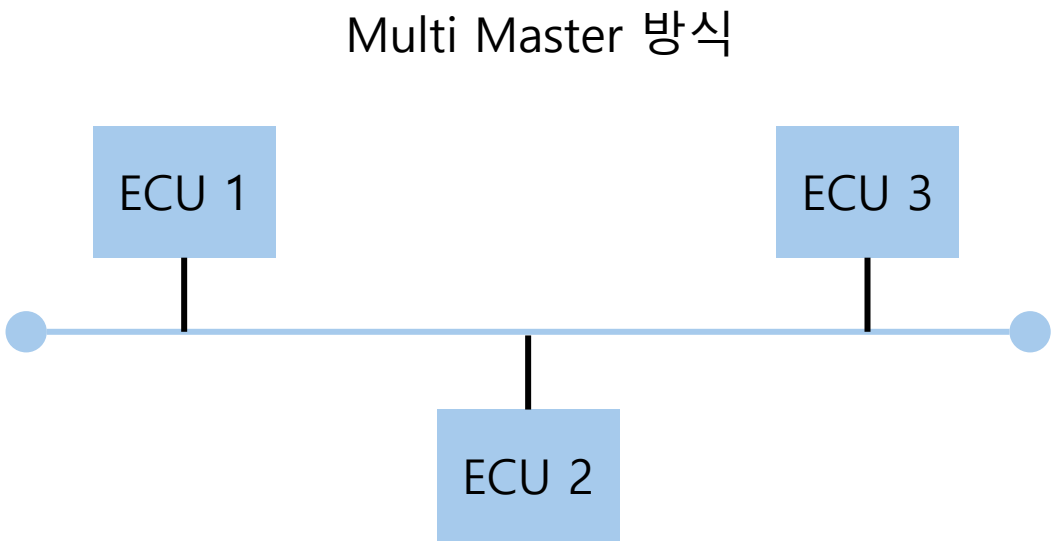
차량 CAN 통신



CAN(Controller Area Network) 통신

- 여러 개의 MCU나 장치들의 통신을 위해 설계된 표준
  통신 규격
- CAN 버스를 통해 데이터를 버스에 보내도 ECU들은
  필요한 데이터에 접근할 수 있다

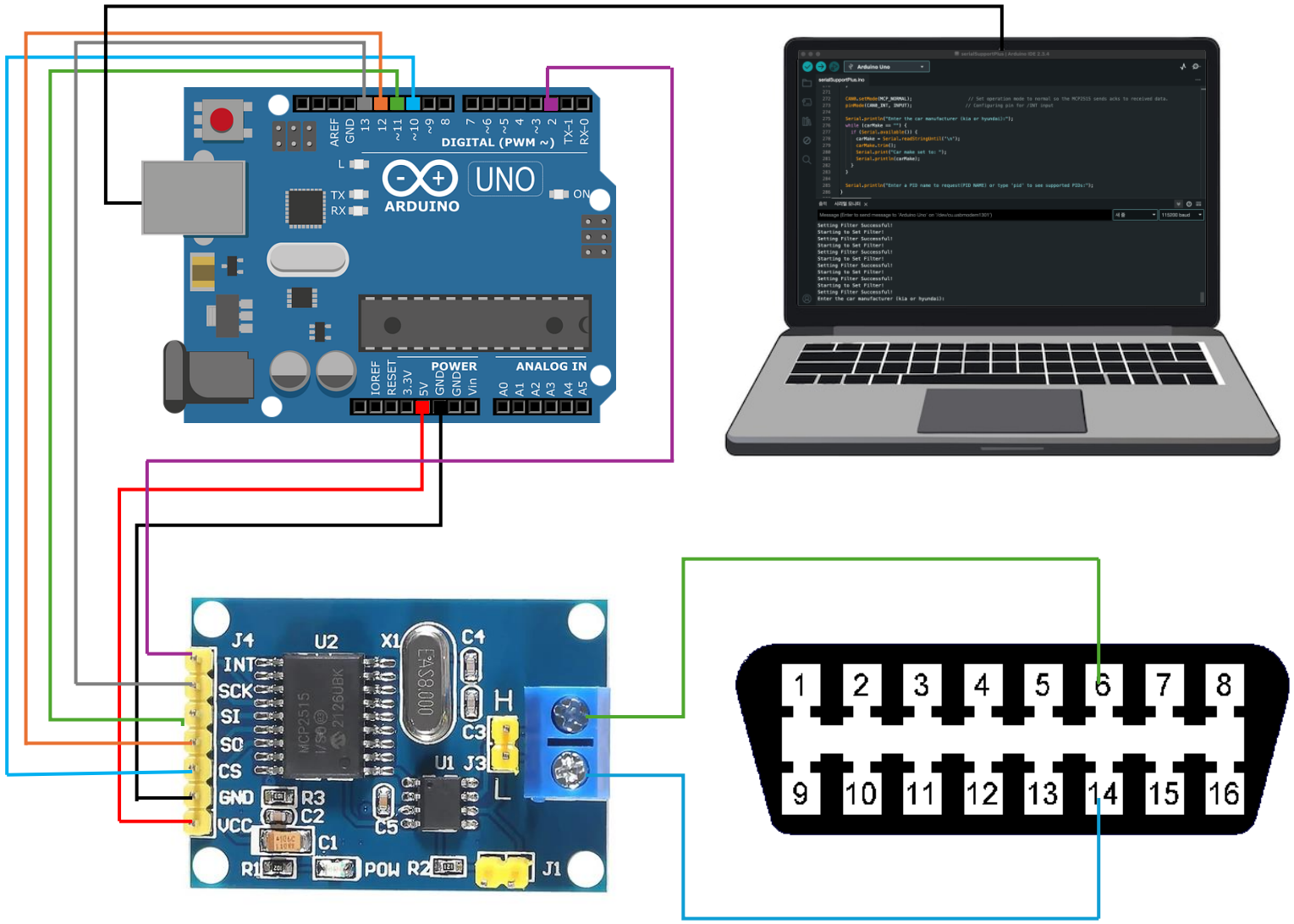
→ 1:1 통신 방식보다 가볍고 효율적인 제어 가능

차량 CAN 통신

CAN(Controller Area Network) 통신

- 통신 전송률은 최대 1Mbps, 저속 모드 10~125kHz, 고속모드 125kbps ~ 1Mbps
- HIGH, LOW 2선으로 통신하는 것을 기본으로 하고, 자동 신호 방식의 직렬 버스 사용
- 다중 마스터 다중 슬레이브 통신 방식 사용
- 모든 노드가 동등한 권한을 가지고 있고, 네트워크상에 버스가 비었다고 판단되면 어떤 노드라고 메시지를 전송할 수 있음
- 모든 노드는 해당 메시지를 읽을 수 있으며, 메시지 수신은 모든 노드가 가능하나, 메시지 종류에 따라 선별적 수신 가능
- 메시지 식별자(ID) 우선순위에 의거하는 메시지 지향 프로토콜
  → ID 기반으로 판단하며, 선별적으로 수신 또는 무시한다
  → ID 값이 낮을 수록 우선순위가 높다
- 반이중 통신, 비동기 방식

하드웨어 연결

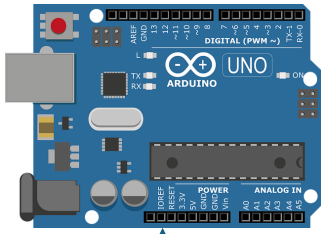


MCP2515	아두이노 UNO
VCC	5V
GND	GND
CS	10
SO	12
SI	11
SCK	13
INT	2

MCP2515	OBD 커넥터
High	6
Low	14

PC	아두이노 UNO
USB 포트	USB 포트

데이터 송수신



ECU 응답 메시지
0x7E8 0x8 0x04, 0x41, 0x2F, 0x7F, 0, 0, 0, 0

요청 메시지 전송
sendPID(PID_FUEL_LEVEL*);

```
tmp[8] = {0x02, 0x01, __pid, 0, 0, 0, 0, 0}; *
sndStat = CAN0.sendMsgBuf(CAN_ID_PID, 0, 8, tmp); *
```

tmp 배열의 인덱스
- 0번: 전송할 데이터 길이
- 1번: 서비스 모드
- 2번: PID
- 나머지: OBD-II 메시지는 8바이트 → 통신 규격을 맞추기 위해 0으로 패딩 처리

응답 메시지 수신
receivePID(PID_FUEL_LEVEL);

```
CAN0.readMsgBuf(&rxId, &len, rxBuf);
if(rxBuf[2] == PID_FUEL_LEVEL) {
    fuel = (100 / 255) * rxBuf[3];
}
```

- rxId: 0x7E8
- len: 8
- rxBuf: {0x04, 0x41, 0x2F, 0x7F, 0, 0, 0, 0}

2F	47	1	Fuel Tank Level Input	0	100	%	$\frac{100}{255}A$

수식열의 문자(A, B, C 등)
데이터의 첫 번째, 두 번째, 세 번째 바이트를 나타냄

* PID(on-board diagnostic Parameter IDs) (https://en.wikipedia.org/wiki/OBD-II_PIDs의 표준 PID 참고)
#define PID_FUEL_LEVEL 0x2F

* 전송할 데이터 프레임
OBD-II 요청 메시지는 항상 8바이트로 고정

* CAN 메시지 전송
- CAN_ID_PID: 전송할 메시지의 CAN ID
- 0: 메시지 전송 형식-표준형 (11비트 ID)
- 8: 데이터 프레임의 길이
- tmp: 실제로 전송할 데이터 프레임

표준 PID 정의 & 서비스 모드

표준 PID

```
#define PID_COOLANT_TEMP 0x05
#define PID_ENGINE_RPM 0x0C
#define PID_VEHICLE_SPEED 0x0D
#define PID_RUNTIME 0x1F
#define PID_FUEL_LEVEL 0x2F
#define PID_ODOMETER 0xA6
```

https://en.wikipedia.org/wiki/OBD-II_PIDs 에 정리된 표준 PID, 서비스 모드, 수식 참고

서비스 모드

서비스 모드 (16진수)	설명
01	현재 데이터 표시
02	정지 화면 데이터 표시
03	저장된 진단 문제 코드 표시
04	진단 문제 코드 및 저장된 값 지우기
05	테스트 결과, 산소 센서 모니터링 (CAN만 해당)
06	테스트 결과, 기타 구성 요소/시스템 모니터링(테스트 결과, CAN 전용 산소 센서 모니터링)
07	보류 중인 진단 문제 코드 표시 (현재 또는 마지막 운전 주기 동안 감지됨)
08	온보드 구성 요소/시스템의 작동 제어
09	차량 정보 요청
0A	영구 진단 문제 코드 (DTC) (지워진 DTC)

PID (서비스 01)

PID (hex)	PID(Dec)	Data bytes returned	Description	Min Value	Max Value	Units	Formula
0D	13	1	속도	0	255	km/h	A
1F	31	2	엔진 시동 후 실행 시간	0	65,535	s	$256A + B$
A6	166	4	누적 주행 거리	0	429,496,729.5	km	$\frac{A(2^{24}) + B(2^{16}) + C(2^{8}) + D}{10}$

초기 – loop() 안에서 반복적으로 자동 호출

```
void loop()
{
//request coolant vehicle speed (속도)
sendPID(PID_VEHICLE_SPEED);
delay(100);
receivePID(PID_VEHICLE_SPEED);

.
.
.

//request coolant temp (냉각수 온도)
sendPID (PID_COOLANT_TEMP);
delay(100);
receivePID(PID_COOLANT_TEMP);
delay(100); //abitrary loop delay
}
```

출력 로그 (기아)

수신 받은 데이터 중 2F(연료값)에 대한 데이터 없음
→ 지원하지 않는 표준 PID

```
PID sent: 0xD
Standard ID: 0x7EF, DLC: 8, Data:  0x03 0x41 0x0D 0x00 0xAA 0xAA 0xAA 0xAA
Vehicle speed (km/h): 0
PID sent: 0x2F
Standard ID: 0x7E8, DLC: 8, Data:  0x03 0x41 0x0D 0x00 0xAA 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0x1F
Standard ID: 0x7E8, DLC: 8, Data:  0x04 0x41 0x1F 0x01 0x0A 0xAA 0xAA 0xAA
Run time (s): 266 (00:04:26)
PID sent: 0xA6
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x1F 0x01 0x0A 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0xC
Standard ID: 0x7E8, DLC: 8, Data:  0x06 0x41 0xA6 0x00 0x0C 0xBF 0xE5 0xAA
PID mismatch
PID sent: 0x5
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x0C 0x0C 0xA0 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0xD
Standard ID: 0x7E8, DLC: 8, Data:  0x03 0x41 0x05 0x44 0xAA 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0x2F
Standard ID: 0x7EF, DLC: 8, Data:  0x03 0x41 0x0D 0x00 0xAA 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0x1F
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x1F 0x01 0x11 0xAA 0xAA 0xAA
Run time (s): 273 (00:04:33)
```

데이터 구조
[0x03, 0x41, 0x0D, 0x00, 0xAA, 0xAA, 0xAA, 0xAA]
[데이터 길이, 서비스 모드, PID, 데이터 값, 나머지는 AA로 패딩(빈자리 채우기 용)]

데이터 딜레이 문제 발생 → 계산 결과 출력 안 됨

데이터 요청 시간이 데이터 준비 시간보다 빨라서 출력이 제대로 안 되는 것으로 판단

데이터 해석

A6 – 오도미터
0x06 0x41 0xA6 0x00 0x0C 0xBF 0xE5 0xAA
→ 83581.3 km

0C – RPM
0x04 0x41 0x0C 0x0C 0xA0 0xAA 0xAA 0xAA
→ 808 RPM

05 – 냉각수 온도
0x03 0x41 0x05 0x44 0xAA 0xAA 0xAA 0xAA
→ 28 °C

0D – 차량 속도
0x03 0x41 0x0D 0x00 0xAA 0xAA 0xAA 0xAA
→ 0km/h

받은 데이터를 계산한 결과 실제 값과 유사

초기 – loop() 안에서 반복적으로 자동 호출

```
void loop()
{
//request coolant vehicle speed (속도)
sendPID(PID_VEHICLE_SPEED);
delay(100);
receivePID(PID_VEHICLE_SPEED);

.
.
.

//request coolant temp (냉각수 온도)
sendPID (PID_COOLANT_TEMP);
delay(100);
receivePID(PID_COOLANT_TEMP);
delay(100); //abitrary loop delay

}
```

출력 로그 (현대) 수신 받은 데이터 중 A6(오도미터)에 대한 데이터 없음
→ 지원하지 않는 표준 PID

```
PID sent: 0xA6
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x1F 0x00 0x17 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0xC
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x0C 0x12 0xD8 0xAA 0xAA 0xAA
Engine Speed (rpm): 1206
PID sent: 0x5
Standard ID: 0x7E8, DLC: 8, Data:  0x04 0x41 0x0C 0x12 0xB8 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0xD
Standard ID: 0x7E9, DLC: 8, Data:  0x03 0x41 0x05 0x49 0xAA 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0x2F
Standard ID: 0x7E8, DLC: 8, Data:  0x03 0x41 0x0D 0x00 0xAA 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0x1F
Standard ID: 0x7E8, DLC: 8, Data:  0x03 0x41 0x2F 0xAD 0xAA 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0xA6
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x1F 0x00 0x1E 0xAA 0xAA 0xAA
PID mismatch
PID sent: 0xC
Standard ID: 0x7E8, DLC: 8, Data:  0x04 0x41 0x0C 0x12 0x70 0xAA 0xAA 0xAA
Engine Speed (rpm): 1180
PID sent: 0x5
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x0C 0x12 0x5C 0xAA 0xAA 0xAA
PID mismatch
```

데이터 딜레이 문제 발생 → 계산 결과 출력 안 됨
데이터 요청 시간이 데이터 준비 시간보다 빨라서 출력이 제대로 안 되는 것으로 판단

데이터 해석

A5 – 냉각수 온도
0x03 0x41 0x05 0x49 0xAA 0xAA 0xAA 0xAA
→ 33 °C

0D – 차량 속도
0x03 0x41 0x0D 0x00 0xAA 0xAA 0xAA 0xAA
→ 0km/h

2F – 연료량
0x03 0x41 0x2F 0xAD 0xAA 0xAA 0xAA 0xAA
→ 67.8%

1F – 런타임
0x04 0x41 0x1F 0x00 0x1E 0xAA 0xAA 0xAA
→ 30s

받은 데이터를 계산한 결과 실제 값과 유사

데이터 구조
[0x03, 0x41, 0x0D, 0x00, 0xAA, 0xAA, 0xAA, 0xAA]
[데이터 길이, 서비스 모드, PID, 데이터 값, 나머지는 AA로 패딩(빈자리 채우기 용)]

확장 PID 정의

확장 PID - 현대

```
#pragma once


#define HYUNDAI_PID_FUEL_LEVEL 0x22B002
#define HYUNDAI_PID_COOLANT_TEMP 0x0167
#define HYUNDAI_PID_ENGINE_RPM 0x22C003
#define HYUNDAI_PID_VEHICLE_SPEED 0x220104
#define HYUNDAI_PID_RUNTIME 0x22C006
#define HYUNDAI_PID_ODOMETER 0x22B002
```

확장 PID - 기아

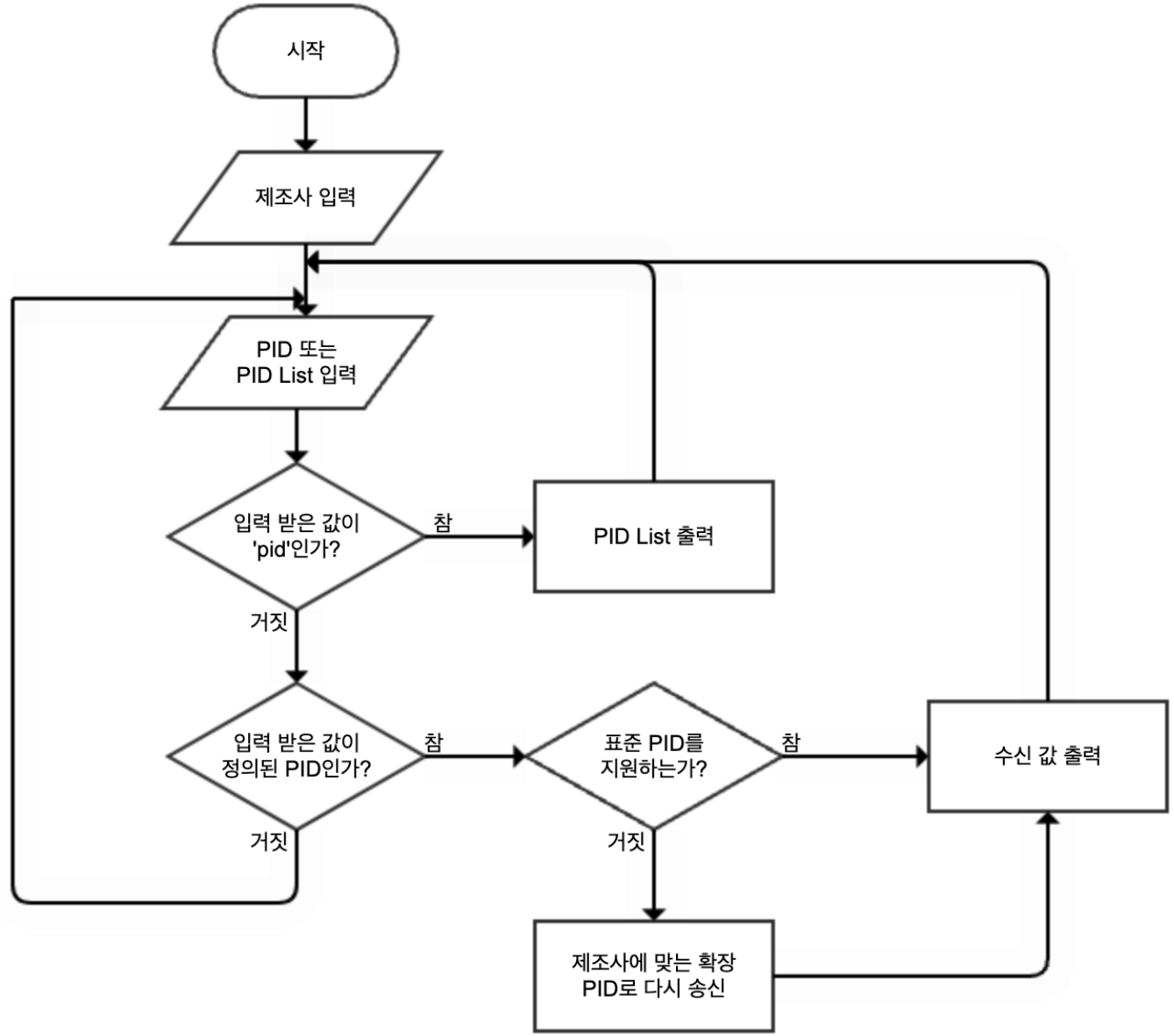
```
#pragma once


#define KIA_PID_FUEL_LEVEL 0x22B002
#define KIA_PID_COOLANT_TEMP 0x0167
#define KIA_PID_ENGINE_RPM 0x22C003
#define KIA_PID_VEHICLE_SPEED 0x220104
#define KIA_PID_RUNTIME 0x22C006
#define KIA_PID_ODOMETER 0x22B002
```

https://github.com/gdincu/HyundaiElantraCN7-OBD2-PIDs/blob/main/CN7.csv 에 정리된 확장 PID 참고
* 기아의 경우 확장 PID를 찾지 못 함 → 현대와 기아는 같은 모기업(현대자동차 그룹)에 속해 있어 PID가 유사할 가능성이 높아 찾은 현대의 확장 PID를 이용해 임의로 정의함

그 외로 찾은 확장 PID
https://github.com/JejuSoul/OBD-PIDs-for-HKMC-EVs/tree/master/Hyundai%20Kona%20EV%20%26%20Kia%20Niro%20EV
https://github.com/JejuSoul/OBD-PIDs-for-HKMC-EVs/commits?author=JejuSoul

최종 – 시리얼 모니터를 이용해 원하는 PID 직접 요청

순서도



최종 – 시리얼 모니터를 이용해 원하는 PID 직접 요청

```
void loop()
{
  if (Serial.available()) {
    if (isPIDSupported(pid)) {
       sendPID(pid);
       delay(500);
       receivePID(pid);
       } else {
         Serial.println("The requested PID is not
supported by the ECU.");
       }
.
.
.
  }
}
```

출력 로그

(입력)

```
Enter the car manufacturer (kia or hyundai): kia
Car make set to: kia
Enter a PID name to request(PID NAME) or type 'pid' to see supported PIDs: pid
Supported PIDs:
PID_COOLANT_TEMP: Engine Coolant Temperature (°C)
PID_ENGINE_RPM: Engine Speed (rpm)
PID_VEHICLE_SPEED: Vehicle Speed (km/h)
PID_RUNTIME: Engine Run Time (hh:mm:ss)
PID_FUEL_LEVEL: Fuel Level (%)
PID_ODOMETER: Odometer (km)
```

① ②

표준 PID를 지원하지 않는 문제를 해결하기 위해 처음에 차 제조사를 입력 받음 → 표준 PID를 지원하지 않을 경우 해당 제조사 PID로 재요청

① 차 제조사 입력 받음

② 시리얼 모니터에 pid를 입력하면 구현해 놓은 PID 리스트 출력
→ 이후 시리얼 모니터에 PID_XXXX 형식으로 요청

최종 – 시리얼 모니터를 이용해 원하는 PID 직접 요청

출력 로그 (입력)

```
Enter another PID name to request(PID NAME): PID_ENGINE_RPM   ①
PID sent: 0xC
PID sent: 0xC
Standard ID: 0x7E9, DLC: 8, Data:  0x04 0x41 0x0C 0x0E 0x1E 0xAA 0xAA 0xAA   ②
Engine Speed (rpm): 903
Enter another PID name to request(PID NAME): PID_VEHICLE_SPEED
PID sent: 0xD
PID sent: 0xD
Standard ID: 0x7EF, DLC: 8, Data:  0x03 0x41 0x0D 0x00 0xAA 0xAA 0xAA 0xAA
Vehicle speed (km/h): 0
Enter another PID name to request(PID NAME): PID_COOLANT_TEMP
PID sent: 0x5
PID sent: 0x5
PID sent: 0x5
Standard ID: 0x7E9, DLC: 8, Data:  0x03 0x41 0x05 0x44 0xAA 0xAA 0xAA 0xAA
Engine Coolant Temp (degC): 28
Enter another PID name to request(PID NAME): PID_FUEL_LEVEL
PID sent: 0x2F
PID sent: 0x2F
PID sent: 0x2F
Standard PID not supported. Trying extended PID...   ③
Extended PID sent: 0x22B002
Standard ID: 0x7CE, DLC: 8, Data:  0x03 0x7F 0x22 0x31 0xAA 0xAA 0xAA 0xAA
```

```
void loop()
{
  if (Serial.available()) {
    if (isPIDSupported(pid)) {
       sendPID(pid);
       delay(500);
       receivePID(pid);
       } else {
         Serial.println("The requested PID is not
supported by the ECU.");
       }
.
.
.
  }
}
```

데이터 딜레이 문제 해결을 위해 시리얼 모니터에 원하는 PID만 요청
→ PID를 최대 3번 송신, 그 안에 요청한 PID가 오지 않을 경우 해당 PID를 지원하지 않는다고 판단

① 원하는 PID 요청 → 요청한 PID에 대한 값을 수신할 때까지 최대 3번 전송

② 3번 안에 응답이 올 경우 출력

③ 3번 안에 응답이 오지 않을 경우, 입력한 제조사에 맞는 확장 PID로 재전송

최종 – 시리얼 모니터를 이용해 원하는 PID 직접 요청

출력 로그

```
Enter another PID name to request(PID NAME):
PID sent: 0x2F
PID sent: 0x2F
PID sent: 0x2F
Standard PID not supported. Trying extended PID...
Extended PID sent: 0x22B002
Standard ID: 0x7CE, DLC: 8, Data:  0x03 0x7F 0x22 0x31 0xAA 0xAA 0xAA 0xAA
```

확장 PID 출력 로그 해석

0x03 0x7F 0x22 0x31 0xAA 0xAA 0xAA 0xAA 해석
0x03: 응답 데이터의 길이 (3바이트)
0x7F: NRC (오류나 요청 실패를 나타내는 응답)
0x22: 서비스 ID (요청한 서비스 ID를 다시 반영)
0x31: NRC 코드
0xAA: 패딩 데이터 (자리 채우기용)

→ 0x22에 대한 요청을 처리하지 못했음, NRC 코드가 0x31인 걸로
요청이 거부된 것으로 확인됨

NRC: 오류의 세부적인 원인을 나타냄
0x10: 요청 메시지가 너무 짧음
0x11: ECU가 요청한 서비스를 지원하지 않음
0x12: 요청한 서브기능을 지원하지 않음
0x13: 메시지 길이가 유효하지 않음
0x21: ECU 임시 오류
0x22: 요청된 데이터가 범위를 초과함
0x31: 요청 거부, 요청한 PID가 ECU에서 지원 X
0x33: 보안 접근이 필요함
0x78:ECU가 바쁜 상태이므로 다시 시도해야 함

오류 응답이 뜨는 이유
1. 지원하지 않는 PID를 요청했을 때
2. 요청을 보낼 때 사용한 CAN ID가 잘못되었을 때
3. 네트워크 충돌로 인해 잘못된 오류가 반환되는 경우
4. CAN ID가 특정 ECU와 일치하지 않는 경우
5. 잘못된 서비스 ID로 요청했을 경우

기아나 현대의 경우 확장 PID 요청에 주로 사용되는 CAN ID, 0x7C6
일반적으로 확장 PID 요청 서비스 ID 0x22
→ 0x7C6, 0x22로 요청했으나 오류가 발생한 것으로 보아
확장 PID가 올바르지 않은 것으로 추정됨