

# 20240516 공부자료

#communication

#MCU

#Sensor

## 통신

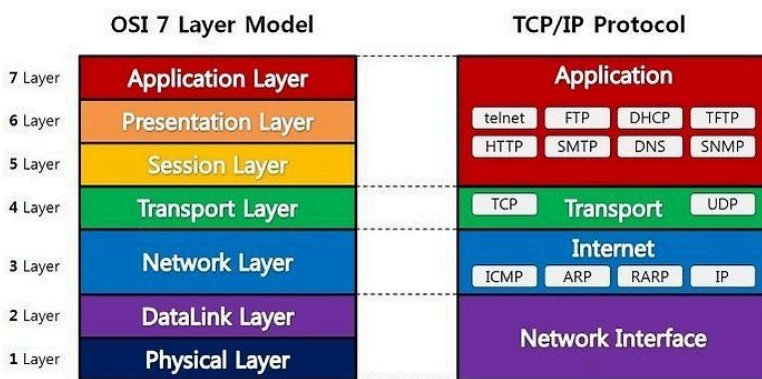
### OSI 7계층 모델

관련 문서 link

#### 정의

- 네트워크에서 통신이 일어나는 과정을 7단계로 나눈 것

#### 7단계



#### 1 Layer - Physical Layer

- 전기적, 기계적, 기능적 특성을 이용해 통신 케이블로 데이터를 전송
- 통신 단위는 비트(1, 0)
- 전달만 할 뿐 에러나 데이터의 종류는 신경 쓰지 않음
- 대표적인 장비: 통신 케이블, 리피터, 허브

#### 2 Layer - DataLink Layer

- Physical Layer로 송수신되는 정보의 오류와 흐름을 관리하여 안전한 정보의 전달을 수행
  - 통신의 오류를 찾고, 재전송 가능
  - MAC 주소를 사용하여 통신
- 통신 단위는 프레임
- Point to Point 간 신뢰성 있는 전송을 보장하기 위해 CRC 기반의 오류 제어와 흐름 제어가 필요

- 대표적인 장비: 브리지, 스위치
- CRC: 순환 중복 검사
- 주소 값은 물리적으로 할당받음 (네트워크 카드가 만들어질 때부터 MAC 주소가 정해짐)
- 주소 체계는 계층이 없는 단일 구조
- 잘 알려진 예: 이더넷, HDLC, ADCCP, 패킷 스위칭 네트워크, LLC, ALOHA

## IP계층

- TCP/IP 상에서 네트워크의 주소(IP 주소)를 정의하고, IP 패킷의 전달 및 라우팅을 담당하는 계층
- IP 계층의 주요 역할
  - 하위 계층인 데이터링크 계층의 하드웨어적 특성에 관계없이 독립적인 역할 수행
- IP 계층의 주요 프로토콜
  - 패킷의 전달을 책임지는 IP
  - 패킷 전달 에러의 보고 및 진단을 위한 ICMP
  - 복잡한 네트워크에서 인터넷워킹을 위한 경로를 찾게 해주는 라우팅 프로토콜

## IP 프로토콜

- 주요 기능
  - IP 계층에서 IP 패킷의 라우팅 대상이 됨 (Routing)
  - IP 주소 지정 (Addressing)
- 주요 특징
  - 신뢰성(에러제어) 및 흐름제어 기능이 전혀 없음 ☞ Best-Effort Service
    - 신뢰성을 확보하려면 상위 트랜스포트 계층에 의존 (예: TCP)
  - 비연결성 데이터그램 방식으로 전달되는 프로토콜 ☞ Connectionless
  - 패킷의 완전한 전달을 보장하지 않음 ☞ Unreliable
  - IP 패킷 헤더 내 수신 및 발신 주소를 포함 ☞ IPv4 헤더, IPv6 헤더, IP 주소
  - IP 헤더 내 바이트 전달 순서: 최상위 바이트(MSB)를 먼저 보냄 ☞ Big-endian
  - 경우에 따라, 단편화가 필요함 ☞ IP 단편화 참조
  - TCP, UDP, ICMP, IGMP 등이 IP 데이터그램에 실려서 전송

## 3 Layer - Network Layer

- 데이터를 목적지까지 가장 안전하고 빠르게 전달하는 기능 (라우팅)
- 경로를 선택하고, 주소를 정하고, 경로를 따라 패킷을 전달
  - IP 주소를 사용
  - 여러 노드를 거칠 때마다 경로를 찾아주는 역할
  - 다양한 길이의 데이터를 네트워크를 통해 전달하고, 전송 계층이 요구하는 서비스 품질(QoS)을 제공
- 수행 내용

- 라우팅
- 흐름 제어
- 세그멘테이션(segmentation/desegmentation)
- 오류 제어
- 인터넷워킹(Internetworking)
- 특징
  - 논리적인 주소 구조(IP), 네트워크 관리자가 직접 주소를 할당하는 구조
  - 계층적(hierarchical)
- 서브네트의 최상위 계층으로 경로를 설정하고, 청구 정보를 관리. 개방형 시스템들 사이에서 네트워크 연결을 설정, 유지, 해제하는 기능을 부여하고, 전송 계층 사이에 네트워크 서비스 데이터 유닛(NSDU)을 교환하는 기능 제공

## 4 Layer - Transport Layer

- 통신을 활성화하기 위한 계층
- 보통 TCP 프로토콜을 이용하여 포트를 열어서 응용 프로그램들이 전송할 수 있도록 함
- 3 Layer → 데이터 → 4 Layer → 데이터를 묶음 → 5 Layer
- 이 계층까지는 물리적인 계층에 속함
  - TCP/UDP 프로토콜 사용
- 양 끝단(End to end)의 사용자들이 신뢰성 있는 데이터를 주고 받을 수 있게 하여 상위 계층들이 데이터 전달의 유효성이나 효율성을 생각하지 않게 함
- 시퀀스 넘버 기반의 오류 제어 방식 사용
- 전송 계층의 특징
  - 특정 연결의 유효성 제어
  - 일부 프로토콜은 상태를 가짐 (stateful)
  - 연결 기반 (connection oriented)
  - 전송 계층이 패킷들의 전송이 유효한지 확인하고 전송 실패한 패킷들을 다시 전송함
    - 예시: TCP

## TCP 프로토콜 (Transmission Control Protocol)

1. 신뢰성 있음 (Reliable)
  - 패킷 손실, 중복, 순서 바뀜 등이 없도록 보장
  - TCP 하위 계층인 IP 계층의 신뢰성 없는 서비스에 대해 다방면으로 신뢰성을 제공
2. 연결지향적 (Connection-oriented)
  - 강한 연결이 아닌 느슨한 연결을 의미
  - 연결 관리를 위한 연결 설정 및 연결 해제 필요
    - TCP 연결 설정, TCP 연결 종료
  - 양단간 어플리케이션/프로세스는 TCP가 제공하는 연결성 회선을 통해 통신

## UDP 프로토콜 (User Datagram Protocol)

1. 비연결성이고, 신뢰성이 없으며, 순서화되지 않은 Datagram 서비스 제공
  - 메시지가 제대로 도착했는지 확인하지 않음 (확인응답 없음)
  - 수신된 메시지의 순서를 맞추지 않음 (순서제어 없음)
  - 흐름 제어를 위한 피드백을 제공하지 않음 (흐름제어 없음)
  - 검사합을 제외한 특별한 오류 검출 및 제어 없음 (오류제어 거의 없음) UDP를 사용하는 프로그램이 자체적으로 오류제어 기능을 갖추어야 함
  - 데이터그램 지향의 전송계층용 프로토콜 (논리적인 가상회선 연결이 필요 없음) 비 연결접속상태 하에서 통신
2. 실시간 응용 및 멀티캐스팅 가능
  - 빠른 요청과 응답이 필요한 실시간 응용에 적합
  - 여러 다수 지점에 전송 가능 (1:多)
3. 헤더가 단순함
  - UDP는 TCP처럼 16 비트의 포트 번호를 사용하나,
  - 헤더는 고정크기의 8 바이트(TCP는 20 바이트)만 사용 즉, 헤더 처리에 많은 시간과 노력을 요하지 않음

## 5 Layer - Session Layer

- 데이터가 통신하기 위한 논리적인 연결
  - 통신을 위한 대문
  - 4계층에서도 연결을 맺고 종료할 수 있어 어느 계층에서 끊어졌는지 판단하기 어려움
  - 4계층과 무관하게 응용 프로그램의 관점에서 봄
  - 기능: 세션 설정, 유지, 종료, 전송 중단 시 복구 등
- 양 끝단의 응용 프로세스가 통신을 관리하기 위한 방법 제공
  - 동시 송수신 방식 (duplex)
  - 반이중 방식 (half-duplex)
  - 전이중 방식 (Full Duplex)
  - 위 통신 방식과 함께 체크 포인트와 유휴, 종료, 다시 시작 과정 등을 수행
- TCP/IP 세션을 만들고 없애는 책임을 짐

## 6 Layer - Presentation Layer

- 데이터 표현이 상이한 응용 프로세스의 독립성을 제공하고, 암호화를 진행
- 코드 간 번역을 담당하여 사용자 시스템에서 데이터의 형식상 차이를 다루는 부담을 응용계층으로부터 덜어줌
  - MIME 인코딩이나 암호화 등의 동작이 이 계층에서 이루어짐
  - 예: EBCDIC로 인코딩된 문서 파일을 ASCII로 변환
  - 해당 데이터가 무엇인지 구분하는 것이 표현 계층의 몫
    - 예: Text, 그림, gif, jpg

## 7 Layer - Application Layer

- 최종 목적지로서 HTTP, FTP, SMTP, POP3, IMAP, Telnet 등과 같은 프로토콜이 존재
- 응용 프로세스와 직접 관계하여 일반적인 응용 서비스를 수행
- 전자메일, 파일 전송, 가상 터미널 등

## HTTP 프로토콜 (HyperText Transfer Protocol)

- 웹 상에서 웹 서버 및 웹 브라우저 상호 간의 데이터 전송을 위한 응용계층 프로토콜
    - 처음에는, WWW 상의 하이퍼텍스트 형태의 문서를 전달하는 데 주로 이용
    - 현재에는, 이미지, 비디오, 음성 등 거의 모든 형식의 데이터 전송 가능
1. 요청 및 응답의 구조
    - 동작 형태가 클라이언트/서버 모델로 동작
  2. 메시지 교환 형태의 프로토콜
    - 클라이언트와 서버 간에 HTTP 메시지를 주고받으며 통신. SMTP 전자메일 프로토콜과 유사
      - HTTP의 응답 및 요청 메시지 구성
      - HTTP 메시지 내 헤더 항목들
  3. 트랜잭션 중심의 비연결성 프로토콜
    - 종단간 연결이 없음 (Connectionless)
    - 이전의 상태를 유지하지 않음 (Stateless)
  4. 전송계층 프로토콜 및 사용 포트 번호
    - 전송계층 프로토콜: TCP
    - 사용 포트 번호: 80번
  5. HTTP 표준
    - HTTP 1.0: RFC 1945 (~1997년)
      - 유용한 초기 개념들 도입
      - HTTP 헤더, HTTP 메서드, HTTP 응답 코드, 리다이렉트, 비지속 연결 등
    - HTTP 1.1: RFC 2068 => RFC 2616 => RFC 7230~7235 (1998년~)
      - HTTP 1.0으로부터 기능 향상
      - HTTP 헤더 내 Host 필드를 필수 항목으로 함 (1개 IP 주소에 다수의 가상 호스팅 가능)
      - HTTP 헤더 내 Accept 필드에 의한 콘텐츠 협상
      - 잘 정의된 캐시 컨트롤
      - 블록 단위 인코딩 전송
      - 지속 연결 회선(킵얼라이브 커넥션)을 통한 재사용 가능
      - 요청 파이프라인을 이용한 병렬 커넥션 처리(실제 사용 거의 없음)

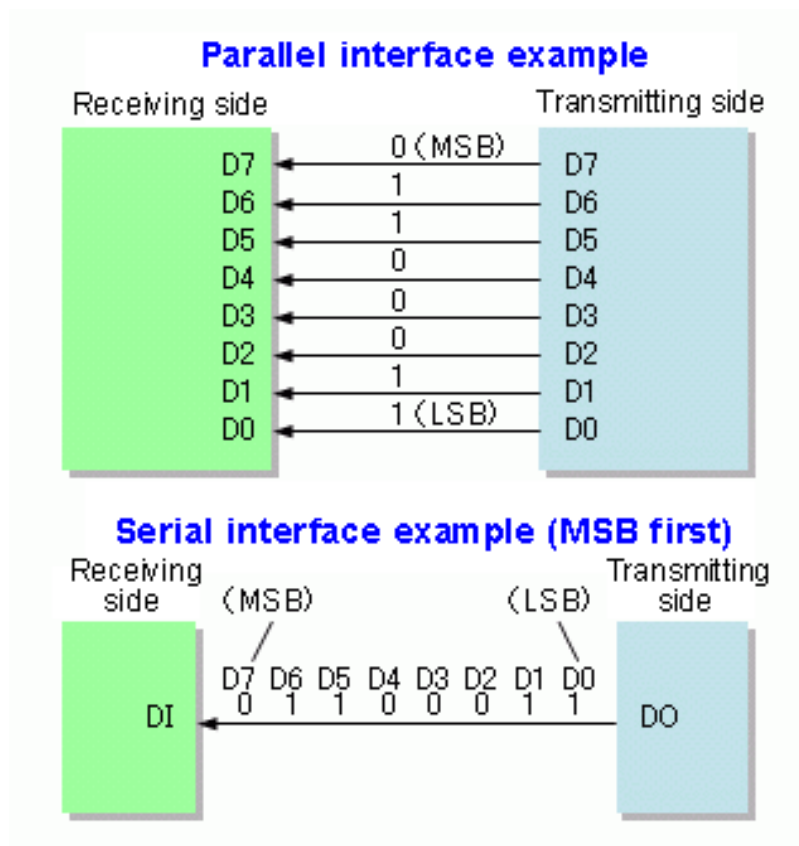
## SCI (Serial Communication Interface)

---

관련 문서 link

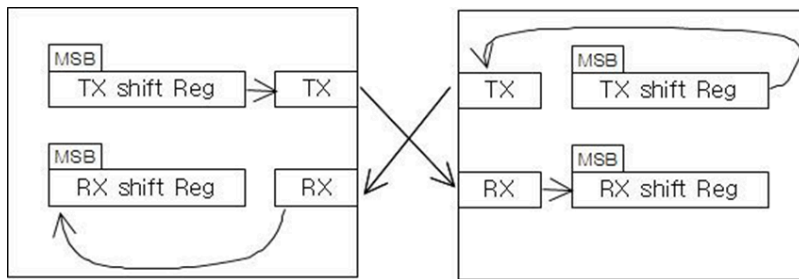
## 정의

- Serial Communication Interface
- 직렬 통신(Serial Bus)
- 연속적으로 통신 채널이나 컴퓨터 버스를 거쳐서 한 번에 하나의 비트 단위로 데이터를 전송하는 것.
- 데이터가 계속 전송되면 각 비트를 구별할 방법이 없기에 비트가 시간적으로 어디서부터 시작이고 끝인지를 알기 위해서 동기 신호를 보내거나 신호 자체에서 데이터 비트를 복원한다.
- 예시 - 모뎀부호, RS-232, RS-422, RS-485, I2C, USB, USART, SATA
- 일반적으로 UART라고 불림
- RX, TX 선을이용해 full-duplex 통신을 지원한다.
- 장거리 통신(보드 간 통신)에 사용되므로 chip의 신호만으로는 강도가 약해 별도의 트랜시버(RS-232, RS-422 등)을 활용해 신호를 증폭해 통신한다.



## 동작 구조

- SCI의 경우 2개의 shift register가 있어 data의 상호 교환이 가능하다. SPI와 달리 SCI는 LSB부터 전송이 되는데, TX shift register의 LSB가 TX 핀으로 넘어가 수신부 RX핀으로 전달된다. 전송된 비트는 RX shift register의 MSB로 넘어간다.
- 비동기식 직렬 통신이기에 통신 속도(band rate)를 맞춰줘야 한다.



## 특징

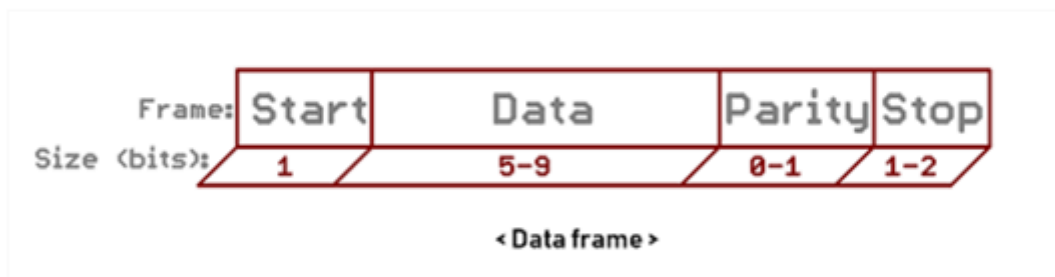
- 2개의 통신용 핀을 갖는다. SCITXD = TX, SCIRXD = RX
- 자체 분주기로 65535가지로 통신 속도 조절이 가능하다.
- 통신 데이터 크기 조절이 가능하다.
- 4가지 방식의 error detection flag를 가진다.
- NRZ(none return to zero) format을 가진다. NRZ: 1101을 보내면 그대로 보내는 것 RZ: 1101 → 10100010 으로 하여 data 구분이 쉽도록 하는 것.
- polling 방식, interrupt방식 모두로 송 수신이 가능하다.
- 16비트 송수신 FIFO를 가지고 있다.
- 자동 Baud rate 감지 기능을 가진다.(한 쪽의 통신속도를 check해서 자동으로 맞춰준다.)

## SCI통신

- SCI는 Serial Communication Interface, 비동기식 직렬통신 = UART
- 데이터 프레임으로 구성 : 시작 비트 + 데이터 비트 + 패리티 비트 + 종료 비트
  - 시작 비트 : 0으로 설정(1bit)
  - 데이터 비트 : 7비트/8비트, 길이는 송수신자 사이에서 미리 설정()
  - 패리티 비트 : 전송중 오류 검출(홀수/짝수)
  - 종료 비트 : 1로 설정(1bit)

패리티 비트는 없을수도 있는 듯

- 데이터 전송 방법
  1. 데이터 준비: 송신자에서 전송할 데이터 대기
  2. 프레임 생성: 데이터 프레임 생성
  3. 전송: 클락 신호의 HIGH, LOW에 동기화되어 전송



UART랑 동일? UART는 직렬통신이랑 같은건가

- 데이터 수신은 데이터 프레임 따라서 잘 하면됨
- Baud Rate가 송신자, 수신자에서 같아야 함
- 송신자, 수신자는 동일한 데이터 프레임을 사용해야 함
- 클락신호가 없음, 시작비트, 클락주기에 따라 동기화

아두이노를 활용한 시리얼 통신 : RX, TX사용

Serial.begin(Baud)

Serial.print()

...

## SPI (Serial Peripheral Interface)

관련 문서 link

### 정의

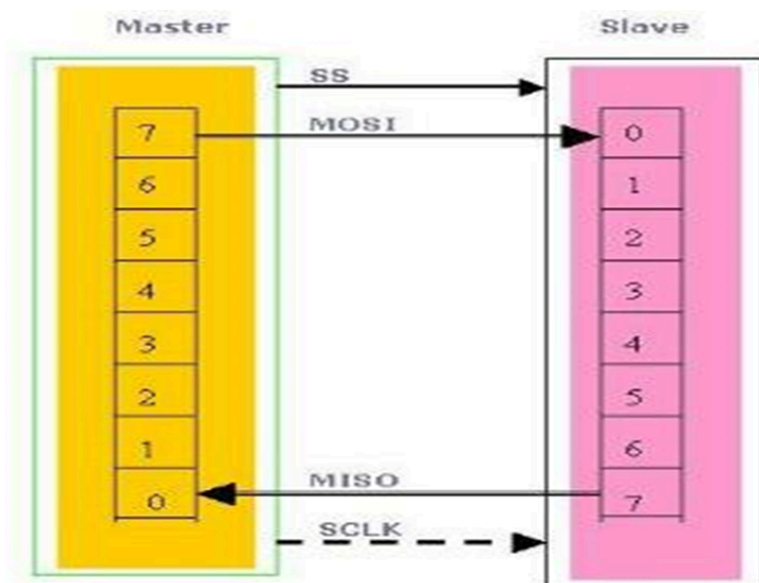
- SPI는 Serial Peripheral Interface의 약자로서 직렬 통신의 한 방법이며 동기식 통신을 한다.(모토로라에서 제안)
- Data 송수신에는 3개의 선이 필요하며 Full-duplex통신이 가능하다 Full-duplex : data 송신부와 수신부가 동시에 data전송, 즉 교환이 가능하다는 의미이다. 이와 대비되는 뜻으로는 Half-duplex가 있으며 간단하게 Full-duplex는 전화기, Half-duplex는 무전기로 보면 된다.



- Single Master 구조를 가지며 Slave는 Multi-slave로 여러 개를 가질 수 있는데 이때는 selection을 위해 선이 더 필요하다.
- 주로 근거리 통신, 즉 chip간 통신에 사용된다.

## 구조

- 4개의 SPI 통신용 핀
  - SPISOMI : SPI slave-output/master-input pin
    - slave : 전송 / master : 수신
  - SPISIMO : SPI slave-input/master-output pin
    - slave : 수신 / master : 전송
  - SPISTE : SPI slave transmit-enable pin
  - SPICLK : SPI serial-clock pin
    - SPI 통신은 동기식이므로 CLK signal이 필요하다



- SPISTE(enable pin)을 제외한 master와 slave 간의 data 송수신 구조만 나타낸 것
  - 옆의 그림에서 ss는 multi-slave에서 slave를 선택하는 pin이다.
  - 그림에서 알 수 있듯이 master와 slave 모두 본인의 MSB부터 전송하며 받은 data는 LSB부터 채워나감을 알 수 있다. 이 때 동작 구조를 선택할 수 있다.
    - master → slave master가 data 전송, dummy 수신
    - master ↔ slave 서로 data 교환
    - master ← slave slave만 data 전송, dummy 수신

## 특징

- master/slave 모드 모두 지원
- 자체 분주기로 125가지 통신속도 조절 가능
- 통신 데이터 크기 조절 가능(1~16bit)
- 4가지 방식으로 동기 클럭을 설정

- Rising edge without delay
- Rising edge with delay
- Falling edge without delay
- Falling edge with delay
- polling or interrupt 방식으로 data 송수신 가능
- 16개의 FIFO를 가진다.

## SPI통신

- Serial Peripheral Interface, **직렬통신?** 동기통신
- Data 송수신에는 3개의 선이 필요 : Single Master, Multi-slave 구조
- 데이터 선 + 클락(송수신 타이밍 맞추기)

핀 구성 : SCK+MISO+MOSI+SS

- 클락신호 만드는 쪽 : Master(cf. Micro Controller)/반대쪽 : Slave

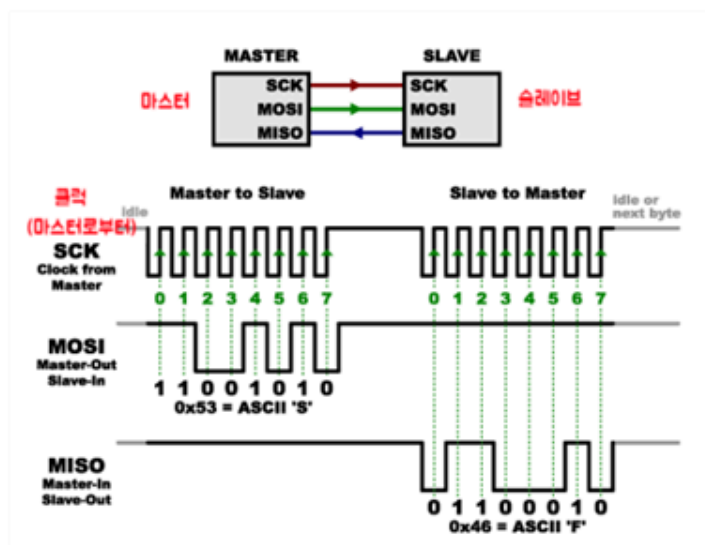
1. 데이터 Master -> Slave

클락은 Master에서 생성 신호(Data)는 MOSI를 따라 Slave에게 전송

2. 데이터 Slave -> Master

클락은 Master에서 생성 신호(Data)는 MISO를 따라 Slave에게 전송

클락은 이미 정해진 주기에 따라 생성



클락은 항상 마스터에서 생성 => 슬레이브가 데이터를 언제, 얼마나 보낼지 미리 알아야 함

Slave 선택(SS) : 여러 Slave중 하나 선택하기



평소에 1(Slave에서 해제된 상태), 통신 시작하면 슬레이브 선택선 신호를 Low로 바꾸어 활성화 : Active Low // Slave는 독립된 SS선을 가짐

## I2C (Inter-Integrated Circuit)

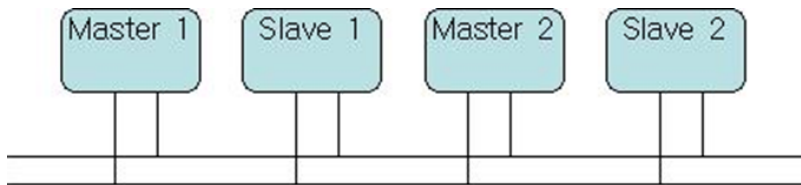
관련 문서 link

## 정의

- Inter-Integrated Circuit의 약자로 동기식 직렬통신의 한 방식
- 통신에는 2개의 선이 필요하며 Half-duplex 통신이 가능하다.
- Multi-master mode를 지원하기에 여러 개의 주변회로와 연결이 가능하다.
- SPI와 마찬가지로 chip간 근거리 통신에 자주 이용된다.

## 동작구조

- 기본적으로 Multi-master와 Multi-slave가 같은 통신 라인에 연결되어 있다.
  - 라인에 연결된 마스터들 중 하나가 Start 신호를 보내면 Bus를 점유한다. 만약, 마스터들이 동시에 보내면 낮은 slave address를 보낸 master가 우선순위를 갖는다.
  - Bus를 점유한 master가 slave address와 Read/Write 정보를 Bus에 전송
  - master가 보낸 address와 일치하는 slave가 ACK 신호를 전송
  - ACK 신호를 받은 마스터가 slave에 data 전송, 전송 후 Bus 점유를 해제
  - I2C의 인터럽트로는 7개의 기본 인터럽트와 2개의 FIFO 인터럽트가 있다.
  - 7개의 기본 인터럽트는 여러 개가 한번에 걸릴 수 없고, Arbitrator에 의해 한 개가 선택되어 걸린다.
  - FIFO 인터럽트는 기본 인터럽트와 별개이므로 기본 인터럽트와 중복되어 걸릴 수 있다.



## I2C통신

- SCL : 데이터 전송 동기화하는 클락신호(HIGH->LOW : 시작, LOW->HIGH : 끝)
- SDA : 데이터 전송
- SCL이 low일 때 데이터 교환 가능
- Master & Slave => SPI통신이랑 역할 동일
- SCL, SDA선이 모두 HIGH일 때 통신가능 => 풀업저항을 연결해 HIGH 상태로 만들자
- 하나의 Master와 하나의 Slave만 통신가능
- 동기화 통신 방식 : 통신속도를 안정해도 됨

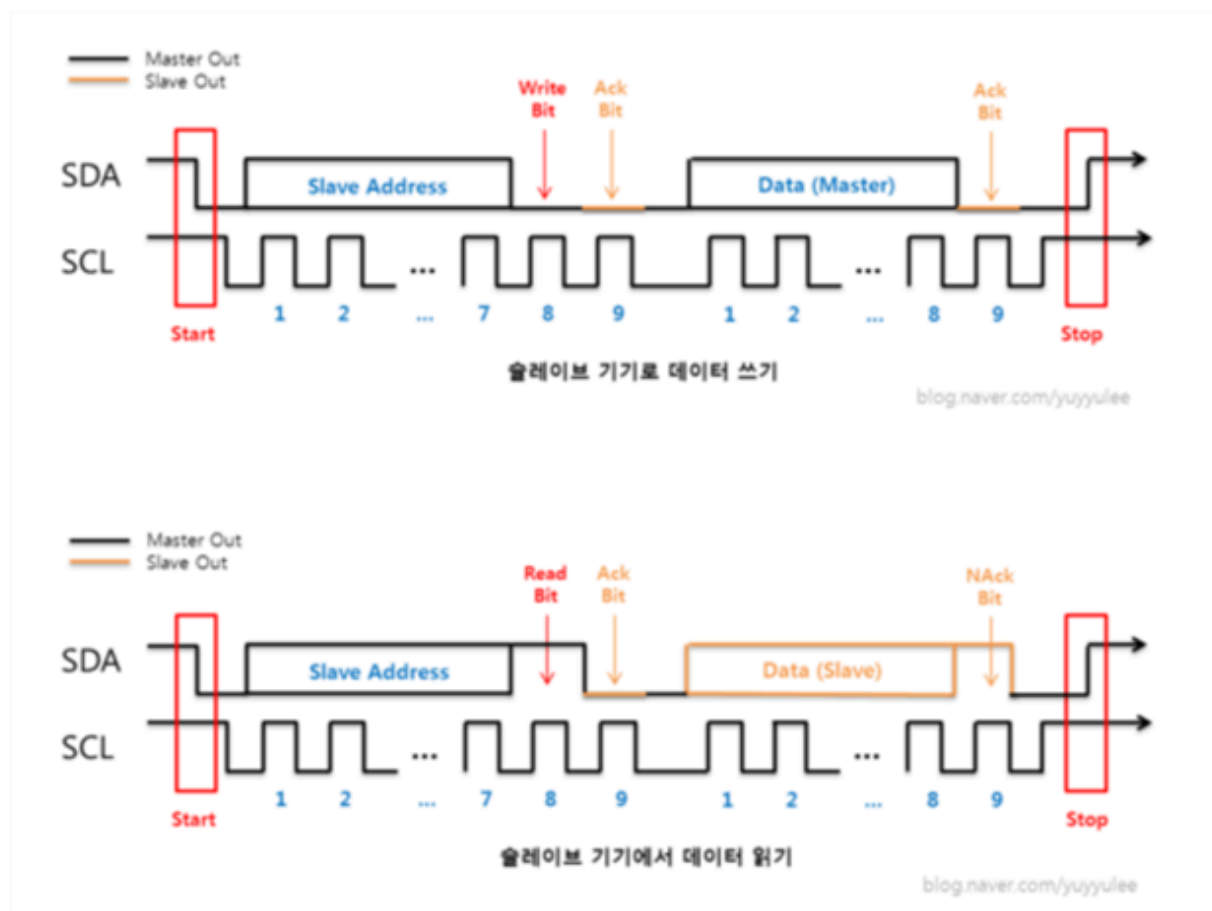
START 1-BIT	ADDRESS 7-BIT	WRITE 1-BIT	ACK	DATA 8-BIT	ACK	STOP 1-BIT
----------------	------------------	----------------	-----	---------------	-----	---------------

〈data write / 주황 : 마스터, 흰색 : 슬레이브〉

START 1-BIT	ADDRESS 7-BIT	READ 1-BIT	ACK	DATA 8-BIT	ACK	STOP 1-BIT
----------------	------------------	---------------	-----	---------------	-----	---------------

〈data read / 주황 : 마스터, 흰색 : 슬레이브〉

## Read/Write 1bit



## I2C모터 드라이버?

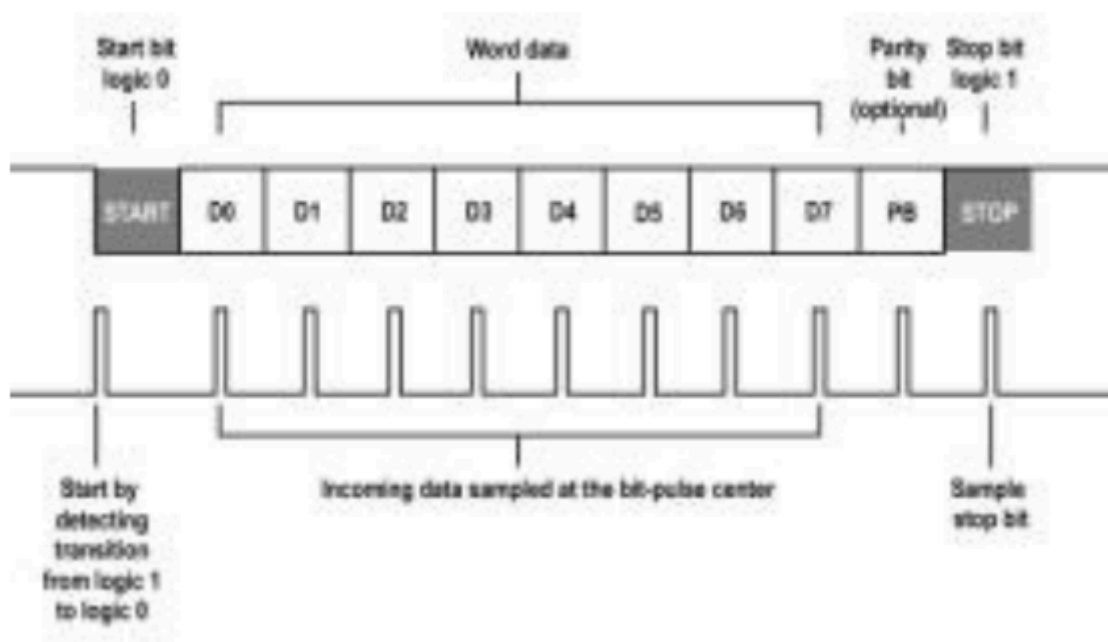
# UART (Universal Asynchronous Receiver/Transmitter)

## 정의

- Universal Asynchronous Receiver/Transmitter(범용 비동기 수신/송신기)는 컴퓨터와 주변 장치 간의 데이터 통신을 위한 하드웨어 장치입니다.
- 비동기식 직렬 통신 방식으로 데이터를 프레임 단위로 전송합니다.
- 시작 비트, 데이터 비트, 패리티 비트, 정지 비트로 구성됩니다.
- 특히, 컴퓨터와 시리얼 포트를 통해 키보드, 마우스 등과의 통신에 널리 사용됩니다.

## 동작 구조

- UART는 송신기(TX)와 수신기(RX)로 구성됩니다.
- 데이터를 송신할 때, 시작 비트로 프레임의 시작을 알리고, 데이터 비트를 전송한 후에는 패리티 비트와 정지 비트를 전송하여 프레임을 완료합니다.
- 데이터를 수신할 때, 시작 비트를 감지하고 데이터 비트를 수신한 후에 패리티와 정지 비트를 확인하여 데이터를 추출합니다.



## 특징

- 비동기식 통신으로, 송수신기 간의 클록을 공유하지 않습니다.
- 간단한 구조로 널리 사용되며, 주변 장치와의 통신에 많이 활용됩니다.
- 표준적으로 8비트 데이터 프레임을 사용하지만, 데이터 비트의 길이는 조절 가능합니다.
- 패리티 비트를 사용하여 오류 검출이 가능합니다.
- 여러 장치 간의 직렬 통신에 유용하게 쓰입니다.
- 저속 데이터 전송에 적합
- Full Duplex 통신 지원
- 하드웨어 및 소프트웨어 구현이 간단

# USART (Universal Synchronous/Asynchronous Receiver/Transmitter)

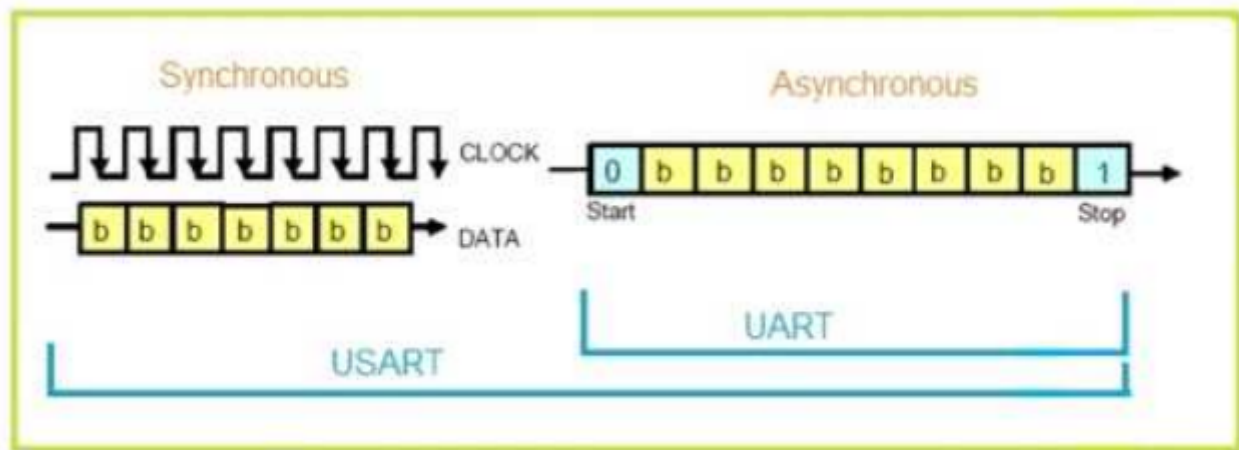
---

[관련 문서 link](#)

## 정의

UART (Universal Asynchronous Receiver Transmitter)의 기능을 확장한 형태로 비동기 통신뿐만 아니라 동기 통신도 지원한다.

## 동작 구조



## 특징

- 동기통신 (Synchronous Communication)에서는 송신자와 수신자가 동일한 클럭 신호를 공유하기에 데이터 전송이 더 빠르고 안정적일 수 있다.  
(비동기 + 동기 통신)
- USART 모듈은 UART (Universal Asynchronous Receiver Transmitter)의 TX, RX 핀 외에도 동기 모드에서 사용하는 클럭신호(SCK) 핀이 추가되었다.  
-> 더 높은 전송 속도를 갖는다.
- 다양한 통신 방식과 전송 속도를 지원하기에 다양한 응용분야에 사용될 수 있다.

# 이더넷 (Ethernet)

---

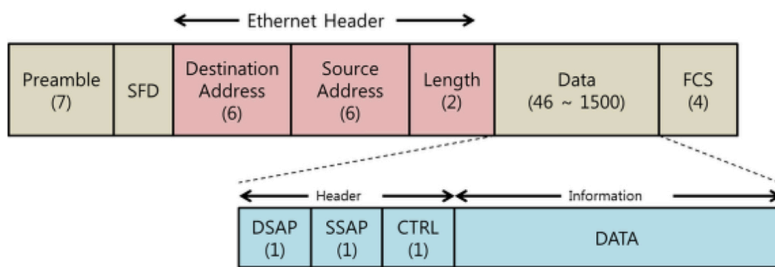
[관련 문서 link](#)

## 정의

- 이더넷은 LAN(Local Area Network)에서 사용되는 네트워크 기술로, IEEE 802.3 표준을 따릅니다.
- 고속 데이터 전송을 지원하며, 유선 네트워크에서 주로 사용됩니다.
- OSI 7계층 모델의 물리 계층에서 신호와 배선, 데이터 링크 계층에서 MAC(Media Access Control) 패킷과 프로토콜 형식을 정의합니다.

- Layer 2의 DLC(Data Link Control) 서비스의 대표적인 프로토콜입니다.

## 동작 구조



- [네이버 블로그 자료](#)
- 이더넷은 CSMA/CD(Carrier Sense Multiple Access with Collision Detection) 방식을 사용하여 데이터 충돌을 감지하고 조정합니다.
- 데이터를 전송하기 전에 다른 장치가 데이터를 전송 중인지 감지하고, 충돌이 발생할 경우 일정 시간 동안 대기한 후 다시 전송을 시도합니다.
- CSMA/CD는 LAN에서의 데이터 전송 충돌을 방지하고, 효율적인 데이터 전송을 보장합니다.

## 특징

- 높은 데이터 전송 속도를 제공합니다.
- 대규모 네트워크 구축이 가능하며, 여러 장치 간의 통신을 지원합니다.
- 안정적인 연결을 제공하여 네트워크의 안정성을 보장합니다.
- 주로 오피스, 가정, 학교 등에서 네트워크 연결에 사용됩니다.

## TCP/IP (Transmission Control Protocol/Internet Protocol)

[관련 문서 link](#)

## 정의

- Transmission Control Protocol/Internet Protocol(TCP/IP)은 인터넷과 같은 네트워크에서 데이터를 전송하기 위한 프로토콜입니다.
- TCP는 연결 지향형 프로토콜로, 데이터의 신뢰성을 보장합니다.
- IP는 패킷 전달과 주소 지정을 담당하여 데이터를 안전하게 전송합니다.

## 동작 구조

- TCP는 데이터 전송을 위해 세그먼트(segment)라는 단위를 사용하며, 연결 설정, 데이터 전송, 연결 종료 등의 기능을 수행합니다.
- IP는 패킷을 전달하기 위해 라우팅과 주소 지정을 담당하여 패킷을 목적지로 안전하게 전달합니다.

- TCP와 IP는 함께 사용되어 데이터를 안정적으로 전송하고, 인터넷 통신을 가능하게 합니다.

## 특징

- 신뢰성 있는 데이터 전송을 보장하여 데이터 손실을 방지합니다.
- 전 세계적으로 사용되는 표준 프로토콜로, 다양한 네트워크 환경에서 사용됩니다.
- 인터넷에서의 통신을 위한 필수적인 프로토콜로, 웹 브라우징, 이메일, 파일 전송 등 다양한 서비스에 사용됩니다.

## UDP (User Datagram Protocol)

---

[관련 문서 link](#)

### 정의

- User Datagram Protocol(UDP)은 비연결형 프로토콜로, 빠른 데이터 전송을 목적으로 합니다.
- 데이터그램(datagram) 단위로 전송되며, 신뢰성을 보장하지 않습니다.
- 신속한 데이터 전송이 필요한 응용 프로그램에 주로 사용됩니다.

### 동작 구조

- UDP는 연결 설정 과정 없이 데이터를 전송하며, 패킷을 목적지로 전달합니다.
- 데이터 전송 후 수신 확인이나 재전송을 하지 않으므로, 신속한 데이터 전송이 가능합니다.
- 주로 실시간 스트리밍, DNS(Domain Name System) 등에 사용되며, 데이터 손실이 발생해도 큰 문제가 없는 경우에 적합합니다.

### 특징

- 낮은 지연 시간을 제공하여 실시간 데이터 전송에 적합합니다.
- 신뢰성을 보장하지 않으므로, 데이터 손실이 발생할 수 있습니다.
- 간단한 헤더 구조를 가지고 있어 오버헤드가 적습니다.

## RS-232

---

[관련 문서 link](#)

### 정의

- RS-232는 직렬 통신 표준으로, 단거리 데이터 전송에 사용됩니다.
- 컴퓨터와 주변 장치 간의 통신에 주로 사용되며, 표준적으로 RS-232 케이블을 통해 연결됩니다.



## 동작 구조

- 1:1 통신 방식으로, 하나의 송수신기와 하나의 수신기가 직접 연결됩니다.
- 최대 50피트 거리까지 데이터 전송이 가능하며, 속도는 보통 20kbps 정도입니다.
- 전송 속도와 데이터 비트 길이 등은 설정 가능합니다.

## 특징

- 저속 통신으로, 주로 컴퓨터와 주변 장치 간의 통신에 사용됩니다.
- 단거리 데이터 전송에 적합하며, 주로 시리얼 포트를 통해 연결됩니다.
- 간단한 인터페이스로 구성되어 있어, 하드웨어 구현이 간단합니다.

## RS-422

---

[관련 문서 link](#)

## 정의

- RS-485는 다중 점 통신을 위한 직렬 통신 표준으로, RS-232의 한계를 극복하기 위해 개발되었습니다.
- 멀티포인트(multi-point) 환경에서 통신할 수 있으며, 멀리 떨어진 여러 장치 간에 통신할 때 적합합니다.
- RS-422와 비슷하지만, RS-485는 멀티포인트 통신을 지원합니다.

## 동작 구조

- RS-485는 멀티포인트 통신을 지원하여 하나의 송신기가 여러 개의 수신기와 통신할 수 있습니다.
- 극성 신호(polarity signaling)를 사용하여 데이터를 전송하며, 장거리 통신에 적합합니다.
- 미국 전기전자기술 협회(ANSI)에서 제정한 표준으로, 산업용 제어 시스템 등에 널리 사용됩니다.
- 최대 32개의 장치가 하나의 버스를 통해 통신할 수 있습니다.

## 특징

- 멀티포인트 통신을 지원하여 여러 장치 간에 데이터를 전송할 수 있습니다.
- 장거리 통신에 적합하며, 노이즈에 강한 특성을 가지고 있어 신뢰성이 높습니다.
- 산업용 제어 시스템, 자동화 시스템, 스마트 빌딩 등에 널리 사용됩니다.
- 강력한 잡음 내성

## RS-485

---

[관련 문서 link](#)

## 정의

RS-422는 차동 신호 전송 방식을 사용하는 직렬 통신 표준으로, 장거리 고속 통신에 적합합니다.

## 동작 구조

1:10 통신 방식을 지원하며, 최대 4000피트 거리까지 데이터 전송이 가능합니다.

## 특징

- 장거리 데이터 전송
- 고속 통신 지원
- 높은 신호 무결성

## USB (Universal Serial Bus)

---

[관련 문서 link](#)

## 정의

- Universal Serial Bus(USB)는 컴퓨터와 주변 장치 간의 데이터 통신 및 전력 공급을 위한 표준 인터페이스입니다.
- Plug and Play 기능을 지원하여 주변 장치를 쉽게 연결하고 사용할 수 있습니다.
- 다양한 종류의 주변 장치를 지원하며, 데이터 전송 속도와 전력 공급량을 조절할 수 있습니다.

## 동작 구조

- USB는 호스트-클라이언트 구조로, 컴퓨터나 장치에 USB 포트를 통해 연결됩니다.
- USB 케이블을 통해 데이터 및 전력 신호를 전송하며, 데이터 전송 속도와 전력 공급량은 USB 버전에 따라 다릅니다.
- 다양한 종류의 주변 장치를 지원하며, 하나의 USB 포트에 여러 장치를 연결하여 사용할 수 있습니다.

## 특징

- Plug and Play 기능을 지원하여 주변 장치를 쉽게 연결하고 사용할 수 있습니다.
- 다양한 종류의 주변 장치를 지원하며, 하나의 USB 포트에 여러 장치를 동시에 연결할 수 있습니다.
- 데이터 전송 속도와 전력 공급량을 조절할 수 있어 다양한 응용이 가능합니다.

## CAN (Controller Area Network)

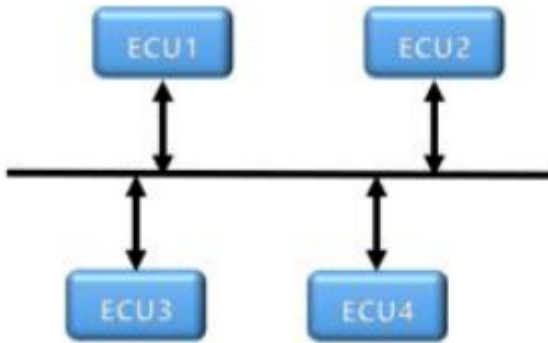
---

[관련 문서 link](#)

## 정의

Controller Area Network는 차량 내 네트워크 통신 표준으로, 실시간 데이터 통신을 지원합니다.

## 동작 구조



<그림 3> CAN 통신 방법

다중 마스터, 메시지 기반 통신을 지원하며, 모든 장치가 동일한 버스를 통해 통신합니다.

## 특징

- 실시간 통신
- 높은 오류 검출 능력
- 차량 및 산업용 애플리케이션에 사용

## CAN Open

---

[관련 문서 link](#)

## 정의

CAN 프로토콜을 기반으로 한 상위 레이어 프로토콜로, 네트워크 관리와 객체 사전 기능을 제공합니다.

## 동작 구조

네트워크의 모든 장치가 상호 운용성을 보장하도록 설계되었습니다.

## 특징

- 모듈 간 상호 운용성
- 표준화된 프로토콜
- 산업 자동화에 사용

## CAN FD (Flexible Data-rate)

---

[관련 문서 link](#)

### 정의

CAN의 확장 버전으로, 데이터 속도를 향상시킨 프로토콜입니다.

### 동작 구조

가변 데이터 속도를 지원하며, 최대 64바이트의 데이터 전송이 가능합니다.

### 특징

- 높은 데이터 처리 속도
- 기존 CAN 네트워크와 호환 가능
- 실시간 애플리케이션에 적합

## 이더넷 (Ethernet)

---

[관련 문서 link](#)

### 정의

EtherCAT은 산업용 이더넷 통신 표준으로, 짧은 사이클 타임과 높은 효율성을 제공합니다.

### 동작 구조

마스터-슬레이브 구조를 사용하며, 실시간 데이터 전송을 지원합니다.

### 특징

- 높은 효율성
- 저지연
- 복잡한 자동화 시스템에 적합

## 동기통신 (Synchronous Communication)

---

[관련 문서 link](#)

### 정의

클록 신호를 사용하여 데이터 전송을 동기화하는 통신 방식입니다.

### 동작 구조

송신기와 수신기가 동일한 클록 신호를 사용하여 데이터를 전송합니다.

- 요청을 보내면 응답을 얻기 전까지 Block 상태가 된다

## 특징

- 높은 데이터 전송 속도
- 정확한 비트 동기화
- 이로 인해서 뒤의 요청이 연결을 맺지 못한다.
- 순서를 보장해주기 때문에 요청에 대한 결과가 변경되는 사항에서 중요하다!

## 비동기통신 (Asynchronous Communication)

---

[관련 문서 link](#)

### 정의

클록 신호 없이 데이터 전송을 수행하는 통신 방식입니다.

### 동작 구조

시작 비트와 정지 비트를 사용하여 비트의 시작과 끝을 구분합니다.

- 응답을 기다리지 않아서 Non Block 상태가 된다.

### 특징

- 저속 데이터 전송
- 단순한 하드웨어 구현
- 순서를 보장하지 않는다.
- 동기 통신에 비해 성능이 좋다.
- 순서를 보장하지 않기에 결과를 보장 받고 처리해야 하는 시스템에는 적합하지 않다.

## Full Duplex

---

[관련 문서 link](#)

### 정의

송신기와 수신기가 동시에 데이터를 주고받을 수 있는 통신 방식입니다.

### 동작 구조

별도의 송수신 채널을 사용하여 데이터를 주고받습니다.

### 특징

- 양방향 통신 가능
- 효율적인 데이터 전송
- 전화 통신, 인터넷 등의 응용 분야에서 사용

## Half-Duplex

---

[관련 문서 link](#)

### 정의

송신기와 수신기가 번갈아 가며 데이터를 주고받는 통신 방식입니다.

### 동작 구조

단일 채널을 공유하며, 한 번에 하나의 방향으로만 데이터 전송이 가능합니다.

### 특징

- 양방향 통신 가능하지만 동시에 송수신 불가능
- 무전기, 토크백 시스템에서 사용

## eCAN

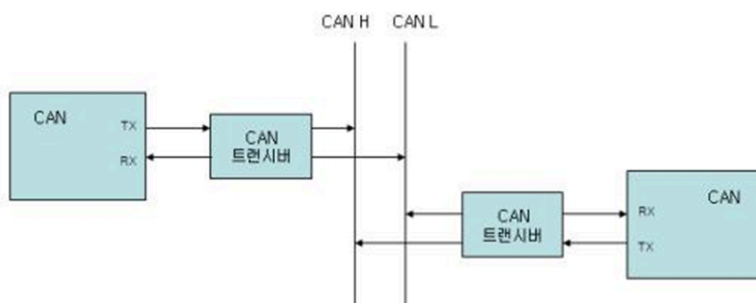
---

[관련 문서 link](#)

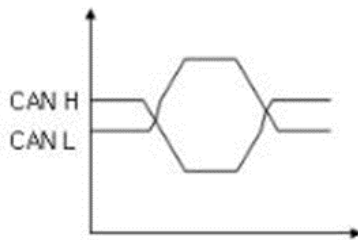
### 정의

- Enhanced Controller Area Network의 약자로 비동기식 직렬 통신의 한 방식이다.
- 2개의 선으로 통신하며 SCI와는 달리 Half-duplex 통신이 가능하다.
- 멀티마스터-멀티슬레이브 구조가 가능해 N:N 통신이 가능하다.
- Differential 신호방식을 사용하므로 노이즈에 강하다.
- 주로 중장거리 보드 간 통신에 사용되므로 이 역시 별도의 트랜시버를 필요로 한다.

### 동작 구조



- CAN통신은 일반적으로 위와 같은 구조를 가진다. CAN을 나온 data는 트랜시버를 거쳐 CAN H, CAN L로 data가 넘어가게 되는데 이 트랜시버를 거치며 사용하는 신호 방식인 differential 신호 방식이 CAN통신의 특징이다.
- 트랜시버에는 TX에서 나온 data를 받아서 Inverting을 하는 데 1번 한 것을 CAN H에, 2번 한 것을 CAN L에 전달하고 이 두 신호의 전위차로 원 신호를 파악하게 된다. 좀 더 자세하게 살펴보기 위해 그래프를 살펴본다.



- 이 그래프는 TX에서 101이 트랜시버를 거쳐서 나왔을 때를 나타낸 것인데 CAN H와 CAN L의 전위차가 작으면 1, 전위차가 크면 0으로 하여 원 신호인 101을 파악할 수 있음을 알 수 있다. 그리고 이러한 신호 방식은 noise가 생기더라도 전위 차이는 바뀌지 않기 때문에 noise에 굉장히 강한 모습을 나타낸다.

ID	Mail Box 1
ID	Mail Box 2
ID	Mail Box 3
ID	Mail Box 4
ID	Mail Box 5

- 그리고 CAN통신에서 실제로 주고 받는 data는 이 Mail box에 저장되고 보내지는데 main box에는 각각의 ID가 있어서 이 ID가 address역할을 하여 특정 mail box의 data를 송수신 할 수 있다.
- CAN에서 실제로 전달되는 메시지 구조는 ID, Data를 제외하고 Start, ACK 등 여러 가지가 포함되는데 이는 따로 저장되어 있지 않고 개발자가 설정하면 자동으로 추가되어 전달된다.
- data를 주고받는 방식은 SPI와 비슷하게 master에서 ID와 data를 같이 송신하면 slave에서 ID를 check하고 만약 ID가 일치하면 data를 받아들이게 된다.

## MCU 기능

### 인터럽트 (Interrupt)

[관련 문서 link](#)

#### 정의

특정 이벤트가 발생했을 때 CPU의 현재 작업을 중단하고 즉시 해당 이벤트를 처리하는 기능입니다.

#### 동작 구조

- 하드웨어 인터럽트: 특정 하드웨어 장치가 CPU에 신호를 보내어 발생
- 소프트웨어 인터럽트: 프로그램 코드 내에서 인터럽트를 발생시킴

## 특징

- 실시간 응답 가능
- 우선 순위에 따라 여러 인터럽트를 관리
- 시스템 자원 효율적 사용

## 쓰레드 (Thread)

---

[관련 문서 link](#)

### 정의

프로세스 내에서 독립적으로 실행되는 실행 단위입니다.

### 동작 구조

하나의 프로세스는 여러 개의 쓰레드를 가질 수 있으며, 각 쓰레드는 독립적으로 실행됩니다.

### 특징

- 동시 실행 가능
- 자원 공유
- 멀티스레딩을 통해 성능 향상

## ADC (Analog to Digital Converter)

---

[관련 문서 link](#)

### 정의

아날로그 신호를 디지털 신호로 변환하는 장치입니다.

### 동작 구조

- 샘플링: 아날로그 신호를 일정한 간격으로 샘플링
- 양자화: 샘플링된 값을 정수값으로 변환
- 인코딩: 양자화된 값을 이진수로 변환

### 특징

- 디지털 처리 가능
- 다양한 해상도와 샘플링 속도 제공



- 센서 데이터 수집에 사용

## PWM (Pulse Width Modulation)

---

[관련 문서 link](#)

### 정의

펄스 폭 변조를 통해 아날로그 신호를 디지털 방식으로 제어하는 기술입니다.

### 동작 구조

신호의 듀티 사이클을 조절하여 출력 전압을 변경합니다.

### 특징

- 모터 속도 제어, LED 밝기 조절 등 다양한 응용
- 효율적인 전력 제어

## DAC (Digital to Analog Converter)

---

[관련 문서 link](#)

### 정의

디지털 신호를 아날로그 신호로 변환하는 장치입니다.

### 동작 구조

저항망, 전류 소스 등을 사용하여 디지털 값을 아날로그 전압으로 변환합니다.

### 특징

- 아날로그 출력 제공
- 오디오 신호 생성, 데이터 변환 등에 사용

## GPIO (General-Purpose Input/Output)

---

[관련 문서 link](#)

### 정의

일반 목적의 입력 및 출력 핀으로, 다양한 장치를 제어하거나 데이터를 수신할 수 있습니다.

### 동작 구조

핀을 입력 또는 출력으로 설정하여 디지털 신호를 처리합니다.

## 특징

- 다목적 사용 가능
- 사용자 정의 가능
- 간단한 인터페이스

## TTL (Transistor-Transistor Logic)

---

[관련 문서 link](#)

### 정의

트랜지스터-트랜지스터 로직 기반의 디지털 회로 기술입니다.

### 동작 구조

바이폴라 전계효과 트랜지스터를 사용하여 논리 연산을 수행합니다.

### 특징

- 고속 스위칭
- 낮은 전력 소비
- 디지털 시스템에서 널리 사용

## USB (Universal Serial Bus)

---

[관련 문서 link](#)

### 정의

범용 직렬 버스로, 데이터 전송 및 전원 공급을 위한 인터페이스입니다.

### 동작 구조

호스트-디바이스 구조로 동작하며, 데이터 전송 및 전원 공급을 동시에 수행할 수 있습니다.

### 특징

- 플러그 앤 플레이
- 고속 데이터 전송
- 다양한 장치 연결 지원

## 센서

---

## 컨버터 (Converter)

---

[관련 문서 link](#)

## 정의

신호를 변환하는 장치로, 아날로그-디지털, 디지털-아날로그 등의 변환을 수행합니다.

## 동작 구조

입력 신호를 원하는 형태로 변환하여 출력합니다.

## 특징

- 다양한 신호 변환 가능
- 전자 시스템에서 필수 요소

## 인버터 (Inverter)

---

[관련 문서 link](#)

## 정의

DC 전원을 AC 전원으로 변환하는 장치입니다.

## 동작 구조

DC 입력을 스위칭 회로를 통해 교류 전원으로 변환합니다.

## 특징

- 재생 가능 에너지 시스템, 전기차 등에 사용
- 다양한 출력 전압 및 주파수 제공

## 엔코더 (Encoder)

---

[관련 문서 link](#)

## 정의

회전각이나 직선 운동을 디지털 신호로 변환하는 장치입니다.

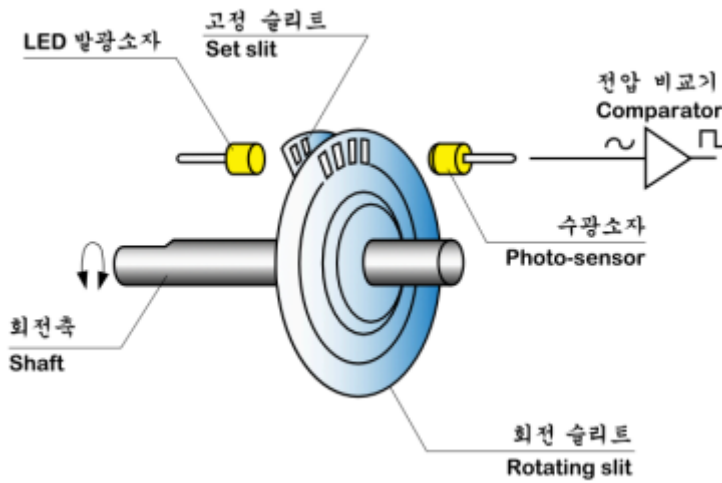
## 동작 구조

### 광학식 엔코더

: 빛을 이용해 회전자의 위치 및 변화를 파악한다.

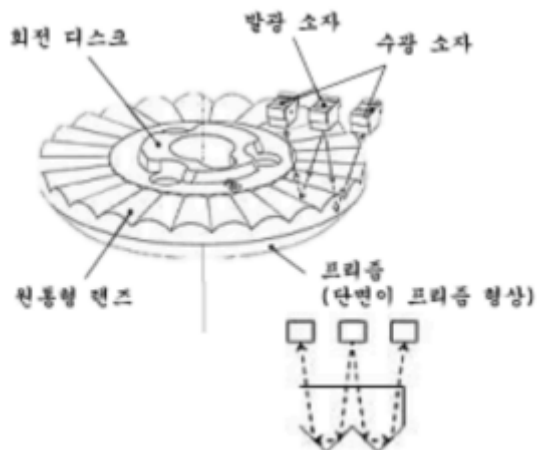
- 특징 : 분해능이 높고, 자기장에 영향을 덜 받는다.

## 1. 투과형 엔코더



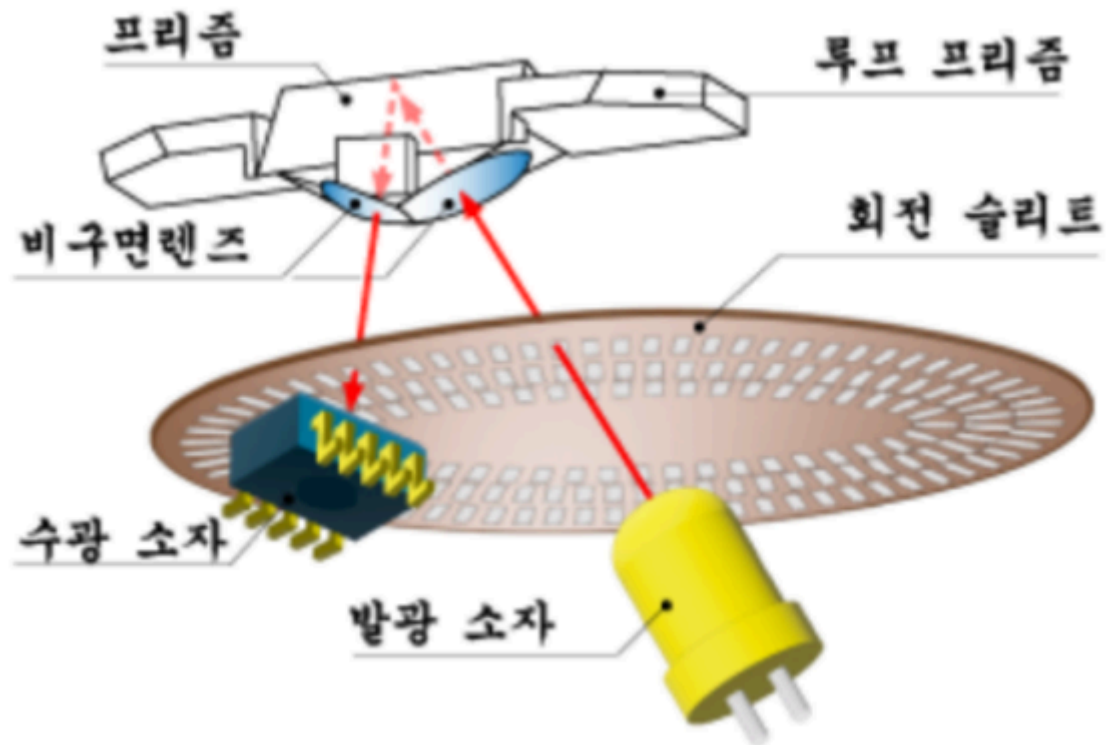
- 작동 원리 : 회전자와 고정자 모두에 슬릿을 뚫어 놓은 뒤, 고정된 광학 소자로부터 나오는 빛을 회전 소자 뒤의 수광 소자가 감지해 그 간격을 활용해 위치와 각속도 등을 측정한다.

## 2. 반사형 엔코더



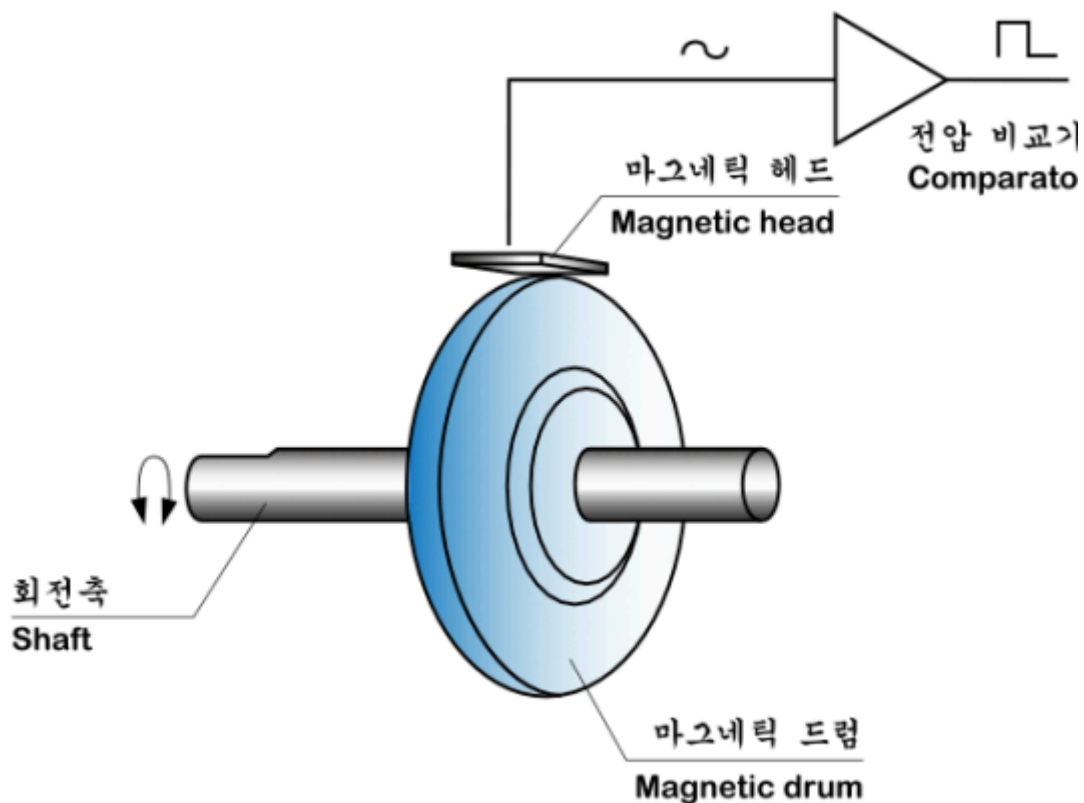
- 작동 원리 : 회전자에는 일정한 간격으로 렌즈와 프리즘이, 고정자에는 하나의 발광 소자 + 2개의 수광 소자가 위치한다. 발광 소자로부터 나온 빛이 회전자에서 반사되어 수광 소자에 도달한다. 이 때, 2개의 수광 소자를 사용하므로 둘의 수광 시간차로 각속도 등을 측정할 수 있다.

### 3. 복합(투과형 + 반사형) 엔코더



- 작동 원리 : 회전자에만 슬릿이 존재하고, 고정자에는 렌즈와 프리즘이 위치한다. 위의 그림과 같이 발광 소자에서 나온 빛이 고정자에서 반사된 뒤, 회전 슬릿과 반사된 상이 일치할 때 고정된 수광 소자로 빛이 도달한다. 이때, 회전자의 슬릿 개수를 조절하여 소형임에도 불구하고 고분해능을 실현할 수 있다.

### 자기식 엔코더



- 작동 원리 : 회전 디스크에 N극과 S극이 교대로 위치하여 회전자가 회전함에 따라 외측에 고정되어 있는 자기저항소자가 이를 감지해 전기 신호로 변환한다.
- 특징 : 광학식에 비해 소비전력이 낮음.

광학 또는 자기 센서를 사용하여 위치나 속도를 측정합니다.

## 특징

- 높은 정확도
- 산업용 로봇, 모터 제어 등에 사용

## 레졸버 (Resolver)

---

[관련 문서 link](#)

## 정의

회전 변위나 속도를 측정하는 아날로그 장치입니다.

## 동작 구조

회전 변위를 정현파 및 여현파 신호로 변환하여 측정합니다.

## 특징

- 견고한 구조
- harsh 환경에서도 안정적인 성능 제공

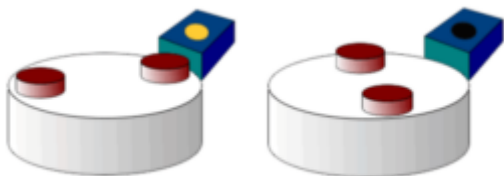
## 홀센서 (Hall Sensor)

---

[관련 문서 link](#)

## 정의

: 전하가 흐르는 도체 또는 반도체가 자기장을 인가하면서 전하 흐름에 수직한 방향으로 전위차가 형성되는 홀효과를 활용한 센서



## 동작 구조

홀 효과를 이용하여 자기장의 세기를 측정하고 전기 신호로 변환합니다.

## 특징

- 비접촉식 측정
- 위치, 속도, 전류 측정에 사용

## BISS-C (Bidirectional Synchronous Serial - C)

---

[관련 문서 link](#)

## 정의

비동기식 직렬 통신 프로토콜로, 엔코더와 같은 장치에서 사용됩니다.

## 동작 구조

데이터와 클럭 신호를 동시에 전송하여 실시간 데이터를 제공합니다.

## 특징

- 높은 데이터 전송 속도
- 실시간 위치 피드백 제공
- 산업 자동화 및 모터 제어 시스템에 사용