

# Command Library

*sCAN / uCAN V3.0*

## 사용자 매뉴얼



## 수정 사항

수정일	문서 버전	페이지	설명
2018. 02. 07.	1.0	전체	최초 작성

Copyright 2018 SystemBase Co., Ltd. All rights reserved.

Website <http://www.sysbas.com/>

Tel 82-2-855-0501

Fax 82-2-855-0580

서울시 구로구 디지털로 288, 대륭포스트타워1차 1601호  
문의사항에 대해서는 [tech@sysbas.com](mailto:tech@sysbas.com)으로 연락바랍니다.

# 목 차

1 장 개요.....	4
1. 이 매뉴얼에 대해.....	4
2. 독자 .....	4
3. 매뉴얼 구성.....	4
4. Command Library 관련 장비.....	5
2 장 Command .....	6
1. Command 요약.....	6
2. Command 설명.....	7
2.1 Active mode.....	7
2.2 Setup Mode .....	9
3 장 C 코드 파일 .....	12
1. 구성 .....	12
2. 구조체 자료형 .....	12
3. API .....	13
4 장 부록.....	16
4. 스위치 설정.....	17

# 1 장 개요

이 장은 시스템베이스의 CAN 통신 전용 Command Library에 대해 소개한다.

## 1.1 이 매뉴얼에 대해

이 매뉴얼은 Command Library의 명령어, 사용 예시와 구조체 자료형에 대한 설명과 사용 방법에 대해 기술되었다.

## 1.2 독자

이 매뉴얼은 sCAN/uCAN V3.0의 사용자와 관리자를 위해 작성되었다. Command Library를 사용하거나 프로그래밍 전에 이 매뉴얼을 읽는 것이 좋으며, 간편하게 디바이스를 제어하기 위해서는 유틸리티 프로그램인 CANView를 사용하여 장비를 제어하고 관리하는 것이 유용할 것이다.

## 1.3 매뉴얼 구성

1장 개요는 Command Library의 일반적인 정보와 소개를 담고 있다.

2장 Command는 모드에 따른 Command에 대한 설명과 예시를 다루고 있다.

3장 C코드 파일은 시스템베이스에서 제공되는 C 코드파일에 대한 설명을 다루고 있다.

4장 부록에서는 Command Library에서 지원되는 Mask, DAR, ABOR에 대한 설명과 sCAN/uCAN V3.0의 스위치 설정에 대해 설명에 대한 내용을 다루고 있다.

## 1.4 Command Library 관련 장비

Command Library와 관련된 장비는 다음과 같다.

문서	설명
sCAN V1.0	시리얼(RS-232) to CAN 컨버터
uCAN V3.0	USB 2.0 to CAN 컨버터
sCAN V1.0 사용자 매뉴얼	sCAN V1.0 제품의 사용법
sCAN V1.0 Spec Sheet	sCAN V1.0 제품의 사양
uCAN V3.0 사용자 매뉴얼	uCAN V3.0 제품의 사용법
uCAN V3.0 Spec Sheet	uCAN V3.0 제품의 사양

모든 문서는 최신 버전으로 업데이트 되어 홈페이지에 게재되고 있으니 참고 바란다. 문서의 내용은 사전 공지 없이 수정될 수 있다.

## 2 장 Command

이 장에서는 모드에 따른 Command에 대한 설명과 예시를 다루고 있다.

### Command 요약

Command Header	Description	S	A
t	Standard CAN Data Frame 전송		●
T	Standard CAN Remote Frame 전송		●
e	Extended CAN Data Frame 전송		●
E	Extended CAN Remote Frame 전송		●
RS	장비의 Serial 통신 환경 로드	●	
RC	장비의 CAN 통신 환경 로드	●	
RH	Active mode의 Command Option 로드	●	
WC	장비의 CAN 통신 환경 설정	●	
WS	장비의 Serial 통신 환경 설정	●	
WH	Active mode의 Command Option 설정	●	
SF	공장 초기화	●	
SV	장비 설정 값 저장	●	
SR	장비 재 부팅	●	
FV	현재 Firmware Version 로드	●	

S: Setup Mode

A: Active Mode

## Command 설명

※ 모든 명령어의 끝은 Carriage Return인 0x0D이며, 일부를 제외한 모든 셋업 모드 명령어는 ACK 리턴이 존재한다.

### 2.1 Active mode

#### - Standard Data Frame Transmit

t(ID)(DLC)(DATA)<CR>	
설명	<ul style="list-style-type: none"> <li>- t: CAN Standard Data Frame을 전송하는 Command Header</li> <li>- (ID): CAN Frame의 Message Identify (ASCII 3자로 구성, 범위: 000~7FF)</li> <li>- (DLC): CAN Frame의 데이터 길이 (ASCII 1자로 구성, 범위: 0~8)</li> <li>- (DATA): CAN Frame의 실제 데이터 (ASCII)</li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul>
예시	t12356162636465<CR> → ID=0x123, Data Length=5, Data1=0x61, Data2=0x62, Data3=0x63, Data4=0x64, Data5=0x65인 CAN Frame을 전송

#### - Standard Remote Frame Transmit

T(ID)(DLC)<CR>	
설명	<ul style="list-style-type: none"> <li>- T: CAN Standard Remote Frame을 전송하는 Command Header</li> <li>- (ID): CAN Frame의 Message Identify (ASCII 3자로 구성, 범위: 000~7FF)</li> <li>- (DLC): CAN Frame의 데이터 길이 (ASCII 1자로 구성, 범위: 0~8)</li> </ul>
예시	T1235<CR> → ID=0x123, Data Length=5, Remote CAN Frame을 전송

#### - Extended Data Frame Transmit

e(ID)(DLC)(DATA)<CR>	
설명	<ul style="list-style-type: none"> <li>- e: CAN Extended Data Frame을 전송하는 Command Header</li> <li>- (ID): CAN Frame의 Message Identify. (ASCII 8자로 구성, 범위: 00000000~1FFFFFFF)</li> <li>- (DLC): CAN Frame의 데이터 길이 (ASCII 1자로 구성, 범위: 0~8)</li> <li>- (DATA): CAN Frame의 실제 데이터 (ASCII)</li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul>
예시	- 예시: e1234567826162<CR> → ID=0x12345678, Data Length=2, Data1=0x61, Data2=0x62 CAN Frame을 전송

- Extended Remote Frame Transmit

E(ID)(DLC)<CR>	
설명	<ul style="list-style-type: none"> <li>- E: CAN Extended Remote Frame을 전송하는 Command Header</li> <li>- (ID): CAN Frame의 Message Identify (ASCII 8자로 구성, 범위: 00000000~1FFFFFFF)</li> <li>- (DLC): CAN Frame의 데이터 길이 (ASCII 1자로 구성, 범위: 0~8)</li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul>
예시	- 예시: E123456782<CR> → ID=0x12345678, Data Length=2 CAN Frame을 전송



## 2.2 Setup Mode

### - Write Header Setting

WH(std_data)(std_remote)(ext_data)(ext_remote)<CR>	
설명	<ul style="list-style-type: none"> <li>- WH: Active Mode에서 사용하는 Command Header를 변경하는 명령어</li> <li>- (std_data): Standard Data을송/수신하는Command Header (ASCII 1Byte: a~z, A~Z, 0~9)</li> <li>- (std_remote): Standard Remote을송/수신하는Command Header (ASCII 1Byte: a~z, A~Z, 0~9)</li> <li>- (ext_data): Extended Data을송/수신하는Command Header (ASCII 1Byte: a~z, A~Z, 0~9)</li> <li>- (ext_remote): Extended Remote을송/수신하는Command Header (ASCII 1Byte: a~z, A~Z, 0~9)</li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul> <p>4개의 Command Header가 중복되면 안됨.</p>
응답	<p>OK&lt;CR&gt;: No error</p> <p>ER:arg: 인자 값이 오류</p>
예시	<p>WHTTeE&lt;CR&gt; → Active 모드 에서 사용되는 Command Header를 CAN Standard Data Frame을 송/수신하는 Command Header는 ‘t’ CAN Standard Remote Frame을 송/수신하는 Command Header는 ‘T’ CAN Extended Data Frame을 송/수신하는 Command Header는 ‘e’ CAN Extended Remote Frame을 송/수신하는 Command Header는 ‘E’로 설정</p>

### - Write Serial Setting

WS(Flow)(D)(P)(S)(Baud rate)<CR>	
설명	<ul style="list-style-type: none"> <li>- WS: Serial 통신 환경을 설정하는 Command Header</li> <li>- (Flow): Serial Flow control (ASCII) N → None Flow Control H → Hardware Flow Control</li> <li>- (D): Serial Data bits (ASCII, 8 Data bits로 고정) 8 → 8 Data bits</li> <li>- (P): Serial Parity bit (ASCII) N → None Parity      O → Odd Parity    E → Even Parity M → Mark Parity      S → Space Parity</li> <li>- (S): Serial Stop bit (ASCII) 1 → 1 Stop bit                      2 → 2 Stop bits</li> <li>- (Baud rate): Serial Baud rate (ASCII, 범위: 300 ~ 921600, 단위 bps)</li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul>
응답	<p>OK&lt;CR&gt;: No error</p>
예시	<p>WSN8N19600&lt;CR&gt; → 장비의 시리얼 통신 환경을 None Flow Control, 8N1, 9600bps로 설정</p>

## - Write CAN Setting

WC(Spec),(Baud rate),(ID)(MASK)(Option)<CR>	
설명	<ul style="list-style-type: none"> <li>- WC: CAN 통신 환경을 설정하는 Command Header</li> <li>- (Spec): CAN 통신 규격. (ASCII) <ul style="list-style-type: none"> <li>A → CAN 2.0 A (ID가 11bits)</li> <li>B → CAN 2.0 B (ID가 29bits)</li> </ul> </li> <li>- (Baud rate): CAN Baud rate (ASCII, 범위: 100 ~ 1000, 단위: Kbps)</li> <li>- (ID): CAN Mask ID (ASCII) <ul style="list-style-type: none"> <li>CAN 2.0 A → 000~7FF</li> <li>CAN 2.0 B → 00000000~1FFFFFFF</li> </ul> </li> <li>- (MASK): CAN Mask (ASCII) <ul style="list-style-type: none"> <li>CAN 2.0 A → 000~7FF</li> <li>CAN 2.0 B → 00000000~1FFFFFFF</li> </ul> </li> <li>- (Option): CAN 통신 옵션(ASCII) <ul style="list-style-type: none"> <li>0 → No Option</li> <li>1 → DAR enable</li> <li>2 → ABOR enable</li> <li>3 → DAR, ABOR enable</li> </ul> </li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul>
응답	OK<CR>: No error
예시	WCA,500,123,7FF,3<CR> → 장비의 CAN 설정을 CAN Spec은 2.0A, Baud rate는 500Kbps, Mask ID는 0x123으로 Mask는 0x7FF, ABOR과 DAR 기능을 설정

## - Read Serial Setting

RS<CR>	
설명	<ul style="list-style-type: none"> <li>- RS: Serial 통신 환경을 로드하는 Command Header</li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul>
응답	(Flow)(D)(P)(S)(Baud rate)<CR>
예시	N8N19600<CR> → 장비의 시리얼 통신 환경이 None Flow Control, 8N1, 9600bps

## - Read CAN Setting

RC<CR>	
설명	<ul style="list-style-type: none"> <li>- RC: CAN 통신 환경을 반환하는 Command Header</li> <li>- &lt;CR&gt;: 0x0D (Carriage return)</li> </ul>
응답	(Spec),(Baud rate),(ID)(MASK)(Option)<CR>
예시	A,500,123,7FF,3<CR> → 장비의 CAN 설정이 CAN Spec은 2.0A, Baud rate는 500Kbps, Mask ID는 0x123으로 Mask는 0x7FF, ABOR과DAR 기능 On

## - Read Header Setting

RH<CR>	
설명	- RH: Active Mode에서 사용하는 Command Header를 반환하는 명령어 - <CR>: 0x0D (Carriage return)
응답	(std_data)(std_remote)(ext_data)(ext_remote)<CR>
예시	tTeE<CR> → 장비의 Active 모드에서 사용되는 Command Header를 CAN Standard Data Frame을 송/수신하는 Command Header는 't' CAN Standard Remote Frame을 송/수신하는 Command Header는 'T' CAN Extended Data Frame을 송/수신하는 Command Header는 'e' CAN Extended Remote Frame을 송/수신하는 Command Header는 'E'

## - Factory Default

SF<CR>	
설명	- SF: 모든 설정 초기화하는 명령어 - <CR>: 0x0D (Carriage return)
응답	OK<CR>
예시	초기화 후 저장 명령(SV)을 실행해야 적용

## - System Reset

SR<CR>	
설명	- SR: 장비를 소프트웨어적으로 재 부팅하는 Command Header - <CR>: 0x0D (Carriage return)
응답	응답없음
예시	SR<CR> → 장비 재 부팅

## - System Save

SV<CR>	
설명	- SV: 변경된 설정을 장비에 저장하는 Command Header - <CR>: 0x0D (Carriage return)
응답	OK<CR>: No error
예시	SV<CR> → Serial, CAN, Option 설정 저장

## - Firmware Version

FV<CR>	
설명	- FV: 현재 Firmware version을 반환하는 Command Header - <CR>: 0x0D (Carriage return)
응답	(Firmware version)<CR>
예시	2.00<CR> → Firmware version 2.00

## 3 장 C 코드 파일

시리얼 데이터 파싱(Parsing) 소스를 제공하며 윈도우, 리눅스, 맥 또는 타겟 보드에서 활용이 가능한 C 코드로 제공 된다.

### 3.1 구성

파일 명	설 명
define.h	시리얼 포트와 자료형에 대한 헤더 파일
CommandLib.h	명령어 파싱 코드의 헤더 파일
CommandLib.c	명령어 파싱 코드의 소스 파일

### 3.2 구조체 자료형

#### 1. CAN\_Struct

구조체 멤버 변수	설 명
UINT8 Format	CAN Frame의 종류 및 송수신을 표현 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 ID (Hex)
UINT8 DLC	CAN Frame의 DLC (0~8)
UINT8 Data[8]	CAN Frame의 Data (8Bytes)

## 2. CANConfigInfo

구조체 멤버 변수	설 명
UINT32 Baudrate	CAN baud rate
UINT32 ID	CAN Mask ID
UINT32 MASK	CAN Mask
UINT8 Spec	CAN Spec
bool DAR	Disable Automatic Retransmit (4장 부록 참고)
bool ABOR	Automatic Bus-Off Recovery (4장 부록 참고)

## 3.3 API

## - endCANTxFrame

함수 원형	quint8 SendCANTxFrame(CAN_Struct Tx, quint8 * SerialBuffer, quint8 BufferSize, quint8 * ConvertSize)
기 능	배열에 CAN을 전송하는 명령어 전달
인자 값	Tx: 전송하려는 CAN Frame 구조체 SerialBuffer: 명령어를 넣을 배열 포인터 BufferSize: 명령어를 넣을 최대 배열 크기 ConvertSize: 명령어를 넣은 배열 크기
결과 값	0x00: No error 0x01: Invalid_Arg

## - SetSerialConfig

함수 원형	quint8 SetSerialConfig(SerialConfigInfo info, quint8 * SerialBuffer, quint8 BufferSize, quint8 * CovertSize);
기 능	배열에 시리얼 통신 환경을 설정 하는 명령어 전달
인자 값	info: 설정하려는 Serial 통신 환경 설정 구조체 SerialBuffer: 명령어를 넣을 배열 포인터 BufferSize: 명령어를 넣을 최대 배열 크기 ConvertSize: 명령어를 넣은 배열 크기
결과 값	0x00: No error 0x01: Invalid_Arg

## - SetCANConfig

함수 원형	quint8 SetCANConfig(CANConfigInfo info, quint8 * SerialBuffer, quint8 BufferSize, quint8 * CovertSize);
기 능	배열에 CAN 통신 환경을 설정 하는 명령어 전달
인자 값	info: 설정하려는 CAN 통신 환경 설정 구조체 SerialBuffer: 명령어를 넣을 배열 포인터 BufferSize: 명령어를 넣을 최대 배열 크기 ConvertSize: 명령어를 넣은 배열 크기
결과 값	0x00: No error 0x01: Invalid_Arg

## - GetSerialConfig

함수 원형	quint8 GetSerialConfig(quint8 * SerialBuffer, quint8 BufferSize, quint8 * ConvertSize);
기 능	배열에 시리얼 통신 환경을 반환하는 명령어 전달
인자 값	SerialBuffer: 명령어를 넣을 배열 포인터 BufferSize: 명령어를 넣을 최대 배열 크기 ConvertSize: 명령어를 넣은 배열 크기
결과 값	0x00: No error 0x01: Invalid_Arg

## - GetCANConfig

함수 원형	quint8 GetSerialConfig(quint8 * SerialBuffer, quint8 BufferSize, quint8 * ConvertSize);
기 능	배열에 CAN 통신 환경을 반환하는 명령어 전달
인자 값	SerialBuffer: 명령어를 넣을 배열 포인터 BufferSize: 명령어를 넣을 최대 배열 크기 ConvertSize: 명령어를 넣은 배열 크기
결과 값	0x00: No error 0x01: Invalid_Arg

## - ParsingSerialInfo

함수 원형	quint8 ParsingSerialInfo(SerialConfigInfo * info, quint8 * SerialBuffer, quint8 BufferSize);
기 능	받은 시리얼 통신 환경 데이터를 시리얼 통신 환경 구조체로 전달
인자 값	info: 시리얼 통신 환경을 넣을 구조체 포인터 SerialBuffer: 받은 시리얼 통신 환경 데이터 BufferSize: 받은 시리얼 통신 환경 데이터 크기
결과 값	0x00: No error 0x01: Invalid_Arg

- ParsingCANInfo

함수 원형	quint8 ParsingCANInfo(CANConfigInfo * info, quint8 * SerialBuffer, quint8 BufferSize);
기 능	받은 CAN 통신 환경 데이터를 CAN 통신 환경 구조체로 전달
인자 값	info: CAN 통신 환경을 넣을 구조체 포인터 SerialBuffer: 받은 CAN 통신 환경 데이터 BufferSize: 받은 CAN 통신 환경 데이터 크기
결과 값	0x00: No error 0x01: Invalid_Arg

- SaveConfig

함수 원형	quint8 SaveConfig(quint8 * SerialBuffer, quint8 BufferSize, quint8 * ConvertSize);
기 능	배열에 설정을 저장하는 명령어를 전달
인자 값	SerialBuffer: 명령어를 넣을 배열 포인터 BufferSize: 명령어를 넣을 최대 배열 크기 ConvertSize: 명령어를 넣은 배열 크기
결과 값	0x00: No error 0x01: Invalid_Arg

## 4 장 부록

### 4.1 Mask

일반적인 CAN 통신에서는 수신 ID와 수신 Mask ID를 조합하여 사용하며 네트워크상의 CAN 데이터 중 수신 하고자 하는 특정 데이터를 필터링 하여 통신 부하를 조절한다.

수신 ID는 CAN 데이터 Frame ID를 뜻하며, 수신 Mask ID는 수신한 데이터의 ID 중 특정 필드의 비트와 이를 비교하여, 일치 시 데이터를 수신하고 불 일치 시 데이터를 수신 치 않는 기능에 활용된다.

### 4.2 DAR(Disable Automatic Retransmit)

일반적인 CAN 통신에서는 CAN 데이터를 전송 후, 여타 CAN 디바이스에서 에러가 발생 시 자동으로 재전송을 하여 데이터 무 결성을 지원한다. 하지만 잦은 재 전송으로 통신 부하와 우선 순위에 따라 데이터 전송에 지연이 발생 될 수 있다. 이런 경우 DAR(Disable Automatic Retransmit) 기능을 통해 통신 부하를 저감할 수 있으며, 데이터 전송이 지연 되는 것을 방지 할 수 있다.

### 4.3 ABOR(Automatic Bus-Off Recovery)

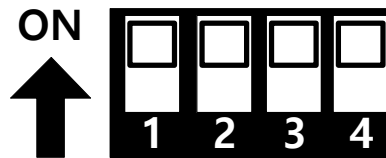
에러가 자주 발생 되는 디바이스로 인해 전체 CAN 네트워크의 불 안전성 증대 되고, 이로 인해 통신 효율 저하 및 데이터 전송 지연이 발생 될 수 있다. 이런 문제를 방지 하기 위해 각 디바이스들은 송신 중 발생 된 에러(TEC: Transmit Error Count)와 수신 중 발생되는 에러(REC: Receive Error Count)를 카운트 하여 특정 수준 이상이 에러 검출 시 네트워크에서 자동적으로 분리되어 통신 효율을 향상시킨다. 이후 에러가 해결 되고 디바이스 재 부팅을 통해 네트워크 복구가 가능하나 ABOR(Automatic Bus-Off Recovery) 기능을 통해 재 부팅을 하지 않고 자동으로 네트워크에 복구가 가능하다.



## 4.4 스위치 설정

sCAN 및 uCAN V3.0 바닥 면의 스위치에 따라 전원 선택, 동작 모드 설정, 종단저항 On/Off를 할 수 있다.

※ sCAN 및 uCAN V3.0 설정을 변경하기 위해서는 제품 바닥 면의 스위치를 Setup 모드로 변경 해야 하며, 설정 변경이 완료 된 후 Active모드로 변경 해야 변경된 설정이 적용된다.



번호	상태	설 명
Switch 1	ON	VDD을 전원으로 사용 (기본)
	OFF	CAN VDD을 전원으로 사용 안 함
Switch 2	ON	USB VBUS를 전원으로 사용 (기본)
	OFF	USB VBUS를 전원으로 사용 안 함
Switch 3	ON	Active Mode (기본)
	OFF	Setup Mode
Switch 4	ON	종단저항 Enable (120Ω) (기본)
	OFF	종단저항 Disable



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

#### 상담 시간

오전 09:00 ~ 오후 06:00

(토요일, 일요일, 공휴일은 휴무입니다.)