ROBO™ BLOCK

ROBOBLOCK SYSTEM

www. roboblock.co.kr

DTMMA7260 Module

ROBOBLOCK SYSTEM Co., Ltd.

(주)로보블럭시스템

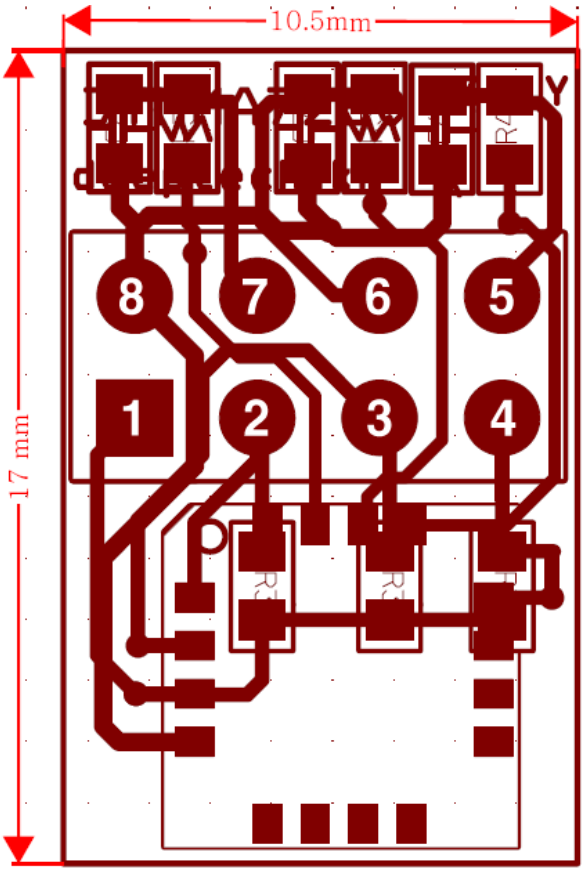
1. DTMMA7260

DTMMA7260은 Freescale사(구 모토로라)의 MMA7260Q를 이용하여 주변 회로를 추가하여 사용할 수 있게 모듈형식으로 만든 제품이다 X,Y,Z 3축의 가속도값을 얻을 수 있으며 전력소비를 줄이기 위해 슬립모드(Sleep Mode) 기능이 있다. 추가적으로 센서의 감도를 조절할 수 있는 제품이다.

1.1. 특징

< DTMMA7260 특징>

동작 전압	:	2.2V ~ 3.6V
가속도 선택	:	1.5g/2g/4g/6g
소비 전류 (Sleep Mode)	:	500uA (3uA)
출력신호	:	Low Pass Filter 사용



Copyright©2009 roboblock System. All right Reserved.

1.2. 핀 설명

핀 번호	핀 명	기 능
1	VDD	전원 (2.2V ~ 3.3V)
2	gSEL1	g-select 1
3	gSEL2	g-select 2
4	/SLEEP	Sleep Mode
5	ZOUT	Z out
6	YOUT	Y out
7	XOUT	X out
8	GND	Ground

gSEL1,gSEL2 핀은 4.7KOhm 저항으로 풀업되어 있어 다른 신호에 연결하지 않을 경우 기본적으로 6g(200mV/g) 로 동작하게 된다.

/SLEEP 핀은 4.7KOhm 저항으로 풀업되어 있어 다른 신호에 연결하지 않을 경우 기본적으로 Normal Mode로 동작하게 된다.

gSEL1, gSEL2 핀 설정에 따라 가속도 값을 결정할 수 있다.

gSEL1	gSEL2	gRange	Sensitivity
0	0	1.5g	800mV/g
0	1	2g	600mV/g
1	0	4g	300mV/g
1	1	6g	200mV/g

Copyright©2009 roboblock System. All right Reserved.

2. 예제 프로그램

2.1. DTES100 사용

DTES100 를 사용하면 쉽게 제어할 수 있다.

DTES100의 X축은 CH0에 Y축은 CH1에 Z축은 CH2에 연결이 되어 있고 4g로 테스트를 한다면 다음과 같은 코드가 될 것이다.

```
    PORT_DirOutput(gSEL1 | gSEL2);
    PORT_SetOutput( gSEL1);
    PORT_ClearOutput( gSEL2);

    struct _DTES_COMM_INFO infoADC0;
    struct _DTES_COMM_INFO infoADC1;
    struct _DTES_COMM_INFO infoADC2;
    struct _DTES_COMM_INFO infoADC3;

    // 채널0,1,2,3을 1Khz로 초기화를 하고
    m_DTES.InitADC(ADC_CH0|ADC_CH1|ADC_CH2|ADC_CH3,1000000);

while(1){  // 값을 읽어온다.
 UINT V0 =0;
 for(iTemp=0;iTemp<infoADC3.iWrPos;iTemp+ =2){
        V0 = infoADC0.chRxData[iTemp+ 1]<<8 |  infoADC0.chRxData[iTemp];
        iData0[iTemp/2] = V0;
        V0 = infoADC1.chRxData[iTemp+ 1]<<8 | infoADC1.chRxData[iTemp];
        iData1[iTemp/2] = V0;
        V0 = infoADC2.chRxData[iTemp+ 1]<<8 | infoADC2.chRxData[iTemp];
        iData2[iTemp/2] = V0;
   }
}
```

< DTES100을 이용한 DTMMA7260 예제 >

Copyright©2009 roboblock System. All right Reserved.

2.2. ATMEGA128 사용

ATMEGA128로 제어하는 소스(코드비전)는 다음과 같다.

```
 // ADC 초기화
int iX; // X축 값
int iY; // Y축 값
int iZ; // Z축 값

          // ADC 초기화
                    ACSR=0x00;
                    SFIOR=0x00;

                    ADCSRA=0x87 ;
                    ADMUX=0x00;
                    Delay_us(400);

                    ADCSRA = 0xD7;

    while(1){
        while(0x10 != (ADCSRA& 0x10)){}
        iX = ADCW;
        ADMUX = 0x01;

        while(0x10 != (ADCSRA& 0x10)){}
        iY = ADCW;
        ADMUX = 0x02;

        while(0x10 != (ADCSRA& 0x10)){}
        iZ = ADCW;
        ADMUX=0x00;
        //iX, iY, iZ 의 값에 X,Y,Z의 값이 저장된다.
    }
```


Copyright©2009 roboblock System. All right Reserved.

ROBOBLOCK SYSTEM Co., Ltd.

Homepage : http://www.roboblock.com

Seoul office : 4th floor, 8-1 Mullaedong 4ga Youngdeungpo-ku, Seoul, Korea

Email : robotoz@hotmail.com

Tel : +82 2 2679 8556

Fax : +82 2 2679 8557

Head Office : Room No. 502, Bucheon Technopark #401, Yakdae-Dong, Wonmi-ku, Bucheon-city, Kyunggi-do, Korea

Tel : +82 - 70-7764-8556

Copyright©2009 roboblock System. All right Reserved.