



---

# JTool 사용 설명서

---

Ver. 1.1

July 2024

Copyright CORERIVER Semiconductor Co., Ltd. 2024.

All Rights Reserved

- ◆ *CORERIVER Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time.*
- ◆ *To discontinue any product or service, CORERIVER should inform customers of that before 3 months through its homepage.*
- ◆ *Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.*
- ◆ *The CORERIVER Semiconductor products listed in this document are intended for usage in general electronics applications. These CORERIVER Semiconductor products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury.*

---

## Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>3</b>
<b>1 JTOOL 사용하기</b> .....	<b>4</b>
1.1 JTool 소개.....	4
1.1.1 JTool의 특징.....	4
1.1.2 구성 및 설명.....	5
1.2 사전 점검 사항.....	8
1.2.1 TCxxx_user.h 내 Configuration.....	9
1.3 F/W 다운로드.....	10
1.4 디버깅 따라하기.....	11
1.4.1 Read.....	12
1.4.2 Write.....	15
<b>2 APPENDIX</b> .....	<b>16</b>

## 1 JTool 사용하기

### 1.1 JTool 소개

JTool 은 miniGenICE 를 사용하여 Debugging 을 할 수 있는 GUI 입니다. I2C 통신을 사용한 Graph 창을 통해 데이터를 쉽게 확인할 수 있습니다. 또한, 모니터링 한 데이터를 엑셀 형식의 로그 파일로, 그래프를 이미지 파일로 저장할 수 있습니다.

#### 1.1.1 JTool 의 특징

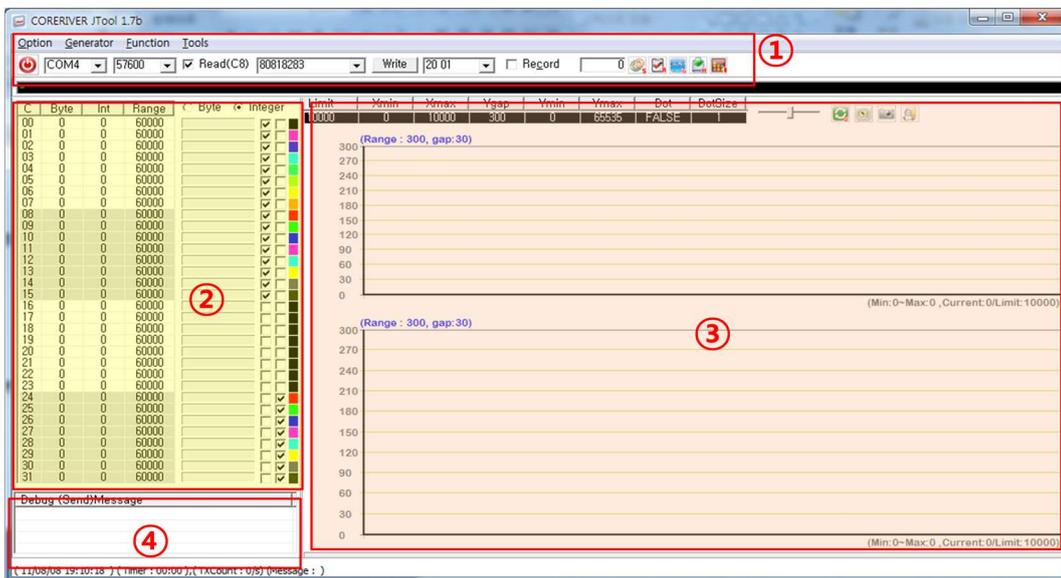
- ◆ 편리하고 직관적인 인터페이스
- ◆ TouchCore series의 모니터링 및 디버깅 지원
- ◆ Touch Key 관련 인터페이스와 툴 지원

## 1.1.2 구성 및 설명

JTool 을 실행하면 아래와 같은 화면이 보입니다.

화면은 크게 ①상단의 메뉴, ②좌측의 데이터 리스트, ③우측의 그래프 창 ④좌측 하단의 디버그 메시지 창으로 구성되어 있습니다.

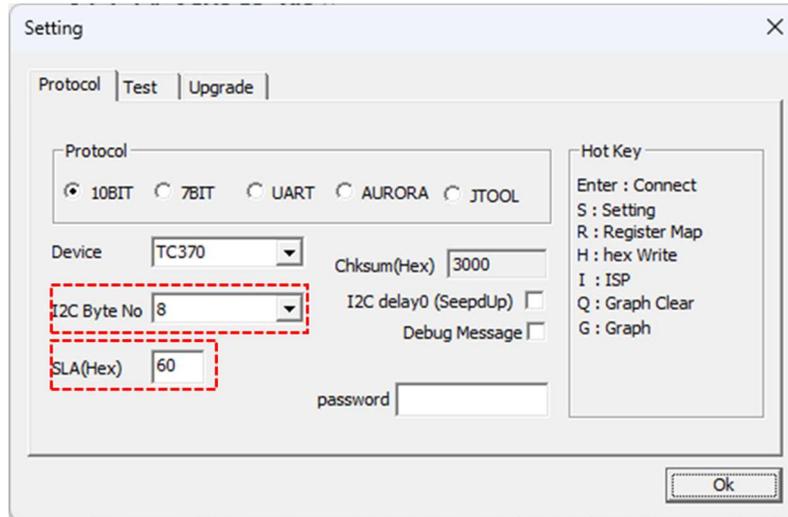
- ◆ 메뉴에서는 JTool이 제공하는 기능들을 실행할 수 있습니다.
- ◆ 데이터 리스트에는 Read하고 있는 메모리의 값들이 표시됩니다.
- ◆ 그래프 창에는 데이터 리스트에서 체크한 값들이 그래프로 표시됩니다.



먼저 사용법을 확인하시려면 "1.2 사전 점검 사항" 란을 참조해주시시오.

### 1.1.2.1 Setting

 또는 메뉴의 [Option]->[Setting]을 선택해 실행할 수 있습니다.

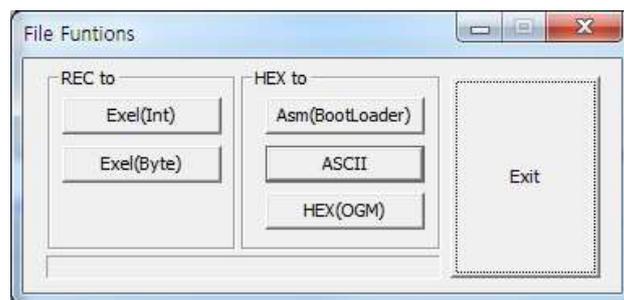


#### ◆ Protocol

- ✓ Target Board와 통신 설정을 합니다. Protocol/Device는 초기값으로 둡니다.
- ✓ I2C Byte No : I2C로 읽어올 Read Byte의 크기를 지정합니다. (Default 8Byte 유지 권장)
- ✓ SLA(Slave Address) : F/W 내 TCxxx\_user.h 내 지정한 I2C Slave Address를 입력합니다.

### 1.1.2.2 File

메뉴의 [Tools]->[File Function]을 선택하여 실행할 수 있습니다. Record 시 File type 을 선택할 수 있습니다.



### 1.1.2.3 Clear(data)

 를 선택하면 실행 됩니다. Read 되었던 데이터 들을 지우고 초기화 해 줍니다. 그래프도 초기화 됩니다.

#### 1.1.2.4 Save to 'xls'

를 선택하여 실행합니다. 지금까지 read 된 그래프에 표시된 값들을 Excel 형식의 파일로 저장합니다. 저장위치를 지정해 주면 파일이 생성됩니다.  
저장은 "...₩YearMonthDay\_₩[YearMonthDay\_HourMinuteSecond].xls"로 이루어집니다.

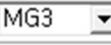
#### 1.1.2.5 Capture

를 선택하여 실행합니다. 현재 그래프를 이미지 파일로 저장합니다. 저장위치를 지정해 주면 파일이 생성됩니다.  
저장은 "...₩YearMonthDay\_₩[YearMonthDay\_HourMinuteSecond].bmp"로 이루어집니다.

#### 1.1.2.6 Record & Open

를 선택하여 실행합니다. 기존에 Record 한 그래프를 불러올 수 있습니다. Record 는  Record 의 체크박스에 체크를 하면 위치 지정 후 시작되고 체크박스를 해제하면 Record 가 중지됩니다.  
저장은 "...₩YearMonthDay\_₩[YearMonthDay\_HourMinuteSecond].rec"로 이루어집니다.

#### 1.1.2.7 Connect & Disconnect

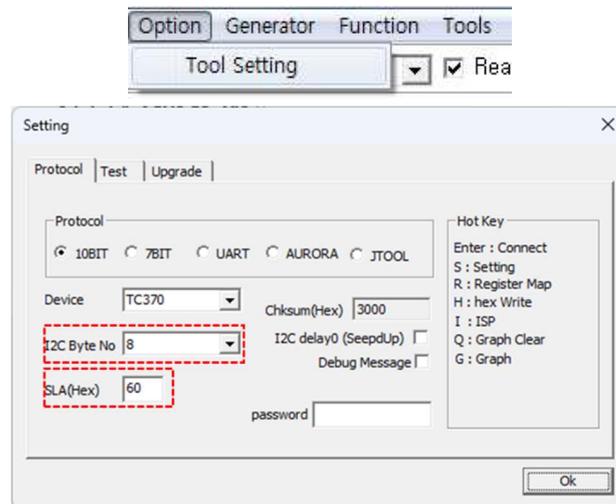
   에서 통신 포트와 기기명을 선택 후 좌측의 버튼을 누르면 Target Board 와 통신이 시작됩니다. 버튼을 다시 누르면 통신이 종료됩니다.

## 1.2 사전 점검 사항

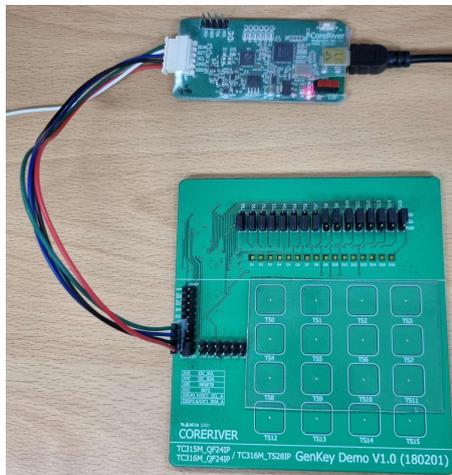
JTool 사용을 위해 miniGenICE3 이 연결 되었는지 확인합니다. 컴퓨터의 [장치관리자]를 실행하여 포트가 정상적으로 인식되었는지 확인하고 몇 번 포트인지 확인합니다.  
 포트가 정상적으로 인식된다면 JTool 의 통신 설정을 합니다.



인식된 포트번호를 선택하고 기기 설정을 합니다.



Option – Tool Setting - [Protocol] 탭에서 **I2C Slave Address** 를 설정해줍니다.  
**SLA** 에 **Slave address** 를 적어 줍니다. (Slave Address 는 Target IC 의 F/W 의 셋팅에 따름)  
**I2C Byte No** 로 I2C 로 읽어올 **Read Byte** 의 크기를 지정합니다. (Default 8Byte 유지 권장)  
 설정을 모두 마쳤으면 [Setting]창을 닫습니다. Target IC 와 miniGenICE3 를 연결합니다.

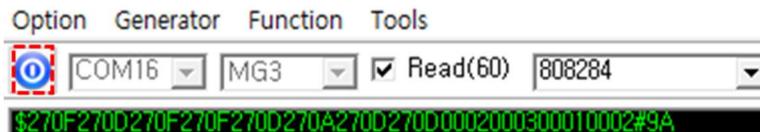


Connect 키를 클릭하여 보드와 연결합니다.

(F/W 를 다운로드 하지 않았으면 정상적으로 진행되지 않을 수 있습니다.)



정상적으로 연결이 됐다면 아래 그림과 같이 아이콘이 변하며 Debug Message 들을 확인할 수 있습니다.



연결이 정상적으로 되지 않으면 위의 사항들을 다시 한번 체크하여 주시고, miniGenICE3 를 전원을 재인가 해보시고, JTool 을 종료 후 재실행 해 보십시오.

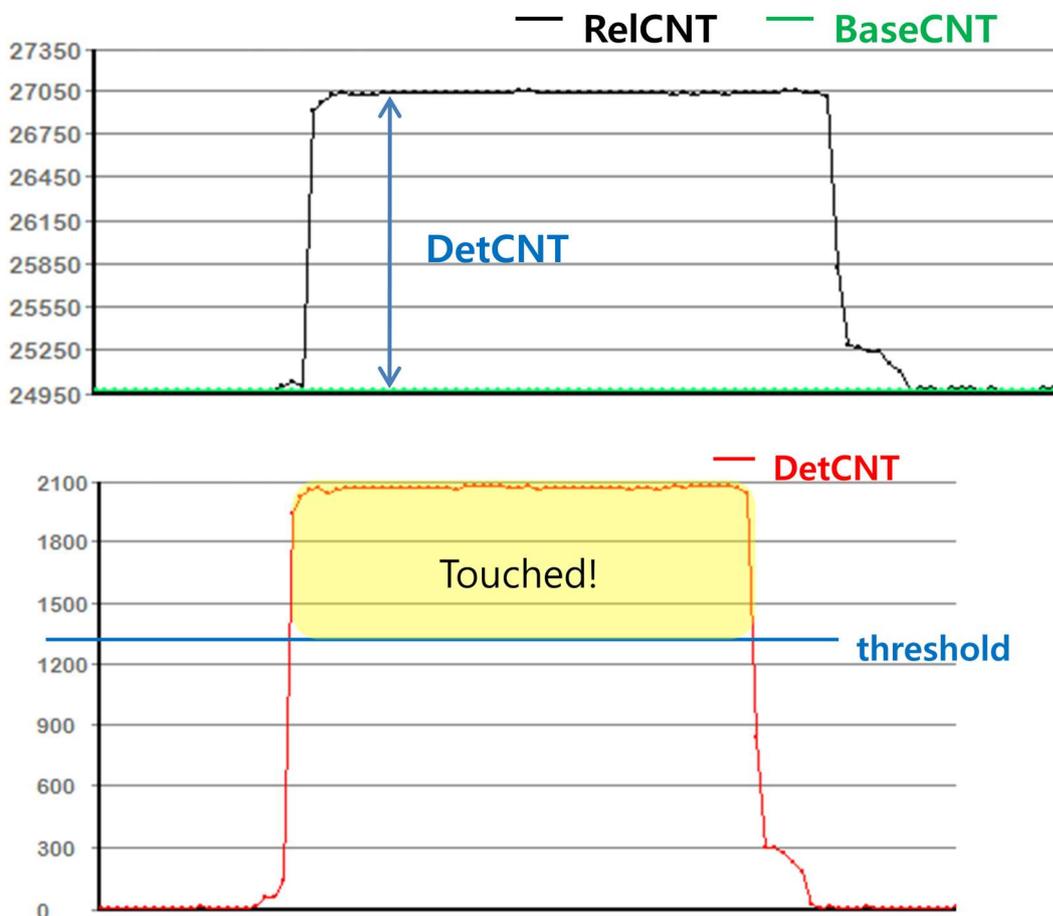
## 1.2.1 TCxxx\_user.h 내 Configuration

터치키 시스템을 구성한 파라미터와 핀 구성을 설정합니다. 아래는 코아리버의 F/W Library 내 설정에 대한 설명입니다.

- ◆ TS\_CH\_NUM
  - 사용할 터치 센서 수
- ◆ SENSING\_ADD\_CNT\_MAX
  - 1회 샘플링 시 센서 값 누적횟수
- ◆ TS\_SR\_DEFAULT\_MODE\_0
  - 터치센서 내부 저항
- ◆ TSENCT\_VAL\_MODE\_0
  - 1회 Sensing 시 사용할 touch sensing pulse의 수
- ◆ I2C\_SLAVE\_ID
  - I2C Slave Address
- ◆ CHx\_MAX
  - x번 Touch Sensor Channel의 터치 판정 최소 값(Threshold)

### ※ Threshold & CNT

**RelCNT** 는 현재 측정된 터치 센서의 값이고, **BaseCNT** 는 터치 검출을 위한 기준 값입니다. **DetCNT** 는 RelCNT - BaseCNT 로 실제 터치 검출에 사용되는 값입니다. 이때 설정해준 Threshold 보다 DetCNT 가 커지게 되면 터치키로 인정이 됩니다. 즉, Threshold 를 설정하여 감도를 조정할 수 있습니다.



## 1.3 F/W 다운로드

[GenICE Manager]를 사용하여 Target Device 에 F/W 를 다운로드합니다. 자세한 내용은 MiniGenICE3 Manual 을 참조하여 주십시오.

## 1.4 디버깅 따라하기

JTool 을 사용하여 target IC 의 메모리에 접근해 Read/Write 를 수행할 수 있습니다.

Target Board 에 연결 후 1.4.1 절, 1.4.2 절과 같이 Read 와 Write 를 사용합니다.



- ◆ 위의 예시와 같이 **I2C address** 와 **2nd address** 를 입력하여 주십시오.  
(**I2C address** 입력은 [Option] – Tool Setting – SLA 란 입력, 반드시 **F/W 설정과 동일**)  
(\*I2C 2nd address 입력은 **아래의 I2C 2nd address 선택 방법**을 참조하십시오.)
- ◆ address 입력 후, 상단 메뉴 좌측의 **Connect** 버튼을 눌러 통신을 시작합니다.

- ◆ I2C 2nd address 선택 방법

```
code UCHAR * i2cBuf[] =
{
    //Jtool
    (UCHAR *)&tc.RelCNT[0],           //80
    (UCHAR *)&tc.RelCNT[4],           //81
    (UCHAR *)&tc.BaseCNT[0],          //82
    (UCHAR *)&tc.BaseCNT[4],          //83
    (UCHAR *)&tc.DetCNT[0],           //84
    (UCHAR *)&tc.DetCNT[4],           //85
    (UCHAR *)&SumBuf2[0],             //86
    (UCHAR *)&SumBuf2[2],             //87
    (UCHAR *)&SumBuf2[4],             //88
    (UCHAR *)&SumBuf2[6],             //89
    (UCHAR *)&tc.ui_touchDownID,      //8A
    (UCHAR *)&tc.ui_cpu1CycleCnt,     //8B
};
```

- 코아리버 Library 의 I2C\_sw.c 내부 i2cBuf 배열의 예시 데이터입니다.
- 사용자가 읽을 데이터를 해당 배열을 통해 수정할 수 있습니다.  
**배열 내용에 따라 읽을 주소가 변동되니, 사용 시 먼저 내용을 확인하여 주십시오.**
- I2C 송수신 시 포인터 배열 i2cBuf 에 접근하며, 해당 Address 부터 순차적으로 송신하게 됩니다. (배열인자 1 개당 **선택한 byte** 수 만큼 Jtool 에 출력)
- **read byte** 단위를 **8Byte** 선택 시, 입력한 **2nd address** 로부터 **8byte** 를 읽습니다.
  - ◆ read byte 단위는 [Option] – Tool Setting – “I2C Byte No”로 설정합니다(**Default 권장**).
- C 언어에서 배열은 메모리에 순차적으로 저장되므로, 2nd address 80 입력 시 tc.RelCNT[0] 의 상위 바이트부터 순차적으로 8byte 를 miniGenICE 에서 수신하게 됩니다.  
Read 예시는 아래와 같으며, 자세한 예시는 13page 의 표를 참조하십시오.  
ex) 80 82 입력 시 → 0~3 번 Channel 의 tc.RelCNT ( tc.RelCNT[0] ~ tc.RelCNT[3] ,8byte)  
+ 0~3 번 Channel 의 tc.BaseCNT ( tc.BaseCNT [0] ~ tc.BaseCNT[3] ,8byte)

## 1.4.1 Read

Read 의 체크박스에 체크☑를 하면 JTool 이 값을 읽어와 변수 리스트 창, 그래프 창에 출력해줍니다.  
Read 옆의 Edit 창에 읽어오고자 하는 2nd Address 를 기입하면 변수 리스트에 표시가 됩니다.  
여러 개의 메모리 값을 읽고 싶을 때는 이어서 번지수를 기입합니다.



다음은 read 절차와 위의 그림에 대한 설명입니다.

1. Read 체크박스 확인
2. I2C address, I2C 2nd address 입력
  - A. I2C address F/W 설정 TCxxx\_user.h - I2C\_SLAVE\_ID
  - B. I2C 2nd address : 80~8F 범위 내 max 4 개
3. 확인할 데이터의 자료형 선택 (Byte / Integer)
4. 그래프에 표기할 데이터 선택
  - A. 변수 리스트 창의 체크박스☑에 체크 시 그래프로 변수 값을 확인할 수 있습니다.
  - B. 좌측 체크박스는 상단의 그래프, 우측 체크박스는 하단의 그래프에 표시를 의미합니다.

자세한 사용 예시는 다음 표와 그래프를 참조하여 주십시오.

■ 아래는 Jtool 을 통해 읽은 예시 데이터입니다. Touch Ch0 터치 on 데이터입니다.

2nd address / Description	I2C_sw.h (Byte 별 데이터 의미)	2Byte	Jtool 데이터 리스트			Int 열
		실제값	C	Byte	Int	데이터 의미
<b>80</b> tc.RelCNT [0]~[3] (Int 형 x 4) <input type="radio"/> Byte <input checked="" type="radio"/> Integer	tc.RelCNT[0] 상위 byte	11279	0	2C	11279	RelCNT[0]
	tc.RelCNT[0] 하위 byte	2C0F	1	0F	10015	RelCNT[1]
	tc.RelCNT[1] 상위 byte	10015	2	27	9952	RelCNT[2]
	tc.RelCNT[1] 하위 byte	271F	3	1F	9948	RelCNT[3]
	tc.RelCNT[2] 상위 byte	9952	4	26	10017	BaseCNT[0]
	tc.RelCNT[2] 하위 byte	26E0	5	E0	9997	BaseCNT[1]
	tc.RelCNT[3] 상위 byte	9948	6	26	9998	BaseCNT[2]
	tc.RelCNT[3] 하위 byte	26DC	7	DC	9996	BaseCNT[3]
<b>82</b> tc.BaseCNT [0]~ [3] (Int 형 x 4) <input type="radio"/> Byte <input checked="" type="radio"/> Integer	tc.BaseCNT[0] 상위 byte	10017	8	27	1265	DetCNT[0]
	tc.BaseCNT[0] 하위 byte	271E	9	1E	19	DetCNT[1]
	tc.BaseCNT[1] 상위 byte	9997	10	27	0	DetCNT[2]
	tc.BaseCNT[1] 하위 byte	270C	11	0C	0	DetCNT[3]
	tc.BaseCNT[2] 상위 byte	9998	12	27	0	
	tc.BaseCNT[2] 하위 byte	270A	13	0A	0	
	tc.BaseCNT[3] 상위 byte	9996	14	27	0	
	tc.BaseCNT[3] 하위 byte	2711	15	11	0	
<b>84</b> tc.DetCNT [0]~[3] (Int 형 x 4) <input type="radio"/> Byte <input checked="" type="radio"/> Integer	tc.DetCNT[0] 상위 byte	1265	16	04	0	
	tc.DetCNT[0] 하위 byte	04F1	17	F1	0	
	tc.DetCNT[1] 상위 byte	19	18	00	0	
	tc.DetCNT[1] 하위 byte	0013	19	13	0	
	tc.DetCNT[2] 상위 byte	0	20	00	0	
	tc.DetCNT[2] 하위 byte	0000	21	00	0	
	tc.DetCNT[3] 상위 byte	0	22	00	0	
	tc.DetCNT[3] 하위 byte	0000	23	00	0	

◆ I2C\_sw.h 에 입력한 순서대로 Jtool 의 "Int"열을 읽으시면 됩니다.

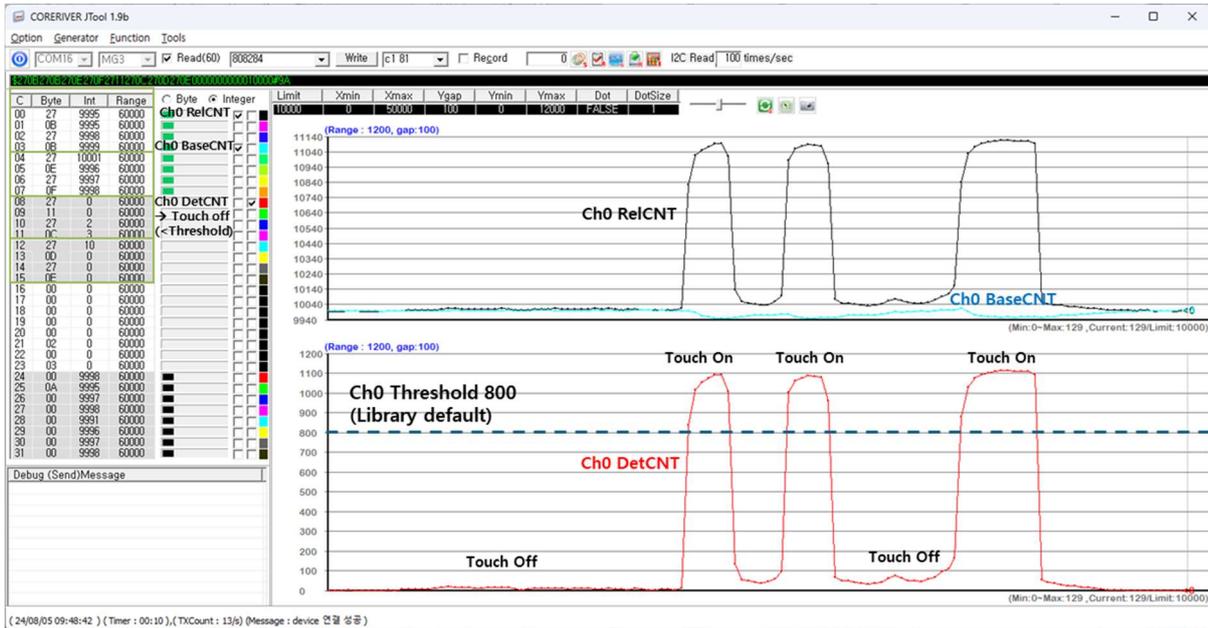
■ Int 열은 2Byte 씩 Int 크기만큼 읽어옵니다. (Int 데이터 내용 참조)

◆ C, Byte, Int 열은 Jtool 변수 리스트 창 출력 내용입니다.

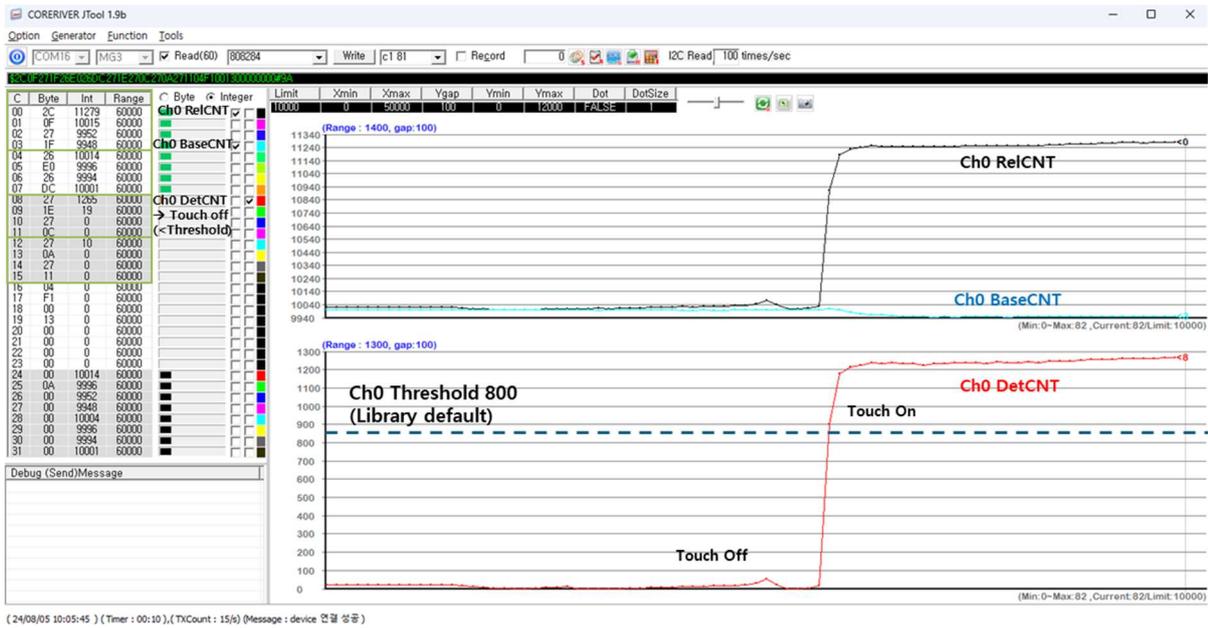
■ 실제 받아온 데이터로, I2C\_sw.h 데이터 의미를 참조하여 읽으시면 됩니다.

◆ 다음 페이지의 Ch0 터치 시 그래프 예시를 참조하여 주십시오.

■ Jtool 을 사용한 예시 화면입니다. 미 터치 시 데이터입니다.

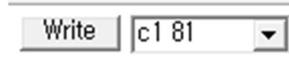


■ Jtool 을 사용한 예시 화면입니다. 13 page 표와 동일하며, Ch0 터치 시 그래프입니다.



## 1.4.2 Write

Write 는 Write Edit 창에 전송하고자 하는 데이터를 입력 후 [Write]버튼을 클릭합니다.



- ① [Connect] 버튼을 눌러 타겟 보드와 통신을 Connect 합니다. 

- ② [Write] 버튼을 눌러 원하는 데이터를 전송할 수 있습니다. 

Write로 전송하는 송신 데이터의 길이는 4byte로 정해져 있습니다.

4Byte보다 짧을 경우 0xFF가 채워져 전송되며, 4Byte보다 길 경우 4Byte 이후 데이터는 전송되지 않습니다.

ex) c1 81 입력(16진수) 후 write 클릭 → c1 81 FF FF 송신

## 2 Appendix

contents	Revision	Date
First released.(국문 버전 생성)	1.00	2011-08-01
Library 업데이트에 따른 TICTOC 내용 제거 및 지원 기능 수정	1.10	2024-08-02