

http://dx.doi.org/10.17703/JCCT.2020.6.1.425

JCCT 2020-2-52

## 소형 드론용 비행 제어기 및 메모리를 위한 고장 감내 시스템 연구

### A Study on Fault Tolerance System for Flight Control Computer and Memory of Small Drones

이정두\*, 조두산\*\*

Jeongdu Lee\*, Doosan Cho\*\*

**요약** 소형무인기(SUAV: small unmanned aerial vehicle) 시장은 기술발전과 함께 다양한 분야에서의 활용이 가능해짐에 따라 급격한 성장세를 보이고 있다. 그러나 소형무인기가 급증하면서 고장, 충돌 및 추락사고 또한 해마다 증가하고 있으며, 이러한 사고를 줄이고 안전성 확보를 위한 기술개발이 활발히 연구되고 있다. 특히 비행 중에 예기치 못한 고장이 발생할 경우 이에 대응키 위한 고장 감내(fault tolerance) 시스템의 적용은 필수적이다. 미 국방부에서 발표한 무인기의 사고원인을 분석한 자료에 의하면 비행제어장치의 오류로 인하여 발생하는 사고가 전체 사고 원인 중에 두 번째로 약 28%의 큰 비중을 차지하고 있다. 본 연구에서는 비행제어장치 고장 감내를 위한 비행제어장치 시스템의 이중구조 설계에 대한 제안을 기술하고 있다.

**주요어** : 소형 무인기, 고장 감내, 비행제어장치, 이중구조

**Abstract** The market for small unmanned aerial vehicles (SUAVs) is growing rapidly as technology advances and makes it possible to use them in various fields. However, due to the rapid increase in small drones, breakdowns, collisions and falls are also increasing year by year, and technologies for reducing accident and securing safety are being actively researched. In particular, the application of a fault tolerance system to cope with unexpected failures during flight is essential. According to data released by the US Department of Defense, accidents caused by errors in flight control computers account for about 28% of all accidents. This paper describes the proposal of flight control computer system's dual structure design to tolerate flight control system failure.

**Key words** : small unmanned aerial vehicle, fault tolerance, flight control computer, dual structure

## 1. 서 론

### 1. 이중 구조

그림 1은 고장 감내 시스템을 구현하기 위해 본 논

문에서 사용 한 이중구조 시스템의 기본 구조이다. 본 논문에서 제안한 이중구조 시스템은 비행제어컴퓨터가 두 개로 구성되며, 한 개의 비행제어컴퓨터는 동작대기 상태에 있다가 고장이 발생될 경우 고장 모듈을 대신하

\*정회원, 국립순천대학교 전자공학과, 연구원 (제1저자)  
\*\*정회원, 국립순천대학교 전기전자공학부 교수 (교신저자)  
접수일: 2019년 10월 18일, 수정완료일: 2019년 11월 02일  
게재확정일: 2019년 11월 12일

Received: October 18, 2019 / Revised: November 02, 2019  
Accepted: November 12, 2019  
\*\*Corresponding Author: mew26@snu.ac.kr  
Dept. of Electronic Engineering, Suncheon National Univ, Korea

여 시스템에 결합되는 동작-대기(active-standby) 방식을 적용하였다 [1]. 그리고 비행제어시스템과 같은 짧은 응답특성을 요구하는 시스템에서는 모듈간의 전환이 발생 할 경우 소비되는 시간을 최소화해야 되기 때문에, 여분의 비행제어컴퓨터는 상시 전원이 인가되어 동작하는 고온 예비 부품(hot standby sparing) 방식을 적용하였다 [2].

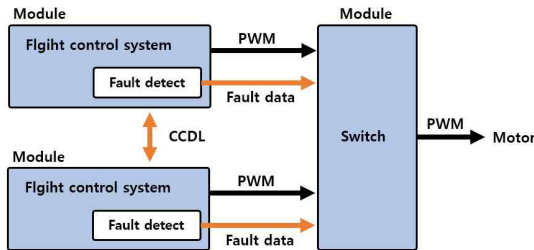


그림 1. 한 개 부품인 예비 부품 시스템  
Figure 1. Standby sparing system with one spare

비행제어컴퓨터를 시스템 관점에서 보면 각각의 동작 장치들은 모듈화 되므로 본 논문에서 제시한 비행제어컴퓨터는 각각 액티브 모듈(active module), 스페어 모듈(spare module), 스위치 모듈(switch module) 등으로 구분하여 기술한다. 여기서 액티브 모듈과 스페어 모듈은 고장을 검출하고, 신호선택기는 스위치 모듈을 사용하여 검출된 결과를 바탕으로 자동 제어되는 구조를 적용하였다. 그리고 고장을 검출하는 기능은 비행제어컴퓨터 내부에 소프트웨어 형태로 구현하고, 실시간으로 고장을 진단한다. 비행제어컴퓨터 상호간에는 교차 채널 데이터 링크(CCDL: cross channel data link)를 통해서 진단 데이터를 공유하고 진단 결과를 스위치로 전송한다. 이런 구조는 액티브 모듈이 동작을 멈춰 진단 결과를 스위치로 전송할 수 없는 상황에서 스위치는 스페어 모듈에서 보내는 진단 결과로 동작한다. 이를 위해 본 논문에서 제안한 이중구조 시스템은 고장의 진단 및 검출과 진단 데이터를 동기화하고 스위치가 자동으로 동작되도록 설계하였다.

## 2. 고장 진단

통합형 구조에서는 롬(ROM : read only memory)과 램(RAM : random access memory)을 포함하고 있는 마이크로컨트롤러(microcontroller)를 사용하고 하나의 인쇄 회로 기판(PCB : printed circuit board)에 센서(sensor)와 전원부가 포함된다. 비행제어컴퓨터에서 고

장 진단은 장치를 추가 구성하지 않을 경우, 마이크로컨트롤러에서 자체 검사로 고장을 진단한다. 그러나 고 사양(high-end) 제품인 경우 하드웨어 방식의 내장 자체 검사(built in test)를 지원하여 고장을 진단하는 회로가 내장되어 있다. 일반적으로 진단기능이 없는 하드웨어 방식의 경우 고장진단용 소프트웨어를 추가해야 하며, 이 경우 저사양(low-end)의 마이크로컨트롤러를 사용하여 진단하는 방법도 연구되고 있다 [3]. 일반적으로 고장 진단에는 부분적으로 발생하는 고장과 완전히 고장이 나서 작동이 중지되는 상태를 분류하여 진단하며, 프로그램 오류로 인해 무한 루프에 빠져 마이크로컨트롤러가 작동이 중지 되는 고장도 함께 고려된다. 그리고 프로세서(processor)의 연산을 진단하는 방법은 산술 연산 장치(ALU : arithmetic logic unit)의 상태 레지스터(status register)를 확인하여 연산 오류를 진단한다. 따라서 코어 내부의 레지스터를 직접 제어해야하므로 진단 코드는 어셈블리어(Assembler)로 작성한다. 램을 진단하는 방법의 경우 메모리에 저장된 내용이 손상되면, 마이크로프로세서는 불가피하게 내부의 모든 메모리 주소를 진단하지 못하는 한계를 가지고 있다. 그리고 롬의 경우 부분적인 손상으로 비행제어컴퓨터의 즉각적인 고장을 발생시키지는 않지만, 비행 제어에 문제가 생기는 원인을 만든다 [4].

## 3. 스위치 분석

픽스호크의 출력 신호는 스위치 모듈을 통해 구동부와 연결되어 모터를 제어한다. 멀티콥터의 구동부는 변속기(ESC: electronic speed controller)로 구성되어있고, 제어 신호는 PWM(pulse width modulation)을 사용한다. PWM 신호의 간격은 2.5msec이고, 제어 신호는 2.5msec중에서 0.9msec와 2msec사이의 펄스를 이용하여 구동부를 제어한다. 이와 같이 변속기 제조사에서는 일반적으로 유효한 제어신호 범위를 제시하며, 범위를 벗어나는 제어신호는 구동부의 꺼짐이나 오동작을 발생한다. 구동부의 제어 신호를 고려하고 고장 감내 시스템을 구현하기 위해 사용한 스위치 모듈의 기본 구조는 그림 2와 같다. 본 논문에서 제안한 스위치 모듈은 두 개의 스위치와 스위치를 제어하는 프로세서로 구성되고, 비행제어컴퓨터에서 보내는 고장 신호를 기반으로 스위치를 제어하여 출력 신호를 선택한다 [5]. 스위치를 제어하는 프로세서에서는 입력 PWM 신호를 모니터링하

고 신호의 동작 상태를 확인하여 제어 주기를 비교한다 [6]. 만약 출력 신호를 선택 할 때 비행제어컴퓨터 상호간의 제어 주기가 다른 경우, 두 개의 스위치를 적절하게 제어하여 출력되는 신호가 변형되지 않도록 한다.

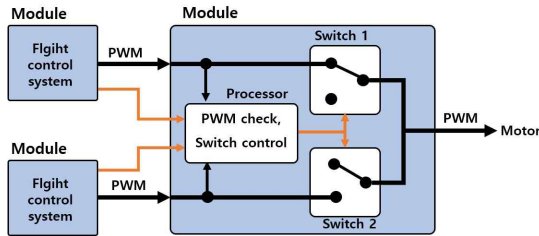


그림 2. 이중 스위치인 PWM 선택기  
 Figure 2. PWM selector with dual switch

## II. 이중 구조 시스템

### 1. 이중 구조 시스템 설계 및 구현

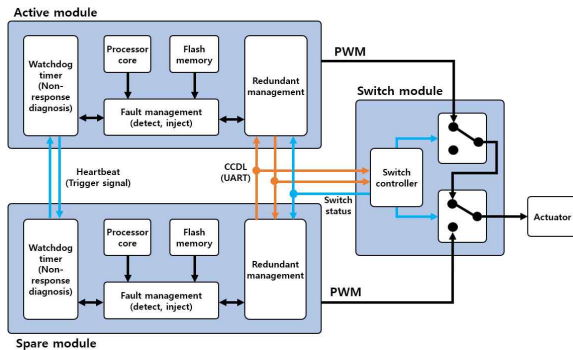


그림 3. 제안된 이중 중복구조 시스템  
 Figure 3. The proposed dual redundancy structure system

본 논문에서 제안한 이중구조 시스템은 그림 3과 같다. 액티브 모듈과 스페어 모듈은 스위치 모듈과 연결되며, 액티브 모듈과 스페어 모듈 내부에는 고장 관리자와 다중구조 관리자가 프로그램 형태로 구현되어 있다. 고장 관리자는 고장을 주입하고 진단 및 검출하는 기능을 하며, 다중구조 관리자는 스위치 모듈을 제어하고 비행제어컴퓨터 상호간의 고장 진단 데이터를 동기화 한다. 고장상태를 모니터링하기 위해 내부에서는 자체적으로 고장을 진단하고 외부에서는 위치독 타이머(WDT : watchdog timer)를 이용하여 무응답을 진단하는 방법을 설계에 반영하였다. 이런 설계를 통해 고장에 대응한 실시간 고장 진단과 고장 분리 및 다중구조 관리가 가능하도록 하였다.

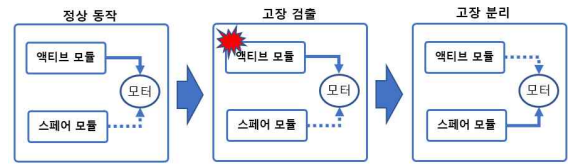


그림 4. 이중 중복 구조 시스템의 동작 절차  
 Figure 4. Operation process of dual redundancy structure system

그림 4는 본 논문에서 설계한 이중구조 시스템의 동작 과정을 나타내고 있으며 고장 진단 및 검출, 분리과정을 수행한다. 그림 5는 실제 구현된 이중구조 시스템이다. 스위치 모듈과 픽스호크 2개를 결합하는 구조이며, 스위치 모듈은 스위치 회로와 스위치를 제어하는 마이크로컨트롤러를 하나의 인쇄 회로 기판에 설계하였다.

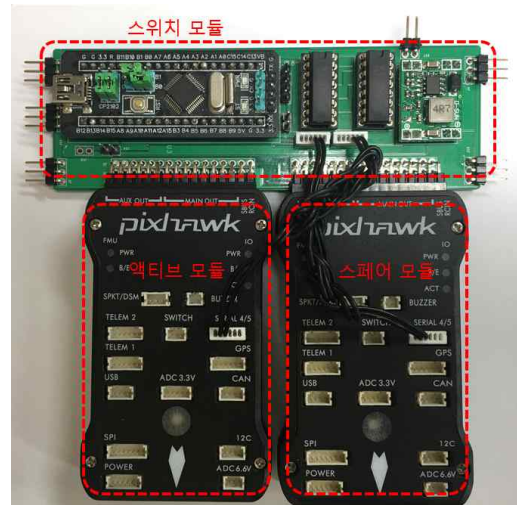


그림 5. 제안된 이중 중복 구조 제품  
 Figure 5. The proposed product of dual redundancy structure

### 2. 이중 구조 시스템 동작 원리

이중구조 시스템은 CCDL를 이용하여 동작되도록 설계하였다. CCDL은 모듈 간에 정보를 전달하는 기능을 수행하며, 데이터 통신은 그림 6와 같이 3개로 구분되어진다. 비행제어컴퓨터의 상호 간에 통신에서는 양방향으로 데이터를 송수신하며 고장 주입 및 실시간 진단 데이터를 동기화 한다. 반면 스위치 모듈은 진단 데이터를 수신하여 스위치를 제어하고 스위치 제어 상태를 다용도 입출력(GPIO : general purpose input output)을 통해서 비행제어컴퓨터로 보낸다.

또한 비행제어시스템 상호간에 구성된 CCDL은 UART(universal asynchronous receiver transmitter)

통신을 이용하여 115200Bps의 전송속도를 가지며, 100Hz 주기로 데이터를 전송한다.

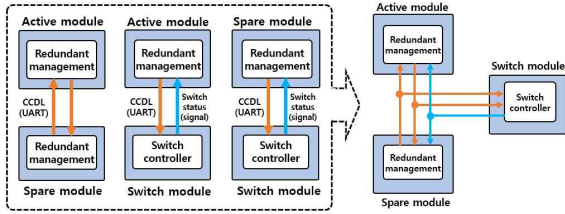


그림 6. CCDL을 이용한 인터페이스 구조  
Figure 6. Interface structure using CCDL

통신 프로토콜(communications protocol)은 비트 방식의 HDLC(high level data link control)을 사용하고, 프레임(frame)을 8Byte 크기로 고정하였다.

### III. 고장 진단

본 연구에서는 고장을 모사하고 동작 상태를 모니터링 하여 고장을 진단하는 방식을 적용하였다. 고장을 진단하는 부분은 크게 내장 자체 검사를 통한 내부 진단과 마이크로컨트롤러의 동작 이상으로 비행제어컴퓨터가 응답이 없는 상태를 진단하는 외부 진단으로 구분한다. 고장을 진단하는 대상은 마이크로컨트롤러 내부 부품으로 산술논리장치와 롬이다. 연산처리를 담당하는 산술논리장치와 펌웨어(firmware)가 저장된 롬은 하드웨어 부품으로서 파손이 되면 수리가 불가능 하다. 따라서 고장 유무를 진단키 위해 본 논문에서는 그림 7와 같이 고장을 진단하는 소프트웨어에 오류를 발생시키는 데이터를 주입하여 고장을 모사한다.

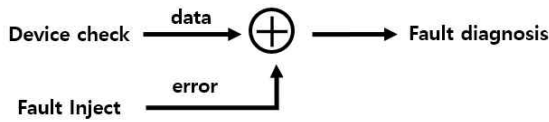


그림 7. 진단을 위한 고장 주입 방법  
Figure 7. Fault injection method for diagnosis

그리고 산술 연산 장치의 연산오류는 상태 레지스터를 확인하여 진단한다. 픽스호크의 펌웨어는 C언어를 기반으로 되어 있어서, 인라인 어셈블리어(inline assembler)를 사용하여 진단 프로그램을 작성하였다 [7]. 설계한 알고리즘은 그림 8과 같다. 진단 프로그램에서는 ALU 이상을 감지하면 'alu\_error\_flag'는 '1'로 설

정되어 진단 결과를 알린다. 진단 과정은 먼저 코어 레지스터에 데이터가 정상적으로 저장되어 있는지 확인한다. 그러기 위해 코어 레지스터에 0x5555를 저장 후 같은 값을 뺄셈 연산 하여 상태 레지스터를 확인한다. 만약 상태 레지스터의 데이터가 정의된 값과 같지 않을 경우에는 고장이 발생한 걸로 판단한다.

롬의 고장 진단 방법은 펌웨어의 모든 데이터의 체크섬을 계산한다. 그리고 데이터가 유효한지를 확인하고 싶을 때는 체크섬을 다시 계산하고 그 결과를 먼저 계산한 결과와 비교하여 오류가 없는지 진단한다. 그림 9는 진단 알고리즘이며 오류를 감지하면 'rom\_error\_flag'는 '1'로 설정되어 진단 결과를 알린다. 미리 연산된 체크섬 코드는 외부 비휘발성 메모리에 보관하고 필요할 경우 불러와서 사용한다. 롬의 저장용량은 1024Kbyte로 바이트(Byte) 단위로 저장되며, 메모리 주소는 최대 0x100000이다. 일반적으로 마이크로컨트롤러는 메모리 맵(memory map) 입출력 방식을 사용한다. 입출력 장치를 접근(access)할 때, 입출력과 메모리의 주소 공간을 따로 분리하지 않고 하나의 메모리 공간에 배치하는 방식이기 때문에 실제 메모리에 접속하는 주소는 다음과 같다. 주소는 0x80000000부터 0x80100000까지이며, 펌웨어의 시작 주소는 0x80004000이다. 체크섬은 순환 중복 검사(CRC : cyclic redundancy check)를 사용한다. 체크섬은 연산 알고리즘에 따라서 오류 검출율과 연산속도가 다르기 때문에 [8], CRC-8과 CRC-16 및 CRC-32를 사용하여 성능을 비교한다.

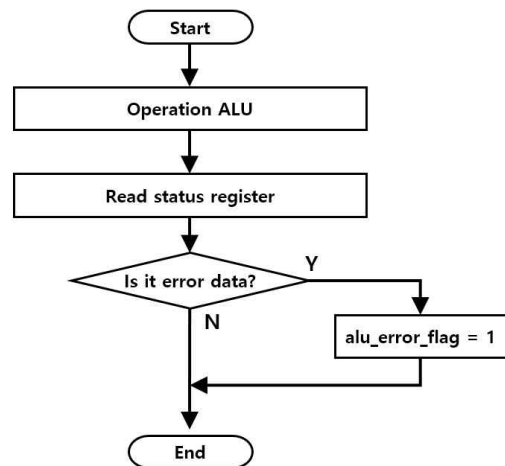


그림 8. ALU 고장 진단 알고리즘  
Figure 8. ALU fault diagnosis algorithm

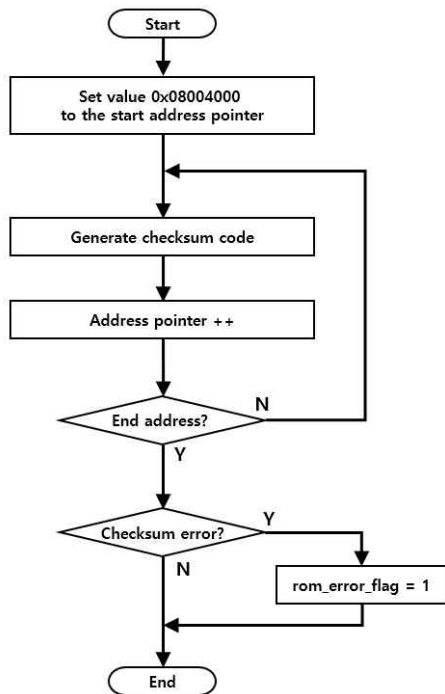


그림 9. ROM 고장 진단 알고리즘  
 Figure 9. ROM fault diagnosis algorithm

무응답의 진단 방법은 그림 10과 같으며, 워치독 타이머를 이용하였다. 마이크로컨트롤러에는 프로그램 오류로 인해 무한 루프에 빠져 있을 경우를 해결하기 위한 워치독 타이머 기능이 있으며, 워치독 타이머를 관리하는 프로그램 함수가 일정한 주기로 동작을 하고 있다. 비행제어컴퓨터의 상호간에는 다용도 입출력으로 연결되어 있으며, 1000Hz 주기로 트리거 신호를 전송한다. 이 신호는 워치독 타이머 관리 프로그램의 정상동작 여부를 판단할 수 있는 신호로서 10msec 이상의 고착상태(stuck at fault)를 확인하여 고장을 진단한다. 오류를 감지하면 'wdt\_error\_flag'는 '1'로 설정되어 진단 결과를 알린다.

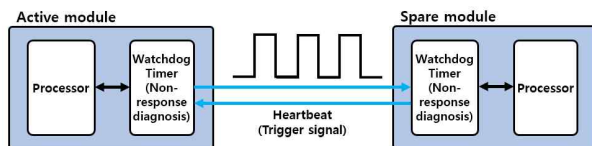


그림 10. 무응답 진단 방법  
 Figure 10. Non-response diagnostic method

#### IV. 실험 및 결과

본 논문에서 구현된 이중구조 시스템의 실험 환경은 그림 11과 같다. 실험을 통해서 고장 진단 성능과 스위치 모듈의 동작 성능을 확인한다.

픽스호크는 외부접속을 위한 터미널 프로그램인 NSH(NuttShell)을 지원한다. 컴퓨터에서 범용 비동기화 송수신기를 통해 NSH에 접속하였고, 제작된 모니터링 함수로 고장 주입 및 진단과 스위치 동작 실험을 진행하였다. 모니터링 함수의 동작 상태는 컴퓨터의 모니터를 통해 확인하며, 신호 분석은 Seead Studio 사의 LA2016 로직 분석기(logic analyzer)를 사용하였다.

##### 1. 고장 진단 실험

산술 논리 장치의 진단 프로그램은 모니터링 프로그램 안에 함수로 구현하였으며, 진단 함수를 실행하는 명령어는 다음과 같다. 일반적인 고장 진단은 'monitoring test alu'이고 고장 주입 후 진단은 'monitoring test fault\_alu'이다. 진단 결과로 연산 데이터와 상태플러그 결과 값을 확인해볼 수 있었으며, 고장을 검출하면 'alu\_error\_flag'는 '1'을 표시한다.



그림 11. 실험 환경  
 Figure 11. Experiment environment

픽스호크는 비행 모드(Armed)와 대기 모드(Disarmed)와 같이 두 동작 모드를 가지며, 동작 모드에 따라서 진단 시간을 비교하기 위해 동작 모드를 분리해서 실험을 진행하고, 1000회의 반복실험으로 데이터를 수집한다. 실험 데이터는 진단 시간을 기준으로 최대와 최소 및 평균으로 정리하였고, 실험 결과는 그림 12과 같다. 비행 모드에서의 진단 시간은 평균 6.42μsec이며, 대기 모드에서의 진단 시간은 평균 6.8μsec이다.

롬 진단 프로그램은 모니터링 프로그램 안에 함수로 구현하였으며, 일반적인 고장 진단은 'monitoring test rom', 고장 주입 후 진단은 'monitoring test fault\_rom' 명령어를 각각 입력하여 테스트를 진행한다. 테스트 결과는 저장된 체크섬과 진단 결과로 확인된 체크섬의 데이터를 나타낸다. 고장을 검출하면 'rom\_error\_flag'는 '1'을 표기한다.

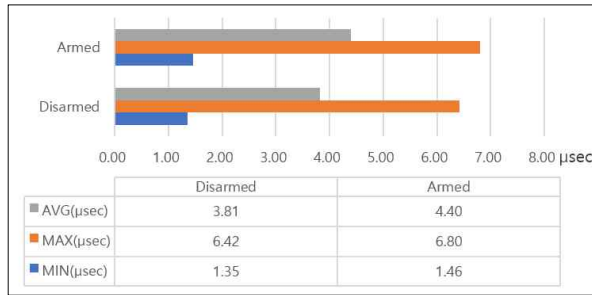


그림 12. 비행 상태에 따른 ALU 진단 시간 비교  
Figure 12. Comparison of ALU diagnostic time according to flight status

실험 결과는 그림 13과 같다. 비행모드에서의 평균 진단 시간은 CRC-8의 경우 294.6msec이며, CRC-16의 경우 357.74msec이다. 그리고 CRC-32는 525.94msec이다. 비행 모드에서의 평균 진단 시간은 대기 모드에서의 평균 진단 시간에 비해 1.08% ~ 1.47%가 증가하였으며, 이는 ALU 고장진단과 마찬가지로 비행 모드에서 프로세서의 연산 처리량이 증가함을 나타낸다. 그리고 진단 프로그램에서는 램에서 데이터만 읽어오는 경우 142.13msec의 처리 시간을 가지며, 전체 고장 진단 시간에서 큰 부분을 차지한다. 고장 진단 시간을 개선시키기 위해서 롬에서 읽는 데이터 크기를 1Byte에서 8Byte로 변경하였고, 실험 결과는 그림 13과 같다. 결과적으로 롬에서 1024Kbyte의 데이터를 불러오는 시간은 26.82msec이며, 기존 프로그램에 비해 5.7배의 빠른 속도를 나타낸다. 그리고 진단 프로그램에 적용한 결과 평균 진단 시간은 CRC-8의 경우 182.91msec이며, CRC-16의 경우 272.52msec이다. 그리고 CRC-32는 380.72msec로 속도가 향상됨을 나타내었다.

무응답 진단 프로그램은 위치독 타이머 프로그램 안에 함수로 구현된다. 무응답의 고장을 주입 방법에는 비행제어컴퓨터를 멈추게 하거나 전원을 차단하는 방법이 있다. 실험에서는 NSH에서 지원하는 재부팅 명령어를

사용하여 무응답의 고장 상태를 모사하였고, 실험 결과는 그림 14와 같다. 액티브 모듈에서는 1000Hz 주기로 트리거 신호를 전송하며, 강제 재부팅(reboot)을 통해 트리거 신호는 고착 상태가 된다. 그리고 스페어 모듈에서는 이상 신호를 확인하고 고착상태가 10msec 이상을 유지하면 고장으로 진단한다.

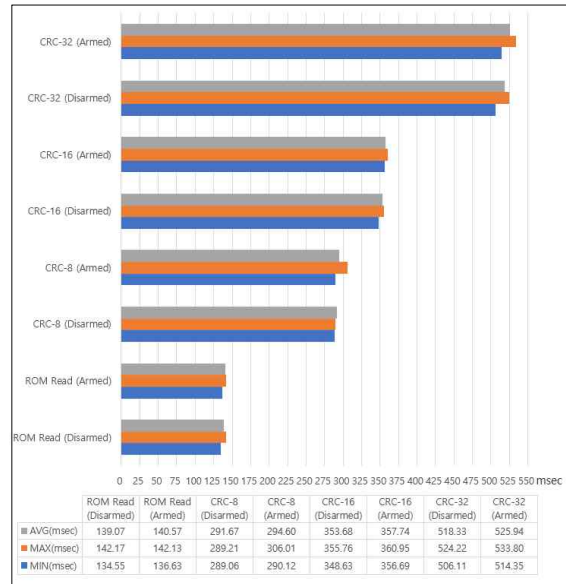


그림 13. 비행 상태에 따른 ROM 진단 시간 비교  
Figure 13. Comparison of ROM diagnostic time according to flight status

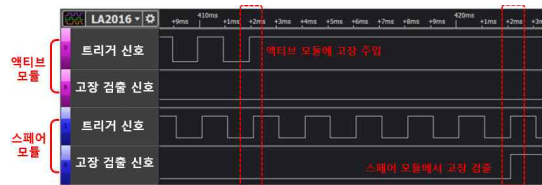


그림 14. 무응답 진단 결과  
Figure 14. Non-response diagnostic results

## 2. 스위치 모듈 실험

그림 15은 제어 신호의 시간 차이를 고려한 스위치 동작 결과이다. 액티브 모듈과 스페어 모듈에서 출력되는 PWM은 스위치 모듈에 입력되어진다. 0번은 액티브 모듈에서 출력되는 PWM 신호이고, 1번은 스페어 모듈에서 출력되는 PWM 신호이다. 그리고 2번은 스위치의 동작에 의해 출력되는 PWM 신호이다. 스위치 모듈은 스위치가 동작을 하지 않을 경우 액티브 모듈의 PWM을 출력하고, 스위치가 동작하게 되면 액티브 모듈에서 스페어 모듈로 전환하여 PWM을 출력한다. 모듈이 전환되는 과정에서 스위치 모듈은 제어 신호 주

기를 모니터링하고, 신호의 변형이 생기지 않도록 액티브 모듈 PWM과 스페어 모듈 PWM 사이에는 제로 전압을 출력한다.



그림 15. 스위치 제어 실험 결과  
 Figure 15. Switch control experiment results

### 3. 이중구조 관리

이중구조 시스템에서 다중구조 관리자는 상호 모듈 간에 100Hz의 주기로 진단 데이터를 공유하고 진단 결과를 스위치로 전송하며, 스위치에서는 모듈에서 보내는 진단 결과를 기반으로 동작한다. 그림 16은 액티브 모듈에 산술 논리 장치에 고장을 주입하여 실험한 결과를 나타내고 있다. 0번은 액티브 모듈의 통신 데이터이고, 1번 신호는 스페어 모듈에서의 통신 데이터이다. 그리고 2번은 스위치의 동작 상태이며, 3번은 액티브 모듈에 고장 주입하여 고장을 검출한 신호이다. 이중구조 시스템에서는 다중구조 관리자를 통해 진단 데이터를 주기적으로 스위치 모듈에 송신하고 있다. 그림 16과 같이 액티브 모듈에서 고장이 검출되면 스위치 모듈에서는 진단 데이터를 기반으로 스위치를 제어한다.

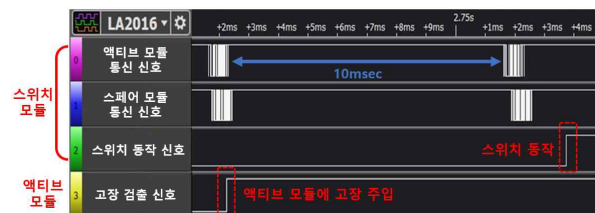


그림 16. ALU 고장 주입 이후 스위칭 동작 상태  
 Figure 16. Switch operating status after ALU fault injection

## V. 결론

본 연구에서는 소형 드론의 비행 안전성을 향상시키기 위해 여분의 비행제어컴퓨터를 추가하는 이중구조의 고장 감내 시스템을 설계 및 제작하였고, 쿼드콥터에 장착하여 비행 실험으로 시스템의 안정성을 확인하였다. 본 논문에서 제안한 이중구조 시스템은 비행제어

컴퓨터가 두 개로 구성되며, 한 개의 비행제어컴퓨터는 동작대기 상태에 있다가 고장이 발생할 경우 고장 모듈을 대신하여 시스템에 결합되는 동작-대기 방식을 적용하였다. 제안된 기술을 [9]와 같은 산림 관리용 드론에 적용할 경우 고장으로 인한 비용 손실을 절감할 수 있을 것으로 예상된다.

## References

- [1] Jun-Hyun Park, "Design and Implementation of Redundant Digital Flight Control Computer", Master Thesis, Chungnam National University, 2011.
- [2] MIL-HDBK-338B, "Electronic Reliability Design Handbook", US Department of Defense, 1998.
- [3] Ad van de Goor, Georgi Gaydadjiev, Said Hamdioui, "Memory Testing with a RISC Microcontroller", 2010 Design Automation and Test in Europe Conference, IEEE, pp. 361-366, 2010.  
DOI: 10.1109/DATE.2010.5457210
- [4] BARR, Michael, "Programming Embedded Systems in C and C++", O'Reilly Media, 1999.
- [5] Elmore Joel D, "Design of an All-In-One Embedded Flight Control System", Ph.D. Thesis, Virginia Commonwealth University, 2015.
- [6] Dikshant Khanke, Dr. P. B. Patil, "Design and Implementation High Reliability PWM Modulator Using Triple Modular Redundancy with spare Arrangement", International Journal of Science and Research, Vol. 3, No. 7, 2014.
- [7] ARM architecture, "Cortex-m4 Devices Generic User Guide", 2011.
- [8] Ali Grami, "Introduction to Digital Communications", Elsevier, 2015.
- [9] Oh, Sun Jin, "Database Design for Management of Forest Resources using a Drone", The Journal of the Convergence on Culture Technology (JCCT), Vol.5 No.3, pp.251-256, 2019.  
DOI: 10.17703/JCCT.2019.5.3.251

※ This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF - 2018R1D1A1 B07050054).