

ZBM 사용자매뉴얼

2007. 6. 25

OL마이크로웨이브

<http://olmicrowaves.com>

목 차

Part 1. ZBM ZigBee 개발 키트 소개

- 1) 제품 개요
- 2) 제품 특징
- 3) 제품 사양

Part 2. 프로그램 개발 환경

- 1) WinAVR 설치
- 2) AVR Studio 설치
- 3) JTAG ICE 설치
- 4) USB(VCP) Driver 설치
- 5) ZBM 설치

Part 3. 예제 프로그램

- 1) 예제 프로그램 설치
- 2) 예제 프로그램 실행
- 3) 예제 프로그램 설명
- 4) 사용자 프로그램 개발

Part 4. ZBM 실습

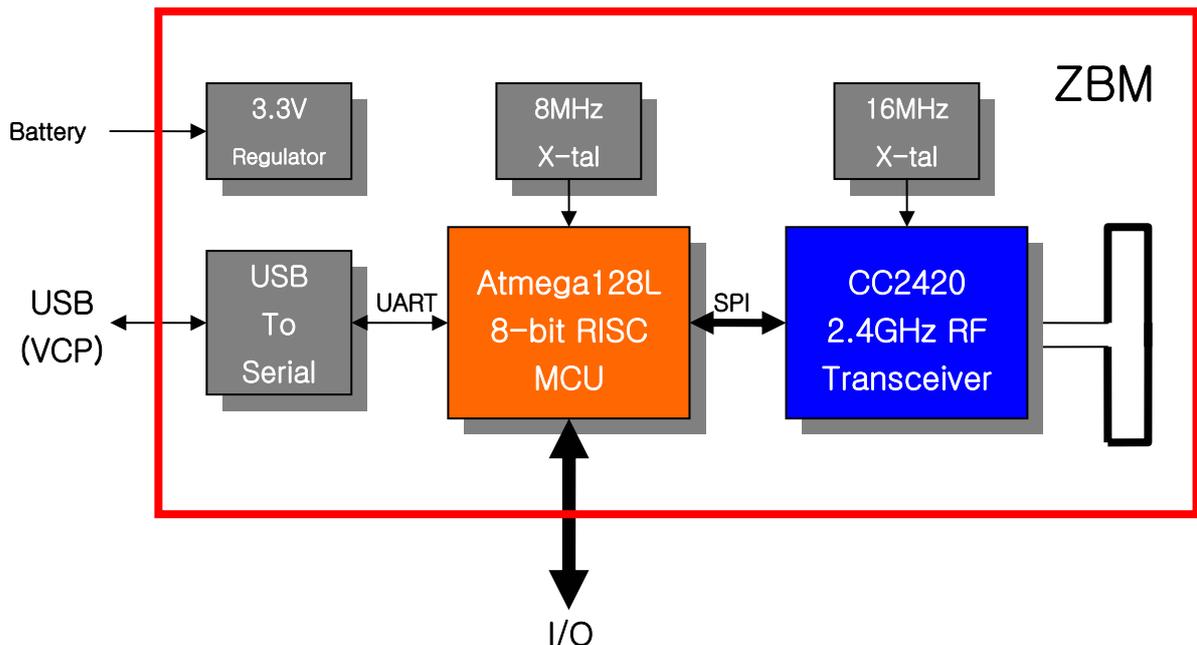
Part 1. ZBM ZigBee 개발 키트 소개

1) 제품 개요

ZBM은 CC2420과 Atmega128L로 구성된 ZigBee 학습 및 개발용 H/W 플랫폼이다.

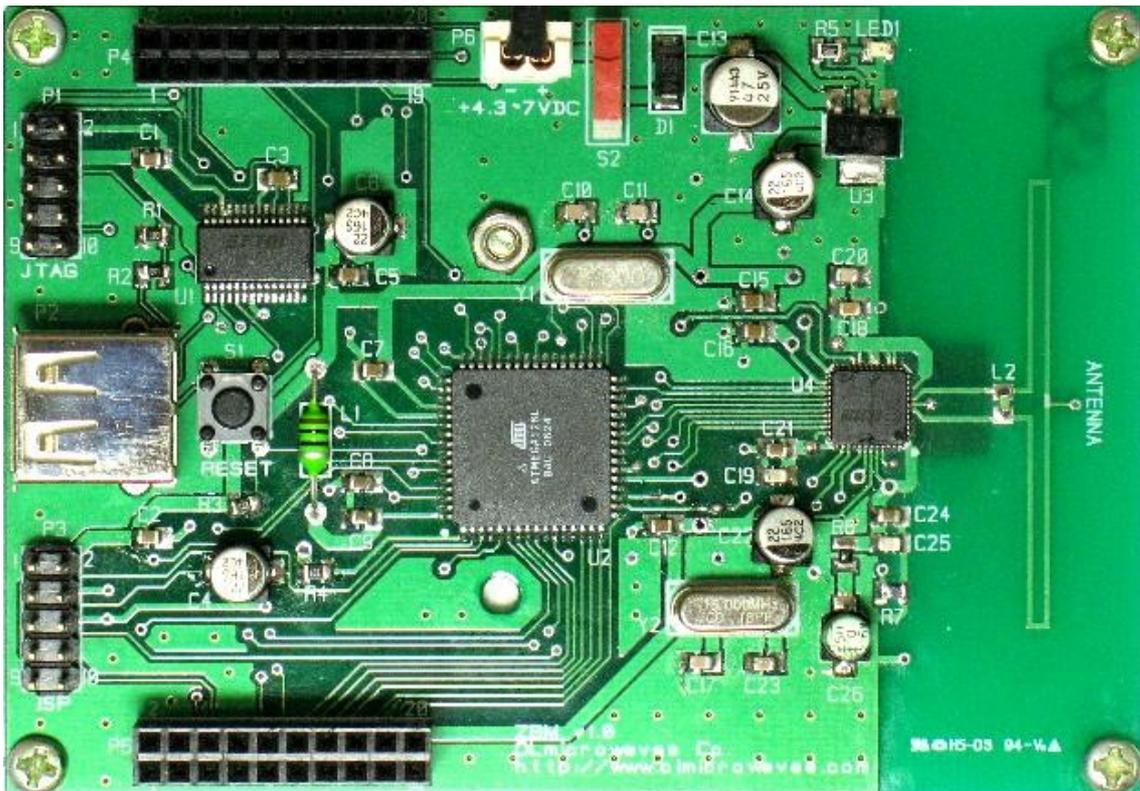
CC2420은 TI사에서 공급하는 RF 트랜시버 칩으로서 IEEE 802.15.4 규격을 만족하는 무선 송수신 기능을 제공한다. Atmega128L은 ATMEL사의 8bit RISC 마이크로프로세서로 국내에서 다양하게 활용되고 있으며 사용자의 입장에서 다른 타입의 마이크로프로세서에 비해 상대적으로 접근이 용이하다고 할 수 있다.

ZigBee 개발용 플랫폼은 현재 다양한 제품들이 나와 있지만 각기 사용되는 RF칩이 상이하고 특히 마이크로프로세서가 임베디드 개발자에게 친숙하지 않은 제품들이 많다. 따라서 극소수의 전문적인 개발자가 아닌 일반 학생이나 초보자의 입장에서는 접근하기가 상당히 어려운 관계로 CC2420과 Atmega128L로 구성된 TI사의 CC2420DBK를 기초로 사용자가 보다 쉽게 기술을 습득하고 개발할 수 있도록 ZBM을 제작하였다.

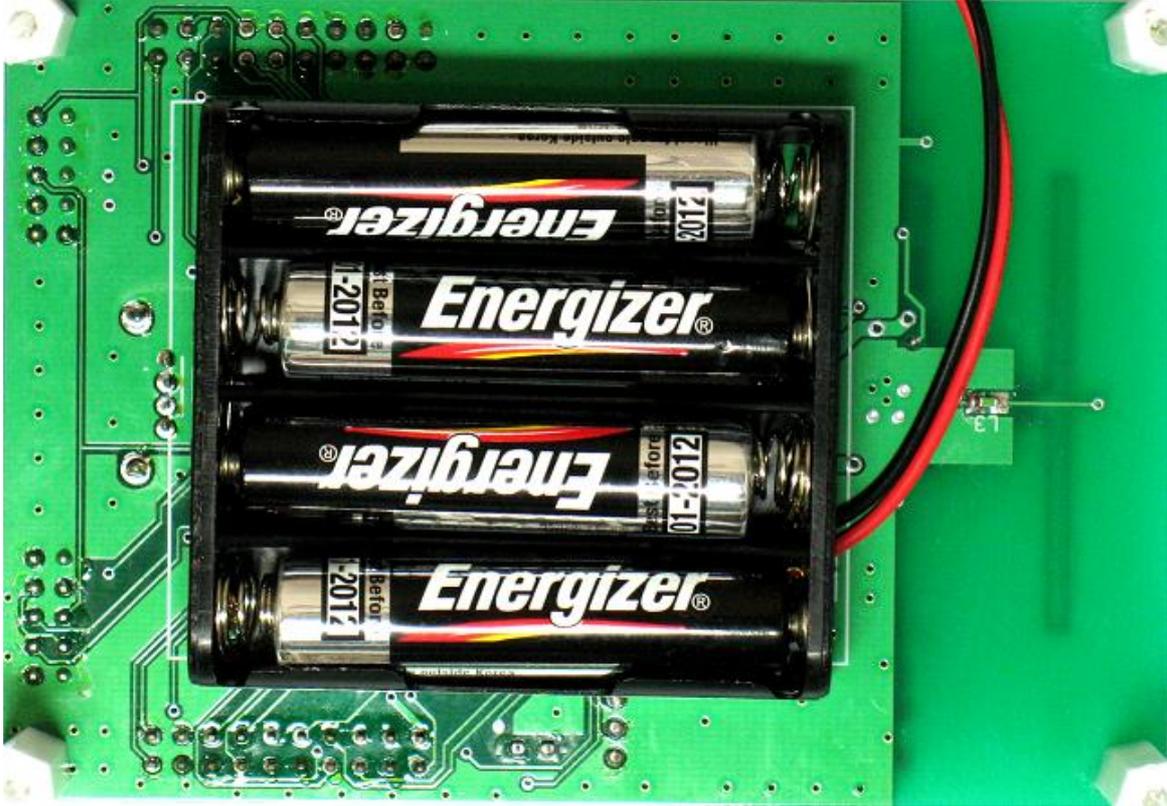


2) 제품 특징

- USB 포트를 통한 가상시리얼 접속 방식으로 9핀 시리얼 포트 없이 시리얼 통신
- 마이콤의 입출력 핀들을 커넥터를 통해 외부 회로와 접속 가능
- JTAG 포트를 통한 프로그래밍
- 고효율의 PCB 내장 안테나
- 전원아답타 없이 일반건전지 사용



[전면]



[후면]

3) 제품 사양

Absolute Maximum Ratings

Parameter	Min.	Max.	Units
Supply voltage	-0.3	7.5	V
Input RF level		10	dBm
Storage temperature range	-50	150	°C

Operating Conditions

Parameter	Min.	Max.	Units
Supply voltage	4.8	7.5	V
Operating ambient temperature range	-30	85	°C

Electrical Specifications

Parameter	Min.	Typ.	Max.	Units
Overall				
RF Frequency Range	2400		2483.5	MHz
Transmit Section				
Transmit bit rate	250		250	kbps
Transmit chip rate	2000		2000	kChips/s
Nominal output power	-3	0		dBm
Programmable output power range		24		dB
Receive Section				
Receiver Sensitivity	-90	-94		dBm
Saturation (maximum input level)	0	10		dBm
Frequency error tolerance	-300		300	kHz
Symbol rate error tolerance			120	ppm
RSSI / Carrier Sense				
Carrier sense level		-77		dBm
RSSI dynamic range		100		dB
RSSI accuracy		± 6		dB
RSSI linearity		± 3		dB
RSSI average time		128		µs

CC2420 Frequency Section				
Crystal oscillator frequency		16		MHz
Crystal frequency accuracy requirement	- 40		40	ppm
Atmega128L Frequency Section				
Crystal oscillator frequency		8		MHz
Crystal frequency accuracy requirement	- 100		100	ppm
Power Supply				
Current Consumption, transmit mode: P = 0 dBm		46		mA
Current Consumption, receive mode		49		mA

Physical Properties

Parameter	Min.
Size	100 mm x 70 mm X 24mm (with battery holder)
Weight	46 / 92 g (without/with battery)

External Pin Assignment

Pin NO.	Pin Name
P1 (JTAG)	
1	TCK
2	GND
3	TDO
4	3.3V
5	TMS
6	/RESET
7	3.3V
8	not connected
9	TDI
10	GND
P2 (USB-UART)	
1	+5 (from PC)
2	-DATA
3	+DATA
4	GND

Pin NO.	Pin Name
P3 (ISP)	
1	PE0
2	3.3V
3	not connected
4	GND
5	/RESET
6	GND
7	SCLK
8	GND
9	PE1
10	GND

※ 주의 : 외부 확장 핀은 3.3V 입·출력 전압 규격임.

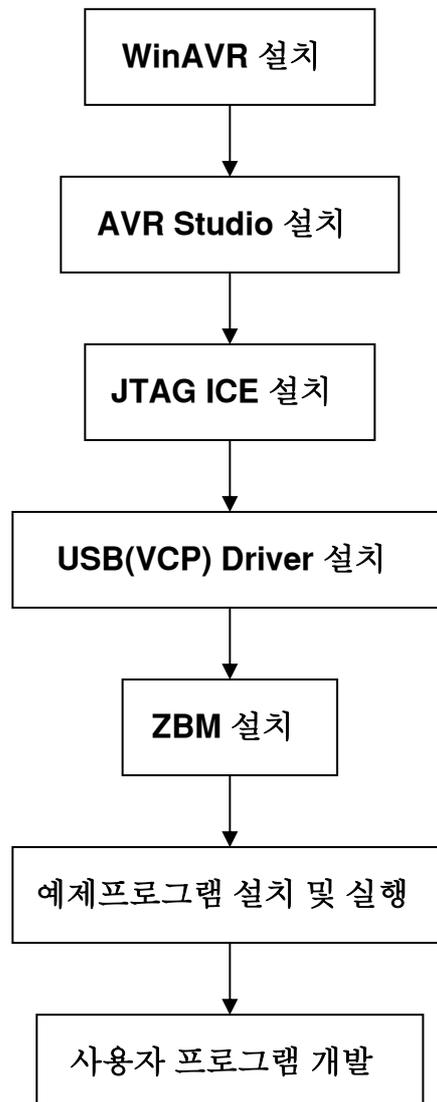
Pin NO.	Pin Name
P4 (GPIO)	
1	PA1
2	PA0
3	PA3
4	PA2
5	PA5
6	PA4
7	PA7
8	PA6
9	PC1
10	PC0
11	PC3
12	PC2
13	PC5
14	PC4
15	PC7
16	PC6
17	PG1
18	PG0
19	GND
20	PG2

Pin NO.	Pin Name
P5 (GPIO)	
1	AREF
2	PF0
3	PF1
4	PF2
5	PF3
6	PE0
7	PE1
8	PE2
9	PE3
10	PE4
11	PE5
12	PE6
13	PE7
14	PB4
15	PB5
16	PB6
17	PB7
18	not connected
19	GND
20	3.3V

※ 주의 : 외부 확장 핀은 3.3V 입·출력 전압 규격임.

Part 2. 프로그램 개발 환경

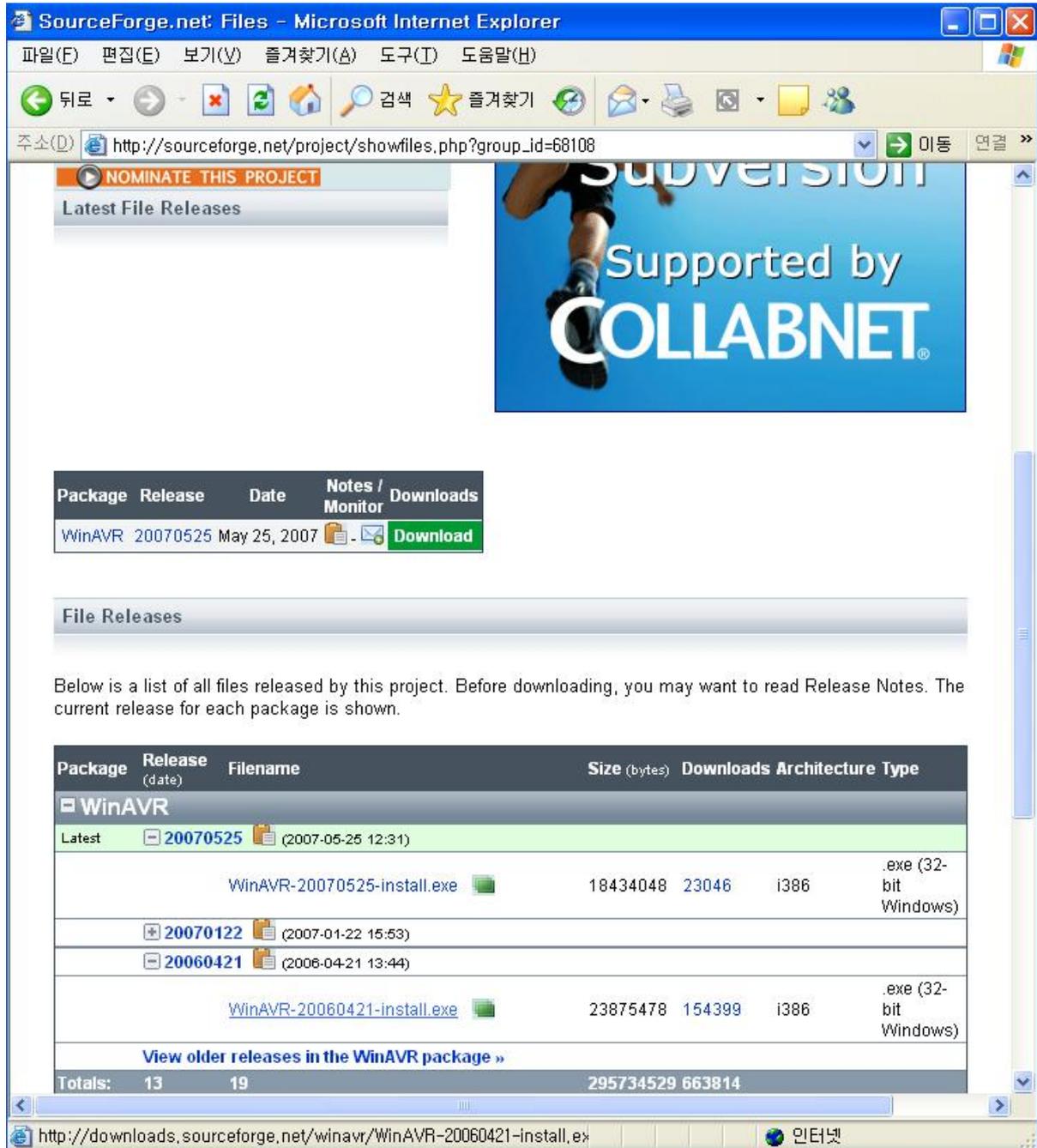
ZBM을 이용하여 프로그램을 개발하는데 필요한 개발 툴과 개발 절차는 다음과 같다.



1) WinAVR 설치

다음의 URL을 방문하여 설치파일을 다운로드하여 설치한다.

- URL : http://sourceforge.net/project/showfiles.php?group_id=68108
- 설치파일 : WinAVR-20060421-install.exe



SourceForge.net: Files - Microsoft Internet Explorer

주소(D) http://sourceforge.net/project/showfiles.php?group_id=68108

Package Release Date Notes / Monitor Downloads

Package	Release	Date	Notes / Monitor	Downloads
WinAVR	20070525	May 25, 2007	 	Download

File Releases

Below is a list of all files released by this project. Before downloading, you may want to read Release Notes. The current release for each package is shown.

Package	Release (date)	Filename	Size (bytes)	Downloads	Architecture	Type
WinAVR						
Latest  20070525  (2007-05-25 12:31)						
		WinAVR-20070525-install.exe 	18434048	23046	i386	.exe (32-bit Windows)
	 20070122  (2007-01-22 15:53)					
	 20060421  (2006-04-21 13:44)					
		WinAVR-20060421-install.exe 	23875478	154399	i386	.exe (32-bit Windows)
View older releases in the WinAVR package »						
Totals:	13	19	295734529	663814		

<http://downloads.sourceforge.net/winavr/WinAVR-20060421-install.exe>

2) AVR Studio 설치

다음의 URL을 방문하여 설치파일을 검색, 다운로드하여 설치한다.

- URL : <http://www.atmel.com>
- 설치파일1 : AVR Studio 4.12 (build 460) (*간단한 등록 과정 필요)
- 설치파일2 : AVR Studio 4.12 Service Pack 4 (build 498)



3) JTAG ICE 설치

사용자 별로 JTAG ICE를 구비하여 설치한다.

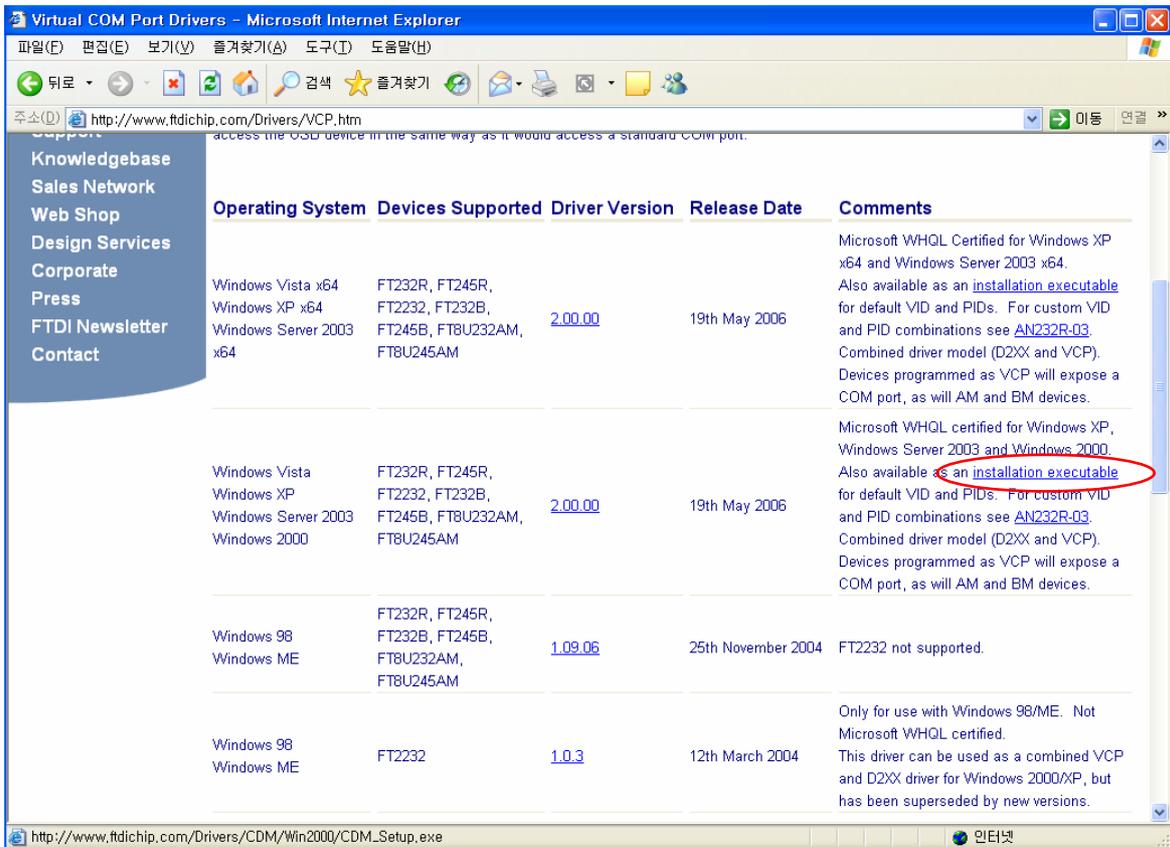
ZBM 에 사용 가능한 JTAG ICE 의 요건은 다음과 같다.

- Atmega128 지원
- ATMEL 10핀 표준 포트를 제공
- +3.3V 타겟보드와 접속

4) USB(VCP) Driver 설치

다음의 URL 을 방문하여 윈도우 버전에 해당하는 설치파일을 다운로드하여 설치한다.

- URL : <http://www.ftdichip.com/Drivers/VCP.htm>
- 설치파일 : CDM_Setup.exe

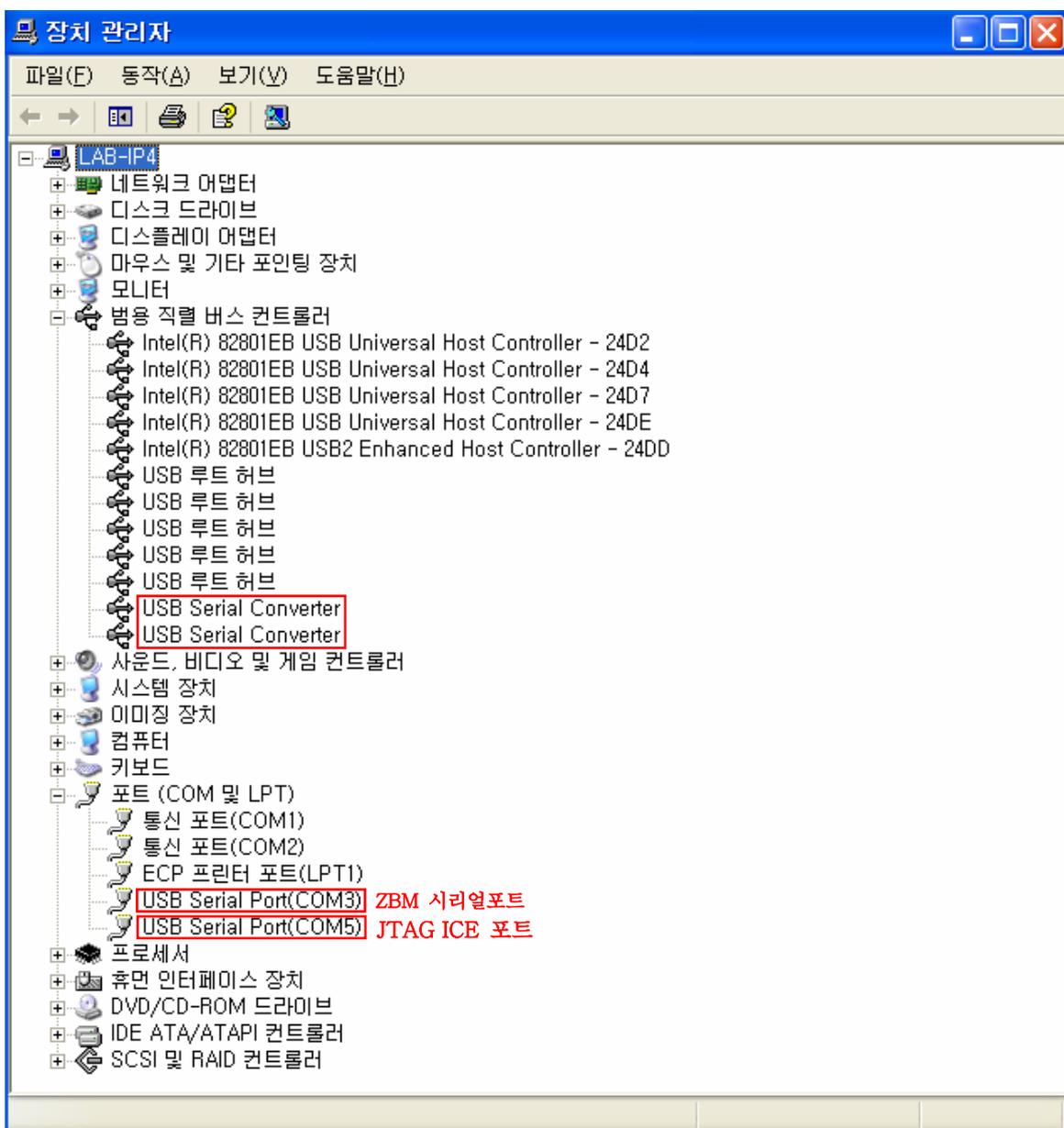


Operating System	Devices Supported	Driver Version	Release Date	Comments
Windows Vista x64 Windows XP x64 Windows Server 2003 x64	FT232R, FT245R, FT2232, FT232B, FT245B, FT8U232AM, FT8U245AM	2.00.00	19th May 2006	Microsoft WHQL Certified for Windows XP x64 and Windows Server 2003 x64. Also available as an installation executable for default VID and PIDs. For custom VID and PID combinations see AN232R-03 . Combined driver model (D2XX and VCP). Devices programmed as VCP will expose a COM port, as will AM and BM devices.
Windows Vista Windows XP Windows Server 2003 Windows 2000	FT232R, FT245R, FT2232, FT232B, FT245B, FT8U232AM, FT8U245AM	2.00.00	19th May 2006	Microsoft WHQL certified for Windows XP, Windows Server 2003 and Windows 2000. Also available as an installation executable for default VID and PIDs. For custom VID and PID combinations see AN232R-03 . Combined driver model (D2XX and VCP). Devices programmed as VCP will expose a COM port, as will AM and BM devices.
Windows 98 Windows ME	FT232R, FT245R, FT232B, FT245B, FT8U232AM, FT8U245AM	1.09.06	25th November 2004	FT2232 not supported.
Windows 98 Windows ME	FT2232	1.0.3	12th March 2004	Only for use with Windows 98/ME. Not Microsoft WHQL certified. This driver can be used as a combined VCP and D2XX driver for Windows 2000/XP, but has been superseded by new versions.

5) ZBM 설치

- ZBM의 JTAG 포트에 JTAG ICE를 연결한다.
 - ZBM의 P2(USB포트)와 PC의 USB 컨넥터를 USB 케이블을 사용하여 연결한다.
 - ZBM의 전원 스위치를 ON한다.
- ⇒ PC에서 자동으로 가상 시리얼 포트를 할당한다.

시작→제어판→시스템→하드웨어→장치관리자에서 포트 설정을 확인한다.



- 하이퍼터미널을 실행한다.

(시작→모든프로그램→보조프로그램→통신/)

- 파일→새연결을 수행한다.

이름 : ex) ZBM monitor

연결에 사용할 모뎀 : COM3 (장치관리자에서 확인된 포트)

비트/초 : 115200

데이터 비트 : 8

패리티 : 없음

정지 비트 : 1

흐름 제어 : 없음

- ZBM 의 리셋 버튼(S1)을 ON/OFF 하면 하이퍼터미널 화면에 ZBM 초기화면이 출력된다.
- '4'를 타이핑하여 ZBM 의 Address 세팅 값을 확인한다.

```
*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number : 4 Read Address

> Address : 00

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number :
```

Part 3. 예제 프로그램

1) 예제 프로그램 설치

다음의 URL을 방문하여 압축파일을 다운로드하여 임의의 작업 디렉토리에 압축 해제한다.

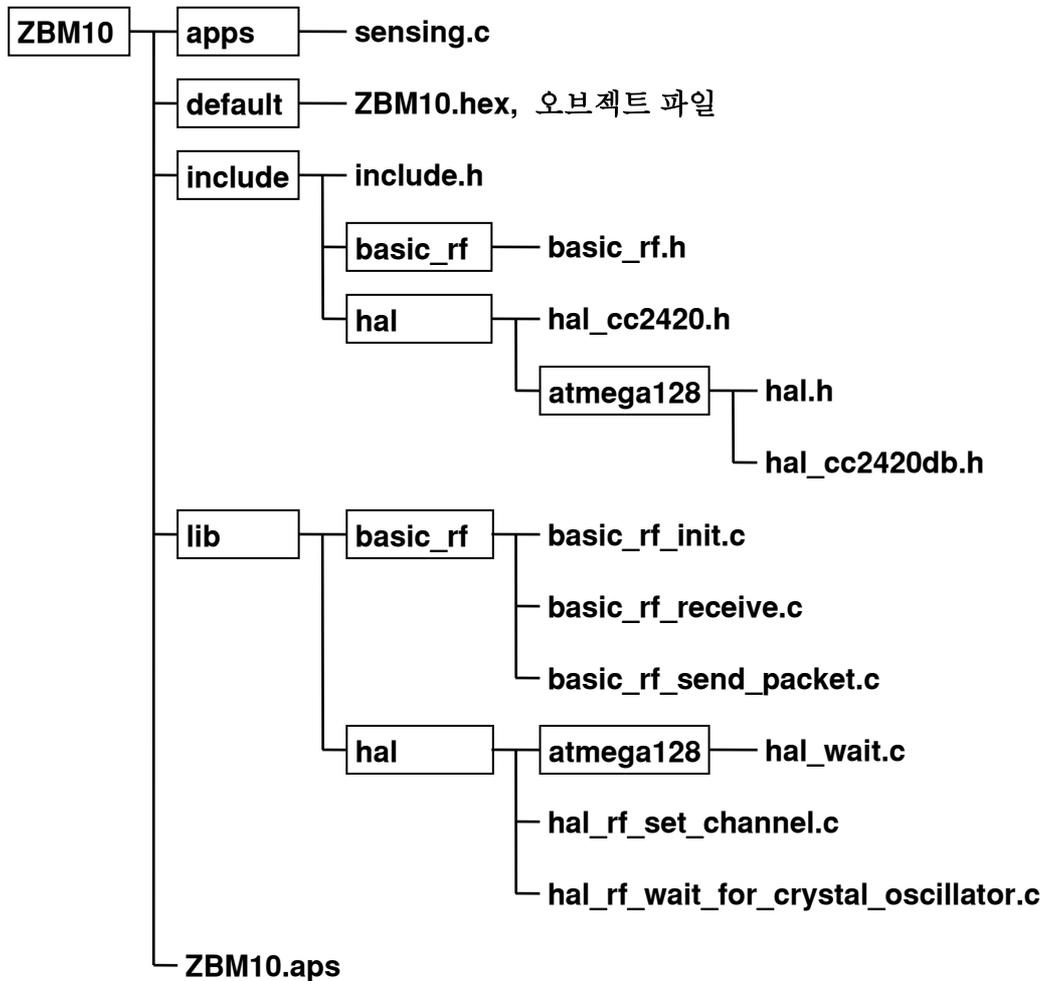
- URL : <http://olmicrowaves.com> => Downloads => ZigBee
- 압축파일 : ZBM10.zip

2) 예제 프로그램 실행

- 'Part 2. 프로그램 개발 환경' 의 '5) ZBM 설치' 과정을 수행하여 ZBM과 JTAG ICE 및 하이퍼터미널을 활성화 한다.
- AVR Studio를 실행한다.
- Project →Open Project 과정으로 압축 해제한 ZBM10 디렉토리 아래에 있는 ZBM10.apc를 로드한다.
- 프로그램을 수정한다. (필요한 경우)
- File →Save All을 수행하여 작업한 파일을 저장한다.
- Build →Build All을 수행하여 실행 파일을 생성한다.
- Tools →Program AVR →Connect를 수행한다.
 - Platform : JTAG ICE
 - Port : COM5 (현재 JTAG ICE가 접속되어 있는 포트)
- 프로그래머 다이얼로그에서 Flash 카테고리의 'Input HEX File'의 경로를 지정한다.
 - [ZBM10/default/ZBM10.hex](#)
- 프로그래머 다이얼로그에서 Flash 카테고리의 'Program' 버튼을 클릭하면 ZBM 내의 MCU로 펌웨어 프로그램이 다운로드 된다.

3) 예제 프로그램 설명

예제 프로젝트 디렉토리 ZBM10은 다음과 같은 파일과 하위 디렉토리로 구성되어 있다.



- **sensing.c** : 응용프로그램으로서, **main()** 함수를 포함하는 기본 프로그램
- **ZBM10.hex** : MCU 다운로드 파일
- **include.h** : 기본적 데이터 타입이 정의되어 있고, 여러가지 헤더파일들을 **include** 시킨다.
- **basic_rf.h** : **basic rf library(lib/basic_rf/*.c)** 파일에 관련된 정의
- **hal_cc2420.h** : CC2420에 관련된 정의
- **hal.h** : CC2420에 접근하기 위한 **SPI** 제어 매크로 정의
- **hal_cc2420db.h** : CC2420과 MCU 주변 장치와의 접속을 정의

- **basic_rf_init.c** : CC2420의 무선 송신 및 수신 동작을 위한 초기화
- **basic_rf_receive.c** : 무선 패킷을 수신하기 위한 인터럽트 서비스 루틴
- **basic_rf_send_packet.c** : 무선 패킷을 송신하기 위한 루틴
- **hal_wait.c** : Delay 함수
- **hal_rf_set_channel.c** : 주어진 채널번호로부터 IEEE802.15.4 무선 주파수 설정
- **hal_rf_wait_for_crystal_oscillator.c** : CC2420 내부 오실레이터의 안정화까지 대기
- **ZBM10.aps** : AVR Studio 프로젝트 파일

※ ZBM 예제 프로젝트(ZBM10)은 TI사의 CC2420DBK 예제 프로그램을 기초로 작성되었다.

TI사의 원본을 다운로드하려면 다음의 URL을 방문하여 압축파일을 다운로드한다.

- URL : <http://focus.ti.com/docs/toolsw/folders/print/cc2420dbk.html>
- 압축파일 : CC2420DBK Libraries (swrc024.zip, 22 KB)

CC2420DBK Examples Release (swrc023.zip, 12 KB)

※ TI 원본에서 ZBM10이 수정된 부분

(A) ZBM10 / include / include.h 수정

```
...
#include <string.h>

#define CC2420DB

#ifdef STK501
    #include "include/hal/atmega128/hal_stk501.h"
#endif

#ifdef CC2420DB
    #include "include/hal/atmega128/hal_cc2420db.h"
#endif
```

```
// HAL include files
#include "include/hal/atmega128/hal.h"
#include "include/hal/hal_cc2420.h"

// BASIC RF include files
#include "include/basic_rf/basic_rf.h"
```

(B) 아래의 각 소스파일에 **include** 경로 수정 : **#include "include\include.h"**

```
zbm / lib / basic_rf / basic_rf_init.c
zbm / lib / basic_rf / basic_rf_receive.c
zbm / lib / basic_rf / basic_rf_send_packet.c
zbm / lib / hal / hal_rf_set_channel.c
zbm / lib / hal / hal_rf_wait_for_crystal_oscillator.c
zbm / lib / hal / atmega128 / hal_wait.c
zbm / apps / basic_rf / rf_blink_led / rf_blink_led.c
```

(C) **hal_cc2420db.h** 수정 사항

< AVR과 CC2420 간의 핀 설정 비교 >

Signal Name	CC2420DBK	ZBM
CSN	PB0	"
SCLK	PB1	"
SI	PB2	"
SO	PB3	"
VREG_EN	PB5	PG3
RESETN	PB6	PG4
CTS	PD7	"
CCA	PD6	"
RTS	PD5	"
SFD	PD4	"
TXD1	PD3	"
RXD1	PD2	"
FIFO	PD1	"
FIFOP	PD0	"

```
// Port B
#define CSN      0 // PB.0 - Output: SPI Chip Select (CS_N)
#define SCK      1 // PB.1 - Output: SPI Serial Clock (SCLK)
#define MOSI     2 // PB.2 - Output: SPI Master out - slave in (MOSI)
#define MISO     3 // PB.3 - Input: SPI Master in - slave out (MISO)
#define OLED     4 // PB.4 - Output: Orange LED
// #define VREG_EN 5 // PB.5 - Output: VREG_EN to CC2420
// #define RESET_N 6 // PB.6 - Output: RESET_N to CC2420
#define GLED     7 // PB.7 - Output: Green LED

// Port G
//-----
#define VREG_EN 3 // PG.3 - Output: VREG_EN to CC2420
#define RESET_N 4 // PG.4 - Output: RESET_N to CC2420

// Port initialization
// Disables pull-up on all inputs!!!
#define PORT_INIT() \
do { \
    SFIOR |= BM(PUD); \
    DDRB = BM(MOSI) | BM(SCK) | BM(CSN) | BM(OLED) | BM(GLED); \
    PORTB = BM(MOSI) | BM(SCK) | BM(CSN); \
    DDRD = BM(UART1_RTS); \
    PORTD = BM(UART1_RTS) | BM(UART1_CTS); \
    DDRE = BM(RLED) | BM(YLED); \
    PORTE = 0x00; \
    DDRG = BM(RESET_N) | BM(VREG_EN); \
    PORTG = BM(RESET_N); \
} while (0)

// CC2420 pin access

// Pin status
#define FIFO_IS_1    (!(PIND & BM(FIFO)))
#define CCA_IS_1    (!(PIND & BM(CCA)))
#define RESET_IS_1  (!(PING & BM(RESET_N)))
```

```
#define VREG_IS_1    (!(PING & BM(VREG_EN)))
#define FIFOP_IS_1  (!(PIND & BM(FIFOP)))
#define SFD_IS_1    (!(PIND & BM(SFD)))

// The CC2420 reset pin
#define SET_RESET_ACTIVE()  PORTG &= ~BM(RESET_N)
#define SET_RESET_INACTIVE() PORTG |= BM(RESET_N)

// CC2420 voltage regulator enable pin
#define SET_VREG_ACTIVE()   PORTG |= BM(VREG_EN)
#define SET_VREG_INACTIVE() PORTG &= ~BM(VREG_EN)
```

(D) hal.h UART 관련 오류 수정

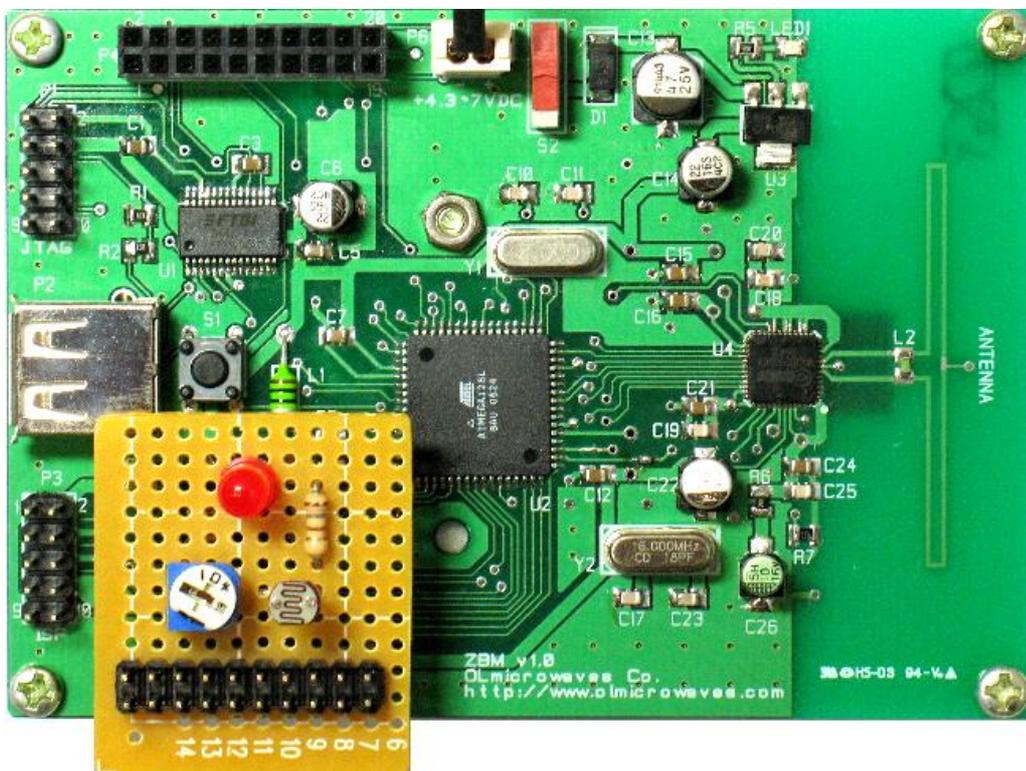
```
#define UART1_WAIT_AND_RECEIVE(x) do{ UDR1 = 0; UART1_WAIT(); UART1_RECEIVE(x); } while (0)
```

(E) hal.h UART 관련 매크로 추가 및 수정

```
#define UART1_WAIT()    do { while (!(UCSR1A & BM(UDRE1))); CLEAR_UART1_TX_INT(); } while (0)
#define UART1_RXWAIT() do { while (!(UCSR1A & BM(RXC1))); CLEAR_UART1_RX_INT(); } while (0)
#define UART1_SEND(x)  do { UDR1 = (x); } while (0)
#define UART1_WAIT_AND_SEND(x) do { UART1_WAIT(); UART1_SEND(x); } while (0)
#define UART1_RECEIVE(x) do { (x) = UDR1; } while (0)
#define UART1_WAIT_AND_RECEIVE(x) do {UDR1=0; UART1_RXWAIT(); UART1_RECEIVE(x); } while(0)
#define UART1_LINEFEED() do { UART1_WAIT_AND_SEND(13); UART1_WAIT_AND_SEND(10); }while(0)
```

4) 사용자 프로그램 개발

- ZBM은 ZigBee 프로토콜을 개발하거나 학습하는 용도로 사용될 수 있으며 뿐만 아니라 외부 확장용 커넥터(P4, P5)를 통해 각종 센서나 제어기와 접속하여 ZigBee 응용시스템을 개발하는 용도로 활용할 수 있다.
- 사용자별로 응용프로그램 개발하기 위해서 외부 확장 커넥터별 신호 구성을 파악한 후 별도의 도터보드를 제작하여 장착한다. (페이지 8 ~ 9 참조)
 - P4 : 범용I/O 및 메모리 제어 신호
 - 외부 메모리를 확장하거나 일반 범용 입출력으로 외부 장치와 접속
 - P5 : ADC, 아나로그 입력, 인터럽트, 범용I/O
 - 센서 장치나 제어 장치와 접속하거나 일반 범용 입출력으로 외부 장치와 접속 (예: 조명제어 응용 실습)



- **MAC**프로그램이나 **ZigBee** 스택을 포팅하고자 하는 경우에는 각 프로그램의 **HAL** 부분을 **Atmega128**에 적합하게 구성한 후, **AVR Studio(WinAVR/avr-gcc)**나 **IAR** 컴파일러 등을 이용하여 컴파일하고, 생성된 **hex** 파일을 **AVR Studio**나 **PonyProg**를 사용하여 **ZBM**의 **Flash ROM**에 다운로드 한다.

TI사에서는 **Z-stack**이라는 **ZigBee** 표준 스택을 제공하며 다음의 **URL**을 방문하여 무료로 다운로드하여 사용할 수 있다. (간단한 사용자 등록 필요)

- **URL** : <http://focus.ti.com/analog/docs/gencontent.tsp?familyId=367&genContentId=31010>

The screenshot shows a Microsoft Internet Explorer browser window displaying the Texas Instruments website. The address bar shows the URL: <http://focus.ti.com/analog/docs/gencontent.tsp?familyId=367&genContentId=31010>. The page title is "Z-Stack™ - ZigBee® Protocol Stack". The navigation menu includes "products", "applications", "design support", and "buy". The breadcrumb trail is "TI Home > RF/IF Components > Z-Stack™ - ZigBee® Protocol Stack". The main content area is titled "RF / IF Components" and includes links for "Cross Reference", "Application Notes", "Training", "Tools & Software", "Block Diagrams", and "ZigBee". Below this, the product name "Z-Stack™ - ZigBee® Protocol Stack" is displayed. A table provides the following details:

	Z-Stack
Name	Z-Stack ZigBee Protocol Stack
Status	ACTIVE
Price (US\$)	Free

Below the table, the "Product Information" section is visible, with a "Description" sub-section. The text reads: "Z-Stack release for the CC2420 + AVR platform. SimpleAPI - designing with ZigBee now made easier! The version 1.4.2 of TI's Z-Stack ZigBee software is now available for download free of charge." Underneath, it says "Download the CC2420 + AVR Z-Stack today:" and lists two download options. The first option, "Download & Register Z-Stack CC2420+AVR Version 1.4.2", is highlighted with a red box. The second option is "Download & Register Z-Stack CC2420+AVR and Application Examples 1.4.2".

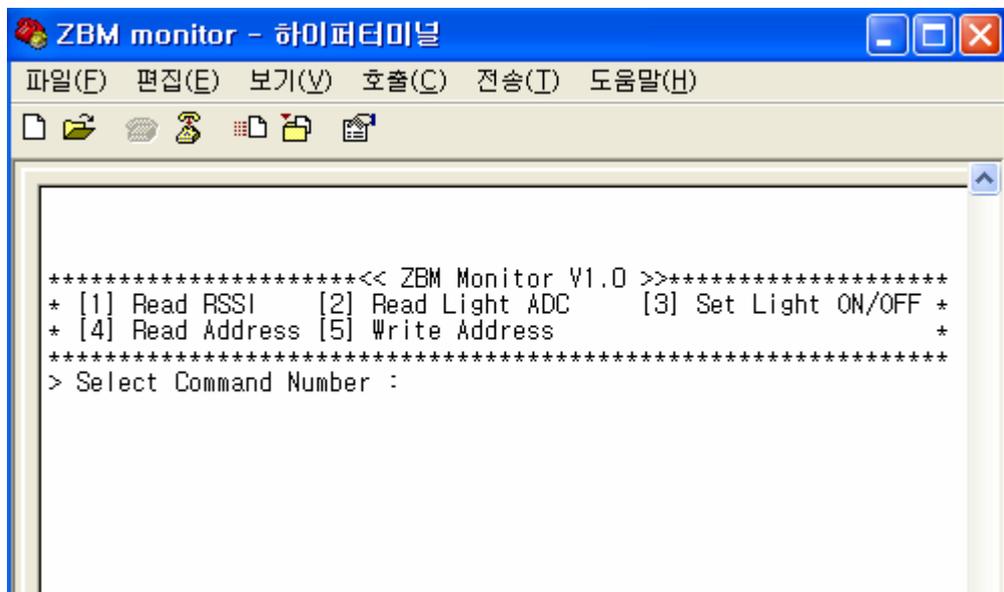
Part 4. ZBM 실습

ZBM에 다운로드 되어 있는 프로그램은 예제 프로젝트인 **ZBM10**이다.

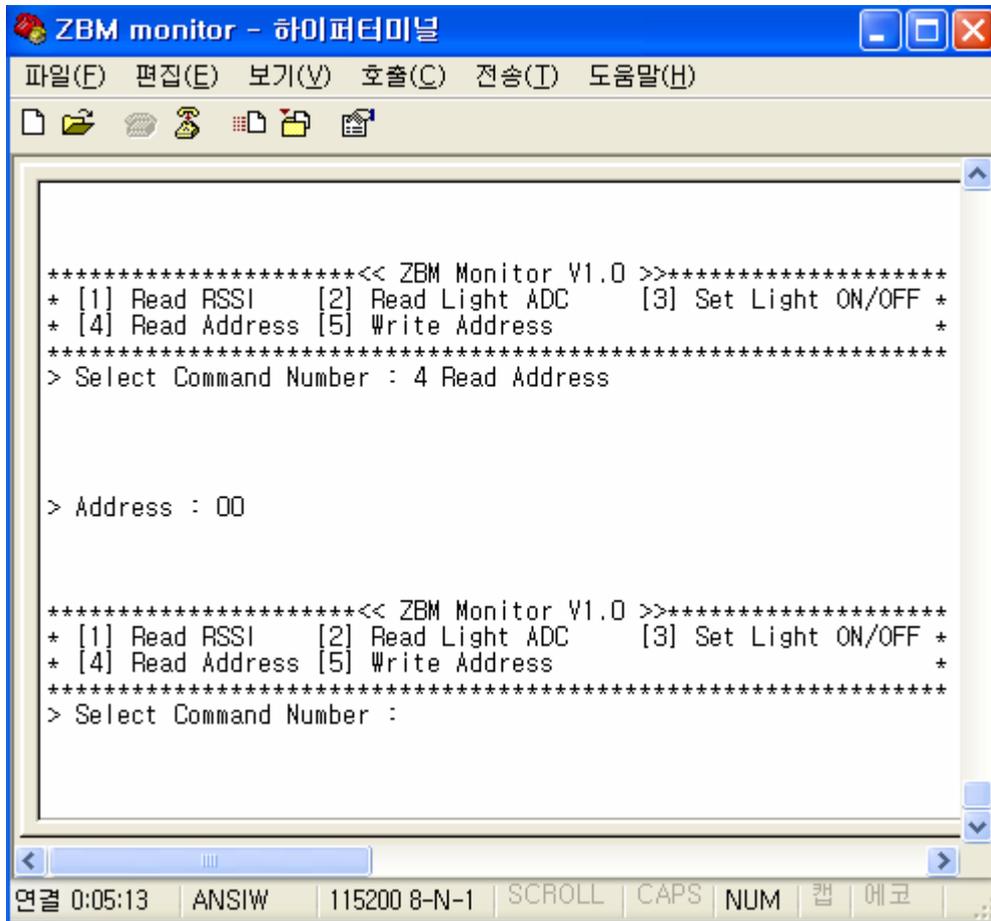
ZBM10에는 두 개의 **ZBM**을 하나는 마스터로 다른 하나는 센서로 간주하고 각각의 **Address**를 설정하여(디폴트로: **00, 99**) 터미널(PC)로부터 마스터를 경유하여 무선으로 센서를 제어하는 동작이 구현되어 있다.

이 예제를 통해 **MCU**를 사용하여 **ZigBee**용 **RF** 트랜시버 칩인 **CC2420**을 제어하는 방법을 습득할 수 있으며 **ZBM**을 홈네트워크나 기타 다양한 분야에 활용할 수 있도록 기본적인 아이디어를 제공한다.

- 마스터로 사용할 **ZBM**을 'Part 2. 프로그램 개발 환경' 의 '5) **ZBM** 설치' 과정 수행하여 셋업한다.
- 리셋 버튼을 누르면 하이퍼터미널 상에 초기 화면이 출력된다.



- '4'를 입력하여 현재 ZBM에 설정된 Address를 확인한다.

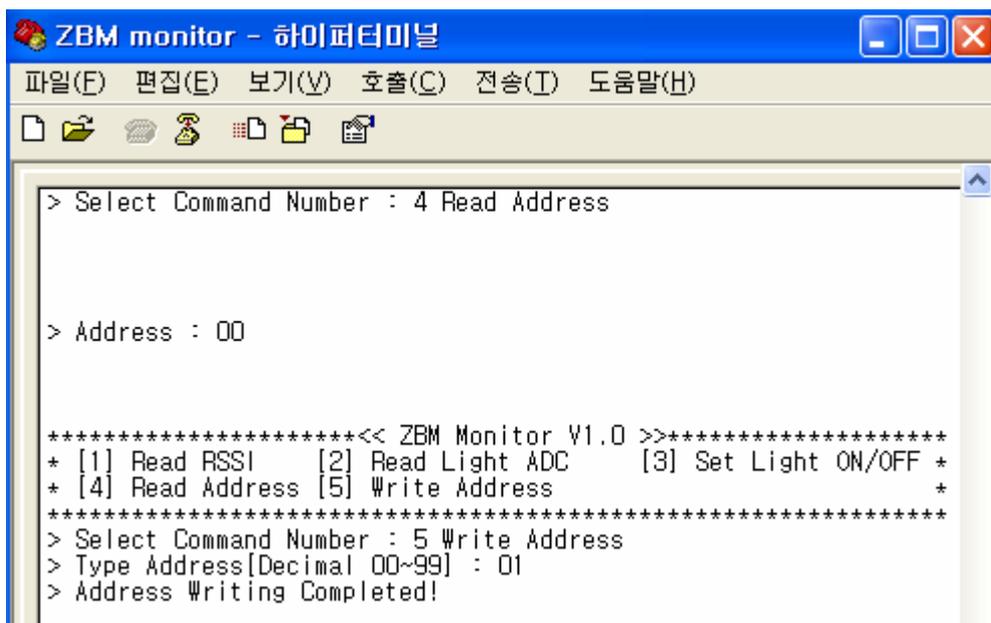


```
ZBM monitor - 하이퍼터미널
파일(F) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)
*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI [2] Read Light ADC [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address *
*****
> Select Command Number : 4 Read Address

> Address : 00

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI [2] Read Light ADC [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address *
*****
> Select Command Number :
```

- 필요한 경우 '5'를 입력하여 Address를 다른 값으로 변경한다.(00 ~ 99 사이의 값)



```
ZBM monitor - 하이퍼터미널
파일(F) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)
*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI [2] Read Light ADC [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address *
*****
> Select Command Number : 4 Read Address

> Address : 00

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI [2] Read Light ADC [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address *
*****
> Select Command Number : 5 Write Address
> Type Address[Decimal 00~99] : 01
> Address Writing Completed!
```

- 동일한 방식으로 센서로 사용할 ZBM의 Address도 확인하고 필요하면 Address를 변경한다.
- 터미널(PC)에 다시 마스터 ZBM을 접속하고, 센서 ZBM을 3미터 정도의 거리에 설치한다.
- 하이퍼터미널로 '1'과 '99'를 차례로 입력하여 Address가 99인 센서 ZBM의 RSSI 값을 읽는다. (센서 Address는 사용자별로 설정한 값을 입력)

```
> Type Address[Decimal 00~99] : 01
> Address Writing Completed!

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number : 1 Read RSSI
> Type Destination Address[Decimal 00~99] : 99

> RSSI : -045[dBm]

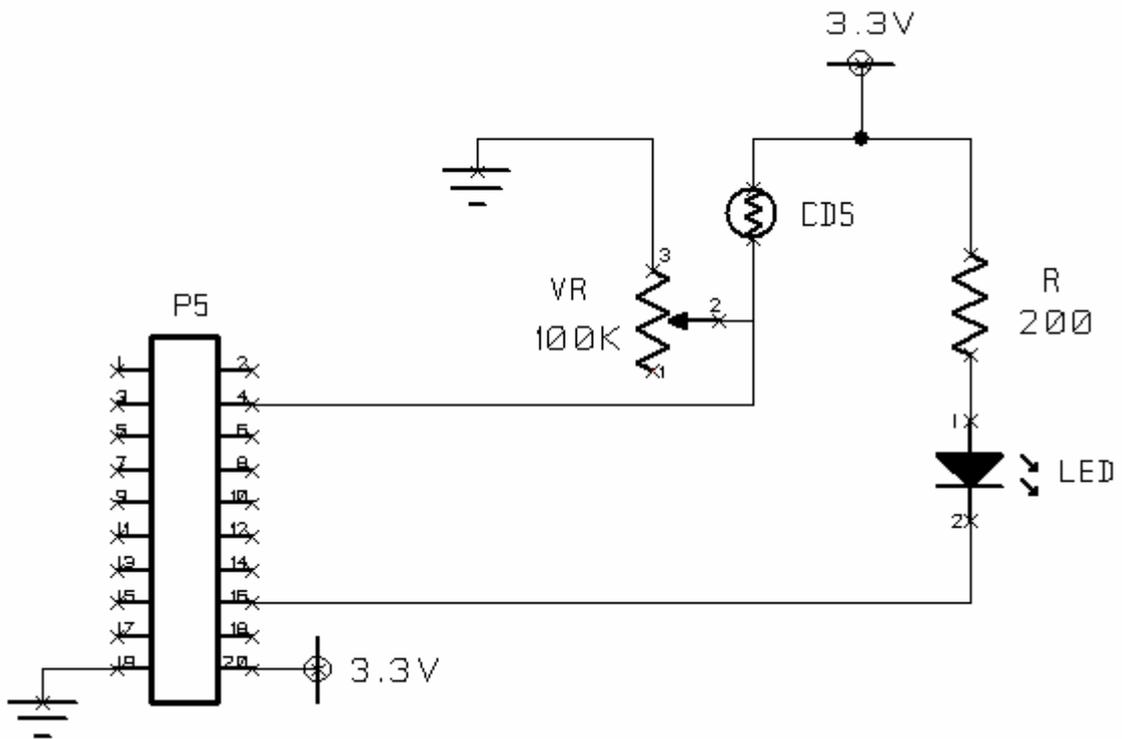
*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number :
```

이 때, 출력되는 RSSI 값은 센서가 마스터로부터 수신하는 신호의 세기를 나타내며, ZBM의 안테나 출력 전력은 약 0dBm(디폴트)으로 설정되어 있다. 따라서 하이퍼터미널 상에 출력되는 RSSI 값은 수신 전력 값인 동시에 해당 거리에서의 경로손실(Path Loss)라고 할 수 있다. 이러한 사실로부터 사용자는 RSSI 관련한 다양한 응용을 할 수 있다. 마스터가 센서로부터 수신하는 신호의 세기도 동일한 값으로 간주해도 된다.

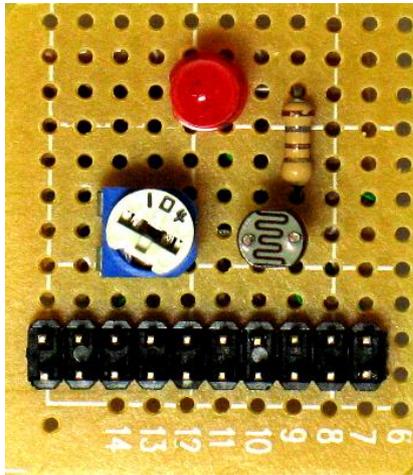
- ZBM monitor의 '2'과 '3' 명령은 조도센서 값을 읽고, LED를 켜거나 끄는 동작을 수행한다. 그러나 조도센서와 LED 회로는 ZBM 모듈에 포함되지 않는다. 따라서 사용자는 다음의 과정을 따라 도터보드를 제작하여 실습을 수행하도록 한다.

◆ 준비물 : 만능기판(HS-01 혹은 기타), 조도센서(A90.09등), LED, 저항(180~200Ω), 가변저항(100KΩ), 핀헤드(2x10), 납땜용구(인두기, 납)

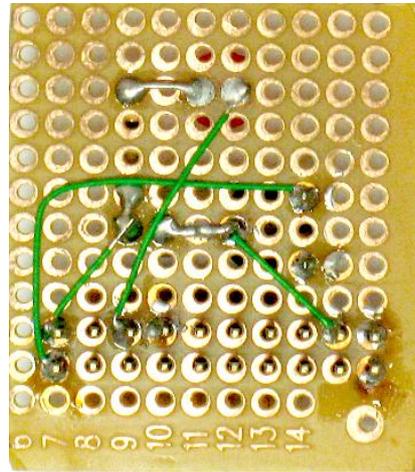
◆ 도터보드 회로도



- ◆ 만능기판을 적당한 크기로 자른다. (3 cm x 3.5 cm 가량)
- ◆ 기판에 회로도를 참조하여 부품을 배치한 후 납땜한다.
 - 핀헤드(P5)는 센서 ZBM의 소켓(P5)에 장착될 수 있도록 한다.

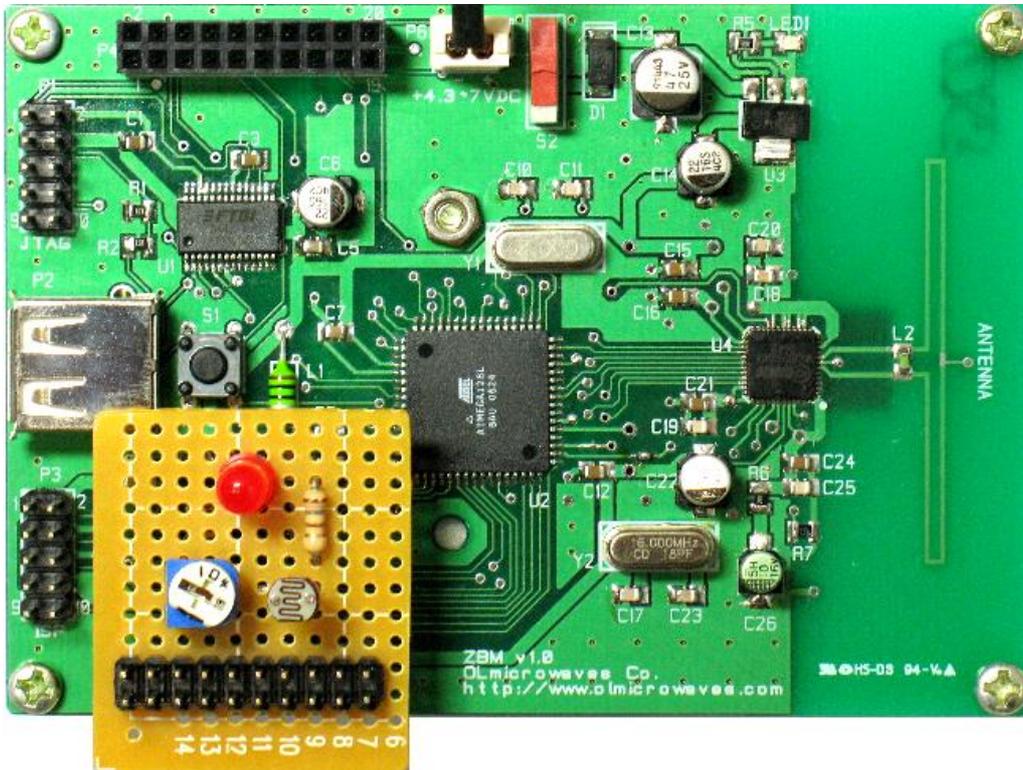


[전면]



[후면]

- 제작한 도터보드를 센서 ZBM 의 소켓(P5)에 장착한다. (방향에 유의)



- 하이퍼터미널로 '2'와 '99'를 차례로 입력하여 Address가 99인 센서 ZBM의 조도값을 읽는다. (센서 Address는 사용자별로 설정한 값을 입력)

```
ZBM monitor - 하이퍼터미널
파일(F) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)
> RSSI : -045[dBm]

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number : 2 Read Light ADC
> Type Destination Address[Decimal 00~99] : 99

> Light ADC : 162

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number : _

연결 2:20:14 ANSIW 115200 8-N-1 SCROLL CAPS NUM 캡 에코
```

조도값의 범위는 0 ~ 255이며 A/D 컨버터에서 읽은 값을 8비트 처리하여 출력한다.
가변저항을 조절하면 특정한 조도에 출력값을 튜닝할 수 있다.

- 하이퍼터미널로 '3', '99', '1'을 차례로 입력하여 Address가 99인 센서 ZBM의 LED를 켜다.

```
> Light ADC : 251

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number : 3 Set Light ON/OFF
> Type Destination Address[Decimal 00~99] : 99
> Select Number[0:off, 1:on] :

> Light Status : ON

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number :
```

센서 ZBM의 LED의 상태를 확인한다.

- 하이퍼터미널로 '3', '99', '0'을 차례로 입력하여 Address가 99인 센서 ZBM의 LED를 끈다.

```
> Light Status : ON

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number : 3 Set Light ON/OFF
> Type Destination Address[Decimal 00~99] : 99
> Select Number[0:off, 1:on] :

> Light Status : OFF

*****<< ZBM Monitor V1.0 >>*****
* [1] Read RSSI    [2] Read Light ADC    [3] Set Light ON/OFF *
* [4] Read Address [5] Write Address      *
*****
> Select Command Number : _
```

센서 ZBM의 LED의 상태를 확인한다.