|  |  |  |  |
| --- | --- | --- | --- |
| 18.02.22 | 전자부 활동 개관 | 전자부:이재린 | No. 1 |
| 요약 | Identity급 II형 발사체 프로젝트 (2018-W)에서 전자부가 진행한 일, 그 결과 및 앞으로 해야 할 사항에 대한 총론 | | |
| **I. Identity-II 발사에서 전자부의 역할**  전자부는 로켓의 발사 과정에서 각종 데이터를 수집하고 로켓의 상태를 실시간으로 전송하는 역할을 맡았다. 이 때 데이터는 다음을 의미한다.   1. GY-91 센서: 로켓의 궤적을 그리기 위한 raw data    1. 압력을 이용한 고도 (altimeter)    2. 3축 선가속도 (accelerometer)    3. 3축 자기장 (magnetometer)    4. 3축 회전각속도 (gyroscope) 2. GPS 센서: 로켓의 실시간 위치 정보 3. 압력 센서: 체임버의 내압   또한 데이터 전송에는 XBee Pro S1 프로토콜 및 모듈을 이용하였다. 발사 실험에서 microSD 카드, XBee 통신 모듈을 이용하여 추출한 데이터를 분석하여 로켓의 궤도를 최종적으로 추론하는 것 (별도 보고서 참조: 궤적 복원) 까지가 전자부의 역할이었다.  **II. 전자부 모듈의 구성**  **1. 계통 구성**  Identity-II 발사체에서 전기 계통(system)은 총 세 개로 이루어져 있는데, 이는 다음과 같다.   1. 외부에서 연결되는 점화 계통 2. Avionics-I 모듈: 앞서 언급한 센서 중 GY-91 센서 및 XBee 통신 통신 모듈 탑재 3. Avionics-II 모듈: GPS 센서 및 압력 센서 탑재, 연소 체임버와 연결되어있으며, coupler 하나를 독립적인 모듈로 승격하여 이용함   개중 1) 점화 계통의 경우 전자부의 역할에서 제외되었으므로 본 보고서에서 더 이상의 언급은 하지 않도록 하겠다.  **2. 각 계통의 구조**  양 avionics 모듈은 크게 중앙 처리장치(프로세서)와 저장 장치(메모리)를 중심으로 각종 센서 장비가 달려있는 구조를 지녔다. Avionics-I (주 모듈)의 경우 Arduino Mega2560 (별도 보고서 참조: Arduino 모듈을 선택하는 방법에 관하여) 을 프로세서로 이용하였으며, Avionics-II (부 모듈)의 경우 Arduino Uno를 프로세서로 이용하였다. 두 모듈 모두 microSD 카드 모듈을 이용하여 데이터를 축적하였으며, 주 모듈의 경우 XBee 통신 모듈을 이용하여 일부 데이터에 한해 지상 통신국 (별도 보고서 참조: 통신 시스템 총론) 에서 실시간으로 정보를 받아볼 수 있도록 하였다. 정확한 각 계통의 회로도는 다음과 같다.  C:\Users\Jaerin Lee\Desktop\KakaoTalk_20180318_182125742.png  그림 1. 전자부 I 회로도. 전체 전자부 회로의 회로도이다. 위의 회로는 CERN에서 이용하는 KiCAD를 이용하여 그렸으며, 동 소프트웨어로 PCB를 작성하였다.  C:\Users\Jaerin Lee\Desktop\KakaoTalk_20180318_182103825.png 그림 2. 전자부 PCB 회로도. KiCAD로 작성하였다.  **III. 발사 실험의 결과**  **1. Xbee 통신 모듈의 통신 결과**  통신 결과 첫 낙하산 사출 때까지 올바른 데이터를 얻을 수 있었다. 그러나 첫 낙하산이 펴진 직후인 작동 후 143.113초 이후 통신이 두절되었다. 다음 그림은 microSD 카드를 이용하여 수집한 데이터를 바탕으로 로켓 (의 전자부 모듈)이 받는 가속도의 크기를 나타낸 것이다. 계단형 그래프는 로켓의 상태변화를 나타내며, 각 계단 위치는 각각 첫 번째와 두 번째의 낙하산이 펴졌음을 의미한다.  그림 3. 로켓의 가속도 변화. 빨간 줄이 통신이 끊어진 순간이다.  위 그림 3에서 볼 수 있듯이, 로켓이 최고점에서 낙하산을 펴고, 잠시 후 낙하산에 의한 강한 끌어당김으로 인한 진동으로 Zigbee 통신 모듈과 연결된 모종의 선이 끊어진 것으로 강하게 추정되는 일이 일어났다. (빨간 수직 선). 안테나가 로켓에 ㄱ자 모양으로 동체 옆에 붙어있었기 때문에 각종 충격에 취약했으며, 내부 배선이 복잡하고 정리가 되어있지 않아 알맞게 붙어있지 않은 부분이 있었다고 생각된다. 안테나의 이러한 모양은 전체 공력에도 영향을 줄 것이며, 이는 또한 추가적인 분석이 필요한 부분이다.  **2. microSD 카드를 이용한 데이터 수집**  microSD 카드를 이용하여 데이터를 수집하게 되면 다음과 같은 문제가 발생하였다. 이는 arduino와 microSD 카드 모듈을 이용한 시스템에서 어쩔 수 없이 맞닥뜨리게 되는 한계점으로서 더 나은 데이터 추출 속도를 얻기 위해서는 새로운 device를 활용해야 할 것이다.  microSD 카드에 접근하기 위해서는 파일 단위로 읽고 쓰는 과정을 수행해야 한다. SD card는 flash memory로서, block 단위로 읽고 쓰는 연산을 수행하는데, 따라서 Arduino 입장에서는 매번 file.close()를 불러와야 파일이 업데이트 되는 구조이다. 하지만 file.close는 매우 비싼 연산으로서, 수십 ms의 정해지지 않는 긴 시간 동안 데이터 추출에 지연을 생기게 한다. 두 번째로, file에 write 또는 print 하는 연산 역시 오랜 시간이 걸리기 때문에 많은 데이터를 한번에 옮기는 것이 효율 면에서 우수할 수밖에 없다. 이러한 관찰을 토대로 가장 빠르게 매 회 53 Bytes의 데이터를 추출한 결과 최소 지연 시간을 6 – 7 ms 까지 줄일 수 있었다. 하지만 매 십 수 회마다 20 – 40 ms의 긴 지연시간이 걸리는 것은 피할 수 없었다. 또한, 데이터 추출 시간이 같은 ms 이내에 이루어졌다는 보장이 없기 때문에 전반적으로 부정확한 데이터를 얻는 데에 그쳤다.  **3. Avionics-II 모듈의 동작 실패**  발사 실험 결과 Avionics-I 의 데이터는 온전히 microSD 카드에 담기게 되었지만, 급조하게 된 Avionics-II 의 데이터는 제대로 담기지 못했다. 남겨진 데이터로 미루어 분석한 결과, 다음과 같은 원인이었음이 밝혀졌다.  현재 쓰고 있는 압력계는 18-32 V의 dc 전압을 걸어주게 되면 압력에 따라 mA 단위로 전류가 나오는 전류원의 역할을 한다. 전류를 재기 위해서는 저항을 연결하여 걸리는 전압을 측정하여야 하는데, 이를 위해서 일반적인 330 Ohm의 저항이 채택되었다. 이는 압력계의 전류 범위 4 – 20 mA에 330 Ohm의 저항을 통과하게 되면, 80 bar 이내에서 5 V 이내의 전압을 확인할 수 있기에 아두이노 등의 보드를 이용하여 측정이 용이하게 되기 때문이다. 하지만, 로켓을 장착한 후, 전압을 십 수 분동안 흘려보낸 결과, 저항이 감당할 수 있는 것 이상의 전류가 흘러 저항이 점차 뜨거워짐에 따라 열저항으로 인해 저항의 크기가 올라갔으며, 이에 측정된 압력이 시간에 따라 점차 증가하다가 0으로 순식간에 떨어지는 모습을 확인하였다. 이 모든 일은 발사 전에 벌어졌으며, 회수 후 저항이 끊어졌음을 확인하였다.  압력계의 전류를 변환하기 위한 저항으로 훨씬 큰 전력소모를 견디는 저항을 이용하여야 한다는 결론을 얻을 수 있었다. GPS 역시 제대로 된 데이터를 측정할 수 없었는데, 이는 GPS 모듈과 짝퉁 아두이노 간의 상호 통신의 어려움으로부터 기인한 것이다.  신기하게도, 아래쪽 모터에 붙어있던 Avionics-II 모듈의 전자부 모든 부분이 연소 후에도 녹지 않고 앞서 언급한 330 Ohm의 저항을 제외하고는 살아있음을 확인하였다. 좋은 단열재를 썼다는 결론을 내렸다.  (별도 보고서 참조: 궤적 복원)  **4. 낙하산이 주는 흔들림에 의한 궤적 복원의 부정확성**  그림 3에서 다음과 같은 사실을 발견할 수 있다. 이번 실험에 쓴 센서는 GY-91로서, 최대 가속도 16g까지를 측정할 수 있는 센서이다. 측정 결과 낙하산에 의한 흔들림에 의해 최대 가속도 이상의 가속도가 수 차례 측정되고 있음을 확인할 수 있었다. 이는 후기 궤적 복원에 약간의 오차를 줄 수 있음을 시사한다. 물론, 발사 중의 궤적에 대해서는 크게 의미가 있지 않으므로 무시할 수도 있을 것이다.  **IV. 결론 및 제언**  다음과 같은 결론을 얻었다.   1. 압력센서에 달 저항은 큰 전류를 견딜 수 있어야 한다. 2. Arduino, Zigbee, microSD card에 의존하는 현재 방식은 향후 더 많은 데이터 추출을 위해서는 심각한 개선이 필요하다. 3. 두 개의 전자부 모듈을 하나로 합쳐야 한다. 이로서 모든 로켓 내부의 전기 계통을 하나로 관리할 수 있어야 한다. 이를 위해서는 낙하산이 한쪽에 달려있어야 할 것이다. | | | |
| 참고문헌 | *없음.* | | |
| 별첨 01 | Identity-II\_Avionics\_Module-I-Code.ino | Arduino Code | |
| 별첨 02 | Identity-II\_Avionics\_Module-II-Code.ino | Arduino Code | |
| 별첨 03 | Identity-II\_Avionics\_XBee\_Log.log | Raw Data | |
| 별첨 04 | Identity-II\_Avionics\_Module-I-SD\_Log.txt | Raw Data | |
| 별첨 05 | Identity-II\_Avionics\_Module-I-SD\_Log.csv | Processed Data | |
| 별첨 06 | Identity-II\_Avionics\_Module-II-SD\_Log1.txt | Raw Data | |
|  |  |  | |
| 의미 | 본 보고서에서 별첨 자료가 가진 의미를 기록  특별한 의미가 없는 경우, 본 보고서에 들어가기 힘든 그림, 표, 그래프 등을 일괄적으로 여기에 첨부할 것  (예시: [그림 1 ~ 5], [표 1 ~ 2]) | | |