

uCAN Programming

사용자 매뉴얼



History

Date (년/월/일)	Version	Revised	Description
2017/12/18	1.1	All	폰트 변경
2014/07/07	1.0	All	초기 버전

Copyright 2017 SystemBase Co., Ltd. All rights reserved.

홈페이지: <http://www.sysbas.com/>

전화: 82-2-855-0501 팩스: 82-2-855-0580

서울특별시 구로구 디지털로 288, 대륭포스트타워 1차 1601호

목차

1. 개요	5
1.1. DLL이란	5
1.2. uCAN DLL.....	5
1.3. uCANDLL구조	5
1.4. uCANDLL 파일 및 폴더 구조	6
1.5. 주의 사항	6
2. 사용법	7
2.1. Visaul C++.....	7
3. 프로그램 작성 순서.....	10
4. 사용자 정의 메시지	11
4.1. ON_CANRX	11
4.2. ON_CANTX	11
4.3. ON_CANERROR.....	11
5. 구조체 자료형.....	12
5.1. CAN_Frame	12
5.2. CAN_BTR	12
5.3. CAN_Mask	12
5.4. CAN_Error.....	13
6. API.....	14
6.1. uCAN_SetNotifyHandle	14
6.2. uCAN_FindDevice	14
6.3. uCAN_Open	14
6.4. uCAN_Close	14
6.5. uCAN_SetSetting	15
6.6. uCAN_GetSetting	15
6.7. uCAN_GetBTR	15
6.8. uCAN_GetBaudrate	16
6.9. uCAN_GetSamplingPoint	16
6.10. uCAN_CAN_Enable.....	16
6.11. uCAN_SetSerial.....	17
6.12. uCAN_GetSerial	17
6.13. uCAN_SendCANTxFrame	17
7. 참고 사항	19

7.1. Baud-rate	19
7.2. Mask.....	20

1. 개요

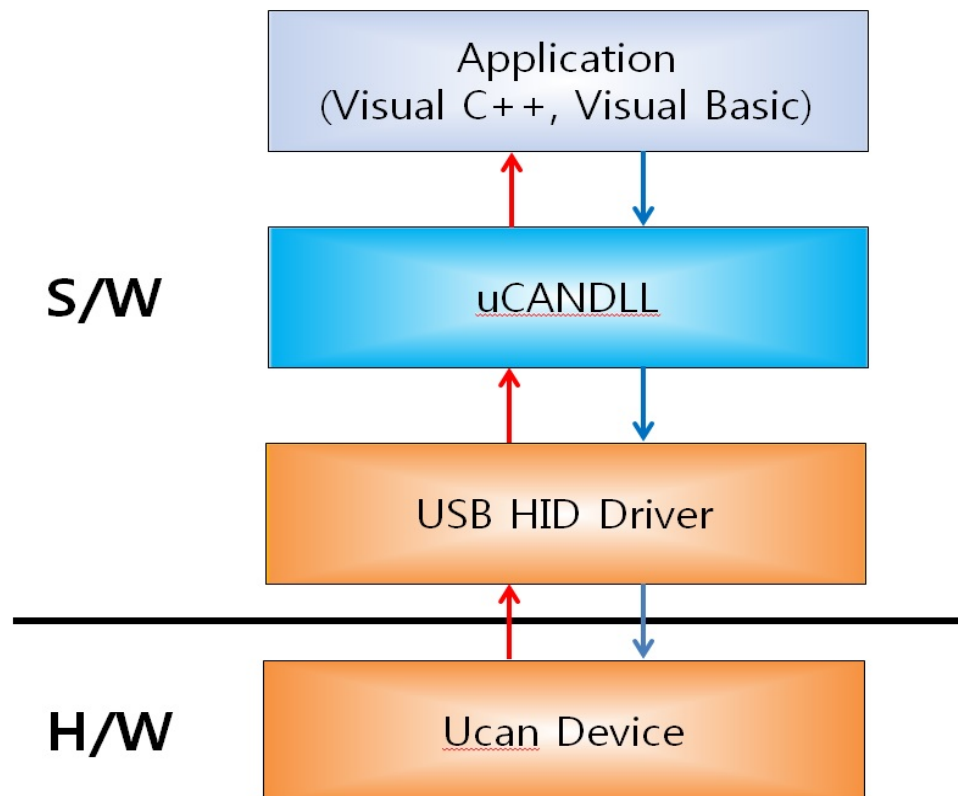
1.1. DLL이란

동적 링크 라이브러리, 줄여서 DLL(Dynamic-link library)은 마이크로소프트 윈도우에서 구현된 동적 라이브러리입니다. DLL은 내부에 다른 프로그램에서 불러서 쓸 수 있는 다양한 함수, 데이터 타입들을 가지고 있습니다. DLL 파일은 거의 “.dll”이라는 파일 확장자를 가지고 있습니다. DLL을 이용하여 개발하면 개발 속도도 빨라지고 신뢰성도 확보할 수 있습니다.

1.2. uCAN DLL

uCAN DLL은 uCAN Device을 이용하는 고객이 Visual C++ 또는 Visual Basic의 프로그래밍 언어를 통해 uCAN Device을 직접 접근하여 사용할 수 있도록 하는 동적 링크 라이브러리입니다. CAN Frame을 송/수신 할 수 있고, 통신 환경 설정을 할 수 있습니다.

1.3. uCANDLL구조



1.4. uCANDLL 파일 및 폴더 구조

파일 및 디렉토리	비고
VC SampleW	Visual C++ Sample Code
uCANDLL.dll	DLL 파일
uCANDLL.lib	Library 파일
uCANDLL.h	Header 파일
uCANDLL.pdf	DLL 매뉴얼 파일

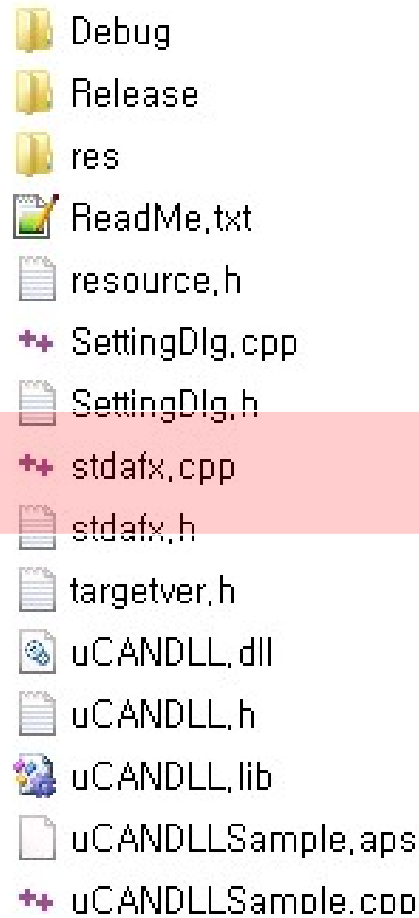
1.5. 주의 사항

CAN Frame을 수신할 때는 이벤트 메시지를 전송하기 때문에 uCANDLL을 초기화할 때 CAN Frame 수신 이벤트 메시지를 받을 수 있는 핸들을 인자 값으로 넘겨받고 있습니다.

2. 사용법

2.1. Visaul C++

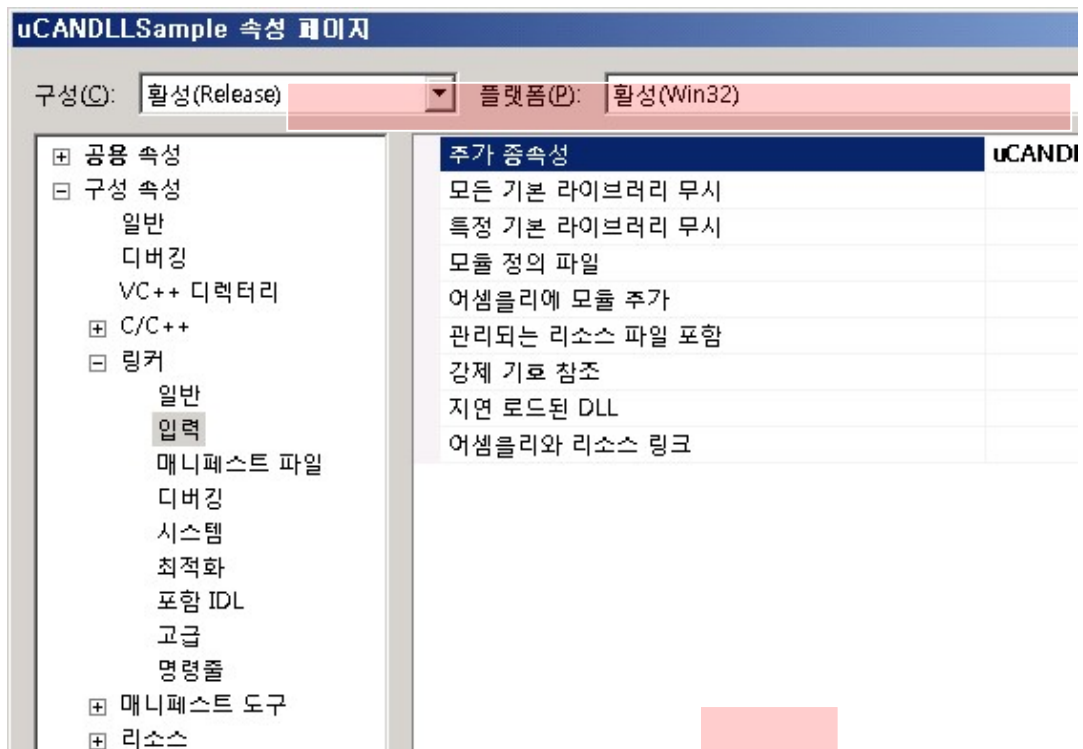
1. 아래 그림과 같이 uCANDLL.dll,uCANDLL.lib,uCANDLL.h 파일을 프로그램 소스 파일 폴더에 옮겨놓습니다.



2. Visual Studio 메뉴에서 “프로젝트” → “속성”을 클릭하여 프로젝트 속성 페이지 창을 엽니다.



3. “링커” → “입력” 항목을 클릭 한 후 “추가 종속성”란에 “uCANDLL.lib”를 입력하고 “확인”을 누릅니다.




4. uCANDLL.h 헤더 파일을 인클루드 합니다.

```
#include "stdafx.h"
#include "uCANDLLSample.h"
#include "uCANDLLSampleDlg.h"
#include "afxdialogex.h"
#include "SettingDlg.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
```

5. 이벤트 함수를 메시지 맵에 등록합니다

uCANDLLSampleDlg.h

 CuCANDLLSampleDlg

```
ULONGLONG m_TxCountValue;
```

```
ULONGLONG m_RxCountValue;
```

```
protected:
```

```
afx_msg LRESULT OnCANRx(WPARAM wParam)
```

uCANDLLSampleDlg.cpp

uCANDLLSampleDlg.h

(전역 범위)

```
ON_MESSAGE(ON_CANRX, &CuCANDLLSampleDlg::
```

```
ON_MESSAGE(ON_CANERROR, &CuCANDLLSampleC
```

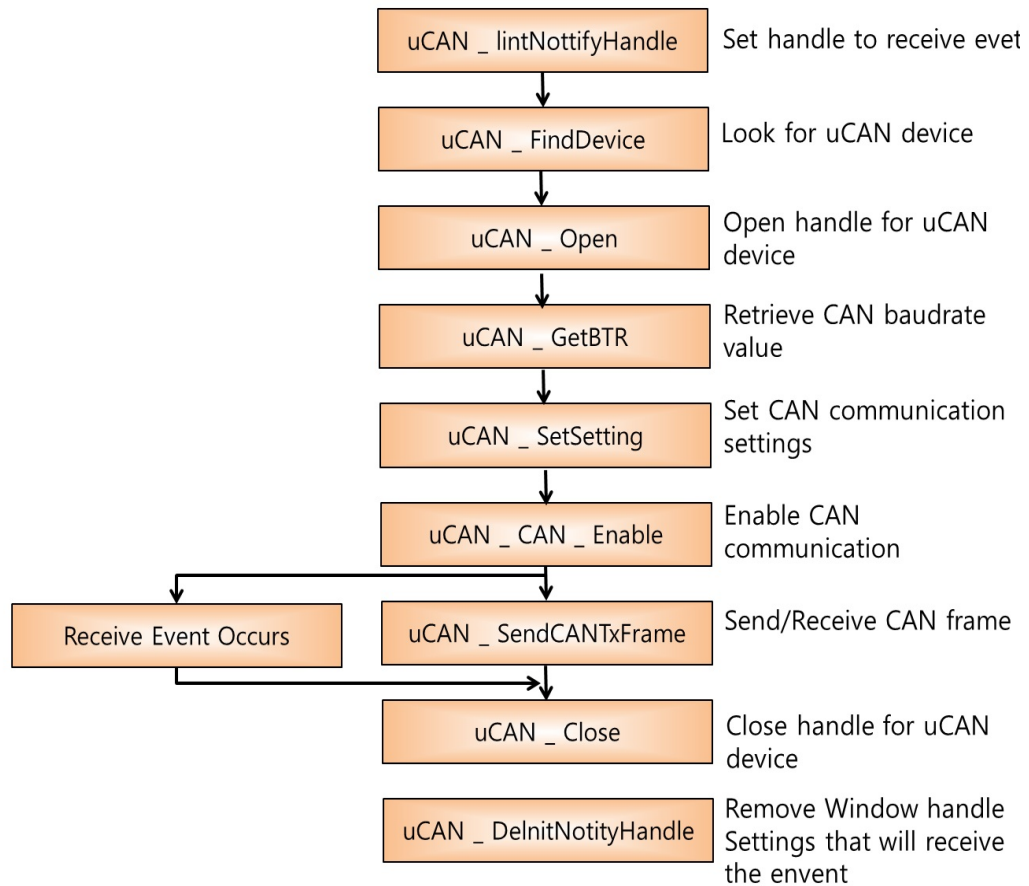
6. 아래 그림과 같이 API에 따라 프로그래밍 합니다.

```
int num;
```

```
CString tmp;
```

```
num = uCAN_FindDevice();
```

3. 프로그램 작성 순서



4. 사용자 정의 메시지

4.1. ON_CANRX

Message Parameter	설 명
wParam	수신된 CAN_Frame 구조체 포인터가 넘어옴.
lParam	Flag가 넘어옴. DLL의 처리 되지 않은 CAN Frame이 DLL 버퍼에 많이 있을 경우 BUSY flag가 설정된다. 0: NORMAL 2: BUSY

4.2. ON_CANTX

Message Parameter	설 명
wParam	송신된 CAN_Frame 구조체 포인터가 넘어옴.
lParam	Flag가 넘어옴. DLL의 처리 되지 않은 CAN Frame이 DLL 버퍼에 많이 있을 경우 BUSY flag가 설정된다. 0: NORMAL 2: BUSY

4.3. ON_CANERROR

Message Parameter	설 명
wParam	CAN_Error 구조체 포인터가 넘어옴.
lParam	None

5. 구조체 자료형

5.1. CAN_Frame

구조체 멤버 변수	설 명
UINT8 Format	CAN Frame의 종류 및 송수신을 나타낸다. 0x04: (수신)Standard Data 0x05: (수신)Standard Remote 0x06: (수신)Extended Data 0x07: (수신)Extended Remote 0x14: (송신)Standard Data 0x15: (송신)Standard Remote 0x16: (송신)Extended Data 0x17: (송신)Extended Remote
UINT32 ID	CAN Frame의 ID를 의미한다. (Hex)
UINT8 DLC	CAN Frame의 DLC를 의미한다. (0~8)
UINT8 Data[8]	CAN Frame의 Data 배열(8Bytes)
UINT32 TimeStamp	CAN Frame이 수신될 때의 TimeStamp(수신전용)

5.2. CAN_BTR

구조체 멤버 변수	설 명
UINT8 BTR	CAN Baudrate의 BTR를 의미한다.
UINT8 SJW	CAN Baudrate의 SJW를 의미한다.
UINT8 TSEG1	CAN Baudrate의 TSEG1을 의미한다.
UINT8 TSEG2	CAN Baudrate의 TSEG2을 의미한다.
UINT8 CLKDIV	CAN Baudrate의 CLKDIV을 의미한다.

5.3. CAN_Mask

구조체 멤버 변수	설 명
UINT32 ID	수신ID을 Masking할 ID를 의미한다.
UINT32 MASK	수신ID을 Masking할 MASK를 의미한다.
UINT8 Format	수신ID Masking할 Format를 의미한다. 0: ALL

	1: Standard 2: Extended
--	----------------------------

5.4. CAN_Error

구조체 멤버 변수	설 명
UINT8 TEC	uCAN의 TEC(Transmit Error Count)를 의미한다.
UINT8 REC	uCAN의 REC(Receive Error Count)를 의미한다.
UINT8 LEC	uCAN의 LEC(Last Error Count)를 의미한다. 0x00: No error 0x01: Stuff error 0x02: Form error 0x03: Ack error 0x04: Bit error 0x05: CRC error
UINT8 Mode	uCAN의 Error 상태를 의미한다. 0x00: active mode 0x01: warning mode 0x02: error passive mode 0x03: bus-off mode
UINT16 Stuff_EC	uCAN에서 현재까지 Stuff Error가 감지된 횟수를 의미한다.
UINT16 Form_EC	uCAN에서 현재까지 Form Error가 감지된 횟수를 의미한다.
UINT16 Ack_EC	uCAN에서 현재까지 ACK Error가 감지된 횟수를 의미한다.
UINT16 Bit_EC	uCAN에서 현재까지 Bit Error가 감지된 횟수를 의미한다.
UINT16 CRC_EC	uCAN에서 현재까지 CRC Error가 감지된 횟수를 의미한다.

6. API

6.1. uCAN_SetNotifyHandle

함수 원형	UINT8 uCAN_SetNotifyHandle(HANDLE hwnd)
기 능	CAN Frame 송/수신 이벤트 메시지를 받을 윈도우 핸들을 설정한다.
인자 값	hwnd : CAN Frame 송/수신 이벤트를 받는 윈도우의 핸들 (in)
결과 값	0x00: 윈도우 핸들 설정 성공 0x01: 윈도우 핸들 설정 실패

6.2. uCAN_FindDevice

함수 원형	UINT8 uCAN_FindDevice(int *MaxCount)
기 능	인식된 uCAN Device의 개수(인덱스)를 얻어온다.
인자 값	MaxCount: 인식된 uCAN Device의 개수 (out)
결과 값	0x00: 정상적으로 인식됨. 0x01: uCAN Device를 정상적으로 찾을 수 없음.

6.3. uCAN_Open

함수 원형	UINT8 uCAN_Open(int index)
기 능	uCAN Device 핸들을 연다.
인자 값	index: 핸들을 열려고 하는 uCAN Device의 인덱스 (in)
결과 값	0x00: 정상적으로 연결됨 0x01: 장치를 찾을 수 없는 인덱스 0x10: 다른 프로세스에서 연결되어 있음

6.4. uCAN_Close

함수 원형	UINT8 uCAN_Close (void)
기 능	uCAN Device 핸들을 닫는다.
인자 값	None
결과 값	0x00: 정상적으로 연결 해제 됨. 0x01: Close Handle Error

6.5. uCAN_SetSetting

함수 원형	UINT8 uCAN_SetSetting(CAN_BTR btr, CAN_Mask mask, UINT8 function)
기 능	uCAN Device의 통신 환경을 설정한다.
인자 값	btr: CAN_BTR 구조체 (in) mask: CAN_Mask 구조체 (in) function: 부가 기능의 정보를 담고 있다. (in) 0x01: AR(automatic retransmission) function enable 0x02: ABOR(automatic Bus-Off recovery) function enable 0x10: 중단저항 enable
결과 값	0x00: 정상적으로 설정됨. 0x10: 장치와 연결 되지 않음. 0x20: uCAN 전송 에러

6.6. uCAN_GetSetting

함수 원형	UINT8 uCAN_GetSetting(CAN_BTR *btr, CAN_MASK *mask, UINT8 *function)
기 능	uCAN Device의 통신 환경을 얻어온다.
인자 값	btr: CAN_BTR 구조체 (out) mask: CAN_Mask 구조체 (out) function: 부가 기능의 정보를 담을 변수 (out) 0x01: AR(automatic retransmission) function enable 0x02: ABOR(automatic Bus-Off recovery) function enable 0x10: 중단저항 enable
결과 값	0x00: 정상적으로 얻어옴 0x01: 핸들이 열려있지 않음 0x02: 설정 값을 얻어올 수 없음.

6.7. uCAN_GetBTR

함수 원형	UINT8 uCAN_GetBTR (UINT32 baud, CAN_BTR *btr)
기 능	CAN baud-rate을 설정하기 위한 CAN_BTR 구조체 값을 계산하여 얻어온다.

인자 값	baud: 설정하고자 하는 CAN baud-rate을 입력 받는다. (in) btr: 설정하고자 하는 CAN baud-rate의 CAN_BTR 구조체 값을 얻어온다. (out)
결과 값	0x00: 정상적으로 계산됨. 0x01: CAN_BTR 값을 계산하지 못함.
비 고	uCAN_GetBaudrate와 반대임.

6.8. uCAN_GetBaudrate

함수 원형	UINT8 uCAN_GetBaudrate (UINT32 *baud, CAN_BTR btr)
기 능	CAN_BTR 값으로 CAN baud-rate 값을 얻어온다.
인자 값	baud: CAN baud-rate값을 저장하는 변수이다. (out) btr: CAN baud-rate값을 얻어오려는 CAN_BTR 구조체 값을 입력 받는다. (in)
결과 값	0x00: 정상적으로 전송됨. 0x01: 계산 오류
비 고	uCAN_GetBTR과 반대임.

6.9. uCAN_GetSamplingPoint

함수 원형	UINT8 uCAN_GetSamplingPoint(float *SamplingPoint, CAN_BTR btr)
기 능	CAN_BTR 값으로 CAN SamplingPoint을 얻어온다.
인자 값	SamlingPoint: CAN SamplingPoint을 저장하는 변수이다. (out) btr: CAN SamplingPoint을 얻어오려는 CAN_BTR 구조체 값을 입력 받는다. (in)
결과 값	0x00: 정상적으로 전송됨. 0x01: 계산 오류

6.10. uCAN_CAN_Enable

함수 원형	UINT8 uCAN_CAN_Enable (UINT8 enable)
기 능	CAN 통신을 활성화/비활성화 한다.
인자 값	enable: CAN 통신을 활성화 또는 비활성화 한다. (in) 0: 비활성화 1: 활성화

결과 값	0x00: 정상적으로 전송됨. 0x01: 허용하지 않는 값. 0x10: 열려있지 않는 핸들 값
------	--

6.11. uCAN_SetSerial

함수 원형	UINT8 uCAN_SetSerial (char *strBuffer, UINT8 strBufferSize)
기 능	uCAN Serial 정보를 저장한다.
인자 값	istrBuffer: Serial 정보를 가지고 있는 변수 (in) strBufferSize: Serial 정보를 가지고 있는 변수의 크기 (in) (10이하.)
결과 값	0x00: 정상적으로 전송됨. 0x01: 허용하지 않는 값. 0x10: 열려있지 않는 핸들 값
비 고	uCAN_GetSerial과 반대임. uCAN_Open을 먼저 호출 후에 사용할 것.

6.12. uCAN_GetSerial

함수 원형	UINT8 uCAN_GetSerial (int index, char *strBuffer, UINT8 strBufferSize)
기 능	uCAN Serial 정보를 읽어온다.
인자 값	index: Serial 정보를 읽어올 index 값 (in) strBuffer: Serial 정보를 저장할 변수 값 (out) strBufferSize: Serial 정보를 저장할 변수의 크기 (in) (10이상.)
결과 값	0x00: 정상적으로 전송됨. 0x01: 허용하지 않는 값. 0x10: 열려있지 않는 핸들 값
비 고	uCAN_SetSerial과 반대임. uCAN_FindDevice를 먼저 호출 후에 사용할 것.

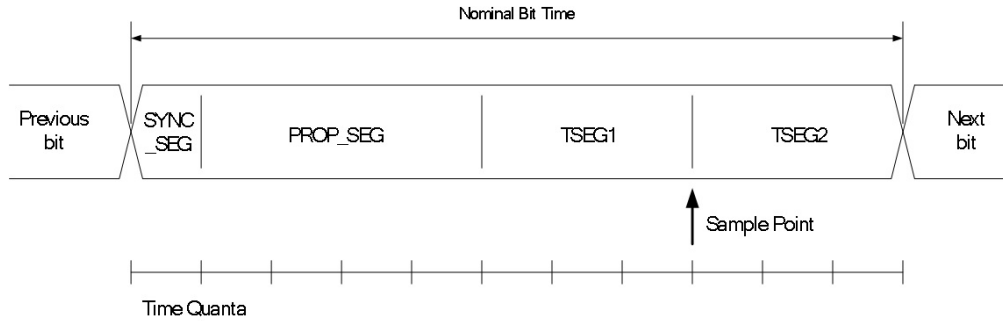
6.13. uCAN_SendCANTxFrame

함수 원형	UINT8 uCAN_SendCANTxFrame (CANTxFrame Tx)
기 능	CAN Frame을 전송한다.
인자 값	Tx: CANTxFrame 구조체 (in)
결과 값	0x00: 정상적으로 전송됨. 0x01: 허용하지 않는 Tx.Format 값

	0x02: 허용하지 않는 Tx.ID 값 0x04: 허용하지 않는 Tx.DLC 값 0x10: 열려있지 않는 핸들 값
--	---

7. 참고 사항

7.1. Baud-rate



Parameter	Range	Function
BRP	1~32	1 t_q 의 크기를 설정한다.
SYNC_SEG	1 t_q	CAN Bus에서 다양한 Nodes를 동기화하는데 사용된다. 크기는 1 t_q 로 고정되어 있다.
PROP_SEG	(1~8) $\times t_q$	물리적 지연 보상(물리적 버스와 내부 CAN node 전달 지연)
TSEG1	(1~8) $\times t_q$	Phase Edge 오류를 위한 보정에 사용. 재동기화 시 줄어드는 시간을 늘림.
TSEG2	(2~8) $\times t_q$	Phase Edge 오류를 위한 보정에 사용. 재동기화 시 늘어난 시간을 단축시킴.
SJW	(1~4) $\times t_q$	TSEG1이 길어지고, TSEG2가 짧아질 수 있는 T_q 를 설정함. TSEG1보다 크면 안됨.

$$\text{PROP_SEG} + \text{TSEG1} + 1 \geq \text{TSEG2}$$

계산식

$$\text{CAN_CLK} = 60000000 / (\text{DIV} + 1)$$

$$\text{CAN Baud rate} = \text{CAN_CLK} / ((\text{BRP} + 1) \times (\text{TSEG1} + 2 + \text{TSEG2} + 1))$$

$$\text{Sample Point} = (\text{TSEG1} + 2) / (\text{TSEG1} + 2 + \text{TSEG2} + 1) \times 100$$

Sample Point는 60~70% 사용 권장(단, CANOpen은 80~90%)

예)

Baud rate	Sample Point	ClkDiv	BTR	SJW	TSEG1	TSEG2
50		3	24	3	6	7
100		2	19	3	6	6
125		1	29	3	6	7
250		1	14	3	6	7
500		1	7	3	6	6
1000		1	3	3	6	6

7.2. Mask

일반적으로 CAN 통신에서는 수신 ID와 수신 Mask ID를 조합하여 CAN 네트워크상의 모든 CAN Frame 중 보고자 하는 CAN Frame을 필터링 하여 통신 처리 부하를 조절합니다.

수신 ID는 보고자 하는 CAN Frame ID를 나타내며, 수신 Mask ID는 수신한 모든 데이터에 대해서 설정한 수신 ID의 해당 비트와 일치하는지를 검사하여 일치하면 데이터를 수신하고 일치하지 않으면 데이터를 수신하지 않습니다.



제품을 사용하시다가 불편하신 점이 있으면 아래 연락처로 상담하여 주십시오.

문의

www.sysbas.com

전화: 02-855-0501

팩스: 02-855-0580

이메일:

- 구매/견적 문의: sales@sysbas.com
- 기술/지원 문의: tech@sysbas.com
- A/S 문의: as@sysbas.com

Copyright © 2020 SystemBase Co., Ltd. All Right Reserved.