

임베디드 시스템 실험 5

목표

◆ Joystick을 이용하여 CAN통신을 통해 DC, Servo Motor 제어

■ CAN통신

- ✓ CAN통신 이론 및 내용 이해
- ✓ CAN통신 코드구현

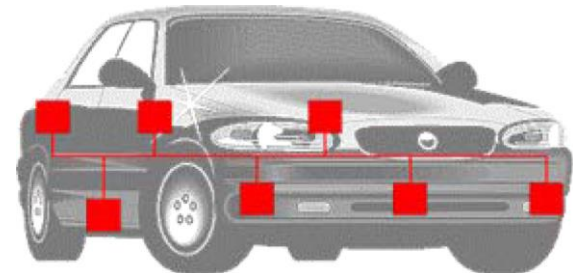
■ Joystick값을 받아, PWM을 이용하여 Motor제어

- ✓ Joystick의 원리 파악
- ✓ 하나의 Node에서 다른 Node로 Data 전송 CAN통신 구현

CAN(Controller Area Network)

◆ CAN통신 이란?

- 1986년 자동차 내부의 서로 다른 3개의 ECU(electronic control unit) 간의 통신을 벤츠(Benz)의 요구로 독일 보쉬(Bosch)가 최초 개발
- UART는 일대일(point-to-point) 통신이므로 서로 다른 3개의 ECU 간의 통신방식으로 적합하지 않아 다중통신(multi-master communication) 방식이 요구되어 CAN 프로토콜 개발
- 최초의 집적화된 CAN 부품은 1987년 인텔(Intel) 생산
- 최근 반도체 및 마이크로프로세서 제조사에서 CAN 프로토콜을 채용한 프로세서를 출시 및 활용



CAN(Controller Area Network)

◆ CAN통신 역사

- 1986 : 독일 BOSCH사 CAN 발표
- 1987 : CAN Chip 상용화(Intel 82526/Philips 82C200)
- 1989 : SAAB Training Target Control에 적용
- 1991 : Extended CAN 발표 / Mercedes S model에 적용
- 1992 : CiA (CAN in Automation) 결성 / CAN Kingdom 발표
- 1993 : ISO 11898 제정
- 1994 : DeviceNet (FA용) 발표
- 1995 : CANOpen (embedded network용) 발표
- 2000 : CAN device 판매량 1억개 돌파
- CAN 적용 차량 : Mercedes-Benz, Volvo, Volkswagen, BMW, Renault, Fiat 등

CAN(Controller Area Network)

◆ CAN통신의 특징

▪ 경제적 솔루션

- ✓ 우수한 가격 대비 성능
- ✓ 자동차와 같은 대량 생산에 사용될 수 있는 저가 디바이스 풍부

▪ 우수한 신뢰성

- ✓ 정교한 오류 검출과 오류 처리 구조 ⇒ 높은 전송 신뢰성
- ✓ 오류 메시지는 반복 재전송
- ✓ 심각한 오류 발생 노드는 버스 상에서 자동 방출
- ✓ 모든 노드에 오류가 알려지므로 시스템의 전체 데이터 일관성 유지
- ✓ differential line ⇒ 전자파 간섭에 대한 강한 내성

▪ 표준화

- ✓ ISO 11898(고속 애플리케이션)
- ✓ ISO 11519-2(저속 애플리케이션 : < 125Kbps)

CAN(Controller Area Network)

- **실시간성**
 - ✓ 최대 데이터 전송률 1M[bps] @ 25[m] bus length
 - ✓ 전송 요청과 실제 전송 시작 사이의 짧은 지연 시간
 - ✓ CSMA/CD 방식 버스 액세스 ⇒ 시간 손실 없이 우선 순위 순서로 메시지 전송
- **유연성**
 - ✓ Multi-Master 구조 가능 : 모든 노드들이 독립적으로 bus access
 - ✓ 노드 연결/분리 용이(Plug & Play)
- **다중 전송/방송 성능**
 - ✓ 노드 지향형이 아닌 메시지 지향형
 - ✓ 메시지 식별자가 메시지 내용과 우선순위 지정
 - ✓ 메시지는 동시에 모든 노드들로 보내지고, 모든 노드들은 공통 데이터를 동시에 수신하고 작업 가능

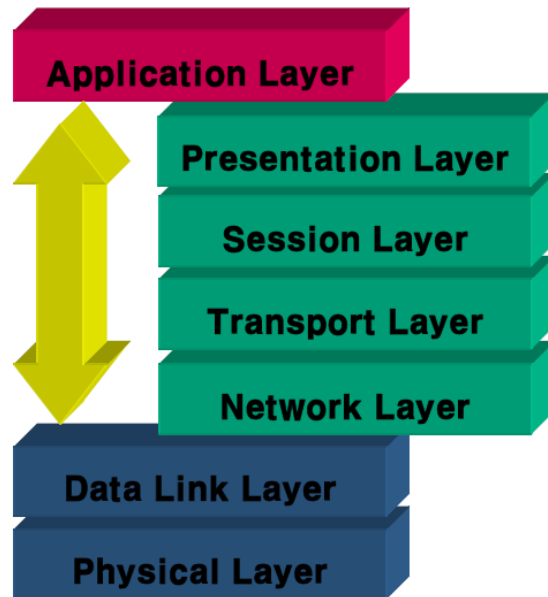
CAN(Controller Area Network)

◆ CAN프로토콜

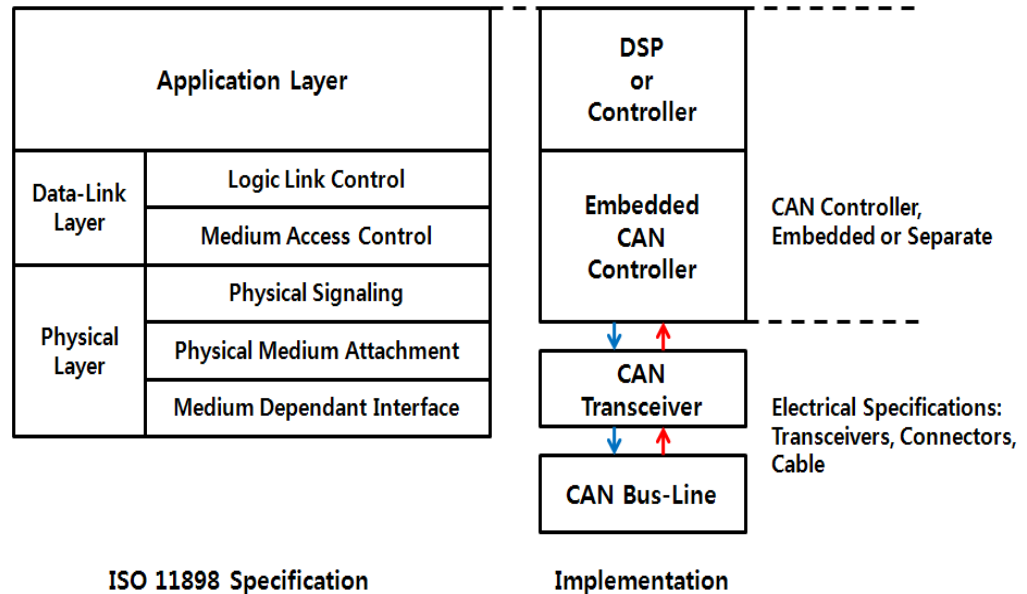
- CAN 메시지에 있는 식별자(ID, identifier)의 길이에 따라 2가지 모드로 구분된다
 - ✓ 표준 CAN 2.0A 버전 : 11 비트 식별자
 - ✓ 표준 CAN 2.0B 버전 : 29 비트 식별자
- ISO 규격에 따라 두 종류의 통신 속도로 구분되며 물리계층에서 차이가 있다.
 - ✓ ISO11898 : 1Mbps 이상의 고속 통신
 - ✓ ISO11519 : 125kbps 까지의 저속 통신

CAN(Controller Area Network)

◆ CAN 계층구조

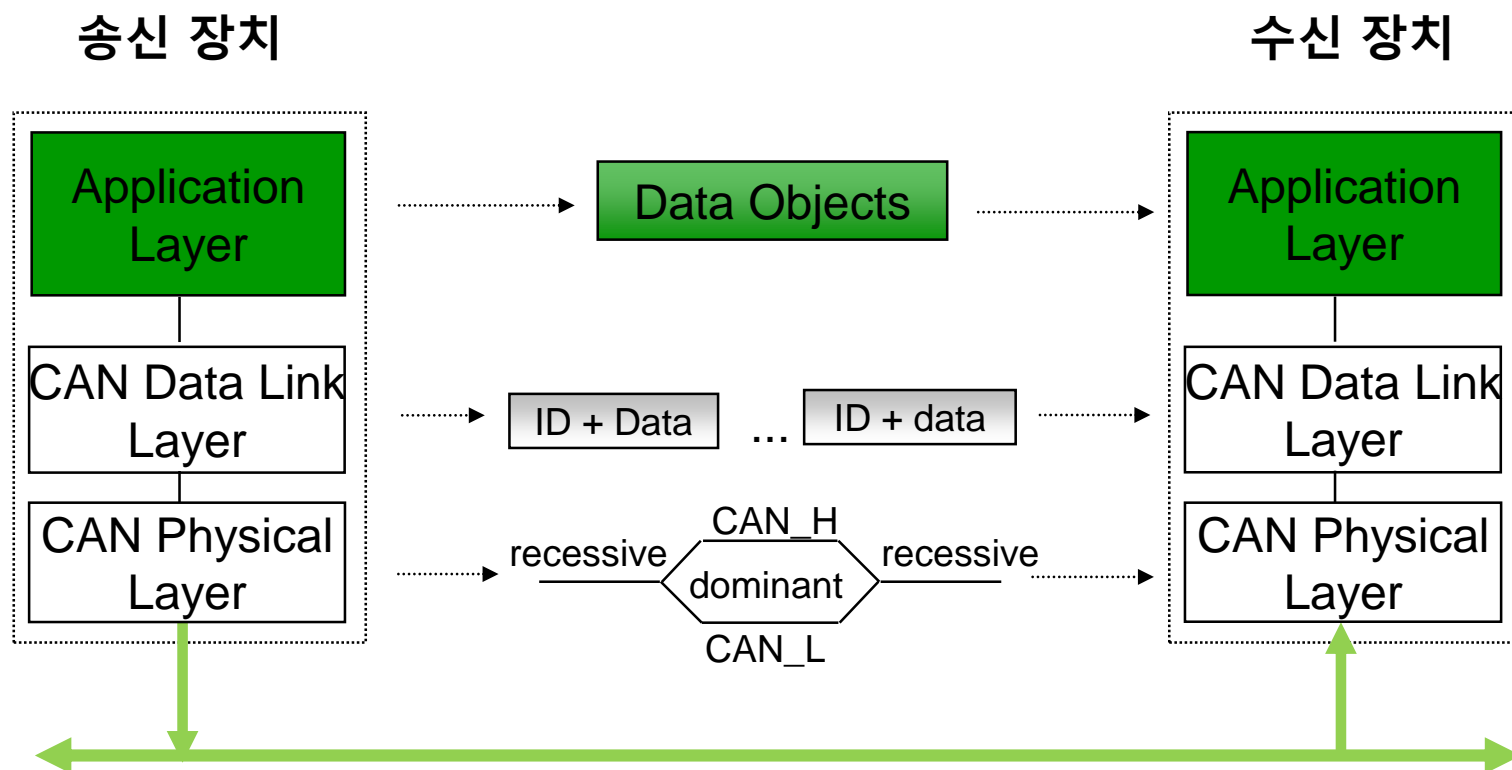


ISO/OSI Reference Model



CAN(Controller Area Network)

◆ CAN 계층별 상호 작용



CAN(Controller Area Network)

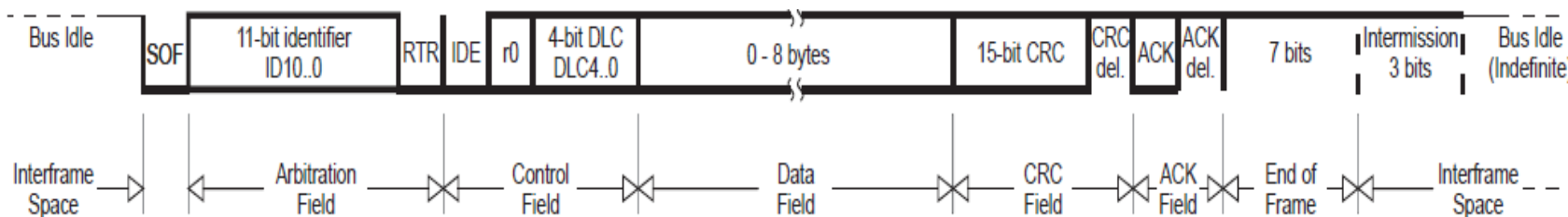
◆ Bus Length

- Maximum 1M[bps] @ 25[m] bus length
- Maximum 650[m] bus length @ 100K[bps]

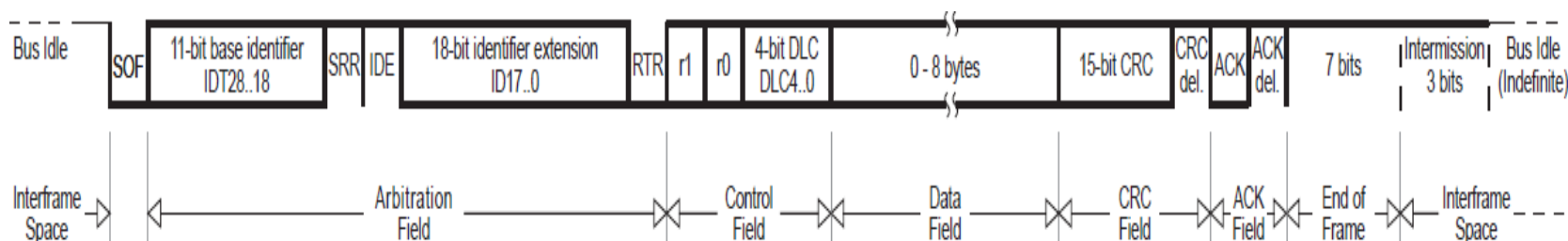
Bus Length [m]	MAX. Rate [Kbps]	Bit Time [usec.]	
25	1,000	1	고속
50	800	1.25	
100	500	2	
250	250	4	
500	125	8	
650	100	10	저속
1k	50	20	
2.5k	20	50	
5k	10	100	

CAN(Controller Area Network)

◆ Data Frame 구조



CAN Standard Data Frame(CAN 2.0 A)



CAN Extended Data Frame(CAN 2.0 B)

CAN(Controller Area Network)

◆ Data Frame 구조

- Start of Frame
 - ✓ 우성 Bit로 Frame 시작을 알림
- Identifier Field
 - ✓ 11 Bit(CAN 2.0A) / 29 Bit(CAN 2.0B)
 - ✓ 메시지의 우선순위와 내용을 반영
 - ✓ Arbitration 수행
- RTR(Remote Transmit Request) Bit
 - ✓ 원격전송요청 Bit
 - ✓ Data Frame(RTR = 0)과 Remote Frame(RTR = 1)을 구분
- IDE(Identifier Extension) Bit
 - ✓ 우성 Bit \Rightarrow Standard Frame
 - ✓ 열성 Bit \Rightarrow Extended Frame

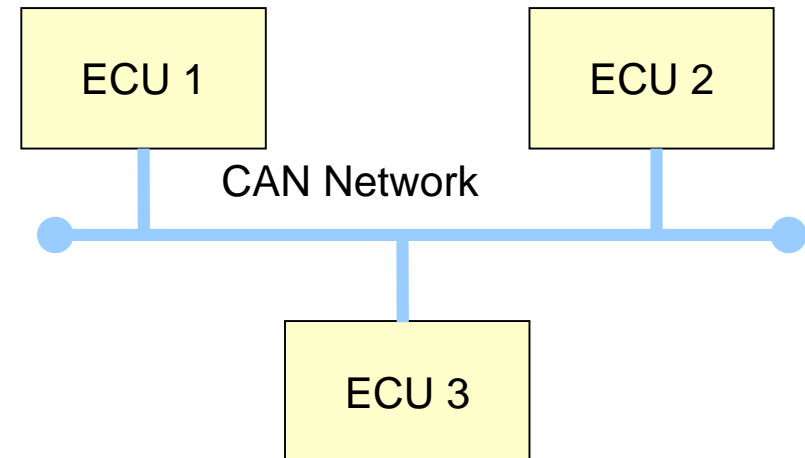
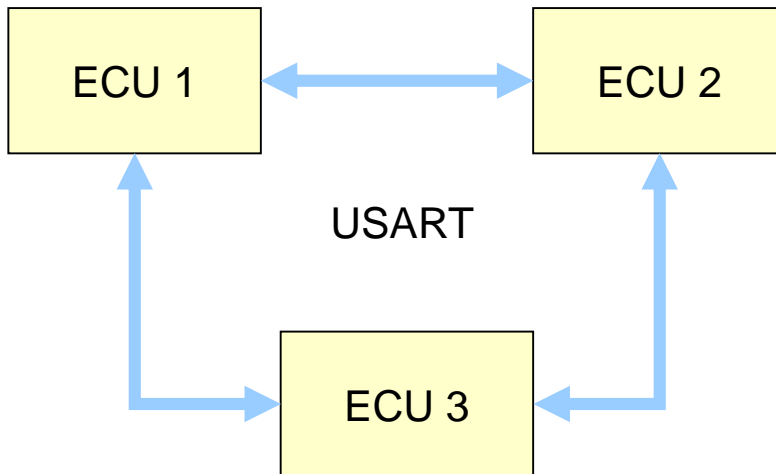
CAN(Controller Area Network)

◆ Data Frame 구조

- DLC(Data Length Code)
 - ✓ 데이터 사이즈 : 0~8 Byte
- CRC(Cyclic Redundancy Code)
 - ✓ 전송오류 탐색을 위한 Checksum
- CRC Delimiter : 열성 Bit로써 CRC Field를 구분
- ACK. Slot
 - ✓ 전송 노드는 열성 Bit 송신
 - ✓ 수신 노드에서 오류 없으면 우성 Bit 송신
- ACK. Delimiter : 열성 Bit로써 ACK. Slot을 구분
- End of Frame : 7개의 열성 Bit로 구성

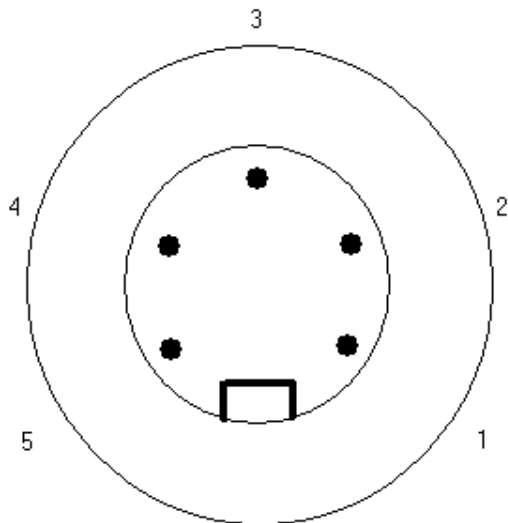
CAN(Controller Area Network)

◆ CAN 통신 방식



CAN(Controller Area Network)

◆ 5-pin Mini-C : DeviceNet 규격 (No Standard)

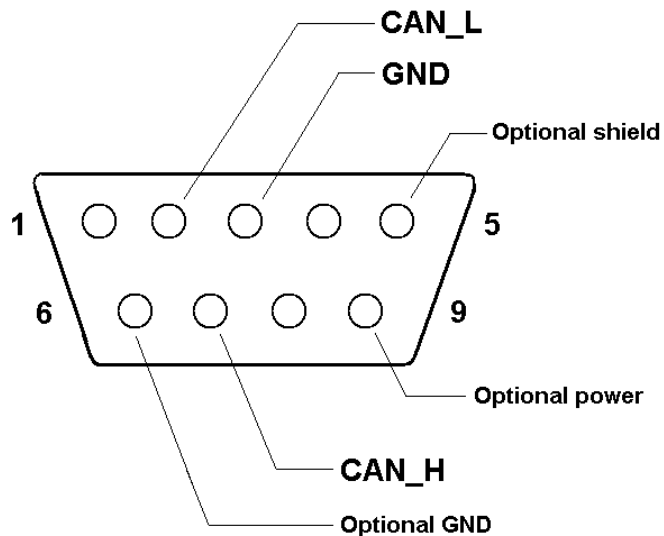


Male (pins)

Pin	Function	DeviceNet Color
1	Drain	Bare
2	V+	Red
3	V-	Black
4	CAN_H	White
5	CAN_L	Blue

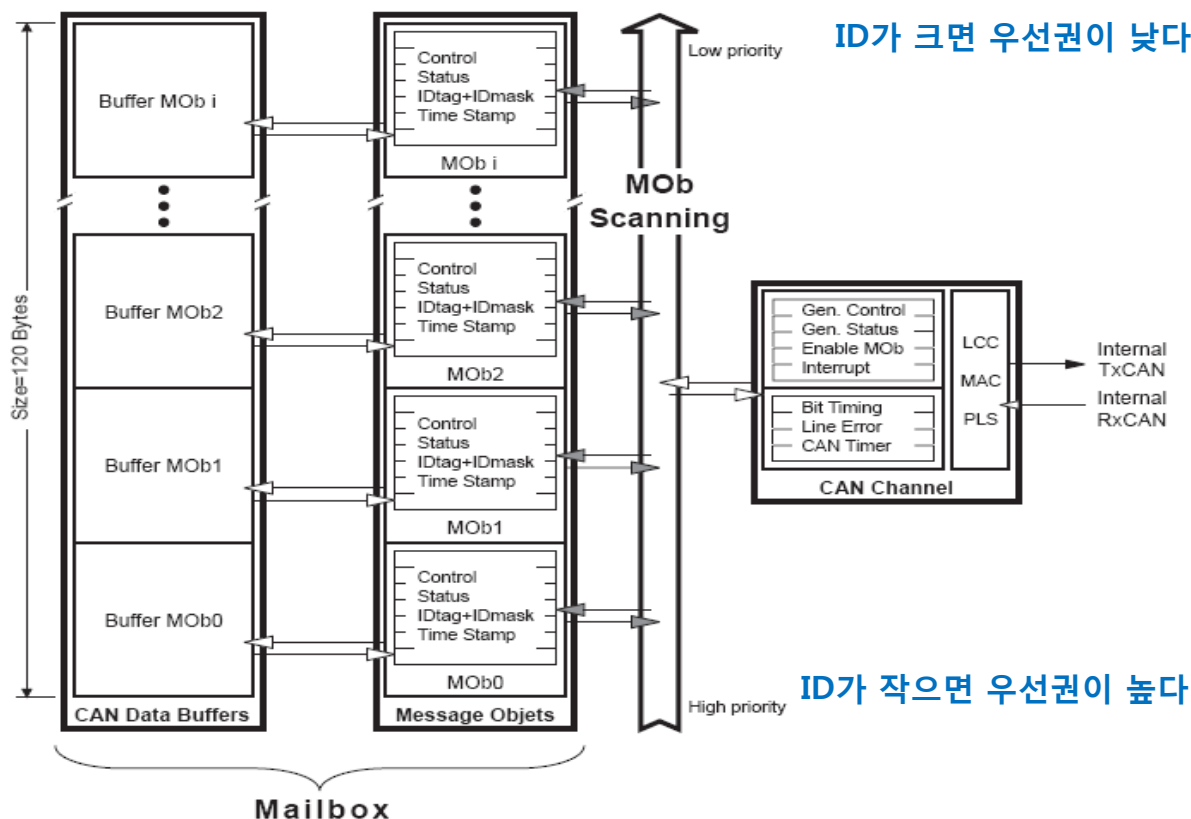
CAN(Controller Area Network)

◆ 9-pin D-sub (CiA 추천)



Pin	Function	Description
1	-	Reserved
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_GND	CAN Ground
4	-	Reserved
5	(CAN_SHLD)	Optional CAN shield
6	(GND)	Optional CAN ground
7	CAN_H	CAN_H bus line (dominant high)
8	-	Reserved (error line)
9	CAN_V+	Optional power

CAN(Controller Area Network)



- The identifier with the lowest binary number has the highest priority.

CAN(Controller Area Network)

◆ Message Objects(MObs)

- CAN 프레임 서술자(descriptor)이다.
- CAN 프레임을 처리할 모든 정보를 포함한다.
- 객체(object)처럼 CAN Message를 서술할 수 있다.
- MOB 세트는 MailBox 에서 전단에 있다.
- Mob는 독립적이며 우선순위가 있다.
- Mob 개수는 15개이다.

◆ MOB 동작 모드

- 불가 모드(Disable mode)
- 전송 모드(Transmit mode)
- 수신 모드(Receive mode)
- 자동응답 모드(Automatic reply)
- 프레임 버퍼 수신 모드(Frame buffer receive mode)

CAN(Controller Area Network)

◆ Operating Modes

Mob configuration		Reply Valid	RTR Tag	Operation Mode
0	0	X	X	Disable
0	1	X	0	Tx Data Frame
		X	1	Tx Remote Frame
1	0	X	0	Rx Data Frame
		0	1	Rx Remote Frame
		1		Rx Remote Frame Tx Data Frame(reply)
1	1	x	x	Frame Buffer Receive Mode

- There is no default mode after RESET. Before enabling the CAN peripheral, each MOB must be configured (ex: disabled mode- CONMOB=00).

CAN(Controller Area Network)

◆ General Registers

- CANGCON- CAN General Control Register
- CANGSTA- CAN General Status Register
- CANGIT- CAN General Interrupt Register
- CANGIE- CAN General Interrupt Enable Register
- CANEN2, CANEN1- CAN Enable MOB Register
- CANIE2, CANIE1- CAN Enable Interrupt MOB Register
- CANSIT2, CANSIT1- CAN Status Interrupt MOB Register
- CANBT1- CAN Bit Timing Register 1
- CANBT2- CAN Bit Timing Register 2
- CANBT3- CAN Bit Timing Register 3
- CANTCON- CAN Timer Control Register
- CANTIML, CANTIMH- CAN Timer Register
- CANTTCL, CANTTCH- CAN TTC Timer Register
- CANTEC- CAN Transit Error Counter Register
- CANREC- CAN Receive Error Counter Register
- CANHPMOB- CAN Highest Priority MOB Register
- CANPAGE- CAN Page MOB Register

CAN(Controller Area Network)

◆ Mob Registers

- CANSTMOB- CAN Mob Status Register
- CANCDMOB- CAN Mob Control and DLC Register
- CANIDT1, CANIDT2, CANIDT3, CANIDT4- CAN Identifier Tag Register
- CANIDM1, CANIDM2, CANIDM3, CANIDM4- CAN Identifier Mask Register
- CANSTML, CANSTMH- CAN Time Stamp Register

◆ CAN Frame과 Register



CAN Register

◆ CAN General Control Register, CANGCON

Bit	7	6	5	4	3	2	1	0
CANGCON	ABRQ	OVRQ	TTC	SYNTTC	LISTEN	TEST	ENA/STB	SWRES
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

- Bit 7 – ABRQ, Abort request
- Bit 6 – OVRQ, overload frame request
- Bit 5 – TTC, time trigger communication
- Bit 4 – SYNTTC, synchronization of TTC
- Bit 3 – Listen, Listening mode
- Bit 2 – Test, Test mode
- Bit 1 – ENA/STB, enable(1) /standby(0) mode
- Bit 0 – SWRES, software reset request

CAN Register

◆ CAN General Status Register, CANGSTA

Bit	7	6	5	4	3	2	1	0
CANGSTA	-	OVFG	-	TXBSY	RXBSY	ENFG	BOFF	ERRP
R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

- Bit 7 – reserved
- Bit 6 – OVFG, overload Frame flag
- Bit 5 – reserved
- Bit 4 – TXBSY, Transmitter Busy
- Bit 3 – RXBSY, Receiver Busy
- Bit 2 – ENFG, enable flag
- Bit 1 – BOFF, bus off mode
- Bit 0 – ERRP, error passive mode

CAN Register

◆ CAN Page MOB Register, CANPAGE

Bit	7	6	5	4	3	2	1	0
CANPAGE	MOBNB3	MOBNB2	MOBNB1	MOBNB0	AINC	INDX2	INDX1	INDX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

- Bit 7:4 – MOBNB3:0, Mob number
 - ✓ Selection of Mob number – 0~ 14(15)
- Bit 3 – AINC, auto increment of FIFO CAN Data Buffer Index(Active Low)
 - ✓ 0 – auto increment of the index, 1 – no auto increment
- Bit 2:0 – INDX2:0, FIFO CAN Data Buffer Index
- Byte location of the CAN Data

CAN Register

◆ CAN Mob Control and DLC Register, CANSTMOB

Bit	7	6	5	4	3	2	1	0
CANCDMOB	DLCW	TXOK	RXOK	BERR	SERR	CERR	FERR	AERR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	-	-	-	-

- Bit 7 – DLCW: Data Length Code Warning
- Bit 6 – TXOK: Transmit OK
- Bit 5 – RXOK: Receive OK
- Bit 4 – BERR: Bit Error (Only in Transmission)
- Bit 3 – SERR: Stuff Error
- Bit 2 – CERR: CRC Error
- Bit 1 – FERR: Form Error
- Bit 0 – AERR: Acknowledgment Error

CAN Register

◆ CAN Mob Control and DLC Register, CANCDMOB

Bit	7	6	5	4	3	2	1	0
CANCDMOB	CONMOB1	CONMOB0	RPLV	IDE	DLC3	DLC2	DLC1	DLC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	-	-	-	-

- Bit 7:6 – CONMOB1:0, Configuration of Message Object
 - ✓ 00 – disable, 01 - enable transmission
 - ✓ 10 - enable reception, 11 – enable frame buffer reception
- Bit 5 – RPLV, Reply Valid
 - ✓ 0 – reply not ready, 1 – reply ready and valid
- Bit 4 – IDE: Identifier extension
 - ✓ 0 - CAN Standard rev 2.0A
 - ✓ 1 - CAN standard rev 2.0 B
- Bit 3:0 – DLC 3:0, Data Length Code

CAN Register

◆ General Register 설정

1. Reset

- CANGCON=0x03
 - ✓ Bit 1- ENA/STB: Enable / Standby Mode
(0- standby mode, 1- enable mode)
 - ✓ Bit 0- SWRES: Software Reset Request
(0- no reset, 1- reset: reset "Ored" with hardware reset)

2. Baud rate 설정

- CANBT1=0x06
 - ✓ Bit6:1- baud rate pre-scaler
 - CANBT2=0x0c
 - ✓ Bit3:1- propagation time segment
 - CANBT3=0x37
 - ✓ Bit6:4- phase segment 2, Bit3:1- phase segment 1, Bit 0- sample point
- ※ 다음과 같은 설정하게 되면 Baud rate는 8 MHz, 125 kbps이 된다.

CAN Register

■ MOB Register 설정

1. CAN Page MOB설정

- MOB 선택, $CANPAGE = I \ll 4$
- 상태레지스터 초기화 $CANSTMOb=0$
- 제어레지스터 초기화 $CANCDMOB=0$
- ID 초기화, $CANIDTn=0$
- Mask 초기화, $CANIDMn=0$
- Message 초기화, $CANMSG=0$

2. CAN enable

- $CANGCON=1 \ll ENASTB$

CAN 통신

■ CAN 송신

1.초기화(Initializing)

- Identifier tag(IDT)
- Identifier extension(IDE)
- Remote transmission request(RTRTAG)
- Data length code(DLC)
- Reserved bit(s) tag(RBnTAG)
- Data bytes of message(MSG)

2.MOb 설정(Mob Configuration)

- CONMOB

3.채널 스캔(Channel scan)

- 송신으로 설정된 Mob 중에서 가장 높은 우선권을 가진 MOB를 찾는다.
- 해당 Mob의 메시지를 전송한다.

4.전송완료 인터럽트 발생(Set Transmission Flag and Interrupt)

- 송신이 정상적으로 완료되면 TXOK 플래그를 1로 설정(set)

5.대기(waiting)

- 새로운 Mob 초기화가 있을 때까지 모든 파라미터와 데이터를 사용할 수 있다

CAN 통신

■ CAN 수신

1.초기화(Initializing)

- Identifier tag(IDT)/Identifier mask(IDMSK)
- Identifier extension(IDE)/Identifier extension mask(IDEMSK)
- Remote transmission request(RTRTAG)/Remote transmission request mask(RTRMSK)
- Data length code(DLC)
- Reserved bit(s) tag(RBnTAG)

2.MOb 설정(Mob Configuration)

- CONMOB

3.채널 스캔(Channel scan)

- 수신으로 설정된 Mob 중에서 가장 높은 우선권을 가진 MOb를 찾는다.
- 해당 Mob의 메시지를 수신한다.

4.MOb 갱신(update MOb)

- IDT, IDE, DLC 일치하면 수신 프레임값으로 갱신한다.

5.수신완료 인터럽트 발생(Set Reception Flag and Interrupt)

- 수신이 정상적으로 완료되면 RXOK 플래그를 1로 설정(set)

6.대기(waiting)

- 새로운 Mob 초기화가 있을 때까지 모든 파라미터와 데이터를 사용할 수 있다.

Acceptance Filtering

- 전체 필터링:

- : CAN 표준 프레임에서 ID=0x317 만 수용함
 - : ID MSK = 111 1111 1111
 - : ID TAG = 011 0001 0111

- 부분 필터링:

- : CAN 표준 프레임에서 ID 0x310부터 0x317까지 수용함
 - : ID MSK = 111 1111 1000
 - : ID TAG = 011 0001 0xxx

- 필터링 없음:

- : CAN 표준 프레임에서 ID 0x000부터 0x7FF까지 모두 수용함
 - : ID MSK = 000 0000 0000
 - : ID TAG = xxx xxxx xxxx

Acceptance Filtering

Mask Register (std ID) 10 0

1	1	1	1	1	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---

(= 0x7f6)

Message Object Arbitration Register

1	0	0	1	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

(= 0x490)

Resulting ID matches (X = don't care)

1	0	0	1	0	0	1	X	0	0	X
---	---	---	---	---	---	---	---	---	---	---

ID's received:

1	0	0	1	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

(= 0x490)

1	0	0	1	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---

(= 0x491)

1	0	0	1	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---

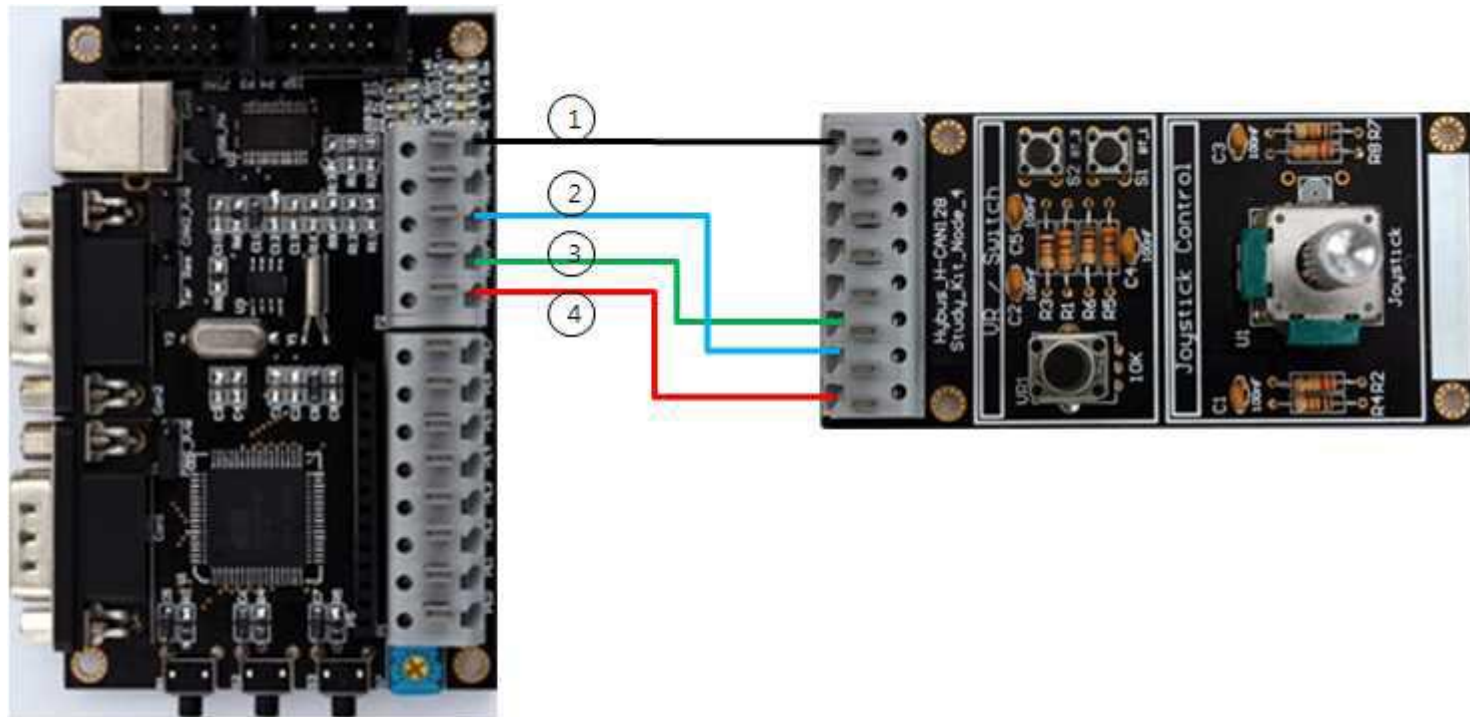
(= 0x498)

1	0	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---

(= 0x499)

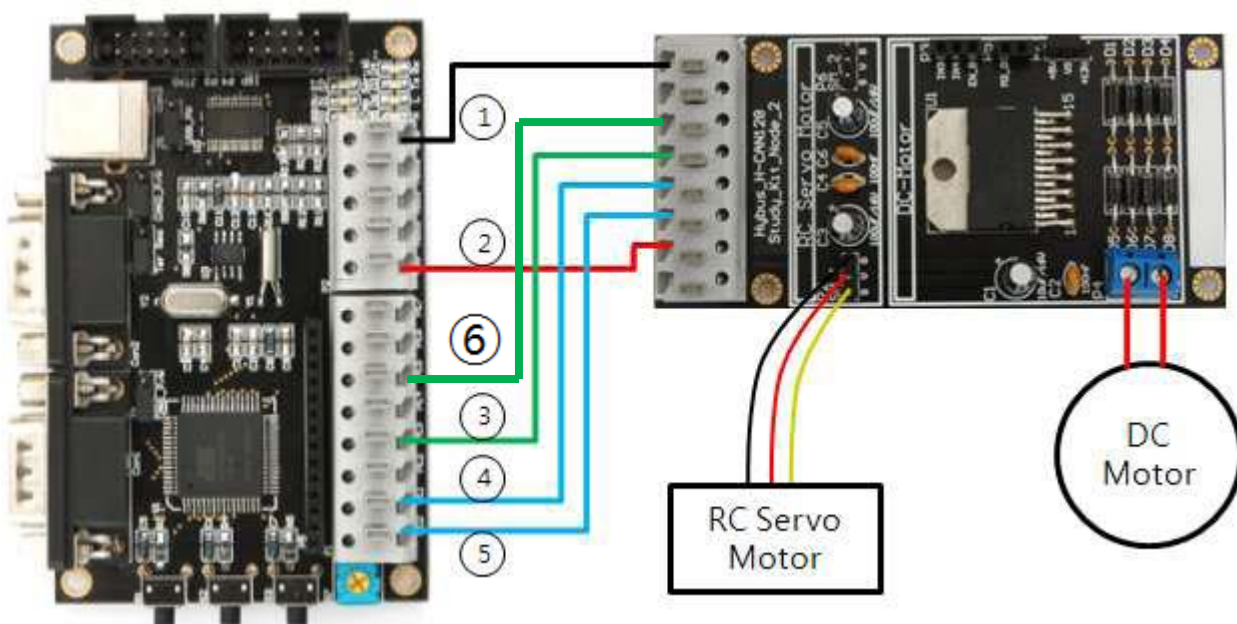
보드 이해

◆ CAN 송신 파트 Node1 연결 방법



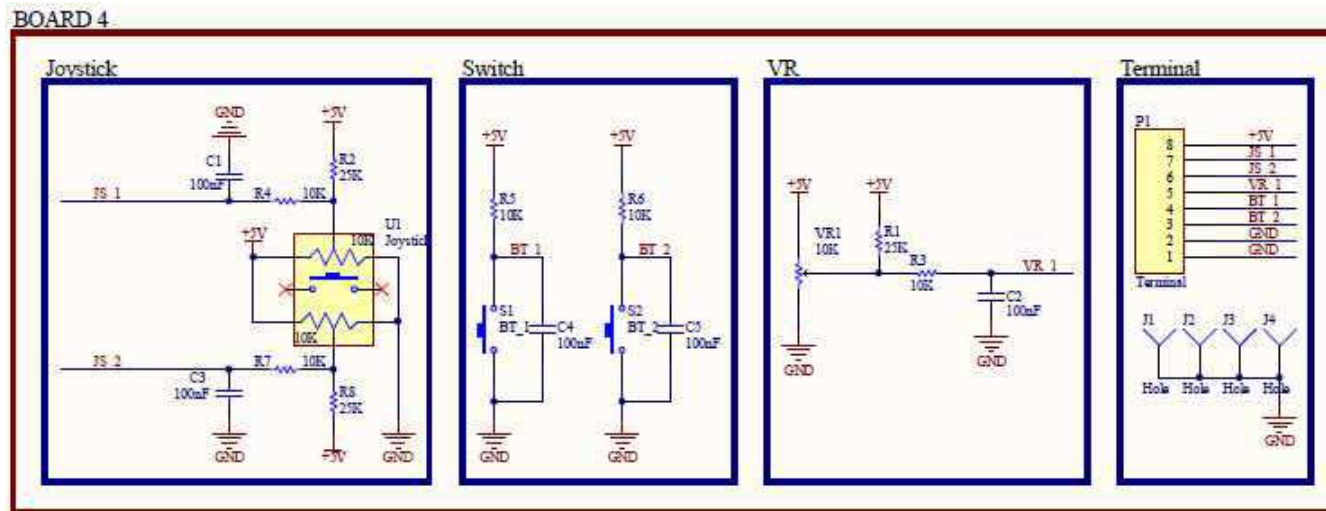
보드 이해

◆ CAN 수신 파트 Node2 연결 방법



CAN통신 회로도

- ◆ CAN Node1에서 조이스틱을 이용하여 데이터를 송신 파트
 - Instruction에 첨부된 보드 회로도 참조

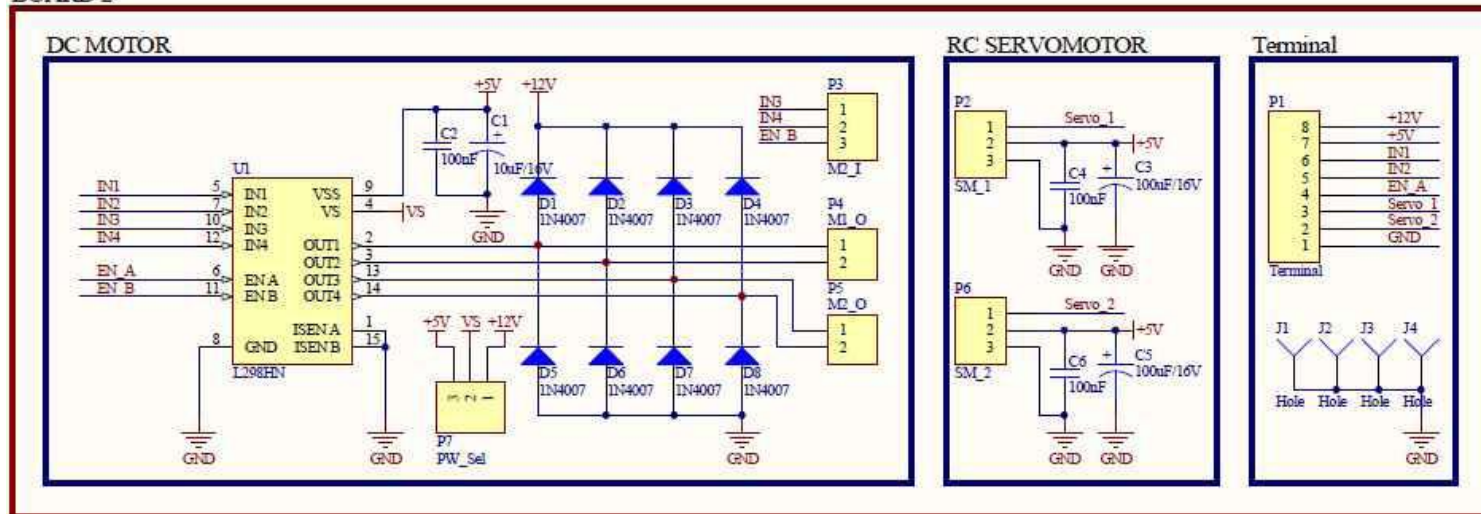


CAN통신 회로도

◆ CAN Node2에서 수신 후 모터 동작 파트

- Instruction에 첨부된 보드 회로도 참조

BOARD 2



실험5

◆ 실험 내용

- 실험 자료에 주어진 문제 해결
- 실험 자료는 실험 전날 공지
- 아래의 내용을 진행할 예정

실험1. CAN송신부에서 조이스틱을 이용하여 X축, Y축으로 CAN통신을 구현한다

실험2. CAN수신부에서 수신된 값을 받아 X축값을 Servo Motor, Y축 값을 DC Motor를 Control 한다.

※각 단계를 완료할 때 마다 조교에게 검사를 받아야 함
- 실험을 한번에 다해서 한번에 검사 받지 말고, 한 단계씩 검사

예비 보고서

◆ 실험6에 대한 예비보고서 준비 (7장 이내, 넘어가면 감점)

▪ Encoder Motor 구동 원리 예습

- ✓ Encoder Motor를 구동하기 위한 예습
- ✓ Encoder에서 나오는 Hall Sensor의 값으로 속도 및 방향 측정하는 방법 예습