In this research, we explore the development of a modular drone which is composed of which when assembled together, achieve a coordinated flight. These individual modules are capable of self-configuration that enables the user to assemble them in any fashion. Once the power is on, these modules communicate with one another, determine their module IDs based on their location, and perform vertical takeoff and flight. We also describe the custom hardware of an individual module, which is constructed out of a 3D printer. We also present the algorithm by which a number of modules perform self-configuration as a whole. We then provide other possible modular drone configurations which can be achieved in future research. Experimental results are included, demonstrating indoor flight tests

System Description: Hardware

·Microcontroller

Handling all computation and PWM generation is a single onboard **STM32F4 Discovery board** featuring 32-bit ARM Cortex M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQGP100 package. The primary electrical components of a module needed for flight consists of a 32-bit microcontroller (see Figure 1) connected to a 9-axis motion tracking sensor (MPU9250), motor speed controller (with BEC), and four inter-module communication interfaces via UART protocol. The general functions of major pin outs are summarized in Table 1.

Figure 1. STM32F4 Discovery Board

Pinouts	Pin Function	Use
PA0	TIM5_CH1	TIM5 Output (PWM)
PA2	TIM9_CH1	TIM9 Output (PWM)
PA5	SPI1_SCK	MPU9250
PA6	SPI1_MISO	MPU9250
PA7	SPI1_MOSI	MPU9250
PA15	TIM2_CH1	TIM2 Output (PWM)
PB4	TIM3_CH1	TIM3 Output (PWM)
PB6	USART1_TX	PORT1 TX
PB7	USART1_RX	PORT1 RX
PC6	USART6_TX	PORT4 TX
PC7	USART6_RX	PORT4 RX
PC10	UART4_TX	PORT2 TX
PC11	UART4_RX	PORT2 RX
PC12	UART5_TX	PORT3 TX
PD2	UART5_RX	PORT3 RX
PD12	TIM4_CH1	TIM3 Output (PWM)
PE3	-	MPU9250

Table 1. Pinout functions summary

·Power Supply and Connectors

Figure 2 is a schematic of a connector used to connect between two modules. Each connector is composed of 4 different wires – GND, VCC, RX and TX. The GND and VCC lines are connected in parallel with each module. The GND and VCC pass BEC to supply stable 5V power to the STM32F4 Discovery Board. For UART protocol, RX and TX lines are connected in cross between two modules.

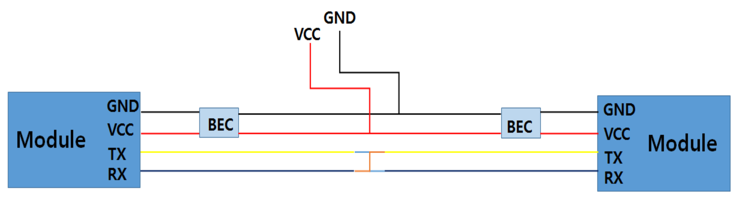


Figure 2. Connector Schematic

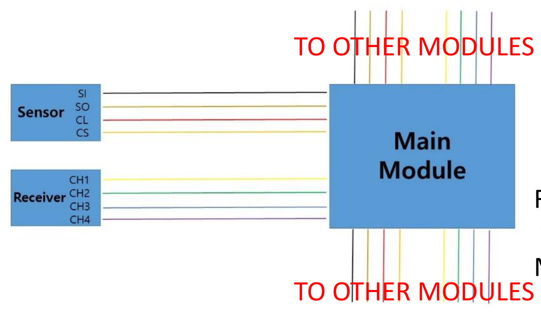


Figure 3. Sensor, Receiver and Main Module

·Frame Design (Final Prototype)

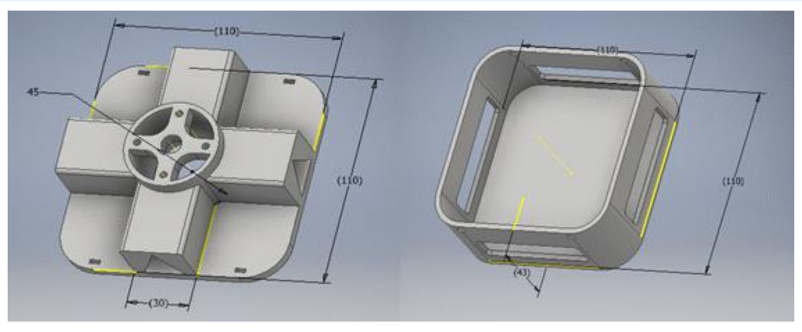


Figure 4. 3D Sketch of Final Prototype (Left: Upper Part for Motor and Rotor, Right: Bottom Part for STM Board and ESC

Figure 5. Final Prototype (with motor and rotor mounted)

General Module Configuration

Each module has four ports starting from its right and going all the way in the **anti-clockwise** direction (Figure 6). Each port is assigned **a specific UART** protocol.

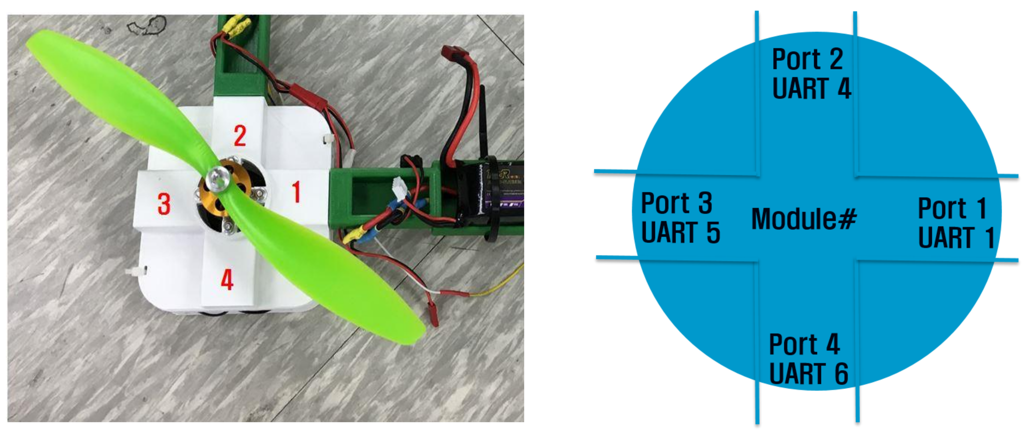


Figure 6. General Module Configuration (Left Actual, Right: Schematic)

·2-by-2 Modular Drone vs ·3-by-2 Modular Drone

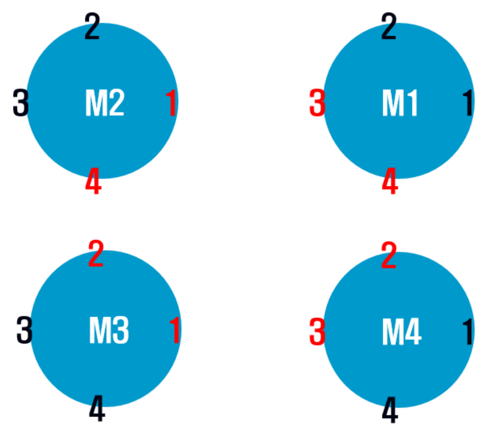


Figure 7. 2-by-2 Modular Drone Configuration

RED indicates successful connection with adjacent module

Motor ID	Connected Ports	Connected Motor
M1	Port 3	M2
	Port 4	M4
M2	Port 1	M1
	Port 4	M3
M3	Port 1	M4
	Port 2	M2
M4	Port 2	M1
	Port 3	M3

Table 2. 2-by-2 Configuration Summary

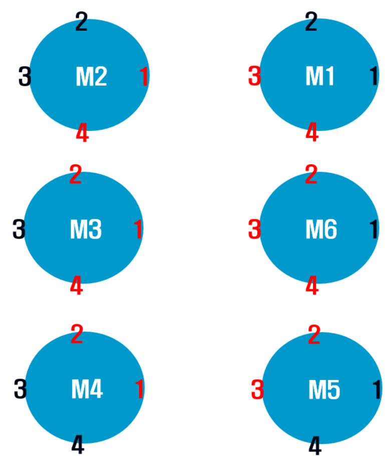


Figure 8. 3-by-2 Modular Drone Configuration

Motor ID	Connected Ports	Connected Motor
M1	Port 3	M2
	Port 4	M6
M2	Port 1	M1
	Port 4	M3
M3	Port 1	M4
	Port 2	M2
	Port 4	M4
M4	Port 2	M3
	Port 1	M5
M5	Port 2	M6
	Port 3	M4
M6	Port 2	M1
	Port 3	M3
	Port 4	M5

Table 3. 3-by-2 Modular Drone Configuration Summary

Self-Configuration Algorithm

A module's connection status can be expressed by the following matrix.

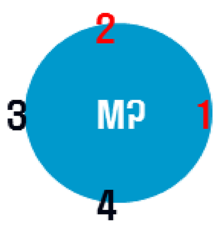
where $M_x = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$

x indicates module ID (1~4),

M_x indicates "module matrix"of #x module, and

a_i indicates connection status at the specified port. (Ex. $a_1 = 1$ indicates successful connection at Port 1 via USART 1)

For example, if a module is stationed as below (red ports are connected ports), the motor matrix is $\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$.



All possible module matrices are stored as module canonical matrix, i.e.

$$M_C = \begin{bmatrix} M_2 & M_1 \\ M_3 & M_4 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{bmatrix}$$

Control Loop Design

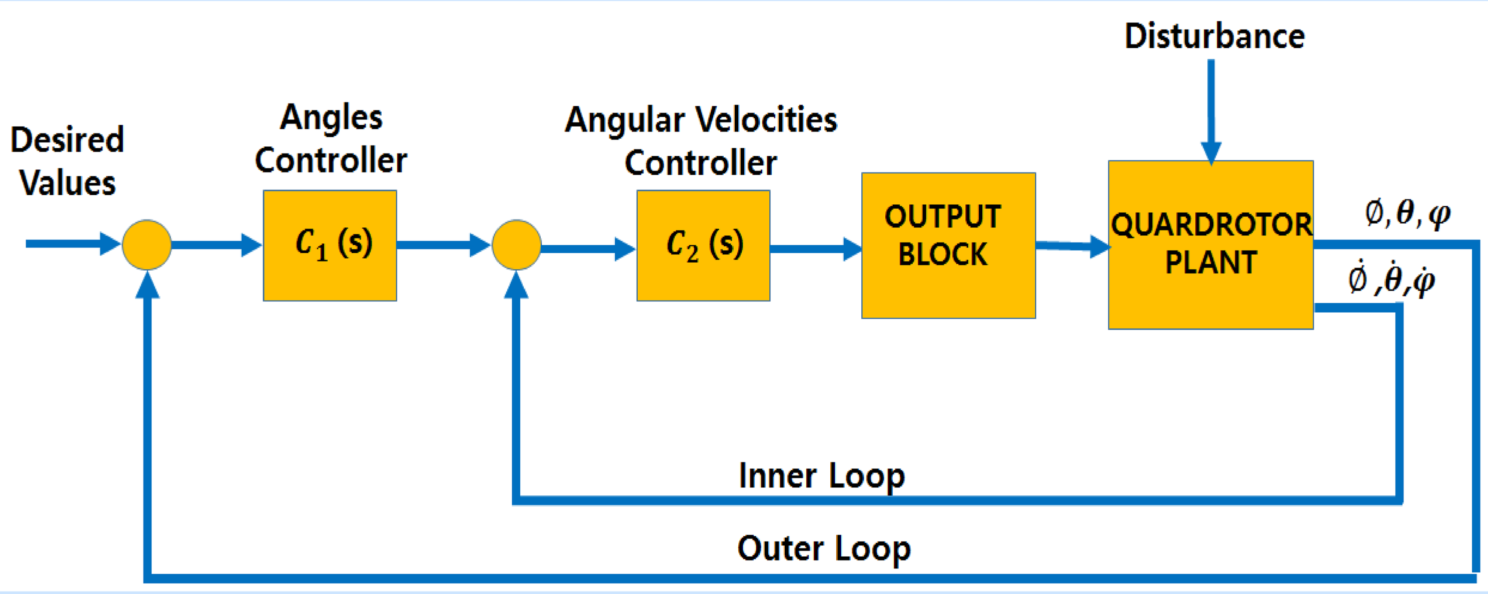


Figure 13. Control Loop Block Diagram

Final Prototype and Flight Test Result

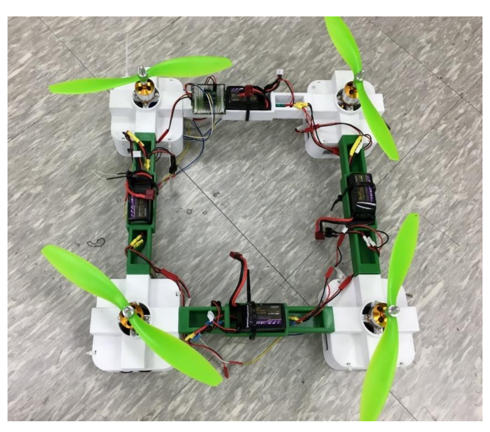


Figure 9. Final Prototype (2-by-2)

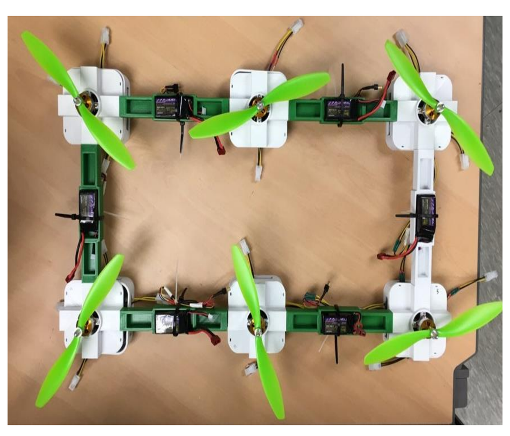


Figure 10. Final Prototype (2-by-3)

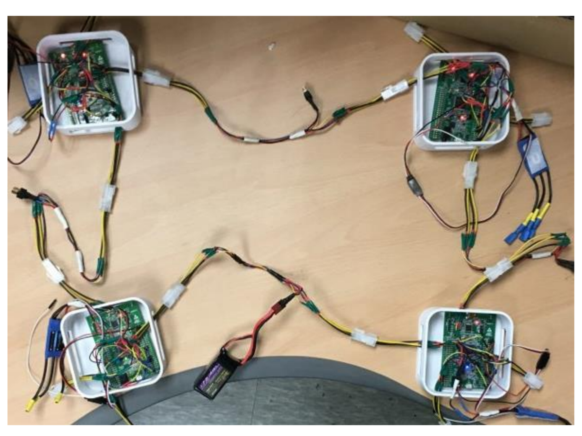


Figure 11. Final Prototype Self-Configuration Test (M1 (red), M2 (Orange), M3 (Green), M4 (Blue))

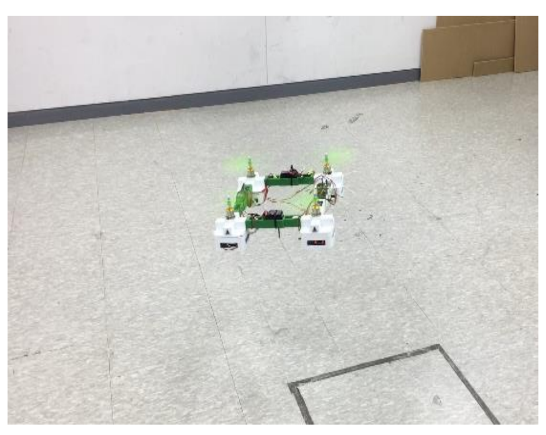


Figure 12. Final Prototype Flight Test

Conclusion and Future Work

In case of a simpliest modular drone, that is a 2-by-2 modular drone, self-configuration process is rather simple and takes fewer time and computational effort. Also, there is a smaller time delay between the user command and actuation. However, when the configuration becomes more complex, for example, a 3-by-2 modular drone, it takes longer time for self-configuration. Also, it was reported that each microcontroller suffers from noise disturbances that hamper self-configuration process. More research needs to be conducted to tackle this increase noise problem for more complex modular drone.