Ch.7 자동차의 Network

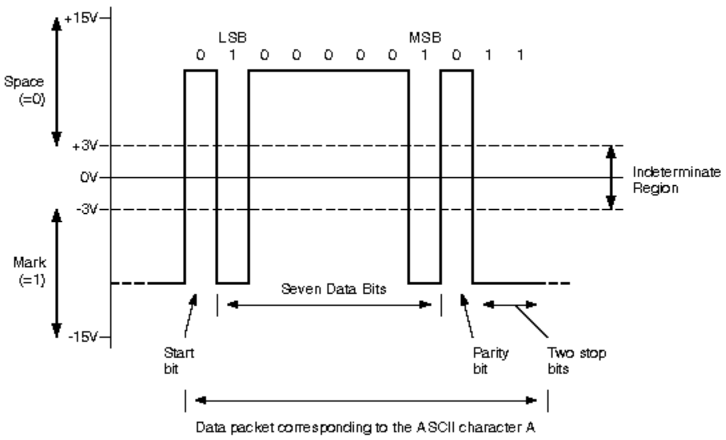
네트웍은 하드웨어 및 소프트 웨어에 따라 다양한 방법이 있다. 본 절에서는 가장 기본적으로 사용되는 통신 방법을 간략히 다루도록 한다.

7.1 Electronic Industries Association (EIA) standard RS-232C

- serial 통신 방법
- 전압변화를 이용하여 1 과 0을 표현
- 최소 3개의 선이 필요

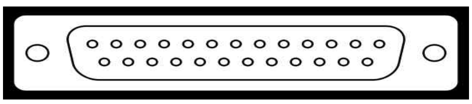
RS232 logic and voltage levels

Data circuits	Control circuits	Voltage
0 (space)	Asserted	+3 to +15 V
1 (mark)	Deasserted	−15 to −3 V



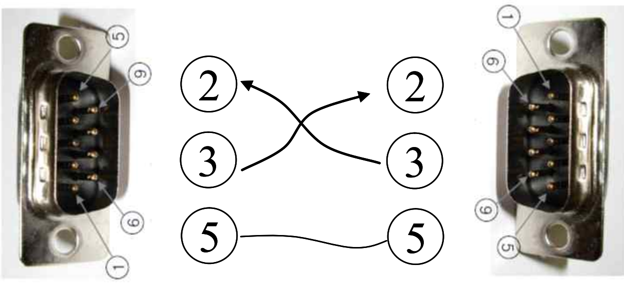
RS-232C 신호

* D Type 9 Pin and D Type 25 Pin Connectors



DB-25 pin connector

- DB9 코넥터를 이용할 경우 연결 방법



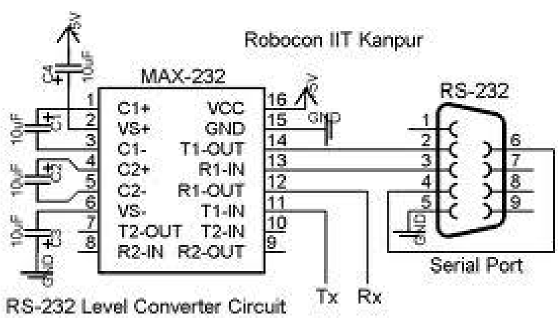
일반적으로 시리얼통신인 RS-232가 많이 사용되는 이유는 송신(3번핀),수신(2번핀) ,신호접지(5번핀)만 사용해도 기본적인 통신을 할 수 있습니다.

참고로 통신의 기본원리로 핸드쉐이크 (악수)라는 의미를 쓰는데 그 이유는 내가 오른손을 내밀면 상대편에서 내손을 맞잡을 수 있는 손은 내밀어야 합니다. 그래야 정상적인 인사가 됩니다.

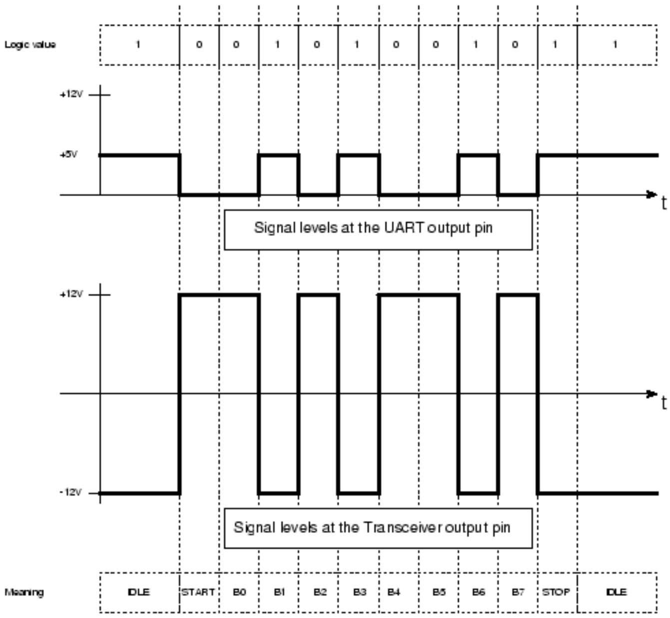
같은 원리로 송신핀은 상대방의 수신핀에, 수신핀은 상대방의 송신핀에 연결해야 합니다.

또한 RS-232는 전송신호의 레벨이 있는데 마이컴에서 발생되는 TTL레벨을 ±12V의 레벨로 통신을 합니다.

그래서 신호변환 회로를 아래와 같이 해주어야 합니다.(신호변환용 칩셋이 필요함)



RS232 Transmission of the letter 'J'



* 통신설정

통신속도(보드레이트); 1200, ... 9600 ...

데이터비트 수; 7, 8

패리티; none, odd, even

스톱비트; 1, 1.5, 2

통신속도의 단위는 bps(bit/sec) 1초당 정송할 수 있는 비트수를 말합니다.

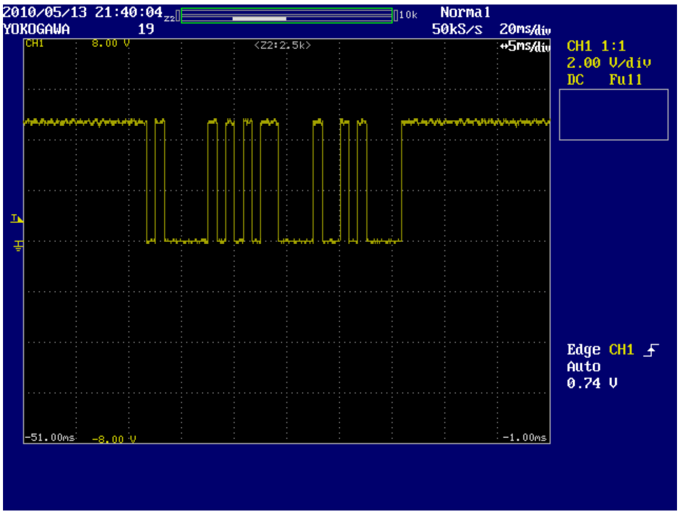
데이터 비트는 전송데이터의 규격을 말하는데 7비트,8비트 에서 설정합니다.

패리티 비트는 데이터 전달과정이 올바른지 체크하기위해서 만든 비트로 none, odd, even 을 선택합니다.

스톱비트는 전송종료를 상대방에게 알리는 비트 입니다. 1,1.5,2 비트중 하나를 선택합니다.

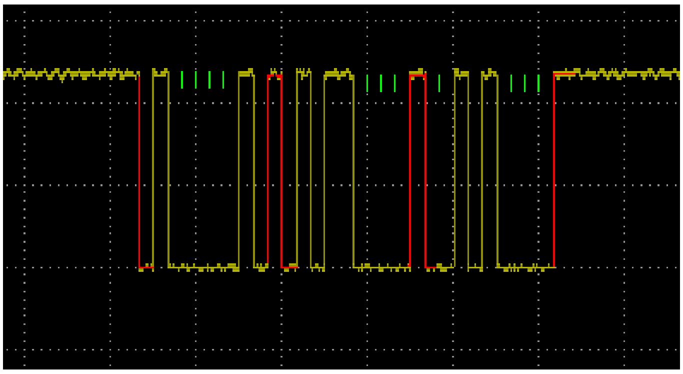
참고로 스타트비트는 1비트가 기본이며 H->L로 떨어지는 falling 신호입니다.

반대로 스톱비트는 L->H로 변하는 rising신호로 변합니다.



실제 측정된 신호는 아래와 같습니다.

위 신호는 아스키코드 A를 전달하는 과정입니다. 자세히 살펴보면 아래와 같습니다.



빨간 부분에서 H->L 부분이 스타트비트 H->L로 변경되는 부분이 스톱비트, 녹색은 제가 비트수를 체크하려고 임으로 그린 것입니다.

첫번째 데이터 값은 1000 0010 최하위비트 부터 전송한다고 했으니까, 역순으로 하면 0100 0001 16진수 41 아스키코드로는 A가 됩니다.

두번째 1011 0000 -> 0000 1101 16진수 0D 아스키 코드 CR(캐리지 리턴),

세번째 0101 0000 -> 0000 1010 16진수 0A 아스키 코드 LF(라인피드) 가 됩니다.

CR, LF는 수신창에서 다음 줄로 개행하는 의미로 하이퍼터미널 등에서 신규 데이터 수신시 다음 줄로 개행하는 기능을 가집니다.

The RS-232 standard defines the voltage levels that correspond to logical one and logical zero levels for the data transmission and the control signal lines. Valid signals are plus or minus 3 to 15 volts; the ±3 V range near zero volts is not a valid RS-232 level. The standard specifies a maximum open-circuit voltage of 25 volts: signal levels of ±5 V, ±10 V, ±12 V, and ±15 V are all commonly seen depending on the power supplies available within a device. RS-232 drivers and receivers must be able to withstand indefinite short circuit to ground or to any voltage level up to ±25 volts. The slew rate, or how fast the signal changes between levels, is also controlled.

For data transmission lines (TxD, RxD) and their secondary channel equivalents) logic one is defined as a negative voltage, the signal condition is called marking, and has the functional significance. Logic zero is positive and the signal condition is termed spacing.

1.	A "Space" (logic 0) will be between +3 and +25 Volts.
2.	A "Mark" (Logic 1) will be between -3 and -25 Volts.
3.	The region between +3 and -3 volts is undefined.
4.	An open circuit voltage should never exceed 25 volts. (In Reference to GND)
5.	A short circuit current should not exceed 500mA. The driver should be able to handle this without damage. (Take note of this one!)

`The electrical specifications of the serial port is contained in the EIA (Electronics Industry Association) RS232C standard. It states many parameters such as -

Above is no where near a complete list of the EIA standard. Line Capacitance, Maximum Baud Rates etc are also included. For more information please consult the EIA RS232-C standard. It is interesting to note however, that the RS232C standard specifies a maximum baud rate of 20,000 BPS!, which is rather slow by today's standards. A new standard, RS-232D has been recently released.

Serial Ports come in two "sizes", There are the D-Type 25 pin connector and the D-Type 9 pin connector both of which are male on the back of the PC, thus you will require a female connector on your device. Below is a table of pin connections for the 9 pin and 25 pin D-Type connectors.

Serial Pinouts (D25 and D9 Connectors)

D Type 9 Pin and D Type 25 Pin Connectors

D-Type-25 Pin No.	D-Type-9 Pin No.	Abbreviation	Full Name
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

Pin Functions

Abbreviation	Full Name	Function
TD	Transmit Data	Serial Data Output (TXD)
RD	Receive Data	Serial Data Input (RXD)
CTS	Clear to Send	This line indicates that the Modem is ready to exchange data.
DCD	Data Carrier Detect	When the modem detects a "Carrier" from the modem at the other end of the phone line, this Line becomes active.
DSR	Data Set Ready	This tells the UART that the modem is ready to establish a link.
DTR	Data Terminal Ready	This is the opposite to DSR. This tells the Modem that the UART is ready to link.
RTS	Request To Send	This line informs the Modem that the UART is ready to exchange data.
RI	Ring Indicator	Goes active when modem detects a ringing signal from the PSTN.

7.2 RS-422

- differential voltage (+6V or －6V)를 이용하여 1 과 0을 표현
- TX+, TX-, RX+, RX- 신호선 이용

7.3 RS-485

- 2개의 선을 이용
- 2개 선의 전압차이를 이용하여 1 과 0을 표현
- 장치마다 국번 (station number)를 부여

RS232, RS422, RS423 and **RS485** are serial communication methods for computers and devices. **RS232** is without doubt the best known interface, because this serial interface is implemented on almost all computers available today. But some of the other interfaces are certainly interesting because they can be used in situations where **RS232** is not appropriate. We will concentrate on the **RS485** interface here.

RS232 is an interface to connect one **DTE**, *data terminal equipment* to one **DCE**, *data communication equipment* at a maximum speed of 20 kbps with a maximum cable length of 50 feet. This was sufficient in the old days where almost all computer equipment were connected using modems, but soon after people started to look for interfaces capable of one or more of the following:

- Connect DTE's directly without the need of modems
- Connect several DTE's in a network structure
- Ability to communicate over longer distances
- Ability to communicate at faster communication rates

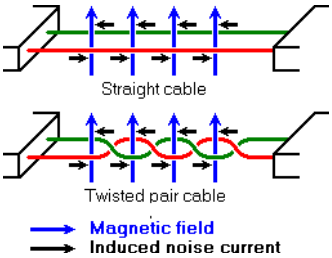
RS485 is the most versatile communication standard in the standard series defined by the EIA, as it performs well on all four points. That is why **RS485** is currently a widely used communication interface in data acquisition and control applications where multiple nodes communicate with each other.

Differential signals with RS485: Longer distances and higher bit rates

One of the main problems with **RS232** is the lack of immunity for noise on the signal lines. The transmitter and receiver compare the voltages of the data- and handshake lines with one common zero line. Shifts in the ground level can have disastrous effects. Therefore the trigger level of the **RS232** interface is set relatively high at

±3 Volt. Noise is easily picked up and limits both the maximum distance and communication speed. With **RS485** on the contrary there is no such thing as a common zero as a signal reference. Several volts difference in the ground level of the **RS485** transmitter and receiver does not cause any problems. The **RS485** signals are floating and each signal is transmitted over a **Sig+** line and a **Sig-** line. The **RS485** receiver compares the *voltage difference* between both lines, instead of the *absolute voltage level* on a signal line. This works well and prevents the existence of ground loops, a common source of communication problems. The best results are achieved if the **Sig+** and **Sig-** lines are twisted. The image below explains why.

Noise in straight and twisted pair cables



In the picture above, noise is generated by magnetic fields from the environment. The picture shows the magnetic field lines and the noise current in the **RS485** data lines that is the result of that magnetic field. In the straight cable, all noise current is flowing in the same direction, practically generating a looping current just like in an ordinary transformer. When the cable is twisted, we see that in some parts of the signal lines the direction of the noise current is the oposite from the current in other parts of the cable. Because of this, the resulting noise current is many factors lower than with an ordinary straight cable. Shielding—which is a common method to prevent noise in **RS232** lines—tries to keep hostile magnetic fields away from the signal lines. Twisted pairs in **RS485** communication however adds immunity which is a much better way to fight noise. The magnetic fields are allowed to pass, but do no harm. If high noise immunity is needed, often a combination of twisting and shielding is used as for example in **STP**, *shielded twisted pair* and **FTP**, *foiled twisted pair* networking cables. Differential signals and twisting allows **RS485** to communicate over much longer communication distances than achievable with **RS232**. With **RS485** communication distances of 1200 m are possible.

Differential signal lines also allow higher bit rates than possible with non-differential connections. Therefore **RS485** can overcome the practical communication speed limit of **RS232**. Currently **RS485** drivers are produced that can achieve a bit rate of 35 mbps.

Characteristics of RS485 compared to RS232, RS422 and RS423

	RS232	RS423	RS422	RS485
Differential	no	no	yes	yes
Max number of drivers Max number of receivers	1 1	1 10	1 10	32 32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m Max speed at 1200 m	20 kbs (1 kbs)	100 kbs 1 kbs	10 Mbs 100 kbs	35 Mbs 100 kbs
Max slew rate	30 V/μs	adjustabl e	n/a	n/a
Receiver input resistance	3..7 kΩ	≧ 4 kΩ	≧ 4 kΩ	≧ 12 kΩ
Driver load impedance	3..7 kΩ	≧ 450 Ω	100 Ω	54 Ω
Receiver input sensitivity	±3 V	±200 mV	±200 mV	±200 mV
Receiver input range	±15 V	±12 V	±10 V	-7..12 V
Max driver output voltage	±25 V	±6 V	±6 V	-7..12 V
Min driver output voltage (with load)	±5 V	±3.6 V	±2.0 V	±1.5 V

What does all the information in this table tell us? First of all we see that the speed of the differential interfaces RS422 and RS485 is far superior to the single ended versions RS232 and RS423. We also see that there is a maximum slew rate defined for both RS232 and RS423. This has been done to avoid reflections of signals. The maximum slew rate also limits the maximum communication speed on the line. For both other interfaces—RS422 and RS485—the slew rate is indefinite. To avoid reflections on longer cables it is necessary to use appropriate termination resistors.

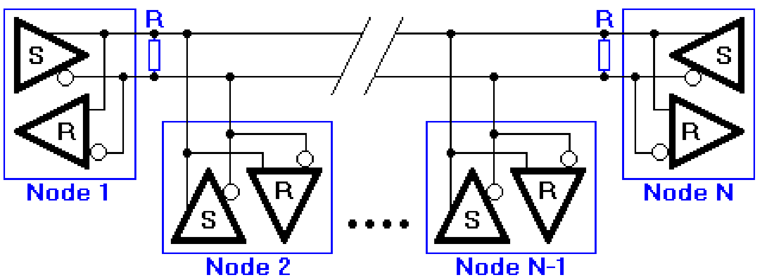
We also see that the maximum allowed voltage levels for all interfaces are in the same range, but that the signal level is lower for the faster interfaces. Because of this RS485 and the others can be used in situations with a severe ground level shift of several volts, where at the same time high bit rates are possible because the transition between logical 0 and logical 1 is only a few hundred millivolts.

Interesting is, that RS232 is the only interface capable of full duplex communication. This is, because on the other interfaces the communication channel is shared by multiple receivers and—in the case of RS485—by multiple senders. RS232 has a separate communication line for transmitting and receiving which—with a well written protocol—allows higher effective data rates at the same bit rate than the other interfaces. The request and acknowledge data needed in most protocols does not consume bandwidth on the primary data channel of RS232.

Network topology with RS485

Network topology is probably the reason why RS485 is now the favorite of the four mentioned interfaces in data acquisition and control applications. RS485 is the only of the interfaces capable of internetworking multiple transmitters and receivers in the same network. When using the default RS485 receivers with an input resistance of 12 kΩ it is possible to connect 32 devices to the network. Currently available high-resistance RS485 inputs allow this number to be expanded to 256. RS485 repeaters are also available which make it possible to increase the number of nodes to several thousands, spanning multiple kilometers. And that with an interface which does not require intelligent network hardware: the implementation on the software side is not much more difficult than with RS232. It is the reason why RS485 is so popular with computers, PLCs, micro controllers and intelligent sensors in scientific and technical applications.

RS485 network topology



In the picture above, the general network topology of RS485 is shown. N nodes are connected in a multipoint RS485 network. For higher speeds and longer lines, the termination resistances are necessary on both ends of the line to eliminate reflections. Use 100 Ω resistors on both ends. The RS485 network must be designed as one line with multiple drops, not as a star. Although total cable length maybe shorter in a star configuration, adequate termination is not possible anymore and signal quality may degrade significantly.

RS485 functionality

And now the most important question, how does RS485 function in practice? Default, all the senders on the RS485 bus are in tri-state with high impedance. In most higher level protocols, one of the nodes is defined as a master which sends queries or commands over the RS485 bus. All other nodes receive these data. Depending of the information in the sent data, zero or more nodes on the line respond to the master. In this situation, bandwidth can be used for almost 100%. There are other implementations of RS485 networks where every node can start a data session on its own. This is comparable with the way ethernet networks function. Because there is a chance of data collision with this implementation, theory tells us that in this case only 37% of the bandwidth will be effectively used. With such an implementation of a RS485 network it is necessary that there is error detection implemented in the higher level protocol to detect the data corruption and resend the information at a later time.

There is no need for the senders to explicity turn the RS485 driver on or off. RS485 drivers automatically return to their high impedance tri-state within a few microseconds after the data has been sent. Therefore it is not needed to have delays between the data packets on the RS485 bus.

RS485 is used as the electrical layer for many well known interface standards, including Profibus and Modbus. Therefore RS485 will be in use for many years in the future.

7.4 Controller Area Network

ISO 11898-1:2003, Part 1: Data link layer and physical signalling
ISO 11898-1:2003/Cor 1:2006
ISO 11898-2:2003, Part 2: High-speed medium access unit
ISO 11898-3:2006, Part 3: Low-speed, fault-tolerant, medium-dependent interface
ISO 11898-4:2004, Part 4: Time-triggered communication
ISO 11898-5:2007, Part 5: High-speed medium access unit with low-power mode
ISO 16845:2004, Conformance test plan

Frames

A CAN network can be configured to work with two different message (or "frame") formats: the standard or base frame format (described in CAN 2.0 A and CAN 2.0 B), and the extended frame format (only described by CAN 2.0 B). The only difference between the two formats is that the "CAN base frame" supports a length of 11 bits for the identifier, and the "CAN extended frame" supports a length of 29 bits for the identifier, made up of the 11-bit identifier ("base identifier") and an 18-bit extension ("identifier extension"). The distinction between CAN base frame format and CAN extended frame format is made by using the IDE bit, which is transmitted as dominant in case of an 11-bit frame, and transmitted as recessive in case of a 29-bit frame. CAN controllers that support extended frame format messages are also able to send and receive messages in CAN base frame format. All frames begin with a start-of-frame (SOF) bit that denotes the start of the frame transmission.
CAN has four frame types:

- Data frame: a frame containing node data for transmission
- Remote frame: a frame requesting the transmission of a specific identifier
- Error frame: a frame transmitted by any node detecting an error
- Overload frame: a frame to inject a delay between data and/or remote frame

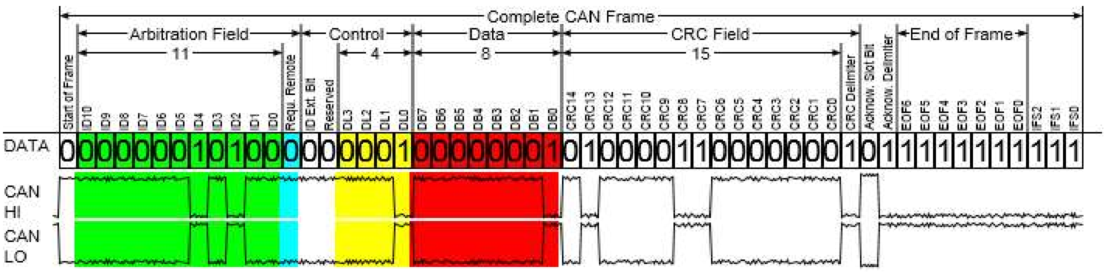
Data frame

The data frame is the only frame for actual data transmission. There are two message formats:

- Base frame format: with 11 identifier bits
- Extended frame format: with 29 identifier bits

The CAN standard requires the implementation must accept the base frame format and may accept the extended frame format, but must tolerate the extended frame format.

Base frame format



CAN-Frame in base format with electrical levels without stuffbits

The frame format is as follows:

Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier (green)	11	A (unique) identifier for the data which also represents the message priority
Remote transmission request (RTR)	1	Dominant (0) (see Remote Frame below)
Identifier extension bit (IDE)	1	Declaring if 11 bit message ID or 29 bit message ID is used. Dominant (0) indicate 11 bit message ID while Recessive (1) indicate 29 bit message.
Reserved bit (r0)	1	Reserved bit (it must be set to dominant (0), but accepted as either dominant or recessive)
Data length code (DLC) (yellow)	4	Number of bytes of data (0–8 bytes)[a]
Data field (red)	0–64 (0–8 bytes)	Data to be transmitted (length in bytes dictated by DLC field)
CRC	15	Cyclic redundancy check
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)

1. It is physically possible for a value between 9–15 to be transmitted in the 4-bit DLC, although the data is still limited to eight bytes. Certain controllers allow the transmission and/or reception of a DLC greater than eight, but the actual data length is always limited to eight bytes.

Extended frame format

The frame format is as follows:

Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier A	11	First part of the (unique) identifier for the data which also represents the message priority
Substitute remote request (SRR)	1	Must be recessive (1). Optional
Identifier extension bit (IDE)	1	Must be recessive (1). Optional
Identifier B	18	Second part of the (unique) identifier for the data which also represents the message priority
Remote transmission request (RTR)	1	Must be dominant (0)
Reserved bits (r0, r1)	2	Reserved bits (it must be set dominant (0), but accepted as either dominant or recessive)
Data length code (DLC)	4	Number of bytes of data (0–8 bytes)[a]
Data field	0 – 64 (0 – 8 bytes)	Data to be transmitted (length dictated by DLC field)
CRC	15	Cyclic redundancy check
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)

1. It is physically possible for a value between 9 – 15 to be transmitted in the 4-bit DLC, although the data is still limited to eight bytes. Certain controllers allow the transmission and/or reception of a DLC greater than eight, but the actual data length is always limited to eight bytes.
2.

The two identifier fields (A & B) combine to form a 29-bit identifier.

Remote frame

- Generally data transmission is performed on an autonomous basis with the data source node (e.g., a sensor) sending out a Data Frame. It is also possible, however, for a destination node to request the data from the source by sending a Remote Frame.
- There are two differences between a Data Frame and a Remote Frame. Firstly the RTR-bit is transmitted as a dominant bit in the Data Frame and secondly

in the Remote Frame there is no Data Field.

i.e.,

RTR = 0 ; DOMINANT in data frame

RTR = 1 ; RECESSIVE in remote frame

In the very unlikely event of a Data Frame and a Remote Frame with the same identifier being transmitted at the same time, the Data Frame wins arbitration due to the dominant RTR bit following the identifier. In this way, the node that transmitted the Remote Frame receives the desired data immediately.

Error frame

The error frame consists of two different fields:

- The first field is given by the superposition of ERROR FLAGS (6 - 12 dominant/recessive bits) contributed from different stations.
- The following second field is the ERROR DELIMITER (8 recessive bits).

There are two types of error flags:

Active Error Flag

six dominant bits – Transmitted by a node detecting an error on the network that is in error state "error active".

Passive Error Flag

six recessive bits – Transmitted by a node detecting an active error frame on the network that is in error state "error passive".

Overload frame

The overload frame contains the two bit fields Overload Flag and Overload Delimiter. There are two kinds of overload conditions that can lead to the transmission of an overload flag:

1. The internal conditions of a receiver, which requires a delay of the next data frame or remote frame.
1. Detection of a dominant bit during intermission.

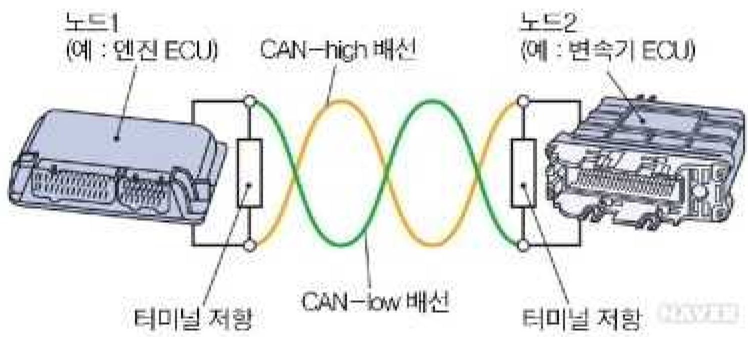
The start of an overload frame due to case 1 is only allowed to be started at the first bit time of an expected intermission, whereas overload frames due to case 2 start one bit after detecting the dominant bit. Overload Flag consists of six dominant bits. The overall form corresponds to that of the active error flag. The overload flag's form destroys the fixed form of the intermission field. As a consequence, all other stations also detect an overload condition and on their part start transmission of an overload flag. Overload Delimiter consists of eight recessive bits. The overload delimiter is of the same form as the error delimiter.

* 참고문헌
CAN [Controller Area Network]
(최신자동차공학시리즈 3 - 첨단자동차전기전자, 2012.9.5, 도서출판 골든벨)

(1) CAN 데이터 버스의 특징

① CAN 데이터 버스 등급 B와 등급 C로 구별된다.
② 최대 데이터 전송률은 등급 B에서는 125kBd, 등급 C에서는 1MBd이다.
③ CAN 데이터 버스 시스템은 2개의 배선을 이용하여 데이터를 전송한다.
④ CAN 등급 B는 단일배선 적응능력이 있다.
⑤ CAN 등급 C는 단일배선 적응능력이 없다.

단일 배선 적응능력(suitability against single line)이란 CAN 데이터 버스 시스템에서 배선 하나가 단선 또는 단락되어도 나머지 1개의 배선이 통신능력을 정확하게 그대로 유지하는 것을 말한다. 그러나 버스 시스템이 단일배선 모드로 바뀌면, 간섭 저항성은 더 이상 보장되지 않는다. 경우에 따라서는 기능장애가 발생할 수도 있다.



(2) CAN 데이터 버스의 구성

CAN 데이터 버스 시스템은 최소한 2개의 노드(node), CAN-low 배선, CAN-high 배선, 그리고 최소한 2개의 터미널 저항(terminal resistor)으로 구성된다.

① 노드(nodes) - 버스 시스템을 구성하는 다수의 스테이션(예 : ECU)
CAN 버스 노드의 내부구조는 LIN-버스 노드의 내부구조와 동일하다. 데이터 전송률을 더 빠르게 하고, LIN-데이터 버스에서와는 다른 전압수준을 버스 배선에 인가하기 위해서 고품질 컨트롤러 및 트랜시버를 사용한다.

② 버스 배선(CAN-high, CAN-low)

노드의 트랜시버에 의해 CAN-high 배선에 우성 수준(Udom)이 형성되면, 이 배선의 전압은 상승한다. 동시에 CAN-low 배선의 전압은 하강한다. 이때의 논리값은 ‘0’이다. 두 배선은 서로 꼬여 있거나 또는 와이어-메쉬(wire mesh)에 의해 차폐되어 있다.

스위칭할 때마다 두 CAN-배선에서 생성되는 자장은, 전압이 서로 반대방향으로 변화하기 때문에 상쇄된다. 따라서 두 배선은 외부에 대해서는 전자적(電磁的)으로 중성이며, 어떠한 외부 간섭도 일으키지 않는다. 즉, 간섭에 대한 저항성이 보장된다.

논리값	LIN	CAN 등급 B		CAN 등급 C	
0	0V	low	1V	low	1.5V
		high	4V	high	3.5V
1	약 12V	low	5V	low	2.5V
		high	0V	high	2.5V

우성(dominant) 수준, 열성(recessive) 수준

③ 터미널 저항(terminal resistors : Abschlusswiderstand)
터미널 저항은 CAN-high 배선과 CAN-low 배선 사이의 회로를 연결한다. 이를 통해 CAN-버스 배선에서 반사(reflection)가 발생되는 것을 방지한다.

터미널 저항은 주로 노드(node)에 설치되어 있다.

터미널 저항이 없는 CAN-버스 배선은, 특히 CAN 등급 C 시스템에서는 기능적인 고장의 원인이 될 수 있다. 따라서 고장의 경우에는 반드시 터미널 저항을 먼저 점검해야 한다.

CAN 등급 C 시스템에서는 저항측정기를 이용하여 CAN-배선의 접점에서 터미널 저항을 테스트할 수 있다.

(3) CAN의 작동원리

① 멀티-마스터(multi-master) 원리
버스 배선을 통해 정보를 송/수신하고 있는 중이 아니라면, 멀티-마스터 원리에 따라 각 노드(예 : ECU)는 버스 배선에 메시지를 전송할 수 있다. 다수의 ECU가 동시에 메시지를 전송하고자 할 경우는, 중재(arbitration)를 통해 가장 중요한 메시지를 가장 먼저 전송한다.

② 중재(仲裁 ; arbitration)

하나의 자원에 대하여 복수의 프로세스나 이용자가 행한, 경합하는 요구를 감시하고 관리, 중재하는 과정(process)을 말한다.

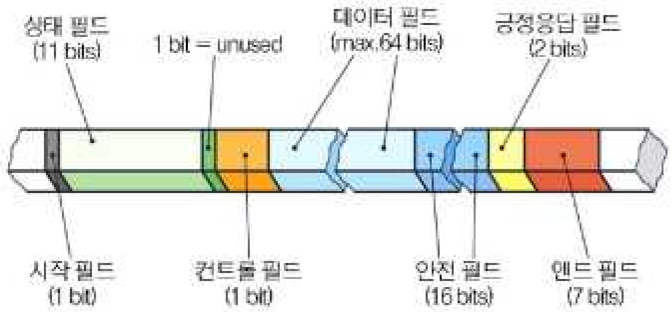
다수의 ECU가 동시에 메시지를 전송하고자 할 때, 데이터 버스 배선으로의 접근을 통제한다. 메시지의 중요도(우선순위)는 식별자(ID)에 의해 정의된다. 식별자(ID)가 낮으면 낮을수록, 우선순위는 더 높다. 중재를 조정(調停 ; arbitration)이라고도 한다.

(4) 데이터 프로토콜(data protocol)의 구성

데이터 프로토콜은 데이터 메시지의 구조를 결정하며, 표준화되어 있다.

CAN 데이터 버스에서, 메시지의 길이는 128bit까지이다. 이들은 서로 연속된 필드(field)로 분할되어 있다. CAN-버스 등급 C에서 클록 주파수가 5kHz인 경우, 128bit의 메시지를 전송하는 데는 약 0.25ms가 소요된다.

[그림 10-9] CAN-메시지의 구조



① 시작 필드(start field)(1bit)
이 필드는 메시지의 시작을 나타내고, 메시지 전송이 시작되었음을 모든 노드에 알린다. 노드들 즉, ECU들은 동기(synchronizing)된다.

② 상태 필드(status field)(11bit)
이 필드는 메시지 식별자(메시지 논리기호)로 구성되어 있다. 노드들은 식별자(ID)에 근거하여, 메시지의 내용을 확인한다. 또 어느 발신자가 먼저 발신해야 하는지 즉, 중재(仲裁)도 식별자(ID)에 근거하여 실행한다.

③ 컨트롤(control) 필드, 안전(safety) 필드 및 긍정응답(acknowledgement) 필드
이들은 데이터의 전송을 보장하는데 사용된다. 메시지 발신자는, 수신자에 의해 메시지가 정확하게 판독되었는지의 여부를 확인하는데 긍정응답 필드를 사용한다. 긍정응답이 없을 경우, 메시지는 반복해서 발신된다. 여러 번 시도해도 수신자로부터 긍정응답이 없을 경우에 발신자는 메시지 발신을 중단한다. 따라서 하나의 노드에 고장이 있을 경우에도, 전체 버스

시스템의 고장은 방지된다.

컨트롤-필드는 6bit, 안전-필드는 16bit, 긍정응답-필드는 2bit로 구성되어 있다.

④ 데이터(data) 필드(max. 64bit)
데이터 필드에는 메시지의 유용한 정보들(예 : 기관회전속도, 기관온도, 스로틀밸브개도 등)이 포함되어 있다.

⑤ 엔드(end) 필드(7bit)
메시지의 끝을 표시하고, 다음 메시지를 위해 버스를 자유로운 대기상태로 만든다.

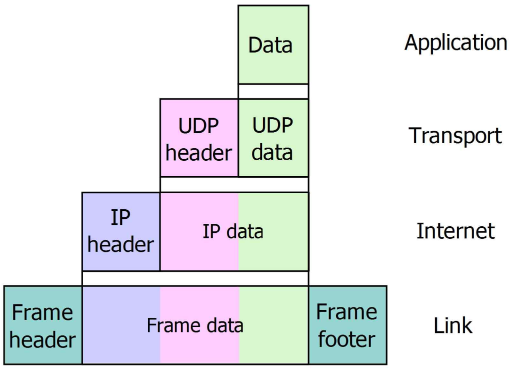
7.5 Ethernet

IEEE 802.3 Standard for Ethernet

방송 방식

10BASE-5; 10 Mbits/s (1.25 MB/s) 가능, thick coex cable 사용, IEEE802.3 standard
10BASE-T; 10 Mbits/s (1.25 MB/s) 가능, twisted pair cable 사용, IEEE802.3i
1000BASE-T; 1 Gbits/s 가능, RJ-45, IEEE802.3ab

TCP/IP; 이더넷을 사용하기 위한 프로토콜 규약
the Transmission Control Protocol (TCP) and the Internet Protocol (IP)



https://en.wikipedia.org/wiki/Internet_protocol_suite

7.6 Field bus

여러가지 장치를 연결하기 위한 산업현장의 통신 방법

- Type 3 PROFIBUS

* filed bus 이외의 산업현장의 통신 방법들

- CAN (Controller Area Network)
  자동차 내의 통신을 위하여 Bosch 가 개발

- EtherCAT