

# 대학생 자작자동차대회 기술부문 신청서



학교명		아주대학교		팀명		A-FA	
지도교수		전용호		팀장		김병찬	
차량	명칭	A12					
	제원	길이	mm × 폭	mm × 높이	mm (무게	kg)	
부문		<input checked="" type="checkbox"/> 기술아이디어 부문 <input type="checkbox"/> 디자인 부문					

상기 참가팀은 KSAE 대학생 자작자동차대회의 기술부문 (기술아이디어 및 디자인) 참가를 신청하며 다음과 같이 보고서를 제출합니다.

년 월 일

대학생 자작자동차대회 조직위원회 귀중

# 대학생 자작자동차대회 기술아이디어 보고서



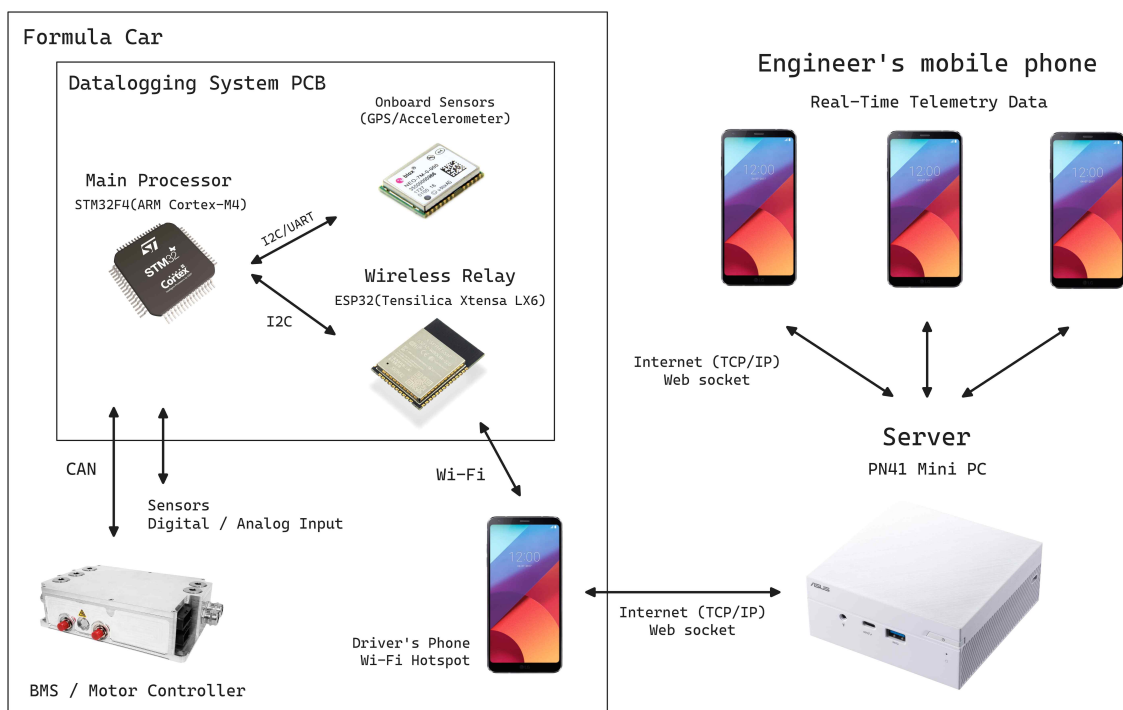
## ■ Summary

본 기술아이디어는 E-Formula 경주 차량의 주행 데이터 수집 및 실시간 원격 모니터링을 위한 임베디드 하드웨어 및 소프트웨어 시스템이다.

KSAE 출전과 같이 특수한 목적을 위해 개발하는 프로토타입 차량들은 제작 후 주행 성능 평가가 개발 과정에서 굉장히 큰 비중을 차지한다. 또한, 주행 중 예상치 못한 상황과 사고가 발생할 확률이 양산 차량보다 훨씬 높다. 이러한 이유로 성능 평가, 사고 원인 분석 및 엔지니어링 등을 위해 차량의 주행 데이터 수집은 반드시 필요하다. 그러나, 현재 찾아볼 수 있는 차량 데이터로깅 장비들은 그 성능과 기능이 파편화되어 있어 하나의 장비로 개발팀의 모든 요구사항을 만족시키기 어렵다. 따라서, 하나의 데이터로깅 플랫폼에서 CAN 트래픽, Real-Time Clock, 아날로그 / 디지털 입력, Pulse Capture, 가속도, GPS 정보 등 기술팀이 필요로 하는 모든 계측 데이터를 제공하여 이러한 문제를 해결할 수 있도록 한다.

또한, 본 대회처럼 사고 위험이 상존하는 주행 환경에서는 SD카드 소실, 데이터 유실 등으로 데이터에 접근할 수 없는 상황이 발생하기 쉽다. 이러한 상황에서도 데이터를 사후에 분석할 수 있도록 수집 즉시 무선으로 서버에 전송하여 데이터 확보를 보장한다. 서버는 전송된 데이터를 DB에 저장하는 동시에 웹 클라이언트에 중계하여 기술팀이 주행 중에도 차량의 데이터를 실시간으로 확인할 수 있도록 한다.

## ■ Conceptual Design



별도로 설계한 PCB를 차량에 탑재하여 차량에서 각종 정보를 수집하고, 이를 SD카드에 저장하는 동시에 무선 통신으로 서버에 전송한다. 드라이버는 차량 탑승 시 Wi-Fi Hotspot 기능을 활성화한 자신의 스마트폰을 휴대하여 무선 전송이 가능하도록 인터넷 연결을 제공한다. 서버는 수신한 데이터를 해석하여 데이터베이스에 저장하며, 웹 클라이언트에 중계한다. 웹 클라이언트는 데이터를 그래픽 인터페이스를 제공하여 기술팀 엔지니어가 데이터를 한 눈에 손쉽게 확인할 수 있도록 하며, 언제 어디서나 접속할 수 있도록 인터넷 웹 페이지를 제공한다.

## ■ Preliminary Design

하나의 데이터로깅 플랫폼으로 차량에서 다음과 같은 기능을 수행한다.

1. CAN 버스 트래픽 모니터링 (배터리 관리 시스템(BMS) 및 모터 컨트롤러 데이터 수집)
2. 6채널 디지털 입력 신호 모니터링 (BMS / BSPD / IMD 오류, HV 활성화, RTD 활성화 신호 수집)
3. 4채널 아날로그 입력 신호 모니터링 (서스펜션 변위 측정 센서 신호 등 수집)
4. 4채널 디지털 펄스 모니터링 (휠 회전 속도 측정 센서 신호 등 수집)
5. 3축 가속도 변화 모니터링
6. GPS 위치 정보 모니터링
7. 자체 CPU 온도 모니터링
8. 무선 시간 동기화 및 RTC(Real-Time Clock)을 통한 시간 정보 유지
9. 드라이버 계기판 LCD 시현 (CAN으로 수집한 속도 및 배터리 잔량 정보 등)
10. 수집한 데이터의 SD카드 로깅
11. 수집한 데이터의 무선 인터넷 전송

각 데이터는 최소 1Hz에서 최대 125Hz 주기로 생성된다. 종합하면 매 초에 약 120개의 로그가 발생한다. 이를 모두 놓치지 않고 기록하기 위해 고성능 프로세서를 사용한다.

## ■ Detail Design

메인 프로세서는 초당 수백 개씩 발생하는 로그를 모두 놓치지 않고 기록할 수 있도록 168MHz 클럭으로 동작하는 32-bit MCU인 STM32F4를 채택하였다. 또한, 펌웨어를 C로 작성하여 시스템 메모리에 바로 접근해 메모리 관리 오버헤드를 최대한 제거하였다.

메인 프로세서가 수집한 데이터는 I2C 통신을 통해 무선통신용 MCU인 ESP32로 전송된다. 기존에는 115200bps UART 통신을 통해 ASCII 문자열로 구성된 수백 Byte 크기의 로그를 전송하였다. 그러나 로그 발생량이 많고 UART 통신 속도가 느려 STM32와 ESP32 간 통신에서 병목이 심하게 발생하였다. 이에 로그 프로토콜을 별도로 정의하여 로그의 크기를 개당 16 Byte로 고정하였으며, I2C 400kHz Fast-Mode 통신을 통해 전송률을 3배 이상 개선하였다. 이를 통하여 초당 2500개 이상의 로그를 전송할 수 있는 대역폭을 확보하였다.

로그 구조는 다음과 같다. 로그 프로토콜의 세부 스펙은 [여기](#)에서 확인할 수 있다.

(<https://github.com/luftaquila/a-fa-telemetry/blob/master/device/ECU%20v2/Core/Inc/logger.h>)

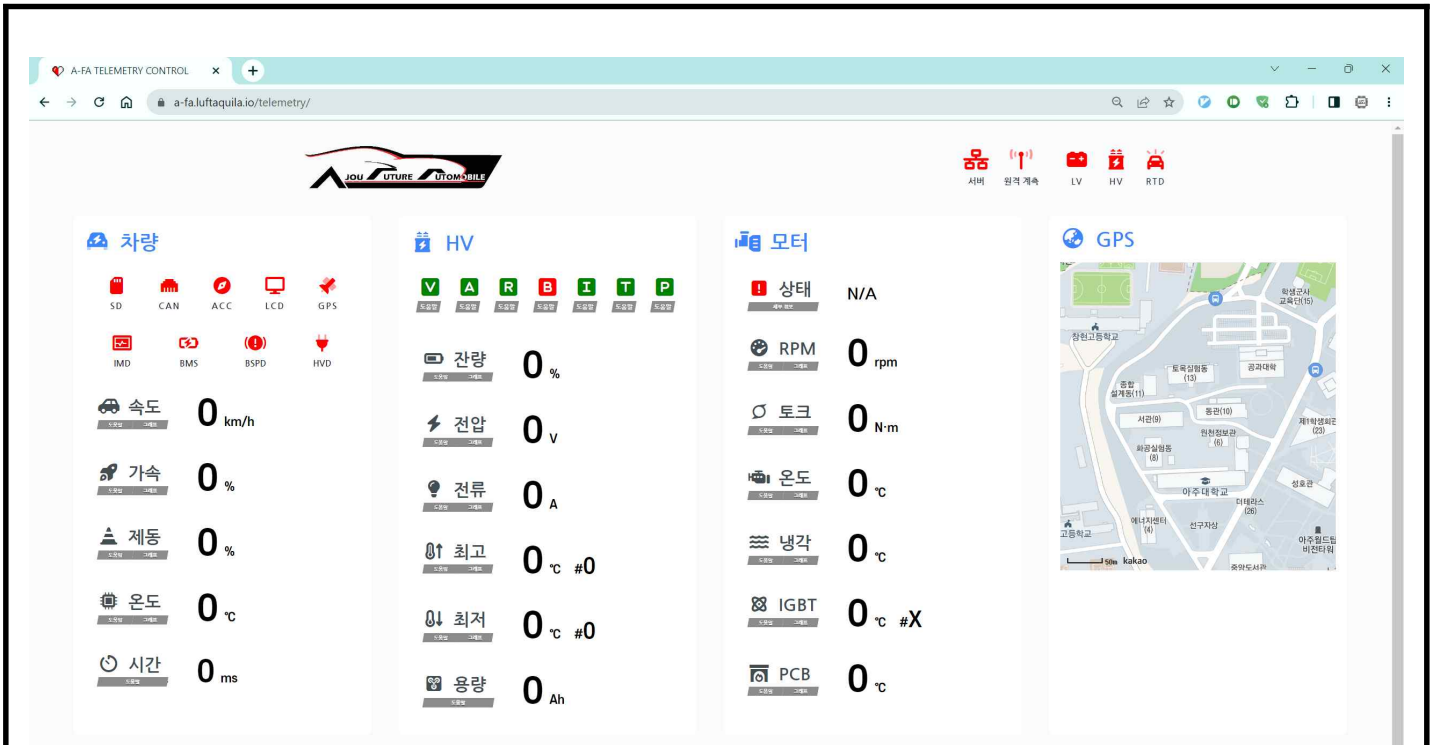
```
1 typedef struct {
2     uint32_t timestamp;
3     uint8_t level;
4     uint8_t source;
5     uint8_t key;
6     uint8_t _reserved;
7     uint8_t value[8];
8 } LOG;
```

ESP32는 드라이버가 휴대하고 탑승한 스마트폰의 Wi-Fi 핫스팟 네트워크를 통해 무선으로 인터넷에 연결된다. STM32 프로세서로부터 I2C를 통해 전달받은 로그를 서버로 전달한다. 통신에는 웹 소켓을 사용하여 HTTP 사용 시 발생하는 매 transaction에서의 TCP handshake 오버헤드 없이 실시간 양방향 통신을 수행한다. 연결 실패나 통신 지연이 많이 발생할 수 있는 Wi-Fi 무선 통신의 특성을 고려하여, 무선 연결 기능이 메인 프로세서의 데이터로깅을 방해하지 않도록 메인 프로세서와 무선통신만을 담당하는 MCU를 분리하여 구현하였다.

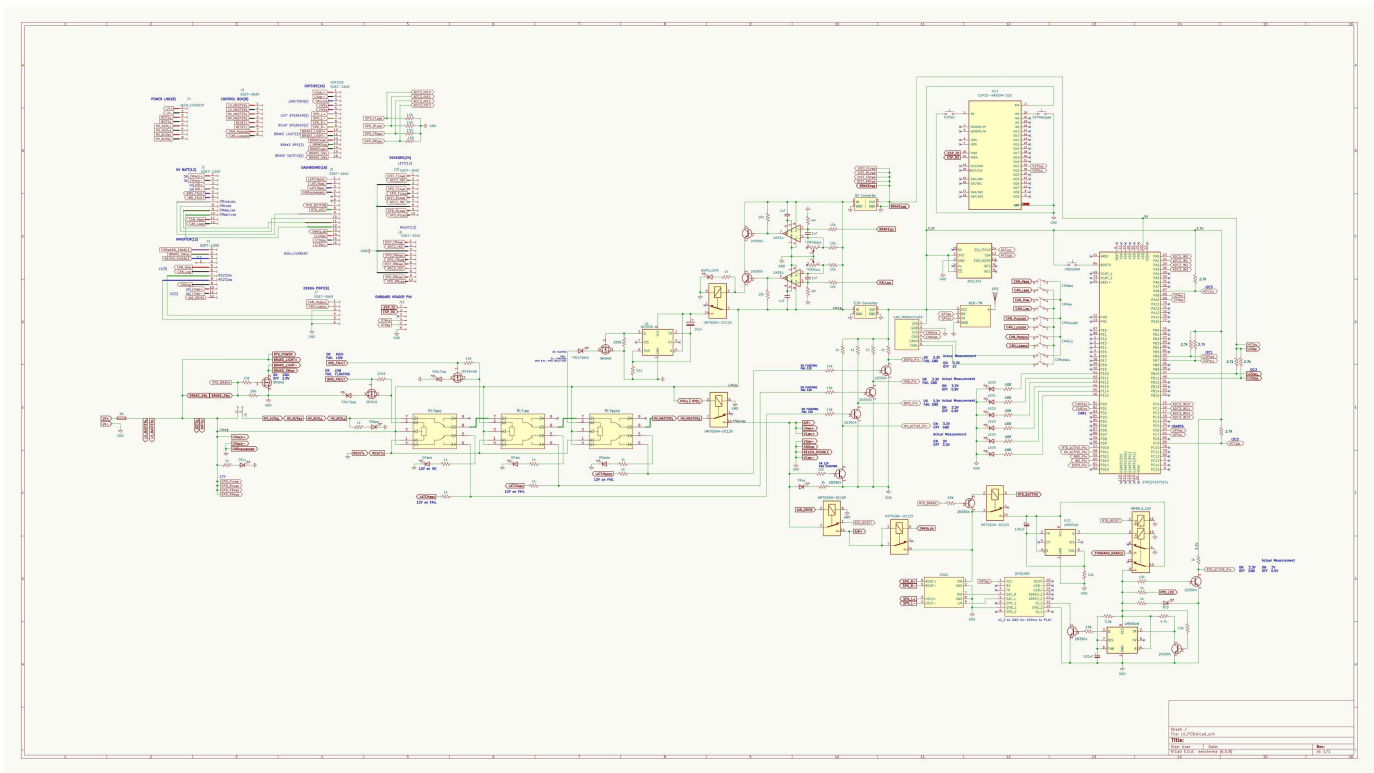
ESP32 또한 240MHz로 동작하는 고성능 2코어 프로세서로, 복잡한 WLAN 드라이버를 구동하여 로그 트래픽을 고속으로 발생시킬 수 있는 성능을 갖추고 있다. 또한, 펌웨어의 메인 코드를 실행하는 코어와 I2C 및 WLAN 수신 인터럽트 이벤트가 발생하는 코어를 분리하여 처리 성능을 극대화하였다. 펌웨어는 C++로 작성하였다.

이러한 16 Byte의 로그는 바이너리 형식으로 사람이 읽을 수 없다. 서버는 이러한 로그 메시지를 수신하여 프로토콜에 맞게 해석하고 데이터베이스에 저장한다. 또한, 서버에 연결된 웹 클라이언트에 수신한 로그를 중계한다. 서버 코드는 Node.js로 작성하였으며, Ubuntu Server 22.04 LTS 환경의 홈 서버에서 실행된다. 또한, 서버 시간을 임베디드 장비에 전달하여 메인 프로세서의 RTC 유닛을 동기화한다. 이를 통하여 STM32가 항상 최신 시간을 유지할 수 있도록 해 로그의 활용성을 높인다.

마지막으로 웹 클라이언트는 위의 서버와 동일한 홈 서버에서 실행되는 NGINX 웹 서버가 서비스한다. 웹 클라이언트는 [URL\(https://a-fa.luftaquila.io/telemetry/\)](https://a-fa.luftaquila.io/telemetry/)을 통하여 기술팀 각 인원이 자신의 스마트폰으로 접근할 수 있으며, 주행 중 실시간으로 확인해야 하는 차량의 주요 지표를 시현한다. 최근 1분 추세 그래프를 제공하여 기술팀이 차량의 데이터 변화를 쉽게 파악할 수 있도록 시각화한다. 또한, 주행 후 SD 카드에 저장된 바이너리 형식의 데이터를 해석하여 읽을 수 있는 JSON, CSV 형식으로 변환하고 다운로드하는 기능도 제공한다. 사용자 인터페이스는 다음과 같다.

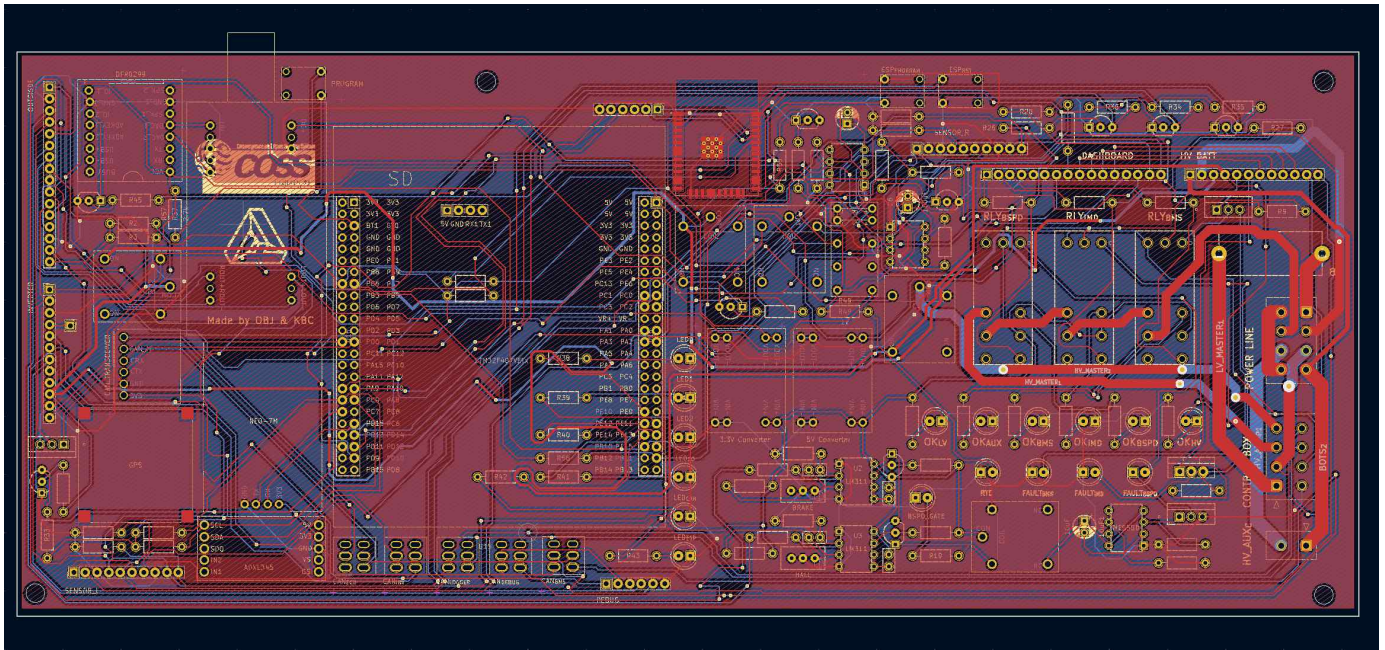


프로세서 및 센서는 문제가 생겼을 경우 쉽게 교체할 수 있도록 헤더 핀을 사용하여 PCB에서 탈부착할 수 있도록 설계하였다. PCB는 KICAD로 설계하였으며, 차량에서 차지하는 부피 및 배선을 줄이기 위해 차량의 차단 회로 등과 같은 PCB 상에 배치하였다. 설계한 회로는 다음과 같다.



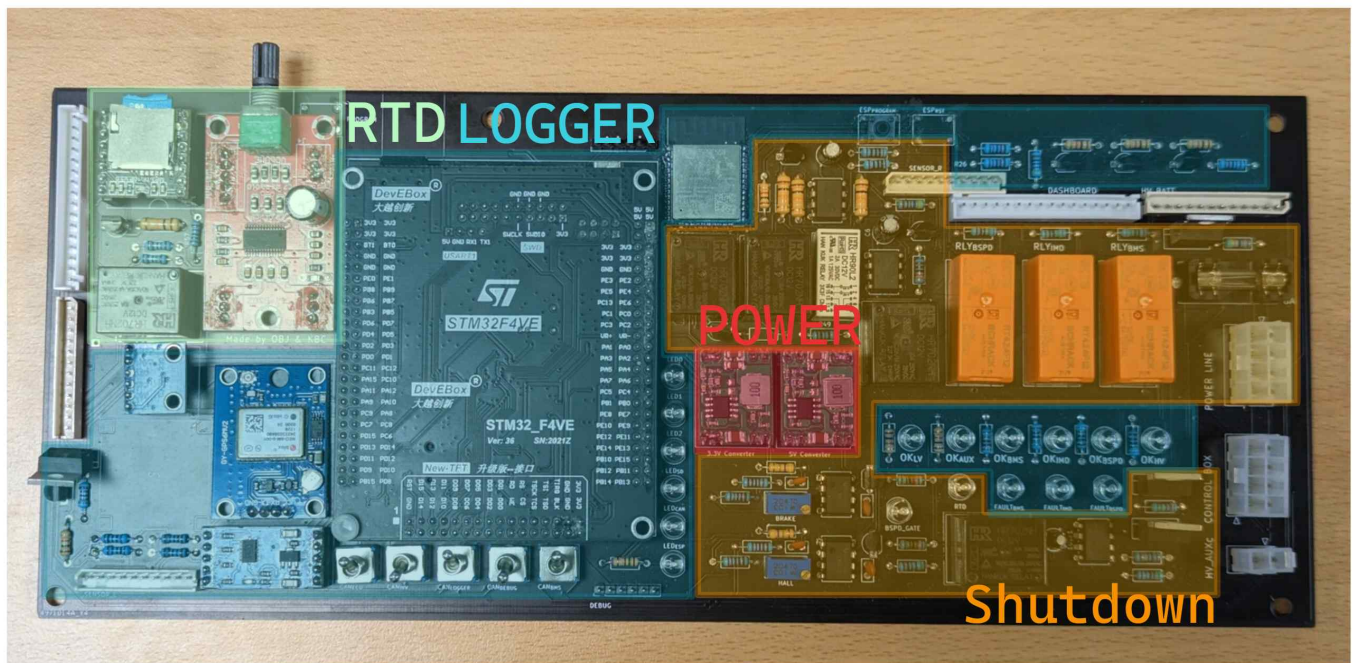


회로를 바탕으로 설계한 PCB 아트워크는 다음과 같다. PCB는 FR4 2-layer로 설계하였다.



## ■ Manufacturing

앞서 설계한 PCB에 부품을 배치하고 납땜하여 제작한 PCB는 다음과 같다. 본 시스템이 차지하는 영역은 LOGGER로 표시된 청색 영역이다. 차량의 모든 CAN 통신선은 본 PCB로 모여 버스를 형성한다. 각 통신 채널마다 토글 스위치를 부착하여 원하는 CAN 채널만 켤 수 있도록 설계하였다.

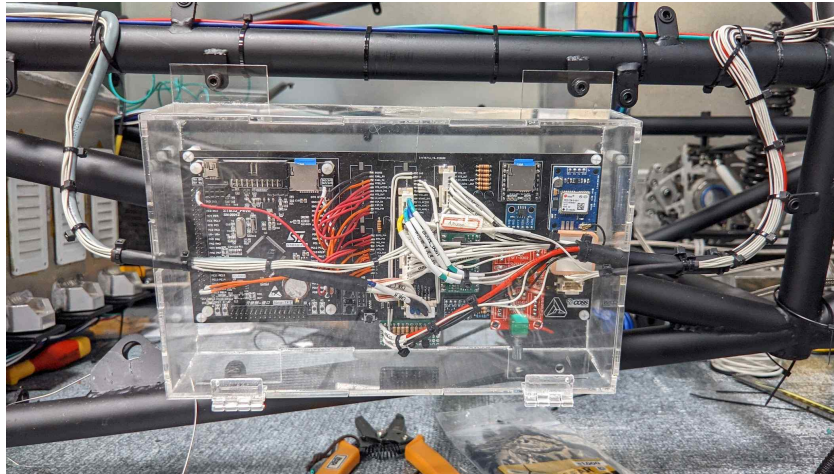


## ■ Conclusion

데이터로깅 자체는 이미 많은 팀들이 필요성을 느끼고 각자 상용품 또는 자체 개발한 데이터로거를 사용하고 있다. 그러나 본 기술아이디어는 하나의 플랫폼에서 CAN, 아날로그 및 디지털, 펄스 등 외부 신호를 수집하며, 자체 GPS 및 가속도 센서, RTC 기능을 지원하고 수집한 로그를 SD카드에 저장하는 동시에 무선으로 송신하여 기술팀 인원 전체가 자신의 스마트폰으로 실시간 주행 정보를 확인할 수 있는 강력한 기능을 제공하여 기존의 데이터로깅 시스템들과 차별점을 두었다. 또한, 개발한 임베디드 펌웨어, 서버 및 클라이언트 소스코드 등을 모두 공개하여 오픈소스 생태계에 기여하고 다른 팀에서도 자유롭게 사용할 수 있도록 하였다. (<https://github.com/luftaquila/a-fa-telemetry>)

## ■ Acknowledgments

본 시스템은 올해 개발중인 차량에 탑재될 예정으로, 아직 완성된 차량에 탑재되어 주행하지 않았다. 그러나, 올해 개선 사항을 반영하지 않은 기존 시스템은 차량에 탑재되어 실제 주행에 사용하였다.



본 기술아이디어의 실제 적용 영상은 다음과 같다.

1. 시스템 작동 영상 1 (주행성능시험): <https://www.youtube.com/watch?v=kZxLHQo4KEE>
2. 시스템 작동 영상 2 (주행성능시험): <https://www.youtube.com/watch?v=0vMqdhWIWpU>
3. 시스템 작동 영상 3 (모터 구동 시험): <https://www.youtube.com/watch?v=ZjLUNqeKCdc>
4. 시스템 작동 영상 4 (모터 구동 시험): <https://www.youtube.com/watch?v=fWfDaCTimpo>

## ■ References

1. [RM0090 STM32F4 advanced Arm®-based 32-bit MCUs Reference Manual](#), STMicroelectronics.
2. [ESP32 Technical Reference Manual Version 4.9](#), Espressif Systems, 2023.
3. [ADXL345 Digital Accelerometer Datasheet Rev. G](#), Analog Devices, Inc., 2022.
4. [NEO-7 u-blox 7 GNSS modules Data Sheet](#), U-BLOX.
5. [Introduction to the Controller Area Network\(CAN\) Application Report](#), Texas Instruments, 2016.