7. 결과 분석

7.1. 기본 상태 변화

7.1.1 OFF->READY

전원 인가 후 초기 상태인 OFF 에서, Zone4 의 IR 센서로 움직임이 감지되면 ID 0x142 로 메시지가 송신하고, 중앙제어기가 이를 받아 READY 상태로 판정한다. 로그 데이터는 2 초에 한번 출력되도록 했다. READY 상태시, 환영등의 개념으로 Zone3 산하의 LED 담당 MCU 가 파란불을 On 시키는 것을 볼 수 있다.

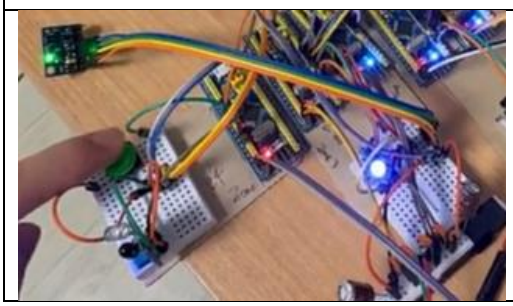
OFF -> READY

```
ajy@ajy:~/AJY/CAN $ sudo ./test
can0 인터페이스 설정 중...
can0 활성화 완료!
모니터링 시작 (2초 주기로 출력합니다...)

--- [7153572] 시스템 상태 요약 ---
현재 모드: OFF
초음파 거리: Z1:    0 cm | Z2:    0 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
--------------------------------

--- [7155576] 시스템 상태 요약 ---
현재 모드: OFF
초음파 거리: Z1:    0 cm | Z2:    0 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
--------------------------------

--- [7157581] 시스템 상태 요약 ---
현재 모드: READY
초음파 거리: Z1:    0 cm | Z2:    0 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
--------------------------------
```

7.1.2 READY -> DRIVE

READY 상태에서 Zone4 의 버튼을 누르면, 0x144 ID 로 CAN 송신한 메시지를 중앙 제어기에서 받아 상태를 DRIVE 로 상태를 전환한다. DRIVE 상태에 들어가면 Zone3 는 LED 를 OFF 하는 명령을 내린다.

READY -> DRIVE



```
--- [7157581] 시스템 상태 요약 ---
현재 모드: READY
초음파 거리: Z1:    0 cm | Z2:    0 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [7159582] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   56 cm | Z2:   74 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 3000
가속도 수신값: 1.020049
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------
```

7.1.3 DRIVE

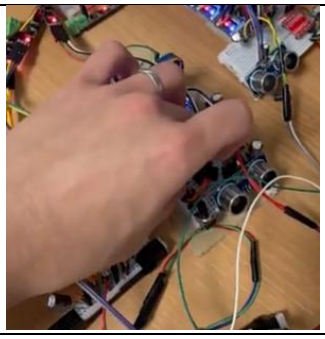
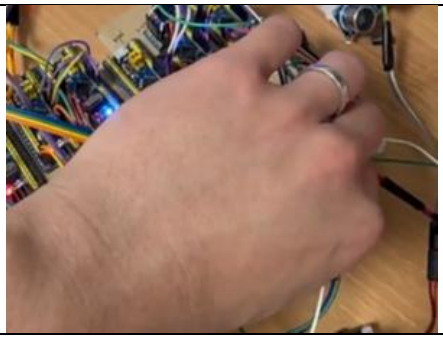
DRIVE 상태에선 초음파거리와 가속도를 측정해 감시 및 출력한다. 우측 candump 로그 데이터를 보면, 0x07x 번대의 zone 하트비트들과 0x004 의 중앙 제어기 하트비트, 그리고 0x012(가속도), 0x040(페달데이터), 0x110(모터 듀티), 0x010(초음파 1) ID 등이 CAN 버스로 메시지 전송한 것을 확인할 수 있다. 하트비트에서는 상태를 데이터로 담는데, 패킷의 데이터 중 04 를 통해 지금이 DRIVE 상태인 것을 알 수 있다. Zone 들은 중앙 제어기의 하트비트 속 상태 데이터를 통해 현 상태를 전파받는다.

페달 관련해서, 악셀 담당 가변 저항과 브레이크 담당 가변 저항 조작을 통해 페달 수신값과 모터 듀티 인가값이 변화하는 것을 확인할 수 있다. 시스템 상태 요약 로그에서는 페달 수신값이라고 되어 있으나, Zone2 의 모터 드라이버쪽 MCU 가 전송한 듀티 인가값을 Zone2 를 거쳐 중앙 제어기로 수신한 값이다. 동영상을 첨부할 수 없는 제한으로, 부득이하게 사진으로 대처한다. 하지만 로그 데이터를 통해 분명히 Zone1 산하의 가변저항이 Zone2 를 거쳐 Zone2 산하의 모터 드라이버 담당 MCU 에게 전달되는 것을 확인할 수 있다. Zone3 의 가속도 센서 값 역시 측정되어 중앙 제어기로 전송되고 있음을 확인할 수 있다.

DRIVE 상태

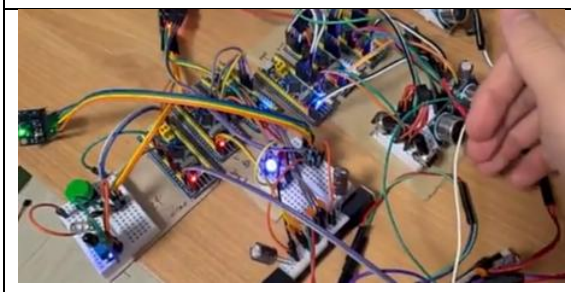
```
--- [7161585] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   56 cm | Z2:   68 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 3000
가속도 수신값: 1.020049
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [7163587] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   66 cm | Z2:   68 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 1172
가속도 수신값: 1.030048
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [7165590] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   66 cm | Z2:   67 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 634
가속도 수신값: 1.010000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------
```

```
can0  010   [8]  09 00 00 00 00 00 AD B6
can0  071   [8]  04 00 00 00 00 00 AE B2
can0  074   [8]  04 00 00 00 00 00 A7 AB
can0  004   [8]  04 00 00 00 00 00 E4 E8
can0  110   [8]  54 04 00 00 00 00 AE 06
can0  072   [8]  04 00 00 00 00 00 AF B3
can0  012   [8]  00 00 03 00 67 00 75 DF
can0  073   [8]  04 00 00 00 00 00 74 78
can0  040   [8]  54 04 00 00 00 00 AF 07
can0  071   [8]  04 00 00 00 00 00 B0 B4
can0  004   [8]  04 00 00 00 00 00 E5 E9
can0  074   [8]  04 00 00 00 00 00 A8 AC
can0  110   [8]  53 04 00 00 00 00 B0 07
can0  072   [8]  04 00 00 00 00 00 B1 B5
can0  012   [8]  00 00 03 00 67 00 77 E1
can0  073   [8]  04 00 00 00 00 00 76 7A
```

악셀 담당 가변 저항 조작	브레이크 담당 가변 저항 조작
	

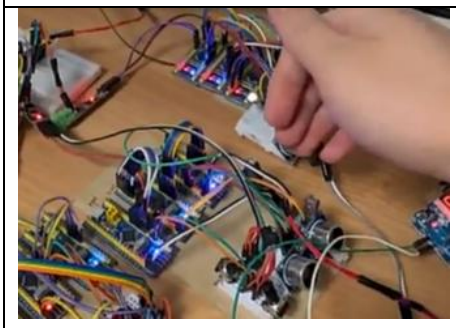
7.1.4 EMERGENCY - 초음파 1(Zone1) 경고

초음파 임계값인 3cm 이하로 초음파 값이 측정된 경우에 중앙 제어기는 EMERGENCY 상태로 판정해 이를 전파한다. 이 경우에는 곧바로 모터를 정지한다. 로그를 통해 모터가 정지한 것을 확인할 수 있다. 해당 케이스에서는 가속도 센서를 정지시켰기에 EMERGENCY 돌입 후 곧바로 READY 상태로 전환된다. READY 상태에서 다시 버튼을 누르면 DRIVE 상태로 재진입할 수 있다.

EMERGENCY - 초음파 1(Zone1) 경고	
	--- [7183618] 시스템 상태 요약 --- 현재 모드: DRIVE 초음파 거리: Z1: 41 cm | Z2: 43 cm 초음파 상태: Z1: 0 cm | Z2 : 0 cm 패달 수신값(0x110): 1104 가속도 수신값: 1.020049 노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE] ---------- [!!!] EMERGENCY STOP: Distance1 detected at 3 cm! --- [7185622] 시스템 상태 요약 --- 현재 모드: READY 초음파 거리: Z1: 0 cm | Z2: 0 cm 초음파 상태: Z1: 0 cm | Z2 : 0 cm 패달 수신값(0x110): 0 가속도 수신값: 1.010049 노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE] ----------

7.1.5 EMERGENCY - 초음파 2(Zone2) 경고

1.4 의 경우와 같이, 초음파 2 데이터로 EMERGENCY 판정하는 케이스이다. 로그 데이터에서 Distance1 과 Distance2 로 구분되는 것을 확인할 수 있다.

EMERGENCY - 초음파 2(Zone2) 경고	
	--- [7193632] 시스템 상태 요약 --- 현재 모드: DRIVE 초음파 거리: Z1: 62 cm | Z2: 7 cm 초음파 상태: Z1: 0 cm | Z2 : 0 cm 페달 수신값(0x110): 1108 가속도 수신값: 1.020049 노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE] ------------------------------ [!!!] EMERGENCY STOP: Distance2 detected at 3 cm! --- [7195634] 시스템 상태 요약 --- 현재 모드: READY 초음파 거리: Z1: 0 cm | Z2: 0 cm 초음파 상태: Z1: 0 cm | Z2 : 0 cm 페달 수신값(0x110): 0 가속도 수신값: 1.030048 노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE] ------------------------------

7.1.6 EMERGENCY - 가속도

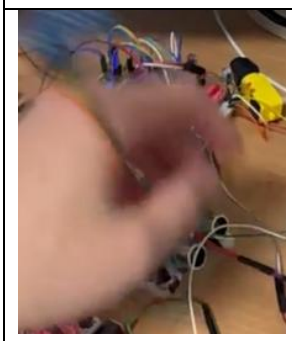
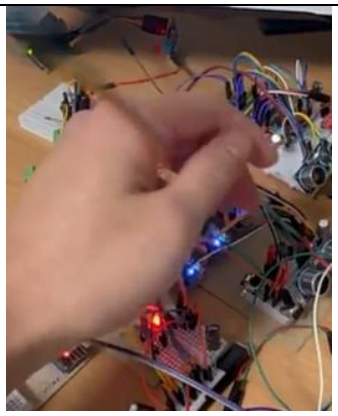
가속도 값이 크기가 3G 가 넘은 경우, 충돌 발생으로 간주해 EMERGENCY 상태로 판정한다. 아래 사진과 같이 가속도 센서를 강하게 흔들면, EMERGENCY 가 판정되는 것을 볼 수 있다.

EMERGENCY - 가속도 기반 충돌 경고	
	--- [7201644] 시스템 상태 요약 --- 현재 모드: DRIVE 초음파 거리: Z1: 66 cm | Z2: 74 cm 초음파 상태: Z1: 0 cm | Z2 : 0 cm 페달 수신값(0x110): 1106 가속도 수신값: 1.765956 노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE] ------------------------------ [!!!] EMERGENCY STOP : Collision Accelaration detected 3.938769g! --- [7203647] 시스템 상태 요약 --- 현재 모드: READY 초음파 거리: Z1: 0 cm | Z2: 0 cm 초음파 상태: Z1: 0 cm | Z2 : 0 cm 페달 수신값(0x110): 0 가속도 수신값: 0.931719 노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE] ------------------------------

7.1.7 EMERGENCY 유지

앞선 EMERGENCY 케이스에서는 가속도 센서를 고의로 정지시켜 곧바로 READY 상태로 전환되었다. 이번에는, 차량이 정지하지 않음을 가정하기 위해 가속도 센서를 조금씩 흔들어서 EMERGENCY 상태가 유지되는 경우이다. EMERGENCY 는 초음파 1 을 사용해 만들었다. 왼쪽 사진의 11 시 방향에 흔들리는 모듈이 가속도 센서이다. 가운데

사진에서, EMERGENCY 상태시 Zone3 가 산하 LED 드라이버 담당 MCU 에게 EMERGENCY 를 나타내도록 해 빨간 불이 토글되는 것을 볼 수 있다.

<table>
<tr><th colspan="3">EMERGENCY</th></tr>
<tr>
<td>  </td>
<td>  </td>
<td>
<pre>
--- [7217677] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   70 cm | Z2:   74 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 1108
가속도 수신값: 0.588727
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]

[!!!] EMERGENCY STOP: Distance1 detected at 3 cm!

--- [7219682] 시스템 상태 요약 ---
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:   3 cm | Z2:   74 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 2.137499
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]

--- [7221686] 시스템 상태 요약 ---
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:   3 cm | Z2:   74 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 2.001449
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]

--- [7223688] 시스템 상태 요약 ---
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:   3 cm | Z2:   74 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 2.095925
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]

</pre>
</td>
</tr>
</table>

7.2. 오류 대응

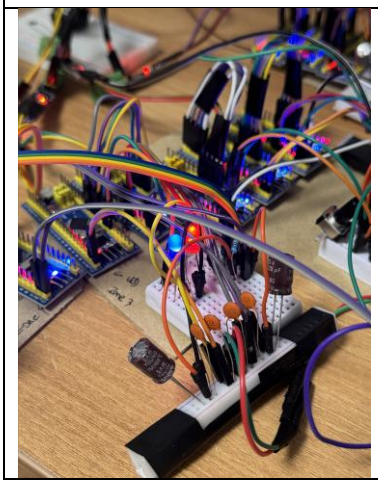
7.2.1 CENTRAL_DOWN

7.2.1.1 하트비트를 통한 감지

각 Zone 에서 중앙 제어기의 하트비트를 일정 시간(200ms) 감지할 수 없는 경우에 내부에서 자체적으로 발생하는 상태이다. 해당 경우에, Zone1 -> Zone2 로의 페달 데이터 전송 이외의 모든 기능은 정지된다. 중앙 제어기 다운시에도 페달 조작을 할 수 있어야 하므로 해당 기능은 유지한다. 중앙 제어기 프로그램을 끔으로써 테스트할 수 있다.

CENTRAL_DOWN 시에는 각 존이 하트비트를 보내지 않아, candump 에 zone1 이 보내는 페달 데이터(ID 0x040)만이 찍히는 것을 볼 수 있다. 각 존에서는 하트비트를 끄고, CAN Filter 를 재구성해 수신 리스트를 조정한다. Zone1, 2, 3 는 모두 수신 거부하고, Zone2 는 0x040 만 수신하고 나머지는 거부한다. Zone3 에서 두 LED 를 모두 깜빡여 CENTRAL_DOWN 임을 표시한다. 아래 로그를 보면, 0x040 즉 Zone1 의 페달 데이터만이 CAN 버스를 사용하는 것을 볼 수 있다. 즉, 하트비트들은 나가지 않는다. 로그속 데이터를 통해 페달 조작이 가능함을 확인할 수 있다.

CENTRAL_DOWN



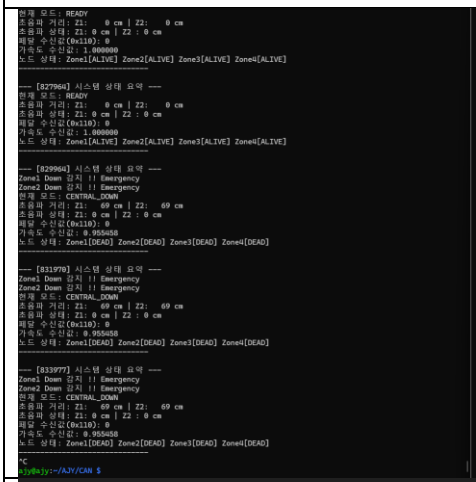
```
--- [13412730] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:  16 cm | Z2:  75 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 1109
가속도 수신값: 1.010000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------

--- [13414736] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:  71 cm | Z2:  75 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 1110
가속도 수신값: 1.010000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------

--- [13416743] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:  71 cm | Z2:  74 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 1109
가속도 수신값: 1.020000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------
^[[A
--- [13418751] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:  16 cm | Z2:  57 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 1110
가속도 수신값: 1.020000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------
^C
ajy@ajy:~/AJY/CAN $
```

```
can0  040  [8]  62 03 00 00 00 00 15 7A
can0  040  [8]  69 03 00 00 00 00 16 82
can0  040  [8]  62 03 00 00 00 00 17 7C
can0  040  [8]  54 03 00 00 00 00 18 6F
can0  040  [8]  D0 02 00 00 00 00 19 EB
can0  040  [8]  3A 03 00 00 00 00 1A 57
can0  040  [8]  A4 02 00 00 00 00 1B C1
can0  040  [8]  DF 02 00 00 00 00 1C FD
can0  040  [8]  59 03 00 00 00 00 1D 79
can0  040  [8]  57 03 00 00 00 00 1E 7B
can0  040  [8]  57 03 00 00 00 00 1F 79
can0  040  [8]  62 03 00 00 00 00 20 85
can0  040  [8]  50 03 00 00 00 00 21 74
can0  040  [8]  5D 03 00 00 00 00 22 82
can0  040  [8]  6C 03 00 00 00 00 23 92
can0  040  [8]  4D 03 00 00 00 00 24 74
can0  040  [8]  41 03 00 00 00 00 25 69
can0  040  [8]  49 03 00 00 00 00 26 72
can0  040  [8]  4D 03 00 00 00 00 27 77
can0  040  [8]  2F 03 00 00 00 00 28 5A
can0  040  [8]  31 03 00 00 00 00 29 5D
can0  040  [8]  3D 03 00 00 00 00 2A 6A
can0  040  [8]  4D 03 00 00 00 00 2B 7B
can0  040  [8]  51 03 00 00 00 00 2C 80
can0  040  [8]  57 03 00 00 00 00 2D 87
can0  040  [8]  62 03 00 00 00 00 2E 93
can0  040  [8]  6D 03 00 00 00 00 2F 9F
can0  040  [8]  6A 03 00 00 00 00 30 9D
can0  040  [8]  6D 03 00 00 00 00 31 A1
can0  040  [8]  6D 03 00 00 00 00 32 A2
can0  040  [8]  6A 03 00 00 00 00 33 A0
can0  040  [8]  6A 03 00 00 00 00 34 A1
can0  040  [8]  6A 03 00 00 00 00 35 A2
can0  040  [8]  6F 03 00 00 00 00 36 A8
can0  040  [8]  71 03 00 00 00 00 37 AB
can0  040  [8]  71 03 00 00 00 00 38 AC
can0  040  [8]  6F 03 00 00 00 00 39 AB
```

7.2.1.2 패킷 분석을 통한 감지

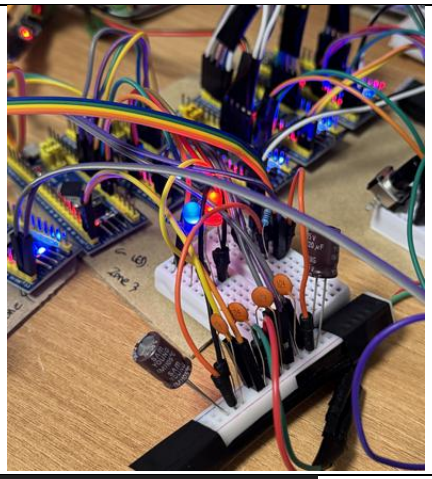
중앙 제어기가 보내는 Heartbeat 패킷의 체크섬, 시퀀스 번호(카운터)를 검증해 연속으로 20 회 패킷 감지시 중앙 제어기나 CAN 버스 고장으로 간주해 CENTRAL_DOWN 으로 전환한다. 고의로 중앙 제어기 하트비트의 시퀀스 번호를 망가뜨려 테스트했다. 따라서 이 시퀀스 번호로 인해 Zone 들의 패킷 검사 로직에 걸린다. 로그를 통해 페달 데이터가 전송되고 있음을 알 수 있다. Zone 들에서는 CENTRAL_DOWN 이 발동되어 하트비트를 보내지 않기에 중앙 제어기에서 이를 감지한다. 모든 Zone 의 하트비트가 오지 않으면 CENTRAL_DOWN 으로 자체 판정한다. CENTRAL_DOWN 이기에 LED 가 둘 모두 토글되는 것을 볼 수 있다.

CENTRAL_DOWN



```
현재 모드: READY
초음파 거리: Z1:   0 cm | Z2:   0 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------

--- [827964] 시스템 상태 요약 ---
현재 모드: READY
초음파 거리: Z1:   0 cm | Z2:   0 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------

--- [829964] 시스템 상태 요약 ---
Zone1 Down 감지 !! Emergency
Zone2 Down 감지 !! Emergency
현재 모드: CENTRAL_DOWN
초음파 거리: Z1:  69 cm | Z2:  69 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 0.955458
노드 상태: Zone1[DEAD] Zone2[DEAD] Zone3[DEAD] Zone4[DEAD]
------------------------

--- [831970] 시스템 상태 요약 ---
Zone1 Down 감지 !! Emergency
Zone2 Down 감지 !! Emergency
현재 모드: CENTRAL_DOWN
초음파 거리: Z1:  69 cm | Z2:  69 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 0.955458
노드 상태: Zone1[DEAD] Zone2[DEAD] Zone3[DEAD] Zone4[DEAD]
------------------------

--- [833977] 시스템 상태 요약 ---
Zone1 Down 감지 !! Emergency
Zone2 Down 감지 !! Emergency
현재 모드: CENTRAL_DOWN
초음파 거리: Z1:  69 cm | Z2:  69 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 0.955458
노드 상태: Zone1[DEAD] Zone2[DEAD] Zone3[DEAD] Zone4[DEAD]
------------------------
^C
ajy@ajy:~/AJY/CAN $
```

```
can0  040  [8]  00 00 00 00 00 00 EA EA
can0  040  [8]  00 00 00 00 00 00 EB EB
can0  040  [8]  00 00 00 00 00 00 EC EC
can0  040  [8]  00 00 00 00 00 00 ED ED
can0  040  [8]  00 00 00 00 00 00 EE EE
can0  040  [8]  00 00 00 00 00 00 EF EF
can0  040  [8]  00 00 00 00 00 00 F0 F0
can0  040  [8]  00 00 00 00 00 00 F1 F1
can0  040  [8]  00 00 00 00 00 00 F2 F2
can0  040  [8]  00 00 00 00 00 00 F3 F3
can0  040  [8]  00 00 00 00 00 00 F4 F4
can0  040  [8]  00 00 00 00 00 00 F5 F5
can0  040  [8]  00 00 00 00 00 00 F6 F6
can0  040  [8]  00 00 00 00 00 00 F7 F7
can0  040  [8]  00 00 00 00 00 00 F8 F8
can0  040  [8]  00 00 00 00 00 00 F9 F9
can0  040  [8]  00 00 00 00 00 00 FA FA
can0  040  [8]  00 00 00 00 00 00 FB FB
can0  040  [8]  00 00 00 00 00 00 FC FC
can0  040  [8]  00 00 00 00 00 00 FD FD
can0  040  [8]  00 00 00 00 00 00 FE FE
can0  040  [8]  00 00 00 00 00 00 FF FF
can0  040  [8]  00 00 00 00 00 00 00 00
can0  040  [8]  00 00 00 00 00 00 01 01
can0  040  [8]  14 00 00 00 00 00 02 16
can0  040  [8]  4B 00 00 00 00 00 03 4D
can0  040  [8]  69 00 00 00 00 00 04 6D
can0  040  [8]  97 00 00 00 00 00 05 9C
can0  040  [8]  AB 00 00 00 00 00 06 B1
can0  040  [8]  C6 00 00 00 00 00 07 CD
can0  040  [8]  EC 00 00 00 00 00 08 F4
can0  040  [8]  12 01 00 00 00 00 09 1C
can0  040  [8]  4C 01 00 00 00 00 0A 57
can0  040  [8]  51 01 00 00 00 00 0B 5D
can0  040  [8]  6C 01 00 00 00 00 0C 79
can0  040  [8]  7B 01 00 00 00 00 0D 89
can0  040  [8]  64 01 00 00 00 00 0E 73
can0  040  [8]  85 01 00 00 00 00 0F 95
can0  040  [8]  85 01 00 00 00 00 10 96
can0  040  [8]  96 01 00 00 00 00 11 A8
can0  040  [8]  9E 01 00 00 00 00 12 AB
can0  040  [8]  98 01 00 00 00 00 13 AC
can0  040  [8]  9A 01 00 00 00 00 14 AF
can0  040  [8]  9E 01 00 00 00 00 15 B4
can0  040  [8]  96 01 00 00 00 00 16 AD
can0  040  [8]  9A 01 00 00 00 00 17 AC
can0  040  [8]  91 01 00 00 00 00 18 AA
can0  040  [8]  90 01 00 00 00 00 19 AA
can0  040  [8]  98 01 00 00 00 00 1A B3
can0  040  [8]  93 01 00 00 00 00 1B AF
can0  040  [8]  8C 01 00 00 00 00 1C A9
```



```
if(current_state == DRIVE) tx_counter = 0; // (2) CENTRAL_DOWN2 의도적으로 드라이브상태에서 카운터 망가뜨리기
tx_frame.data[0] = current_state;
tx_frame.data[6] = tx_counter++;
tx_frame.data[7] = calculate_checksum(tx_frame.data);
write(s, &tx_frame, sizeof(struct can_frame));

usleep(10000); // 10ms
```

7.2.2 Zone1 에러

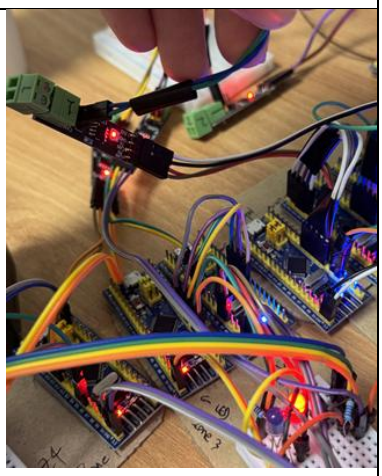
7.2.2.1 BROKEN_SELF & EMERGENCY

Zone1 내부에서 CAN 송신 불가 판정 내리는 경우이다. 해당 상태에선 하트비트를 내보내지 않기 때문에, 중앙 제어기에서 일정 시간 이상 ZONE1 의 하트비트가 오지 않는다면 Zone1_DOWN 판정을 내리고 EMERGENCY 상태로 들어간다. Zone2 는 해당 상태에서 듀티 값 0 을 드라이버 MCU 에게 전달한다. Zone1 의 CAN 을 분리했을 시, 로그를 통해 Zone1 DOWN 이 감지되어 EMERGENCY 가 발생한 것을 볼 수 있다. 우측 로그에서도 상태가 04(DRIVE)에서 08(EMERGENCY)로 바뀌고, 모터 인가 피드백은 0 으로 바뀌어 모터가 정지함을 볼 수 있다. Zone3 의 LED 파츠에서 빨강불이 토글되며 EMERGENCY 임을 알리는 것 역시 볼 수 있다.

Zone1 에러 - EMERGENCY

```
--- [1083319] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:  57 cm | Z2:  50 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 534
가속도 수신값: 1.020000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [1085326] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:  57 cm | Z2:  74 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 471
가속도 수신값: 1.030048
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [1087334] 시스템 상태 요약 ---
Zone1 Down 감지 !! Emergency
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:  50 cm | Z2:  56 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 0.980016
노드 상태: Zone1[DEAD] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [1089342] 시스템 상태 요약 ---
Zone1 Down 감지 !! Emergency
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:  50 cm | Z2:  56 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 0.990454
노드 상태: Zone1[DEAD] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [1091347] 시스템 상태 요약 ---
Zone1 Down 감지 !! Emergency
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:  50 cm | Z2:  56 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
배달 수신값(0x110): 0
가속도 수신값: 1.000650
노드 상태: Zone1[DEAD] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------
```

```
can0  110   [8]  D5 01 00 00 00 00 18 EE
can0  072   [8]  04 00 00 00 00 00 19 1D
can0  012   [8]  00 00 00 00 65 00 E4 49
can0  073   [8]  04 00 00 00 00 00 E3 E7
can0  074   [8]  04 00 00 00 00 00 27 2B
can0  004   [8]  04 00 00 00 00 00 01 05
can0  072   [8]  04 00 00 00 00 00 1A 1E
can0  073   [8]  04 00 00 00 00 00 E5 E9
can0  012   [8]  00 00 00 00 65 00 E6 4B
can0  074   [8]  04 00 00 00 00 00 28 2C
can0  004   [8]  04 00 00 00 00 00 02 06
can0  072   [8]  04 00 00 00 00 00 1B 1F
can0  073   [8]  04 00 00 00 00 00 E7 EB
can0  110   [8]  00 00 00 00 00 00 3D 3D
can0  072   [8]  08 00 00 00 00 00 3E 46
can0  073   [8]  08 00 00 00 00 00 23 2B
can0  012   [8]  F9 FF 04 00 5E 00 24 7E
can0  074   [8]  08 00 00 00 00 00 42 4A
can0  002   [8]  08 00 00 00 00 00 21 29
can0  110   [8]  00 00 00 00 00 00 3F 3F
can0  072   [8]  08 00 00 00 00 00 40 48
can0  073   [8]  08 00 00 00 00 00 25 2D
can0  012   [8]  F9 FF 04 00 5E 00 26 80
can0  074   [8]  08 00 00 00 00 00 43 4B
can0  002   [8]  08 00 00 00 00 00 22 2A
```



7.2.2.2 EMERGENCY

Zone1 이 보내는 패킷이 연속적으로 오류가 발생한 경우에 대한 EMERGENCY 이다. 중앙 제어기 측에서 체크섬, 시퀀스번호(카운터)를 검사해 오류가 10 회 연속 발생하면 Zone1_down 판정을 내린다. 만약 정상 패킷이 수신되면 오류 카운터는 0 으로 초기화된다. EMERGENCY 이기에 LED3 가 이를 나타낸다.

Zone1 에러 (데이터 변조) - EMERGENCY


```
        break;
    case ID_1_HB: // heartbeat
        payload.data16.data = state_global;
        payload.data16.counter = 0; // 의도적으로 (4) EMEGENCY2 일으키기
```

zone1 의 main.c

7.2.3 PEDAL_ERR

Zone1 이 페달 담당 MCU 와의 SPI 통신에 문제가 생긴 경우에, 이를 판정하고 중앙 제어기에게 0x022 ID 로 송신해 이를 알린다. 중앙 제어기 측에서는 이를 받았을 때 PEDAL_ERR 로 상태를 전환시키고 전파한다. PEDAL_ERR 에서는 Zone1 은 페달 데이터 전송을 중지하고, Zone3 는 이를 알리는 LED 명령을 LED 드라이버에게 전송한다. Zone2 는 모터 드라이버 담당 MCU 에게 0 을 전송해 중지 명령을 내린다. Zone3 는 PEDAL_ERR 의 경우 CENTER_DOWN 과 같은 LED 명령을 내려 알린다.

Zone1 의 페달 담당 MCU 가 보내는 SPI 패킷의 SEQ 필드를 강제로 0 으로 유지시켜 테스트했다.

PEDAL_ERR STATE

```
--- [3966523] 시스템 상태 요약 ---
현재 모드: READY
초음파 거리: Z1:   0 cm | Z2:   0 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
----------------------------
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
```

```
can0  073   [8]  02 00 00 00
can0  004   [8]  02 00 00 00
can0  071   [8]  02 00 00 00
can0  072   [8]  02 00 00 00
can0  074   [8]  02 00 00 00
can0  144   [8]  CC 00 00 00
can0  073   [8]  02 00 00 00
can0  004   [8]  04 00 00 00
can0  022   [8]  00 00 00 00
can0  010   [8]  48 00 00 00
can0  071   [8]  04 00 00 00
can0  011   [8]  3A 00 00 00
can0  072   [8]  04 00 00 00
can0  074   [8]  04 00 00 00
can0  073   [8]  04 00 00 00
can0  012   [8]  00 00 00 00
can0  004   [8]  20 00 00 00
can0  022   [8]  00 00 00 00
can0  071   [8]  20 00 00 00
can0  072   [8]  20 00 00 00
can0  073   [8]  20 00 00 00
```

```
--- [3968527] 시스템 상태 요약 ---
현재 모드: PEDAL_ERR
초음파 거리: Z1:   76 cm | Z2:  120 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 1.010891
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
----------------------------
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
pedal down
```

```
tx_buf.pkt.header = 0xAA;
tx_buf.pkt.data = pedal;
tx_buf.pkt.status = 0;
tx_buf.pkt.seq = g_seq++; // 여기서 시퀀스 증가
tx_buf.pkt.seq = 0; // PEDAL_ERR 유도
tx_buf.pkt.reserved = 0x00;
tx_buf.pkt.crc = crc_xor((uint8_t*)&tx_buf.pkt, 6);
tx_buf.pkt.tail = 0xED;
```

Zone1 산하 PEDAL MCU 코드

7.2.4 Zone2 에러

7.2.4.1 BROKEN_SELF & EMERGENCY

Zone2 의 CAN 고장을 Zone2 내부에서 자체적으로 판단한다. 중앙 제어기 측에서 하트비트 감시를 통해 Zone2 의 CAN 송신을 감시해, 일정 시간 이상 오지 않는다면 ZONE2_DOWN 처리해 EMERGENCY 를 판정한다. 만일 Zone2 의 수신 시에도 망가졌다고 가정하여도, 그렇다면 CENTRAL_DOWN 판정 로직(하트비트 감시, 수신 패킷 체크섬, 카운터 검증)으로 걸리기 때문에 모터는 정지한다. EMERGENCY 이기에 ZONE3 의 빨간불이 토글된다

Zone2 에러 - EMERGENCY

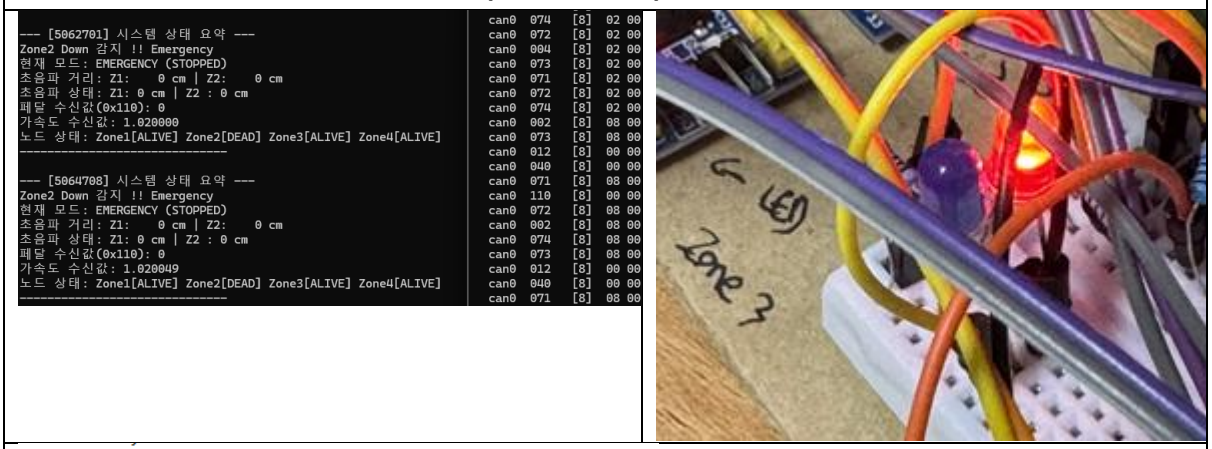
```
--- [4258956] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   64 cm | Z2:   58 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 408
가속도 수신값: 1.010049
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
----------------------------

--- [4260956] 시스템 상태 요약 ---
Zone2 Down 감지 !! Emergency
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:   60 cm | Z2:   45 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 406
가속도 수신값: 1.020196
노드 상태: Zone1[ALIVE] Zone2[DEAD] Zone3[ALIVE] Zone4[ALIVE]
----------------------------

--- [4262960] 시스템 상태 요약 ---
Zone2 Down 감지 !! Emergency
현재 모드: EMERGENCY (STOPPED)
초음파 거리: Z1:   60 cm | Z2:   45 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 406
가속도 수신값: 1.030437
노드 상태: Zone1[ALIVE] Zone2[DEAD] Zone3[ALIVE] Zone4[ALIVE]
----------------------------
```

```
can0  012   [8]  00 00 FE FF 64 00 97 F8
can0  074   [8]  08 00 00 00 00 00 85 8D
can0  040   [8]  00 00 00 00 00 00 F4 F4
can0  071   [8]  08 00 00 00 00 00 F5 FD
can0  002   [8]  08 00 00 00 00 00 85 8D
can0  073   [8]  08 00 00 00 00 00 98 A6
can0  012   [8]  00 00 FE FF 64 00 99 FA
can0  074   [8]  08 00 00 00 00 00 86 8E
can0  040   [8]  00 00 00 00 00 00 F6 F6
can0  071   [8]  08 00 00 00 00 00 F7 FF
can0  002   [8]  08 00 00 00 00 00 86 8E
can0  073   [8]  08 00 00 00 00 00 9A A2
can0  012   [8]  00 00 FE FF 65 00 9B FD
can0  074   [8]  08 00 00 00 00 00 87 8F
can0  040   [8]  00 00 00 00 00 00 F8 F8
can0  071   [8]  08 00 00 00 00 00 F9 01
can0  002   [8]  08 00 00 00 00 00 87 8F
can0  073   [8]  08 00 00 00 00 00 9C A4
can0  012   [8]  00 00 FE FF 65 00 9D FF
can0  074   [8]  08 00 00 00 00 00 88 90
can0  040   [8]  00 00 00 00 00 00 FA FA
can0  071   [8]  08 00 00 00 00 00 FB 03
can0  002   [8]  08 00 00 00 00 00 88 90
can0  073   [8]  08 00 00 00 00 00 9E A6
can0  012   [8]  00 00 FE FF 65 00 9F 01
can0  040   [8]  00 00 00 00 00 00 FC FC
can0  071   [8]  08 00 00 00 00 00 FD 05
can0  074   [8]  08 00 00 00 00 00 89 91
```

7.2.4.2 EMERGENCY

Zone2 가 중앙 제어기에게 보내는 패킷에 변조가 연속 발생시 발생한다. 중앙 제어기에서 패킷을 검증해 EMERGENCY 를 판정한다. 만일 Zone2 의 수신 시에도 망가졌다고 가정하여도, 그렇다면 CENTRAL_DOWN 판정 로직(하트비트 감시, 수신 패킷 체크섬, 카운터 검증)으로 걸리기 때문에 모터는 정지한다. EMERGENCY 이기에 Zone3 에서 LED 중 빨간불을 토글하게 한다. 이때, Zone1 과는 다르게 MOTOR_STATUS 도 의도적으로 카운터를 망가뜨린다. 왜냐하면, 중앙 제어기에서 정상 can 메시지 수신시 에러 카운트를 초기화하기 때문이다. Zone1 의 경우, 초음파와 페달 데이터가 하트비트에 비해 매우 주기가 느리므로 하트비트만 망기뜨렸지만, Zone2 의 경우 모터 피드백이 하트비트와 동일 주기이므로 모터 상태 메시지 역시 카운터를 망가뜨려야 한다.

Zone2 에러 (데이터 변조) - EMERGENCY




```
    case ID_2_MOTOR_STATUS:
        payload.data16.data = g_pedal_from_driver; //페달측정값
        payload.data16.counter = 0; // 의도적인 고장 EMERGENCY
        break;
    case ID_2_MOTOR_SPI_ERR:
        payload.data16.data = 0;
        break;
    case ID_2_HB:
        payload.data16.data = state_global; //HEARTBEAT론 현 반영 상태 보냄
        payload.data16.counter = 0; // 의도적인 고장 EMERGENCY
```

Zone2 제어기의 main.c

7.2.4.3 MOTOR_ERR

Zone2 의 모터 담당 MCU 가 보내는 SPI 패킷의 SEQ 필드를 강제로 0 으로 유지시켜 테스트했다. Zone2 는 SPI 마스터고, 모터 담당 MCU 는 SPI 슬레이브로, Full-Duplex 모드에서 Zone2 는 Zone1 에서 받은 페달값을 전달하고, 모터 MCU 는 인가한 듀티를 Zone2 에게 돌려준다. 이때 모터 MCU 가 돌려주는 패킷에 오염을 일으켰다.
Zone2 에서 패킷 검증을 통해 검출해내고, 일정 횟수 이상 연속으로 오염 발생 시 플래그를 세워 canTask 에서 ID_2_MOTOR_SPI_ERR 로 중앙 제어기에게 에러를 알린다.

Zone3 는 MOTOR_ERR 의 경우 CENTER_DOWN 과 같은 LED 명령을 내려 알린다. MOTOE_ERR 상태에서는 아래 사진 중 부가 설명 코드에서와 같이, 조건문에 걸려 0 의 듀티를 모터 담당 MCU 에게 전달한다. 아래 로그를 통해, ID0x020 으로 모터단의 SPI 에러를 중앙 제어기에게 알리는 것을 볼 수 있다. zone 들은 하트비트에 0x10 을 데이터로 보내 지금 MOTOR_ERR 상태라고 나타내준다.

Zone2 내부에서 플래그가 세워진 경우, 의도적으로 모터 MCU 에게 SPI 전송하는 패킷 필드 중 status 에 0xFF 를 심어 에러 상황을 알린다.

MOTOR_ERR

```
MOTOR DOWN
MOTOR DOWN

--- [5828566] 시스템 상태 요약 ---
현재 모드: MOTOR_ERR
초음파 거리: Z1:   71 cm | Z2:   56 cm
초음파 상태: Z1: 0 cm | Z2 : 0 cm
페달 수신값(0x110): 0
가속도 수신값: 1.129115
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------
MOTOR DOWN
MOTOR DOWN
MOTOR DOWN
MOTOR DOWN
MOTOR DOWN
```

```
can0  012   [8]  01 00 01 00 66 00 1B 83
can0  020   [8]  00 00 00 00 00 00 14 14
can0  072   [8]  10 00 00 00 00 00 15 25
can0  004   [8]  10 00 00 00 00 00 53 63
can0  074   [8]  10 00 00 00 00 00 6E 7E
can0  071   [8]  10 00 00 00 00 00 06 16
can0  073   [8]  10 00 00 00 00 00 1C 2C
can0  012   [8]  01 00 01 00 66 00 1D 85
can0  020   [8]  00 00 00 00 00 00 16 16
can0  072   [8]  10 00 00 00 00 00 17 27
can0  004   [8]  10 00 00 00 00 00 54 64
can0  071   [8]  10 00 00 00 00 00 07 17
can0  074   [8]  10 00 00 00 00 00 6F 7F
can0  073   [8]  10 00 00 00 00 00 1E 2E
can0  012   [8]  01 00 01 00 66 00 1F 87
can0  020   [8]  00 00 00 00 00 00 18 18
```

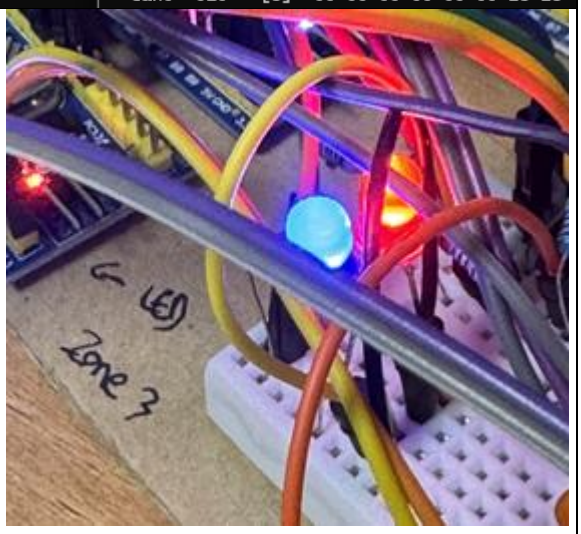
```
        }
        tx_buf.pkt.data = duty;
        tx_buf.pkt.seq = tx_cnt++;
        tx_buf.pkt.seq = 0; // (8) MOTOR_ERR 강제 유도
        tx_buf.pkt.crc = crc_xor((uint8_t*)&tx_buf.pk
    }
```

Zone2 산하 모터 담당 MCU 의 코드

부가 설명*

Zone2 의 spiTask 중 일부

```
if((state_local != DRIVE) && (state_local != CENTRAL_DOWN)) g_pedal = 0;//

g_pedal = (g_pedal < 0) ? 0 : ((g_pedal > 3000) ? 3000 : g_pedal); // 랩 저
// SPI DMA 보낼 패킷 구성
pedal_tx.pkt.header = 0xAA;
pedal_tx.pkt.data = g_pedal;
pedal_tx.pkt.status = SPI2_ERR_FLAG? 0xFF : 0x00;
pedal_tx.pkt.seq = seq++;
pedal_tx.pkt.reserved = 0x00;
pedal_tx.pkt.crc = crc_xor(pedal_tx.bytes, 6);
pedal_tx.pkt.tail = 0xED;
```



7.3. 기타 에러

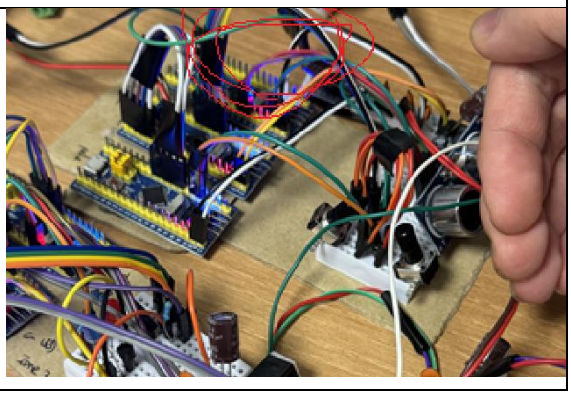
7.3.1 초음파 1 Isolation 1 & 초음파 2 Isolation 1

초음파 SPI 에 문제가 생긴 경우 발생한다. Zone1 은, 내부에서 큐를 통해 CAN Task 에서 초음파 데이터를 사용하기에 초음파 데이터가 오지 않으면 자동적으로 초음파 데이터를 중앙 제어기에게 전송하지 않는다. 따라서 중앙 제어기는 1 초 이상 초음파 데이터가 오지 않는 경우에 이를 표시하고, 충돌 판정 로직에서 초음파 1 을 분리한다. 해당 Isolation 기능은 DRIVE 모드에서만 동작한다. 사진으로 초음파 센서에

손가락을 가져다대도 Zone3 에서 비상등 모드가 발동되지 않았기에, EMERGENCY 판정 로직에서 제외된 것을 알 수 있다. 빨간 동그라미친 부분과 같이 SPI 연결 배선을 제거해 테스트했다.

초음파 에러 - Isolation

```
--- [6672424] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:    0 cm | Z2:   71 cm
초음파 상태: Z1: 0 | Z2 : 0
페달 수신값(0x110): 0
가속도 수신값: 1.020000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[AL
------------------------------

--- [6674428] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:    0 cm | Z2:   71 cm
초음파 상태: Z1: 1 | Z2 : 0
페달 수신값(0x110): 0
가속도 수신값: 1.010000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[AL
------------------------------
```

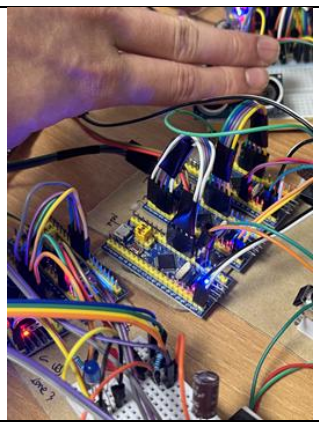


7.3.2 초음파 2 Isolation 2 & 초음파 2 Isolation 2

초음파 1 과 같이 테스트할 수 있으나, 초음파 데이터의 변조가 연속 발생했을 때에도 알 수 있다. Zone2 산하 초음파 담당 MCU 의 SPI 전송 패킷을 고의로 시퀀스 필드를 0 으로 고정해 테스트했다. Zone2 의 초음파를 대표로 테스트했다. 해당 경우에, Zone2 내부에서 5 회 이상 에러가 발생했을 때 플래그를 세워 고장을 알리고, 큐 업데이트를 중지한다. 큐 업데이트가 중지되면 자동적으로 CAN Task 에서 초음파 2 ID 의 송신이 이루어지지 않는다. Z2 = 1 로 나타낸 것을 알 수 있다. 첨부한 사진으로, Zone2 산하의 초음파 센서에 손가락을 가져다대도 Zone3 의 LED 가 DRIVE 즉 OFF 상태인 것으로 EMERGENCY 가 판정나지 않았다는 것을 볼 수 있다. 즉 Zone2 의 초음파가 충돌 판정 로직에서 제외되었음을 알 수 있다.

초음파 에러(데이터 변조) - Isolation

```
--- [6341404] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   73 cm | Z2:    0 cm
초음파 상태: Z1: 0 | Z2 : 1
페달 수신값(0x110): 0
가속도 수신값: 1.020000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------

--- [6343409] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   67 cm | Z2:    0 cm
초음파 상태: Z1: 0 | Z2 : 1
페달 수신값(0x110): 0
가속도 수신값: 1.020000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[ALIVE] Zone4[ALIVE]
------------------------------
```



```
void send_distance_to_master(uint8_t status) {
    // SPI 패킷 구성
    tx_buf.pkt.header   = 0xAA;
    tx_buf.pkt.distance = g_distance; // 마스터에서 (h<<8 | l)로 읽으므로
    tx_buf.pkt.status   = status;
    tx_buf.pkt.seq      = g_seq++;
    tx_buf.pkt.seq      = 0; // (10) 테스트 위해서 강제로 0으로 만듦
    tx_buf.pkt.reserved = 0x00;
    tx_buf.pkt.crc      = crc_xor(tx_buf.bytes, 6); // 0~5번까지 계산
    tx_buf.pkt.tail     = 0xED;
```

Zone2 산하 초음파 담당 MCU 의 SPI 송신 패킷 구성 코드

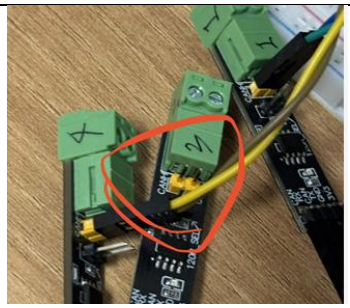
7.3.3 Zone 3 Isolation

Zone3 의 CAN 에 문제가 발생한 경우이다. Zone3 는 가속도와 LED 가 포함되어 있어, 안전과는 직접적인 영향이 없다. 따라서 단순히 가속도 센서를 충돌 로직에서 제외시키는 정도로 처리한다. 아래 로그를 통해, Zone3 가 분리되어도 모터, 페달 초음파 등이 정상 동작하는 것을 볼 수 있다.

Zone3 에러 - Isolation

```
--- [7100159] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   67 cm | Z2:   59 cm
초음파 상태: Z1: 0 | Z2 : 0
페달 수신값(0x110): 0
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[DEAD] Zone4[ALIVE]
------------------------------

--- [7102160] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   68 cm | Z2:   82 cm
초음파 상태: Z1: 0 | Z2 : 0
페달 수신값(0x110): 651
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[DEAD] Zone4[ALIVE]
------------------------------

--- [7104161] 시스템 상태 요약 ---
현재 모드: DRIVE
초음파 거리: Z1:   72 cm | Z2:   83 cm
초음파 상태: Z1: 0 | Z2 : 0
페달 수신값(0x110): 356
가속도 수신값: 1.000000
노드 상태: Zone1[ALIVE] Zone2[ALIVE] Zone3[DEAD] Zone4[ALIVE]
------------------------------

--- [7106166] 시스템 상태 요약 ---
현재 모드: DRIVE
```

```
can0  071   [8]  04 00 00 00 00 00 0F 13
can0  110   [8]  51 01 00 00 00 00 15 67
can0  072   [8]  04 00 00 00 00 00 16 1A
can0  074   [8]  04 00 00 00 00 00 D3 D7
can0  040   [8]  54 01 00 00 00 00 10 65
can0  071   [8]  04 00 00 00 00 00 11 15
can0  004   [8]  04 00 00 00 00 00 A3 A7
can0  110   [8]  51 01 00 00 00 00 17 69
can0  072   [8]  04 00 00 00 00 00 18 1C
can0  040   [8]  54 01 00 00 00 00 12 67
can0  010   [8]  10 00 00 00 00 00 13 23
can0  071   [8]  04 00 00 00 00 00 14 18
can0  004   [8]  04 00 00 00 00 00 A4 A8
can0  074   [8]  04 00 00 00 00 00 D4 D8
can0  011   [8]  39 00 00 00 00 00 19 52
can0  072   [8]  04 00 00 00 00 00 1B 1F
can0  110   [8]  54 01 00 00 00 00 1A 6F
can0  040   [8]  53 01 00 00 00 00 15 69
can0  071   [8]  04 00 00 00 00 00 16 1A
can0  004   [8]  04 00 00 00 00 00 A5 A9
can0  074   [8]  04 00 00 00 00 00 D5 D9
can0  110   [8]  54 01 00 00 00 00 1C 71
can0  072   [8]  04 00 00 00 00 00 1D 21
can0  040   [8]  53 01 00 00 00 00 17 6B
can0  071   [8]  04 00 00 00 00 00 18 1C
can0  004   [8]  04 00 00 00 00 00 A6 AA
can0  074   [8]  04 00 00 00 00 00 D6 DA
can0  110   [8]  53 01 00 00 00 00 1E 72
can0  072   [8]  04 00 00 00 00 00 1F 23
```



* 아래부턴 보고서 분량 한계상 사진을 첨부하지 못했습니다. 다만, 앞선 에러 케이스들과 사실상 에러를 일으키는 방법과 감지하는 방법, 그리고 대응이 똑같기 때문에 생략해도 무관하다고 판단했습니다.

7.3.4 가속도 센서 & LED Isolation

Zone3 산하의 가속도 모듈 I2C 통신이 망가진 경우에 동작한다. 가속도 센서 값은 충돌 로직에서 제외시키는 정도로 처리한다. 이때, 가속도 센서 고장 시 데이터에 0xFFFF 를 담아 중앙 제어기에 알린다.

7.3.5 Zone4 Isolation

7.3.5.1 Zone4 의 하트비트를 중앙 제어기에서 일정 시간 이상 확인할 수 없을 때이다. 단순히 Zone4 down 을 출력한다.

7.3.5.2 Zone4 의 CAN 패킷을 중앙 제어기에서 검증해, 일정 횟수 이상 연속으로 에러가 발생한 경우이다.

8. 고찰

해당 프로젝트는 상태를 기반으로 전체 시스템을 운용하는 임베디드 시스템을 구현한 프로젝트이다. 이를 위해 중앙 제어기와 4 개의 Zone 제어기를 배치해, 중앙 제어기가 관리하는 상태에 따라 시스템이 동작하도록 시스템 전체를 구성했다.

특히, 수업에서 배운 SPI 통신 인터페이스를 활용해 한 Zone 과 그 산하 모듈 간의 통신 시스템을 구현했다. 수업에서 배우지 않은 CAN 통신을 학습해 적용하기도 하고, 여러 모듈을 계층 구조 시스템에 집적하는 경험을 했다. 단순히 충돌 대응 및 페달로 모터를 조작하는 기능을 넘어, 노드들의 고장에 대비하는 Fail-safe, 그리고 초음파 센서를 2 개 배치에 Fault-Tolerance 기능, 그리고 오류 번짐을 고려해 모듈과 기능을 분리해 보다 안전한 시스템을 구현했다.

발생 가능한 오류에 대처하기 위해 오류 주체, 오류 원인, 오류 대응법으로 분류해 총 15 가지의 에러 상황을 설정해 이를 감지하고 대응하는 기능 로직을 구현했다. 이처럼, 수업에서 배운 TIM 등의 하드웨어와 SPI, I2C 통신 인터페이스를 활용함을 넘어 이를 응용해 RTOS(CMSIS V2) 환경에서 보호하고 안전하게 운용하는 시스템으로 확장했다.

이전 실습 혹은 이전 학기들에서의 프로젝트와 다르게 이번 프로젝트는 기획과 설계에 많은 시간을 할애했다. State 설계, Zone 내부 Task 구동 방식 설계, CAN 통신 및 SPI 통신 설계에 많은 기한을 사용했다. 설계도 위에서 코드 작업을 진행해 매우 수월하고 빠르게 작업을 마무리할 수 있었다. 또한, 중앙 제어기, Zone 제어기, Zone 산하 제어기가 분리되어 있기에 각각을 작업한 후 합치는 방식으로 작업해 팀원과 작업 분할에 유리했다.

해당 프로젝트에서 임베디드 시스템을 설계하며, RTOS 환경에서의 개발시 주의 사항으로 데이터 일관성, 동시성, 실시간성 등을 고려하며 설계하는 경험을 쌓았다. 또한, Task 간 데이터 및 상황 공유를 위해 (osMessageQueue)큐, (osThreadFlags)플래그 등을 도입하였다. 동시성을 고려해, 공유 자원인 SPI, CAN 에 접근은 오직 한 Task 에서만 가능하도록 설계하였다. 뿐만 아니라, Cortex M3 의 32bit

architecture 를 고려해 32bit 를 넘어가는 공유 전역 변수 업데이트는 해당 구간만 interrupt 를 off 해 락을 걸어 처리했다. 매우 짧은 업데이트를 위해 Mutex 의 오버헤드를 감수하는 것 보다, 매우 짧은 시간이기에 단순히 레지스터를 조작해 인터럽트를 enable/disable 시키는 것이 효율적이 판단했다.

시스템 전체가 FSMD 기반으로 동작하고, Zone 내부에서도 기능별 Task 가 구분되어 있고 디버깅 과정도 매우 원활했다. 예를 들어, 중앙 제어기의 로그에서 Zone1 의 초음파 데이터를 취득하지 못하는 문제에 대해 빠르게 Zone1 의 CAN 통신 -> SPI 통신 순으로 동작을 관찰했다. 통신 과정에서는 문제를 발견할 수 없어 산하의 초음파 제어기 STM32F103C8T6 파트를 관찰했고, 또 그 내부에서 main 의 루프가 정상 실행함을 확인해 결국 초음파 센서 문제로 특정지을 수 있었다. 다른 센서로 교체해도 문제가 동일해, 전원 문제를 의심했고 멀티미터로 확인한 결과 초음파 센서에 5V 보다 낮은 전압이 인가되고 있음을 확인했다. 이를 토대로 Zone1 자체의 전원을 분석한 결과, 전원 공급이 불안정해 보드가 계속 리셋되는 것이 원인임을 확인할 수 있었다. 따라서 전류 보충을 위한 0.1uF 바이패스 커패시터를 전원 연결 사이사이에 달고, 양단에는 220uF 을 달아 전압과 전류를 안정화시켜 보드에 전원이 확실히 공급되도록 해 문제를 해결했다.

또한, Zone2 가 OFF 상태에서 READY 가 되어도 의도와 다르게 Heartbeat 가 나가지 않고 CAN 통신이 먹통이 되는 것을 관찰했다. 이에 task 들의 우선 순위는 동일하니 starvation 문제는 제외시켰다. 따라서 Heart beat 를 송신하는 canTask 를 문제 원인으로 삼았고 로직을 관찰한 결과 fall-through 설계에 실수가 있어 Heart beat 송신이 되지 않는 것을 확인할 수 있었다.

마지막으로, SPI 에러를 가정하기 위한 패킷 오염 테스트에서 패킷 검증단에 문제가 있음을 발견했다. 함수 반환값인 플래그를 비교할 때, 플래그 비교에 &로 플래그 중첩을 고려하는 것이 익숙해 &를 사용했으나, SPI 의 타임아웃 에러 플래그가 0xFFFF 인 관계로 &를 사용했을 때 & osFlagsErrorTimeout 으로 처리하면 무조건 에러가 발생한다는 것을 고려하지 못했다. spiTask 내부를 관찰해 이를 잡아냈고, & 대신 ==를 도입해 오직 타임아웃 에러 플래그만을 잡아내도록 수정했다.

이외의 많은 문제들에 대해서도 이처럼 모듈, 그리고 기능별로 잘 분리가 되어있도록 설계해놨기에 디버깅이 매우 빠르고 쉽게 진행되었다. 보다 복잡한 시스템을 설계할 때, 표준에 따르거나 모듈화를 철저히 하는 것의 중요성을 배웠다.

아쉬운 점 역시 있다. Zone Architecture 에서 Zone 제어기들 간의 통신은 배선 용이성과 무게 감소로 인한 연비 향상 이점을 위해 CAN 통신을 주로 사용한다고 배웠다. 따라서 CAN2.0 을 실습해보고자 CAN 통신으로 중앙 제어기와 Zone 들 간의

통신을 구현했다. 하지만, Zonal Architecture 에서는 Zone 제어기와 중앙 제어기간의 통신은 Ethernet 으로 구현하나, Ethernet 모듈을 별도로 구매하기에 부담스러워 포기한 점은 아쉬운 점이다. 또한, 해당 프로그램에서 에러 대응은 주로 독립 처리로 구현했다. 자동차에서 에러 복구를 시도해 시간을 소모하는 것 보다 곧바로 비상 정지에 나서거나 아예 시스템에서 방출하는 것이 옳다고 생각했다. 하지만, 초음파 센서나 가속도 센서 등은 분리 후 복구를 기다리면 되었는데 단순히 분리 처리만 해서 아쉬움이 남는다. 또한 비용과 기한 문제로 카메라나 라이다 센서를 넣지 못했다. 이후 해당 임베디드 시스템의 완성도를 높이고 대량의 데이터를 시스템에서 처리해보는 경험을 위해 별도로 카메라 모듈을 Raspberry Pi zero 2W 에 추가 연결해 EMERGENCY 조건으로 추가해보고자 한다.

그럼에도 결과적으로, 목표한 기능이 STATE 가 전환되가며 동작함을 확인했기에, 프로젝트의 목표를 달성했다고 판단한다.

9. 참고 문헌

1) STM32F1 시리즈 HAL & LL DRIVER

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.st.com/resource/en/user_manual/um1850-description-of-stm32f1-hal-and-lowlayer-drivers-stmicroelectronics.pdf

2) STM32F103C8T6 Datasheet

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.st.com/resource/en/datasheet/stm32f103c8.pdf

3) 그 외 STM32 문서

https://www.st.com/en/microcontrollers-microprocessors/stm32f103/documentation.html

4) HC-SR04 Datasheet

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

5) ADXL345 Datasheet

https://www.alldatasheet.co.kr/datasheet-pdf/view/254714/AD/ADXL345.html

6) TB6612FNG Datasheet

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://cdn.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf