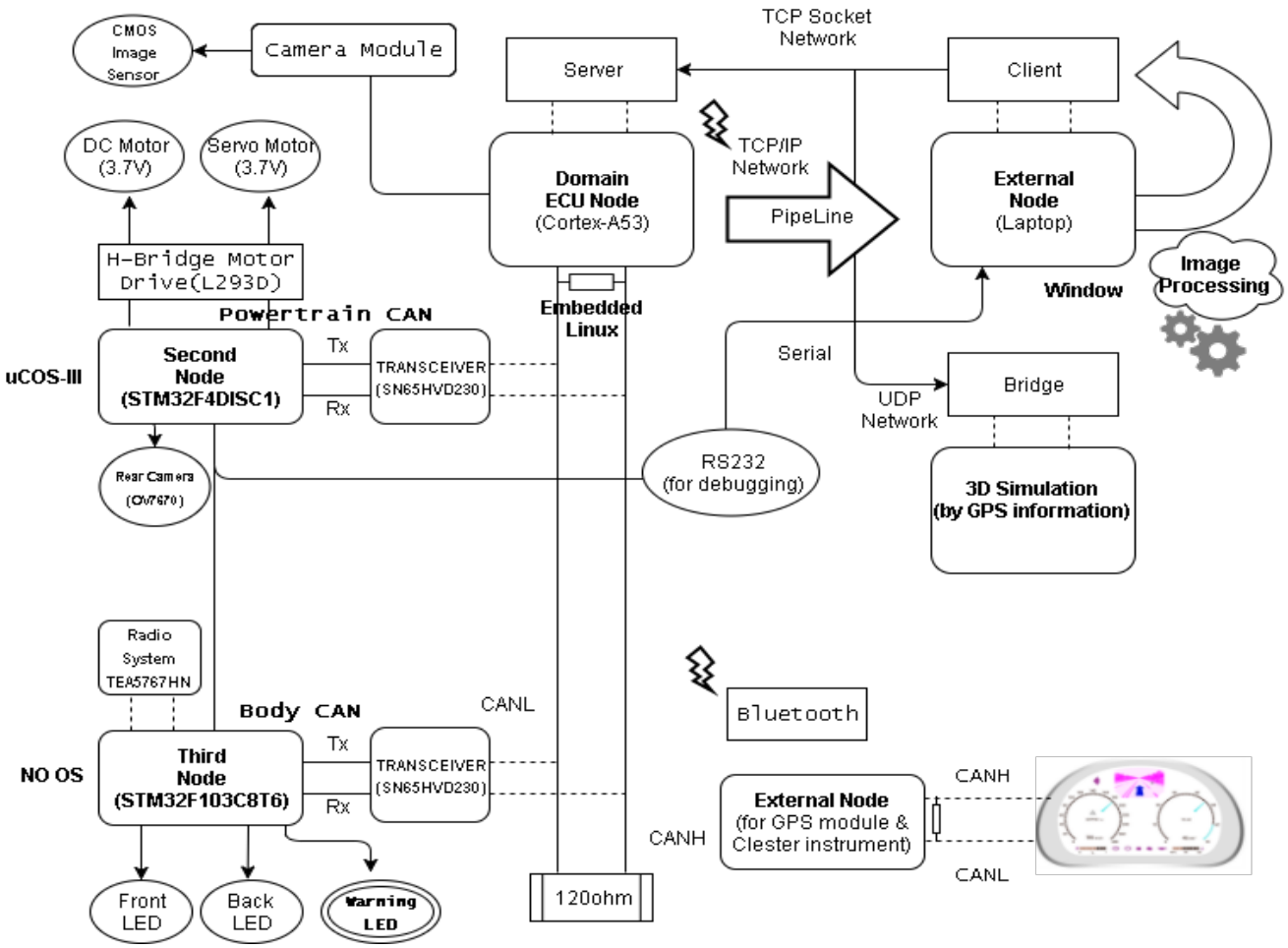
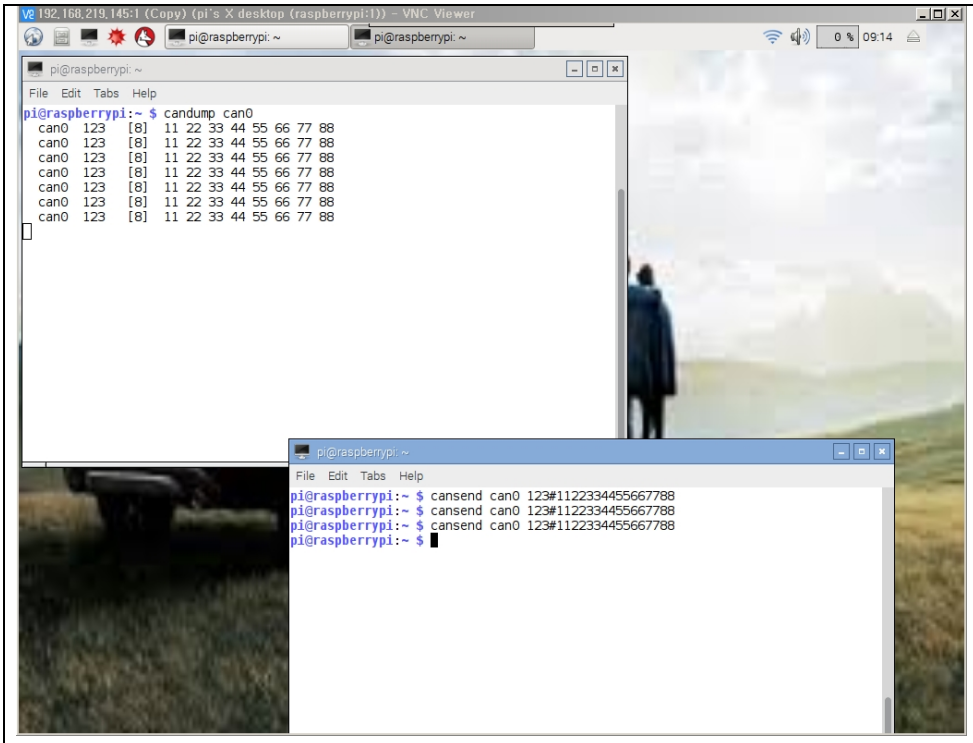
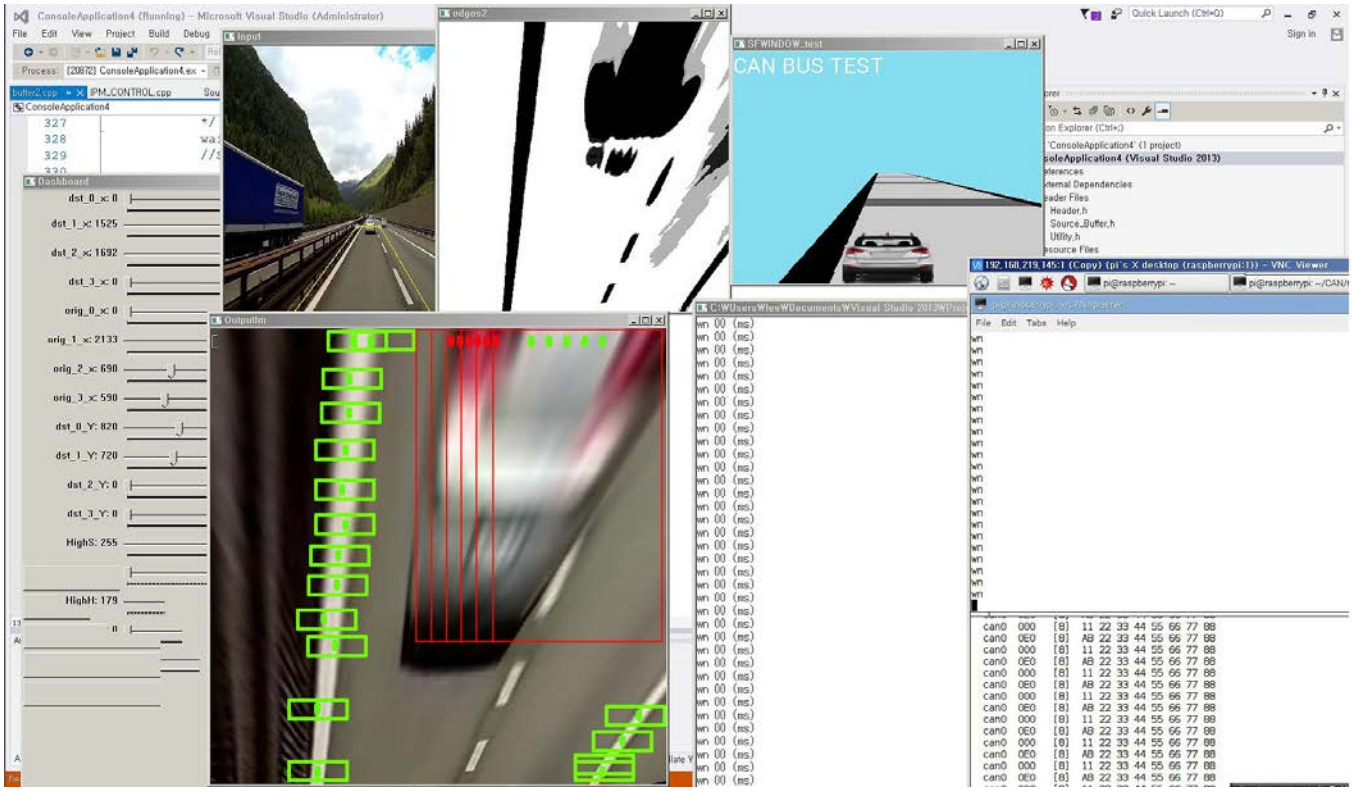
포트폴리오(수행기간 : 2016.11~2017.06) – **시연 동영상 및 소스코드 tinyurl.com/control-can**	
수행목적	**임베디드환경을 구축하는 법을 배우고, C/C++ ARM 그리고 RTOS 리눅스 관련 일을 수행하기 위해, 제일 기초적인 사항들을 공부하는 데 있어서 흥미를 돋우고, 또 목표가 있으면 좀 더 학습 효율이 생길 것 같아 프로젝트를 하게 되었습니다.**
개발목표	모의 차량에 CMOS 이미지 센서를 통한 입력받은 바이너리를 임베디드 리눅스 환경에서 프로세싱 후 다른 프로세스 노드로 네트워크 전송을 수행합니다.(C 언어 및 리눅스 환경을 배우기 위한 작업) 그 데이터를 바탕으로 모의차량을 CAN 통신을 통해 제어하기 위해 임베디드신호를 다시 서버로 전송합니다. 동시에 자동차 인포테인먼트 시뮬레이션을 구축합니다. (C/C++를 배우기 위한 작업) 마지막으로 STM 보드와 ATMEGA 보드를 통해 CAN 통신 기반으로 모터 및 LED 의 동작을 정의합니다. (C 언어 및 ARM 프로세서/RTOS 를 공부하기 위한 작업)
개발환경	C/C++언어, TCP/UDP 프로토콜, CAN 프로토콜, I2C(라디오), RS232(터미널-디버깅), ATMEGA328, CORTEX-M4, M3, A53, µC/OS-III LOGIC-ANALYZER, KEIL, IAR, OPENCV(이미지 처리), GSTREAMER, ETC.
구현기능	1. 카메라모듈로부터 받은 이미지 바이너리를 성능이 높은 Laptop 에서 처리하기 위해 TCP 프로토콜을 통해 비압축 전송하였습니다 2. TCP 로 받은 데이터를, 실시간으로(latency 100ms 이하) 처리를 할 수 있도록 재가공하였습니다. 3. 재가공된 데이터를 CAN 통신 하기 전 데이터처리 (Canny, Hough, bird eyes view, Ipm inversing 등) 4. 이미지처리 후 해당하는 CAN 신호 발생 (Accel, Break, Warning, Left, Right 등)후 3D 시뮬레이션과 연동 및 MATLAB 을 통해 SAFETY SYSTEM 구현 및 검증하였습니다. 5. 다른 프로세스에서는 C++ SFML 프레임워크를 통한 3D reconstruction.(현재위치, 주변차량 복원 등) 6. 도메인 컨트롤 유닛을 기반으로 각각의 프로세서 Component 들을 제어. CAN 통신(통신모듈 MCP2515 MCP2551 SN65HVD230 등 사용)을 통해 2 개의 모터(for Steering & Rear), LED 등을 제어 (각각의 노드들은 CanTX 및 RX 출력 및 트랜시버 통한 CANH 와 CANL 배선 위에서 통신하게 함) 추가적인 기타 라디오, GPS(WAVE 모사를 위한 준비중) 7. 도메인 Node-» External Node(Atmega) 블루투스를 통해 최종적으로 RPM, 좌, 우회전 등의 데이터를 받아 CAN 통신으로 계기판에 출력하게 하였습니다.





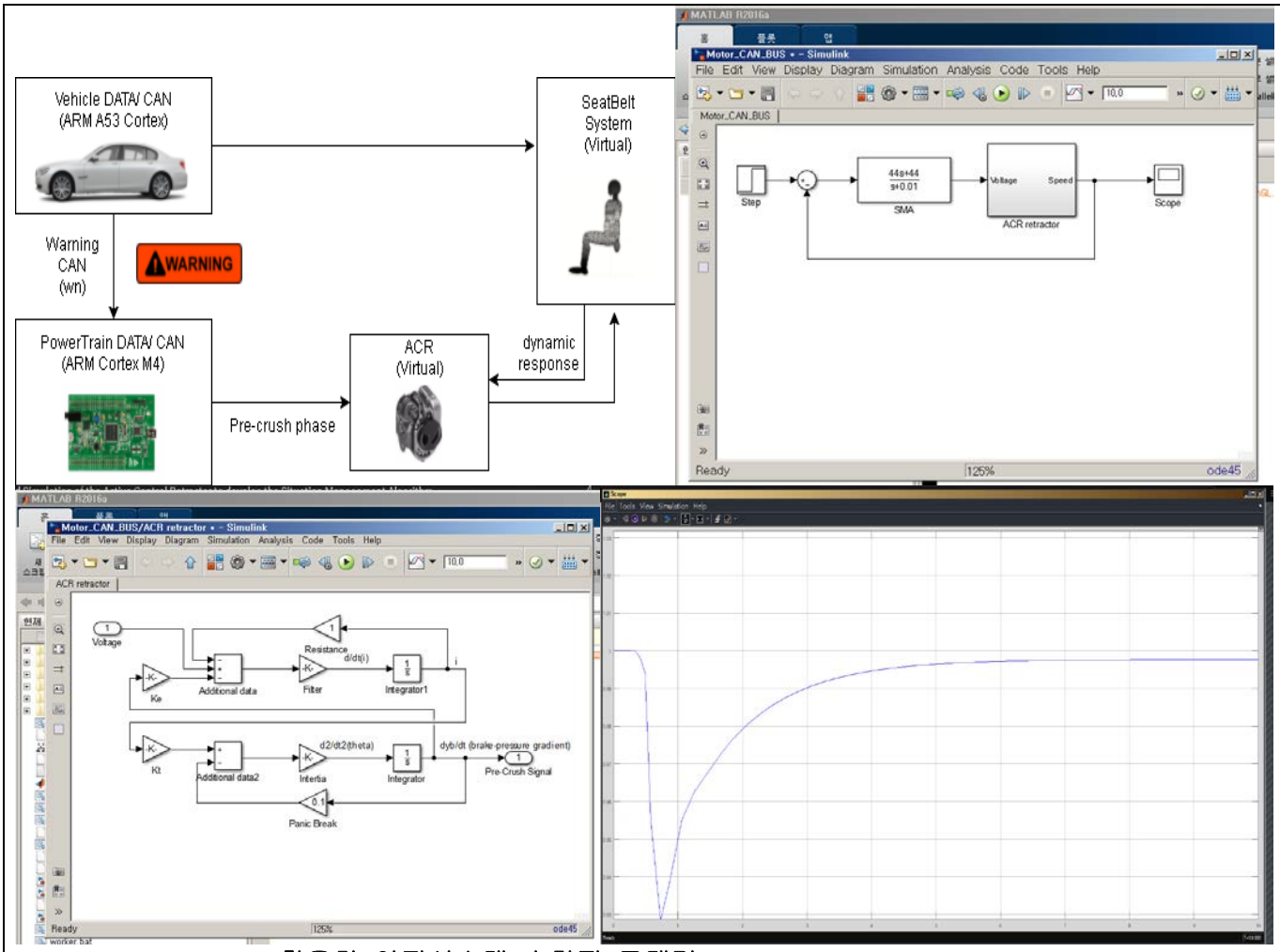
[Linux A53 ARM Processor]

- 최소한 vi 에디터, 동,정적 라이브러리와 MAKEFILE 로 프로그램 빌드 하는 법 만큼은 실전에서 잘 쓸 수 있도록 숙달하고자 노력하였습니다. 또 드라이버 트리 보는법과 다른 호스트(우분투)에서 재빌드하여 타겟에 커널을 올리는 방법 등을 배웠습니다. 폭스바겐사에서 제공하는 Socket-CAN 라이브러리를 바탕으로 부족한 실력으로 임베디드 리눅스 환경에서 CAN 통신을 위한 프로그램(TCP 서버에 CAN 라이브러리 포팅후 ARM 레지스터를 통한 GPIO 제어 등)을 만들어 사용하였습니다. (위의 화면은 LOOPBACK 테스트)



[접근 경고 CAN SIGNAL GENERATING, IMAGE PROCESSING]

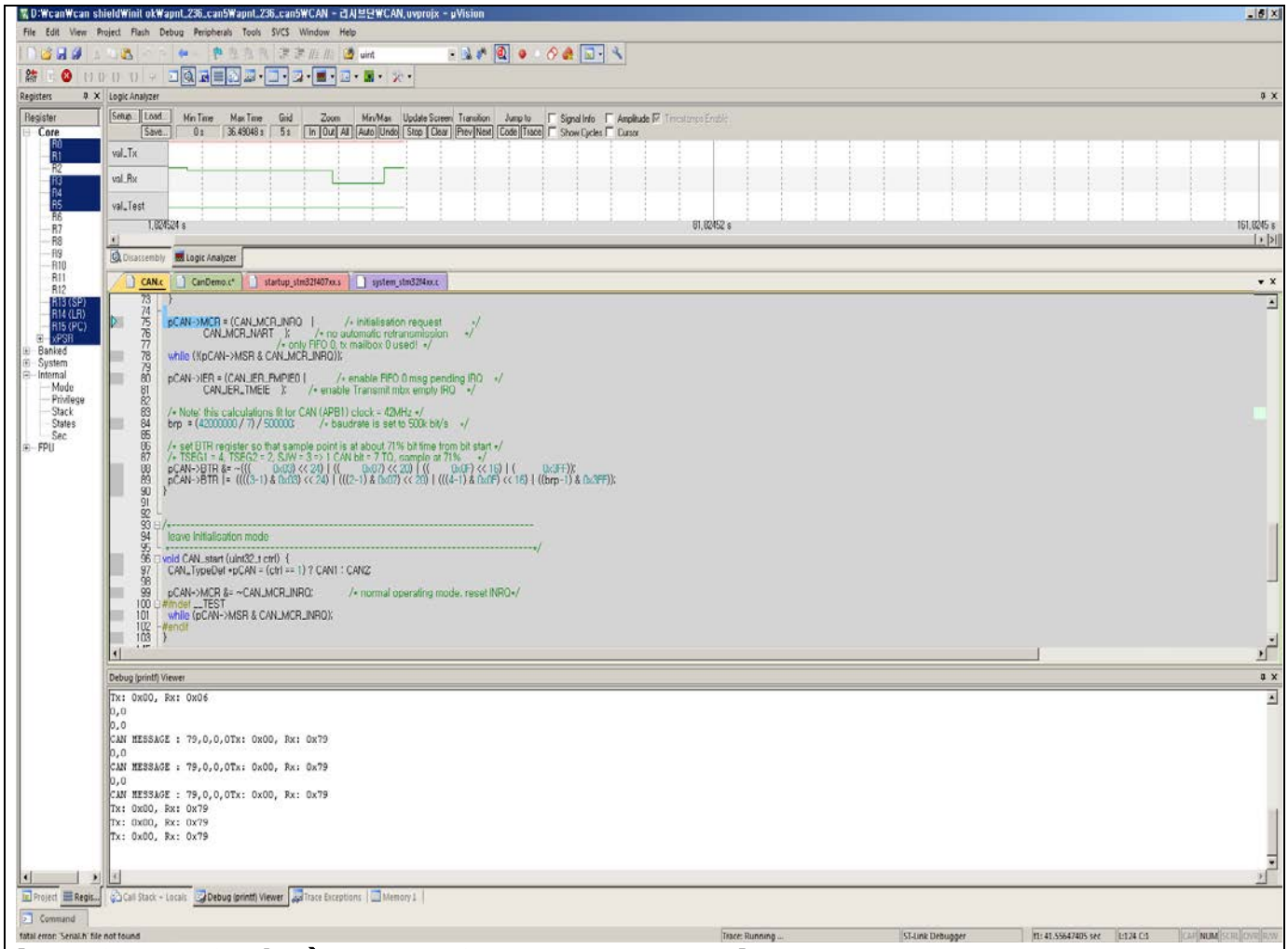
- 최소한의 C++에서 기초가 되는 클래스를 통한 객체화, 스마트 포인터 만큼은 잘해보자 마음먹고, 이 부분에 집중적으로 초점을 맞추고 사용법을 숙달하였고 또한, 이미지 프로세싱을 통해 WN(warning can id=0x0E0)신호가 발생을 통한 현실상의 WARNING LED 작동 등을 하게 하였습니다.



[MATLAB Simulink 활용한 안전시스템 수학적 모델링]

- MATLAB 을 배우고 싶었으나, 관련 지식이 전무 하였기 때문에 해당 내용이 포함된 해외 논문을 참고하였습니다.
(http://ac.els-cdn.com/S1474667015342543/1-s2.0-S1474667015342543-main.pdf?_tid=84f5e95e6b4d11e7882200000aacb361&acdnat=1500336814_f43fca986275ecb5871279de867e90b9)

CAN 통신으로 WN(WARNING) 신호가 발생하였을 때의 상황을 MATLAB SIMULINK 로 모사하였습니다.
우선 도메인 ECU 에서 WN 신호가 CANBUS 로 들어가면 PowerTrain Node 인 CortexM4 에서는 CAN id 필터링을 통해 id = 0E0(위험신호)를 필터링하게 되고 관련 신호를 ACR 에 전달하며 Pre-crush phase 로 들어갈 수 있게 합니다. 즉 Active Control Retractor 장치를 Retractor 하여 부상을 방지하는 시뮬레이션을 모사하였습니다.



[Domain CAN Signal → Powertrain CAN & BODY CAN]

- 베이스는 ST 사에서 제공해주는 예제소스를 참고하였습니다. 데이터 시트를 참고하여 CAN 통신 펌웨어프로그램을 인터럽트 기반으론 구현해보고자 하였지만 실패하였고 하지만 포기하지 않고, 수많은 디버깅과 시행착오를 거쳐, 폴링기반으로는 구현에 성공하였습니다. M3 는 M4 보드같이 디버깅용 STlink 가 달려있지 않아, 소형보드용 j link 를 통해 디버깅하였습니다. Keil 사에서 제공해주는 디버깅 툴을 이용해 PC 상에서 로직, 인터럽트 레지스터 변화 등을 확인할 수 있었습니다. Domain CAN 에서 뿌려주는 CAN 신호 중에서 필요한 신호를 Filtering 하여 각각 기능에 맞게 GPIO 를 통해 LED 를 키거나 모터를 동작시켜 주는 것과 같은 기능상의 역할을 분담하였습니다.

[Wireless(Bluetooth) → Atmega328 (CANH, CANL) → Cluster Instrument]

- 마지막으로 Domain 노드 에서 -> External (Atmega328)로 블루투스 를 통해 신호가 들어올경우 External processor 가 프로세싱을 통해 적절한 id 를 발생시켜 계기판을 시뮬레이션 하였습니다.

사실 이 분야에 업무경험이 없어 코드 디자인이 미숙하고,또 제가 직접 작성하기 보단 인터넷 상에 존재하는 라이브러리들을 대부분 참고하였기 때문에, 기술의 수준 및 깊이도 많이 낮습니다.
하지만 프로젝트를 하면서 항상 기본에만 충실하자는 생각으로 [리눅스/C/C++] 기본 능력을 배양하는데 초점을 맞추었고, 이러한 튼튼한 기본을 바탕으로 저의 장점인 끈기와 열정을 가지고 귀사에서 업무에 지장이 없도록 충실히 노력을 하여 성과를 낼수 있는 항상 기본을 중요시하는 노력형 인재가 되겠습니다.