COM_SCOPE 사용설명서 V1.0

기다리던 바로 그 !!! 오실로스코프.

전력전자 제품 개발에 최강의 광 통신 오실로스코프 출시 !!!

RealSYS

작성일자:2010.5.12

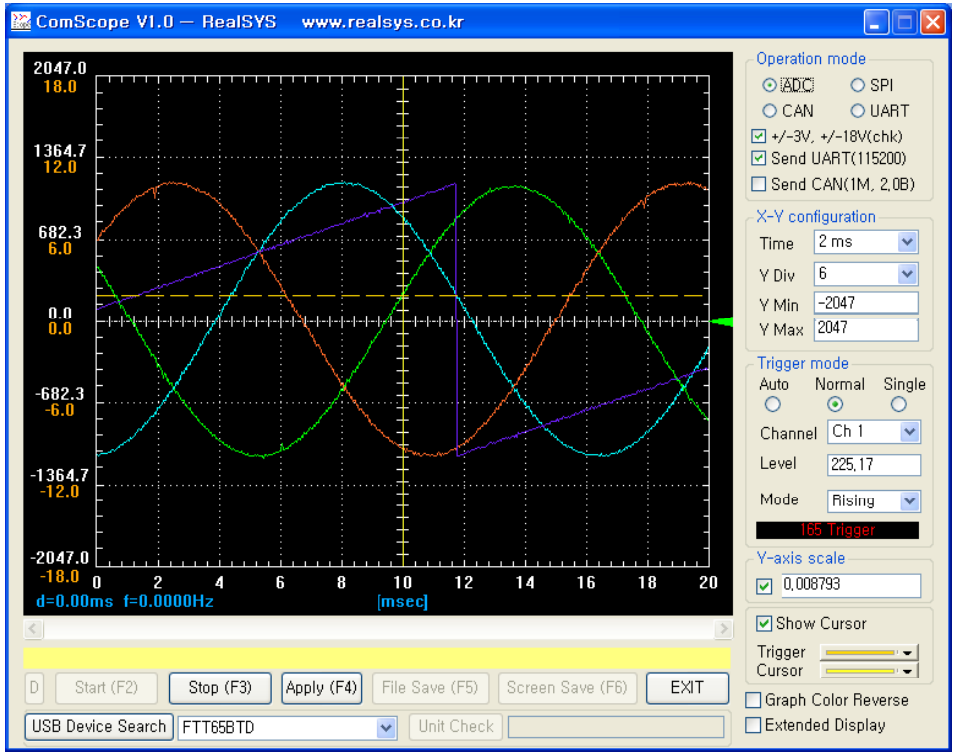
COM_SCOPE의 특징:

(전위차, 전기적 절연 및 노이즈에 강한)

- 4 채널 광통신 오실로스코프 (PC 형, USB 인터페이스, 광 케이블 사용)
- 다양한 신호 입력 방법: AIN(아날로그 신호), SPI 통신, CAN 통신, UART(비동기) 통신
  - AIN : 4채널 아날로그 입력 (12비트 분해 능, -18V~+18V 입력 범위, 최대 샘플링 5usec)
  - SPI 통신: 쉬운 통신 프로토콜(4워드 연속 전송), 16비트, 5Mbps, 15usec 주기 관찰 가능
  - CAN 통신: 1Mbps, 2.0A/B 선택 가능. 4워드(8바이트) 데이터, 150usec 주기 관찰 가능
  - UART 통신: 쉬운 통신 프로토콜, 115200/38400bps 선택, self test 기능
- 데이터 저장 기능: 화면 캡 쳐, 엑셀(csv 포맷) 형태 저장, 장시간 데이터 실시간 저장 기능
- 화면 배경 및 신호 색 변경 가능, 트리거 설정, 에지 설정 등
- 타임 조정: 격자 간격 100usec ~ 1sec (전체 화면 10초)
- DSP, ARM, AVR 등 MCU에 독립적 사용 가능
- 소형 크기: 2개의 모듈 사용, 모듈 크기: 56.5 x 62 x 22 mm

컴스코프 본체

컴퓨터 인터페이스 모듈



기본 제공 케이블: 사용자 보드에 맞추어 적절히 변경 사용

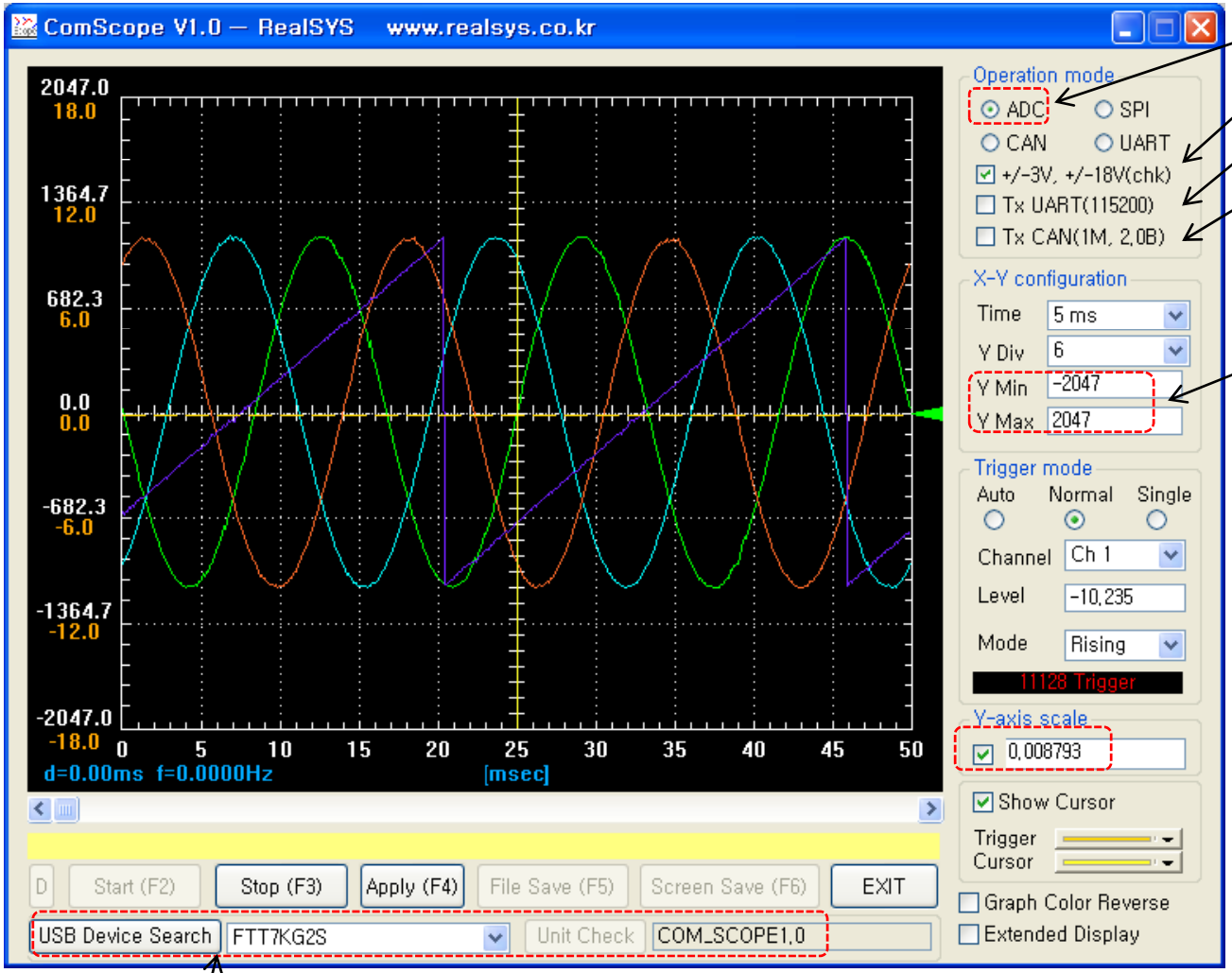
인터페이스 신호

15	GND	5V	TXD	CANH	CLK	GND	AIN2	AIN0	1
16	GND	5V	RXD	CANL	SIMO	STE	AIN3	AIN1	2
	전원입력		UART	CAN	SPI		AIN		

리얼시스 www.realsys.co.kr Tel:031-342-3000, Fax:031-343-0003 경기도 안양시 동안구 호계동 1027 안양 IT밸리 504호

아날로그 입력 모드 선택

AIN 모드는 일반 오실로스코프처럼 아날로그 입력 신호를 표시합니다.
일반 오실로스코프에 비해서 변환시간(최대 5usec)이 느리지만 전류 신호 관찰 등에 활용 가능합니다. A/D 변환 분해능:12 비트



ADC 선택

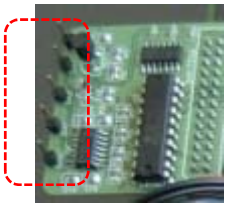
입력 전압 범위 선택

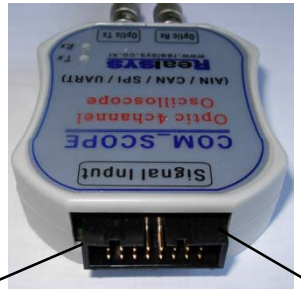
AIN 데이터를 UART로 송출 가능(10msec 주기)

AIN 데이터를 CAN으로 송출 가능(1msec 주기)

아날로그 신호 관찰 예:
DAC의 4채널 데이터 관찰

12비트 A/D
사용:
0 ~ 4096
또는
-2047~+2047





18V / 2047

GND	5V					AIN2	AIN0
GND	5V					AIN3	AIN1

DAC의 4채널 데이터 관찰 예

PC와의 인터페이스 모듈(CNV_OPTIC)은 다른 용도로도 사용이 가능하므로
모듈 선택 및 장치 확인 버튼 필요

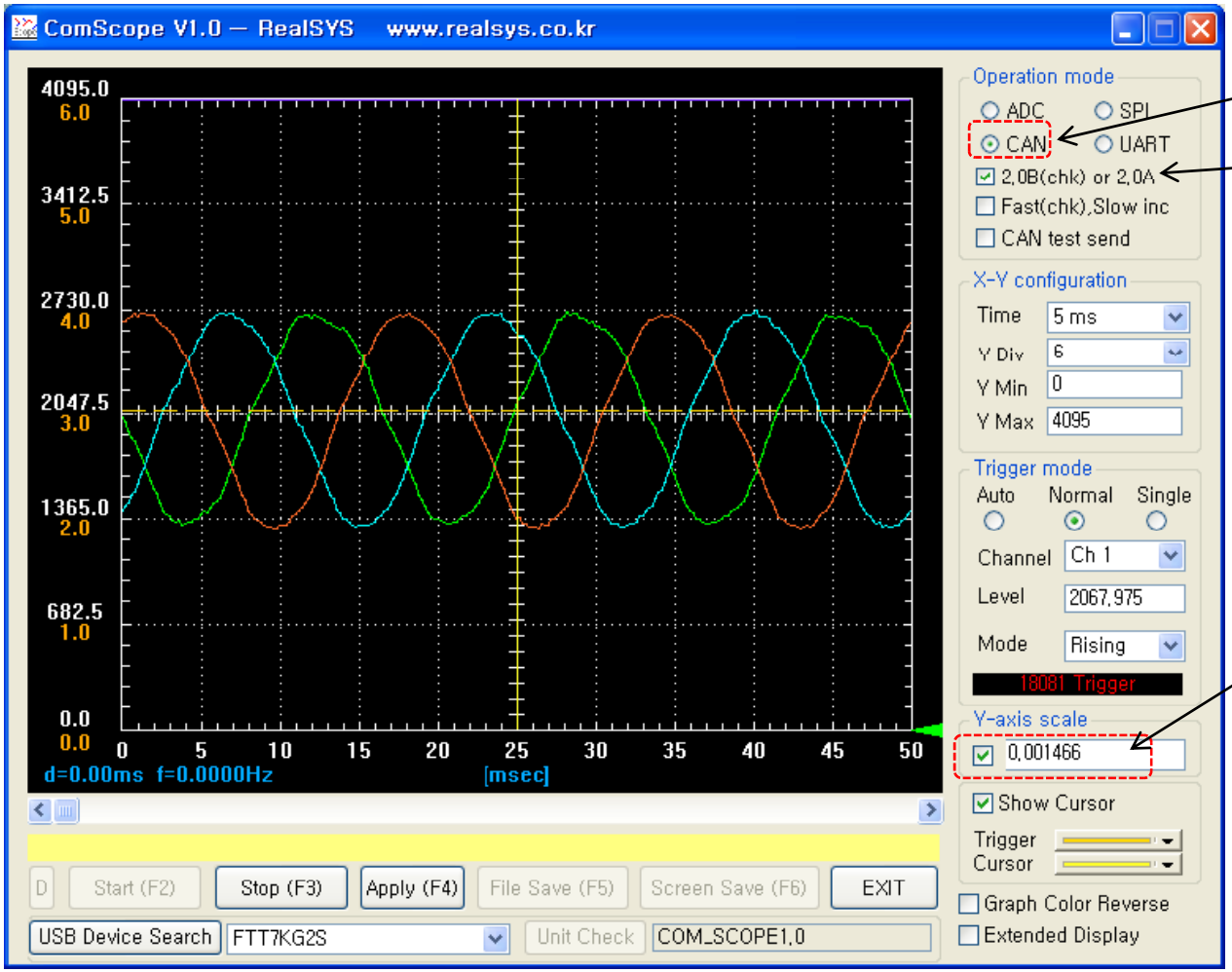
5V 전원 공급 필요

아날로그 신호입력

리얼시스 www.realsys.co.kr Tel:031-342-3000, Fax:031-343-0003 경기도 안양시 동안구 호계동 1027 안양 IT밸리 504호

CAN 통신 모드 선택

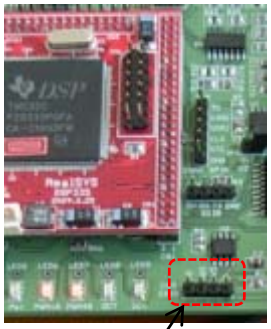
CAN 통신을 사용한 전류 파형 관찰 예



CAN 선택

2.0B 사용

3V / 4095



MOT335 에서 CAN 통신
접속 커넥터 예

GND	5V	11	CANH	7	5	3	1
GND	5V	12	CANL	8	6	4	2

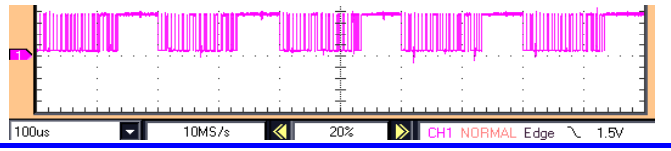
5V 전원 공급 필요

CAN 통신

CAN 통신 : 1Mbps, 2.0A 또는 2.0B 선택, 8바이트를 4워드로 사용

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
채널0 데이터		채널1 데이터		채널2 데이터		채널3 데이터	

CAN 통신 파형: 데이터 전송 시간 약 130usec 정도 걸림

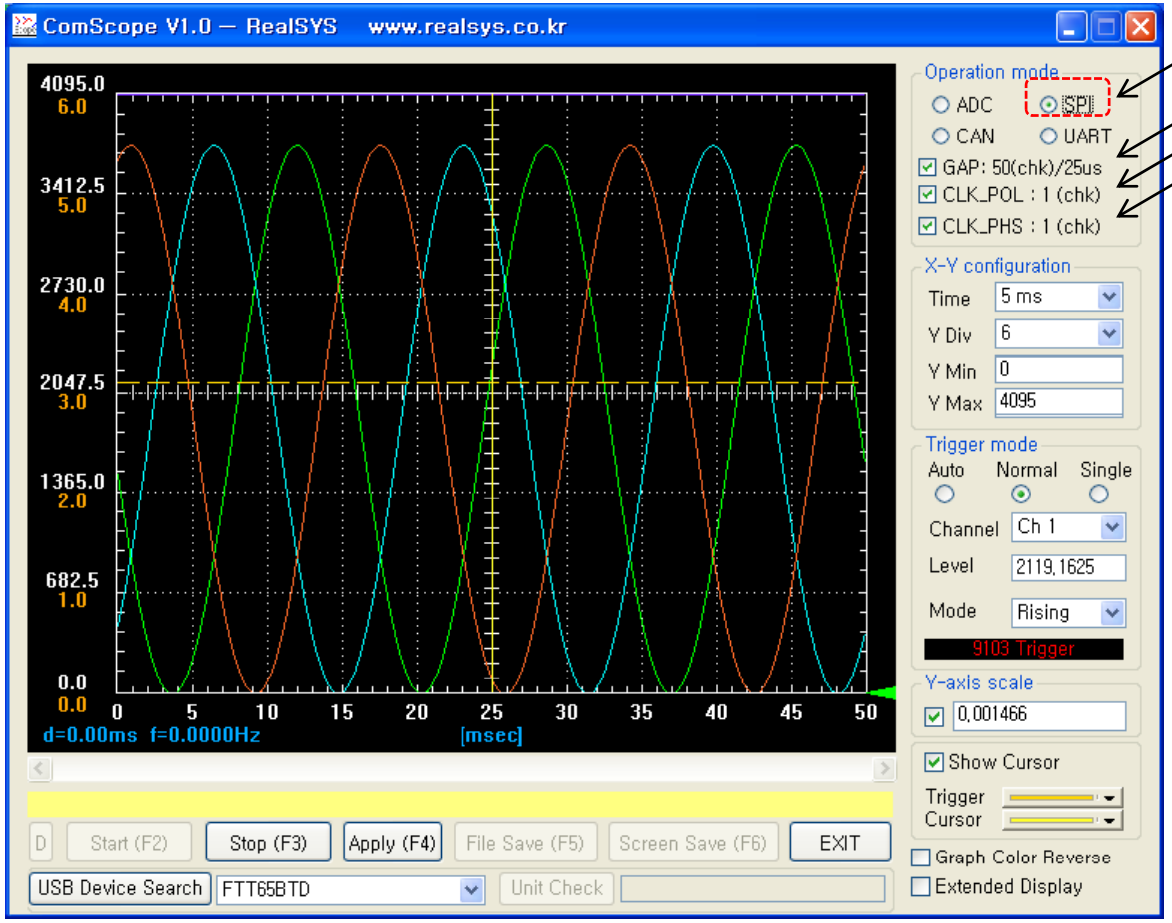


리얼시스 www.realsys.co.kr Tel:031-342-3000, Fax:031-343-0003 경기도 안양시 동안구 호계동 1027 안양 IT밸리 504호

SPI 통신 모드 선택

SPI 통신 동작모드는 보다 빠른 (5Mhz 정도) 샘플링으로 데이터 관찰이 가능하며,
COM_SCOPE는 슬래이브 모드로 동작하므로 클럭 등 신호는 외부 장치가 출력 제어합니다.

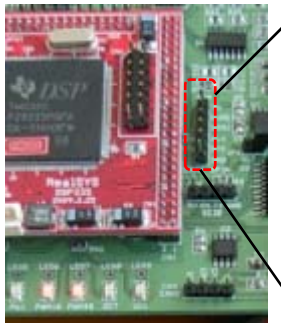
SPI 통신 프로토콜: 4워드 데이터를 연속으로 전송하면 각 채널 데이터로 인식, 만일 2워드 값만 보내면 2개의 채널 값으로 인식.



SPI 모드선택

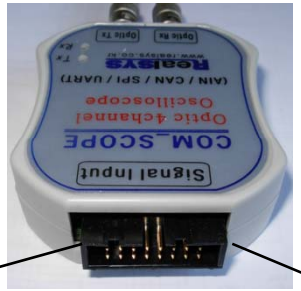
데이터 프레임 간격 선택
SPI 클럭 극성 선택
SPI 클럭 위상 선택

MOT335보드 접속 예



DSP B/D	COM_SCOPE
5V	5V
SIMO	SIMO
SOMI	
CLK	CLK
STE	STE
GND	GND

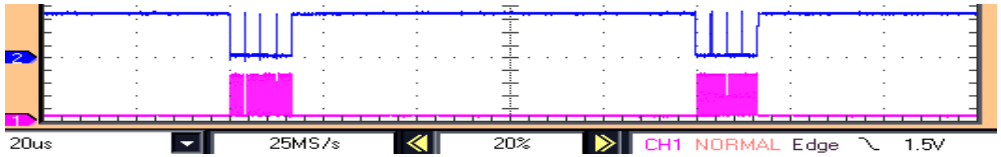
SPI 신호는
3.3V 로직 레벨 임



GND	5V	11	9	CLK	GND	3	1
GND	5V	12	10	SIMO	STE	4	2

5V 전원 공급 필요 SPI 통신

SPI 파형 : 4워드 전송 시간(15usec 정도), CAN보다 섬세한 관찰 가능



SPI 클럭은 내부 최대 샘플링 주기(5usec)를 고려하여 5.0MHz 이내 사용 권장(왼쪽 파형은 5.33MHz 정도에서 동작 예)

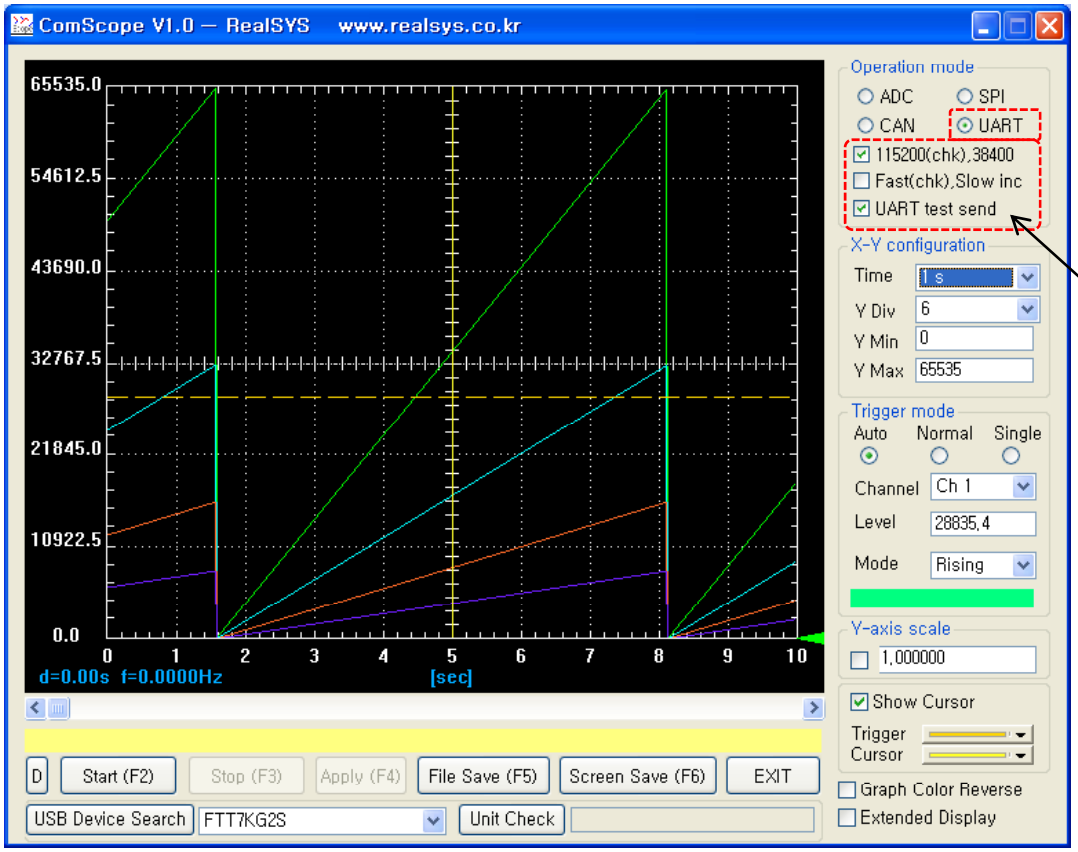
리얼시스 www.realsys.co.kr Tel:031-342-3000, Fax:031-343-0003 경기도 안양시 동안구 호계동 1027 안양 IT밸리 504호

COM_SCOPE: 비동기 통신 프로토콜

COM_SCOPE에는 비동기통신(UART)으로도 모니터링이 가능한데, 아래와 같이 매우 쉬운 구조로 되어 있습니다.

HHHH	HHHH	HHHH	HHHH	CR

- HHHH는 WORD(16비트) ASCII-HEX 형태 임
- 최대 4채널 입력, 데이터 사이에 SPACE(' ') 또는 컴머(',')가 들어가도 됨
- CR(0x0d)이 오면 새로운 시작을 알림
- Check Sum 기능 없음
- 매우 간단한 프로토콜
- 통신 속도: 115200 / 38400 선택 가능, 8비트 데이터, 1스톱, No-parity



GND	5V	TXD	9	7	5	3	1
GND	5V	RXD	10	8	6	4	2

UART 신호: TTL 레벨

자체 시험 모드가 있어서 "UART test send"를 체크하면 아래와 같은 형태로 데이터가 송신되고 TXD와 RXD를 연결해주는 경우에 왼쪽 파형과 같은 톱니 파 형태가 표시됩니다.
UART로 모니터링을 원하는 경우 RXD에 통신 신호를 공급해주면 됩니다.

```
BF44 5FA2 2FD1 17E8
BFA8 5FD4 2FEA 17F5
C00C 6006 3003 1801
C070 6038 301C 180E
C908 6484 3242 1921
C96C 64B6 325B 192D
C9D0 64E8 3274 193A
CA34 651A 328D 1946
CA98 654C 32A6 1953
CAFC 657E 32BF 195F
CB60 65B0 32D8 196C
CBC4 65E2 32F1 1978
CC28 6614 330A 1985
```

SPI 통신 프로그램 예

상세한 응용 예제는 향후 기술강좌 형태로 진행 예정

SPI 초기화 함수, 데이터 출력 함수

```
void   init_spiMon(void){
   EALLOW;
   /* Enable internal pull-up for the selected pins */
   GpioCtrlRegs.GPAPUD.bit.GPIO16 = 0;   // Enable pull-up on GPIO16 (SPISIMOA)
   GpioCtrlRegs.GPAPUD.bit.GPIO17 = 0;   // Enable pull-up on GPIO17 (SPISOMIA)
   GpioCtrlRegs.GPAPUD.bit.GPIO18 = 0;   // Enable pull-up on GPIO18 (SPICLKA)
   GpioCtrlRegs.GPAPUD.bit.GPIO19 = 0;   // Enable pull-up on GPIO19 (SPISTEA)

   GpioCtrlRegs.GPAQSEL2.bit.GPIO16 = 3;   // SPISIMOA : 0= Sync to SYSCLKOUT, 1= 3 sa
   GpioCtrlRegs.GPAQSEL2.bit.GPIO17 = 3;   // SPISIMOA : 0= Sync to SYSCLKOUT, 1= 3 sa
   GpioCtrlRegs.GPAQSEL2.bit.GPIO18 = 3;   // SPISIMOA : 0= Sync to SYSCLKOUT, 1= 3 sa
   GpioCtrlRegs.GPAQSEL2.bit.GPIO19 = 3;   // SPISIMOA : 0= Sync to SYSCLKOUT, 1= 3 sa

   /* Configure SPI-A pins using GPIO regs*/
   GpioCtrlRegs.GPAMUX2.bit.GPIO16 = 1;  // Configure GPIO16 as SPISIMOA
   GpioCtrlRegs.GPAMUX2.bit.GPIO17 = 1;  // Configure GPIO17 as SPISOMIA
   GpioCtrlRegs.GPAMUX2.bit.GPIO18 = 1;  // Configure GPIO18 as SPICLKA
   GpioCtrlRegs.GPAMUX2.bit.GPIO19 = 1;  // Configure GPIO19 as SPISTEA

   EDIS;

   // Disable the FIFO operation
   SpiaRegs.SPIFFTX.all = 0xe000;
   SpiaRegs.SPIFFRX.all = 0x2040;
   SpiaRegs.SPIFFCT.all = 1;   // FIFO delay : 1bit

// SpiaRegs.SPICCR.all =0x000F;     // Reset on, rising edge, 16-bit char bits
// SpiaRegs.SPICTL.all =0x000E;     // Enable master mode, one half-cycle delay phase,I

   SpiaRegs.SPICCR.all =0x004F;     // Reset on, falling edge, 16-bit char bits
   SpiaRegs.SPICTL.all =0x0006;     // Enable master mode, normal phase,Enable TALK, SI

// SpiaRegs.SPIBRR =0x0009;         // Maximum SPI speed(LSPCLK(37.5MHz)/10 = 3.75MHz)
   SpiaRegs.SPIBRR =0x0006;         // Maximum SPI speed(LSPCLK(37.5MHz)/7 = 5.357MHz)
   SpiaRegs.SPICCR.all =0x008F;     // Relinquish SPI from Reset
   SpiaRegs.SPIPRI.bit.FREE = 1;    // Set so breakpoints don't disturb xmission
}

void out_spiMon(WORD *p){
   if(SpiaRegs.SPIFFTX.bit.TXFFST != 0) return;
   SpiaRegs.SPITXBUF = p[0];
   SpiaRegs.SPITXBUF = p[1];
   SpiaRegs.SPITXBUF = p[2];
   SpiaRegs.SPITXBUF = p[3];
}
```

내부의 큐를 사용하여 4개의 데이터를 연달아서 전송 가능하므로 DSP 처리 부단이 별로 없음

SPI 데이터 송신 프로토콜: 매우 단순 함
4워드를 연속하여(50usec 이내로) 전송하면
첫 번째는 채널0값, 두 번째는 채널1 등으로 인식 함
통신 속도는 5Mbps 이내로 권장

SPI 출력 예)

```
interrupt void epwm1_timer_isr(void){
   pwm_cnt++;
   cur_u = AdcRegs.ADCRESULT0 >> 4;
   cur_v = AdcRegs.ADCRESULT1 >> 4;
   cur_w = AdcRegs.ADCRESULT2 >> 4;
   vr = AdcRegs.ADCRESULT3 >> 4;
   tz1_flag = TZ1FLAG; // trip check
   if((tz1_flag & 0x04)==0) pwm123_vf();
   if(sel_data==0){     // ref
       c_txd[0] = uu;
       c_txd[1] = vv;
       c_txd[2] = ww;
       c_txd[3] = vr;
   }
   else{    // cur,
       c_txd[0] = cur_u;
       c_txd[1] = cur_v;
       c_txd[2] = cur_w;
       c_txd[3] = vr;
   }
   out_spiMon(c_txd);   // 100usec period
   if(t_pwm123 ^= 1){   // 200usec period
           canB_data_mon(c_txd);
   }
   ADC_START;
   // Clear INT flag for this timer
   EPwm1Regs.ETCLR.bit.INT = 1;
   // Acknowledge this interrupt to receive
   PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
}
```

여러 변수의 값을 선택적으로 전송 처리도 가능

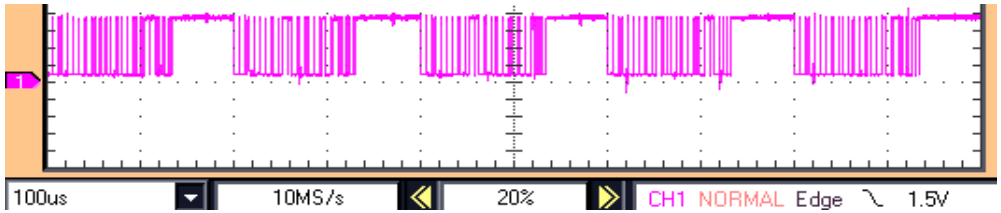
리얼시스 www.realsys.co.kr Tel:031-342-3000, Fax:031-343-0003 경기도 안양시 동안구 호계동 1027 안양 IT밸리 504호

CAN 데이터 모니터링 송신 함수

```
BOOL canB_data_mon(WORD *p){
    if(ECanbRegs.CANTRS.bit.TRS1) return 0;
    ECanbRegs.CANME.bit.ME1 = 0;     // MBX1 disable
    ECanbMboxes.MBOX1.MSGID.all = 0x80000000;        // EXT_TYPE, ID=0
    ECanbRegs.CANME.bit.ME1 = 1;     // MBX1 enable

/* Write to the mailbox RAM field */
    ECanbMboxes.MBOX1.MDL.all = ((long)p[1]<<16) | p[0];
    ECanbMboxes.MBOX1.MDH.all = ((long)p[3]<<16) | p[2];
    ECanbMboxes.MBOX1.MSGCTRL.bit.DLC = 8;
    ECanbRegs.CANTRS.bit.TRS1 = 1;
    return 1;
}
```

CAN 통신 신호 관찰



통신 속도 1Mbps, 2.0B (ID=0사용)로 송신하는 경우 전송 시간이 130usec 정도 소요되므로 데이터를 200usec 주기로 전송하면 바람 직.

따라서 보다 고속의 전송을 원하는 경우는 CAN보다 SPI를 사용하는 것이 좋습니다.

노이즈 면에서는 CAN이 유리하리라 생각.

CAN 통신 데이터 형태: 8바이트를 2바이트씩 구분하여 4워드 전송

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
채널0 데이터		채널1 데이터		채널2 데이터		채널3 데이터	

CAN 통신의 송신 처리도 하드웨어적으로 전송이 되므로 DSP 처리시간에 별로 부담을 주지 않음.

CAN 출력 예)

```
interrupt void epwm1_timer_isr(void){
    pwm_cnt++;
    cur_u = AdcRegs.ADCRESULT0 >> 4;
    cur_v = AdcRegs.ADCRESULT1 >> 4;
    cur_w = AdcRegs.ADCRESULT2 >> 4;
    vr = AdcRegs.ADCRESULT3 >> 4;
    tz1_flag = TZ1FLAG; // trip check
    if((tz1_flag & 0x04)==0) pwm123_vf();
    if(sel_data==0){     // ref
        c_txd[0] = uu;
        c_txd[1] = vv;
        c_txd[2] = ww;
        c_txd[3] = vr;
    }
    else{    // cur.
        c_txd[0] = cur_u;
        c_txd[1] = cur_v;
        c_txd[2] = cur_w;
        c_txd[3] = vr;
    }
    out_spiMon(c_txd);  // 100usec period
    if(t_pwm123 ^= 1){  // 200usec period
        canB_data_mon(c_txd);
    }
    ADC_START;
    // Clear INT flag for this timer
    EPwm1Regs.ETCLR.bit.INT = 1;
    // Acknowledge this interrupt to receive
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
}
```

프로그램 설치 및 입력 커넥터 정보

실행 프로그램 구성은 아래와 같이 USBCOM Driver와 ComScope.exe 파일로 되어있습니다. USB 드라이버 설치는 일반적인 방법(USB 장치를 연결하였을 때 드라이버 위치를 알려주는)으로 설치하시면 되고, ComScope.exe는 setup 파일이 아니고 단독으로 실행하는 파일로 원하는 위치(바탕화면)에 놓고 사용하시면 됩니다.

실행 파일 및 드라이버 디렉터리:

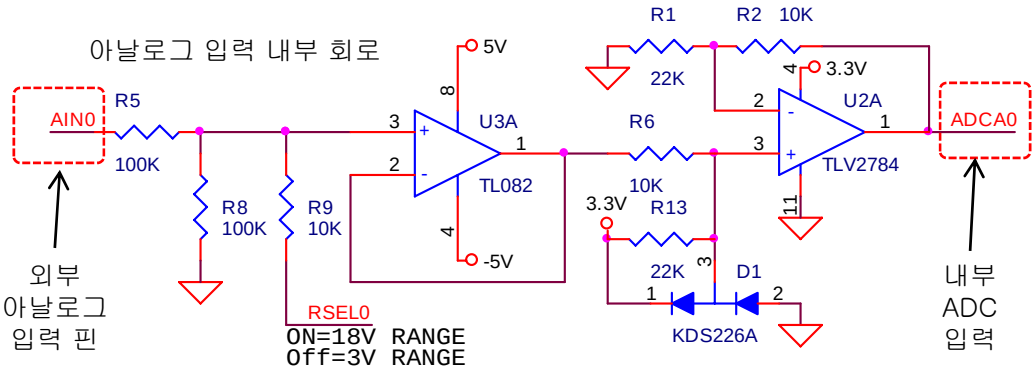
이름	크기	종류	수정한 날짜
USBCOM Driver		파일 폴더	2010-04-21 오후 1:38
ComScope.exe	344KB	응용 프로그램	2010-05-12 오전 8:38

드라이버 디렉터리 내용:

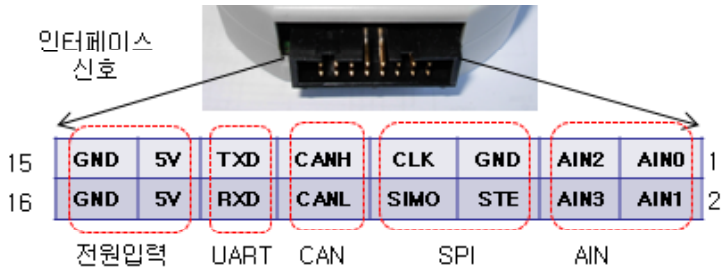
이름	크기	종류	수정한 날짜
amd64		파일 폴더	2010-04-21 오후 1:38
i386		파일 폴더	2010-04-21 오후 1:38
CDM 2 06 00 Release Info.rtf	102KB	서식 있는 텍스트	2009-10-27 오전 11:21
ftd2xx.h	23KB	C/C++ Header File	2008-10-29 오후 4:59
ftdibus.cat	12KB	보안 카탈로그	2009-11-03 오후 1:33
ftdibus.inf	3KB	설치 정보	2010-04-21 오전 11:21
ftdiport.cat	11KB	보안 카탈로그	2009-11-03 오후 1:33
ftdiport.inf	5KB	설치 정보	2010-04-21 오전 11:22
ftdiunin.exe	184KB	응용 프로그램	2006-05-24 오전 10:40
LogoVerificationReport.pdf	42KB	Adobe Acrobat 7.0 D...	2009-11-03 오후 2:22

아날로그 입력은 +-3V 또는 +-18V 영역인데 더욱 높은 전압 범위를 원하는 경우 아래 회로를 참조하여 외부에 저항을 연결하여 사용바랍니다.

아날로그 입력 내부 회로



16핀 커넥터

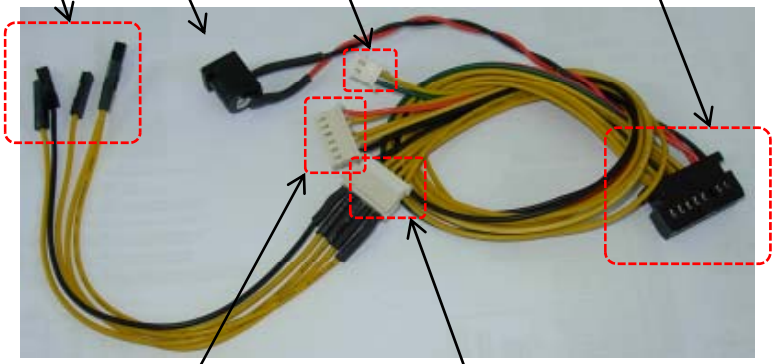


5V 어댑터 커넥터
(중앙:5V)

CAN 커넥터

1	2
노란색	초록색
CANH	CANL

AIN 개별
커넥터



AIN 커넥터

1	2	3	4	5
검정색	노란색	노란색	노란색	노란색
GND	AIN3	AIN2	AIN1	AIN0

SPI 커넥터

1	2	3	4	5	6
적색	노란색		노란색	노란색	검정색
5V	SIMO		CLK	STE	GND

리얼시스 www.realsys.co.kr Tel:031-342-3000, Fax:031-343-0003 경기도 안양시 동안구 호계동 1027 안양 IT밸리 504호