

IoT FND8 7-Segment Module

<http://www.mangoboard.com/>

<http://cafe.naver.com/embeddedcrazyboys>

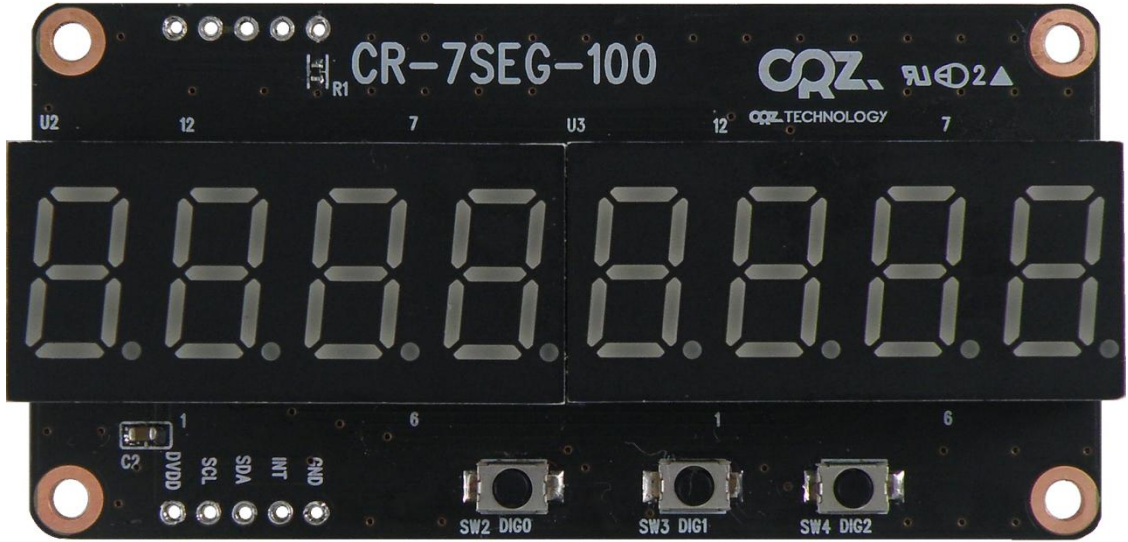
Crazy Embedded Laboratory

Document History

Revision	Date	Change note

1. IoT FND8 7-Segment Module은?	4
1.1. LED Driver	4
1.1.1. as1115란.....	4
1.1.2. AS1115의 응용 다이어그램.....	4
1.2. seven segment	5
1.2.1. WCN4_0036SR_C11 내부 구조.....	5
1.2.2. WCN4_0036SR_C11 외부 구조.....	5
2. I2C Device Address	5
2.1. I2C Device Address Byte	6
2.2. Self Addressing Register	6
2.3. void zF_CR7SEG_SelfAddressModeControl (uint8_t addressMode,struct cr7seg_struct cr7seg_setting)	6
3. Shutdown Mode	6
3.1. zF_CR7SEG_ShutdownModeControl(uint8_t shutdownMode, struct cr7seg_struct cr7seg_setting)	6
4. Basis Setting	6
4.1. Decode Control	7
4.1.1. Decode Mode Examples.....	7
4.1.2. Decode Mode.....	7
4.1.3. No Decode Mode.....	8
4.1.4. zF_CR7SEG_SetDecodeModeControl(uint8_t decodeSetting, struct cr7seg_struct cr7seg_setting).....	10
4.2. Scan_Limit Register	10
4.2.1. zF_CR7SEG_SetScanLimitControl(uint8_t scanLimitSetting, struct cr7seg_struct cr7seg_setting).....	10
4.3. Brightness Control	11
4.3.1. Intensity Register Format.....	11
4.3.2. Intensity Register Address.....	12
4.3.3. zF_CR7SEG_SetBrightnessControl(uint8_t brightnessSetting, cr7seg_setting);.....	12
5. Function	12
5.1. zF_CR7SEG_Code_B_Font_Print(struct cr7seg_struct cr7seg_setting)	13
5.2. zF_CR7SEG_Test_Brightness_Print(struct cr7seg_struct cr7seg_setting)	14
5.3. zF_CR7SEG_Choice_Character (struct cr7seg_struct cr7seg_setting)	14
5.4. zF_CR7SEG_Blink_Print(struct cr7seg_struct cr7seg_setting)	15
5.5. zF_CR7SEG_Scroll_Right(struct cr7seg_struct cr7seg_setting)	15
5.6. zF_CR7SEG_Scroll_Left(struct cr7seg_struct cr7seg_setting)	15
5.7. zF_CR7SEG_SetBasisSetting(cr7seg_setting)	16
5.8. zF_CR7SEG_SetClean(struct cr7seg_struct cr7seg_setting)	16

IoT FND8 7-Segment Module은?



1.1. LED Driver

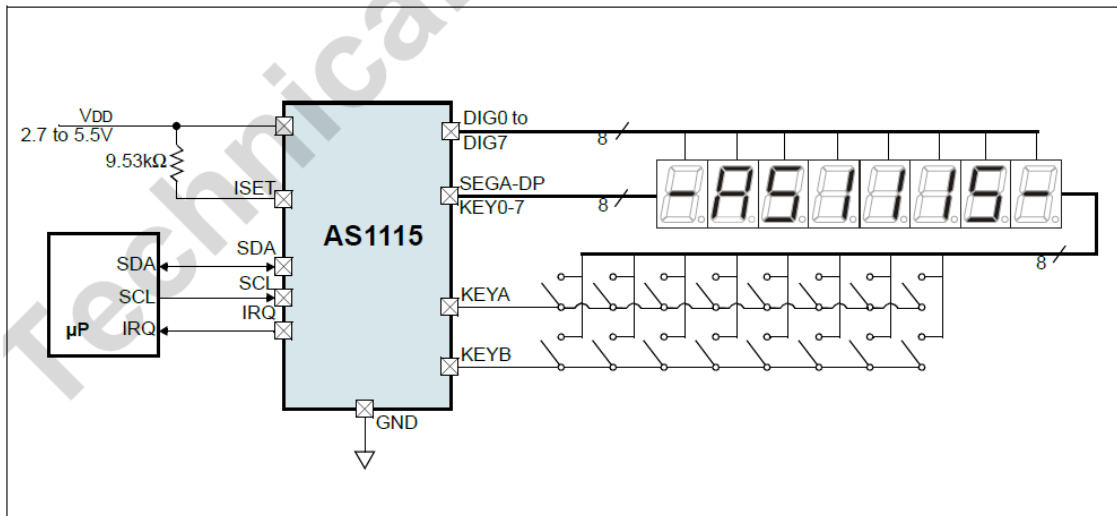
IoT FND8 7-Segment Module은 AS1115드라이버를 사용하여 SEVEN SEGMENT를 제어합니다. AS1115의 응용 다이어그램을 토대로 제작되었습니다.

1.1.1. as1115란

64개의 LED가 I2c 인터페이스된 LED 드라이버와 Keyscan입니다.

1.1.2. AS1115의 응용 다이어그램

Figure 1. AS1115 - Typical Application Diagram



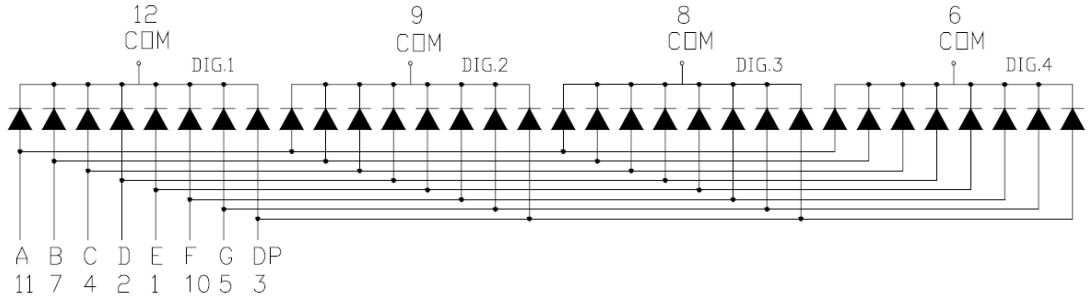
AS1115의 응용 다이어그램

1.2. seven segment

WCN4_0036SR_C11를 2개 사용하였습니다.

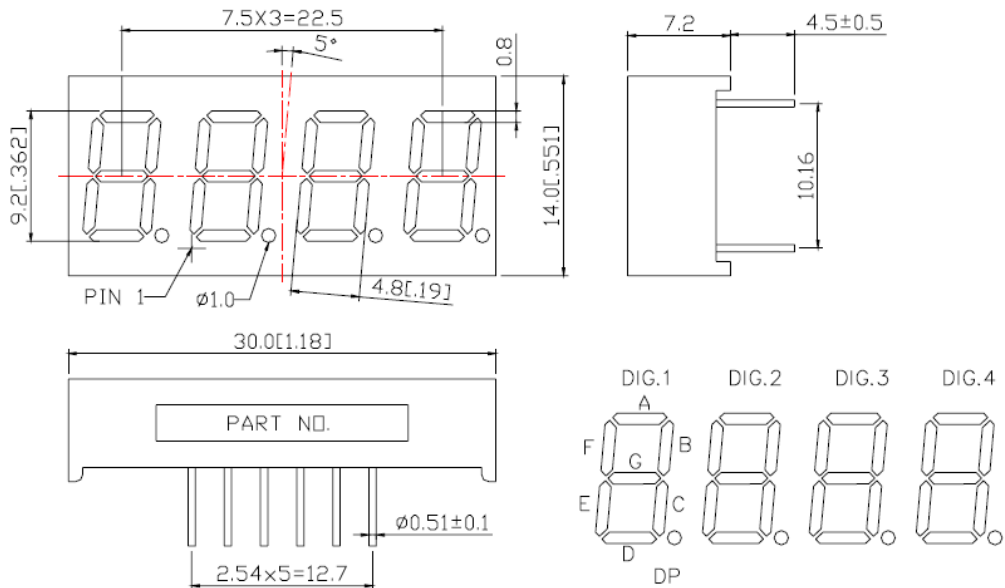
1.2.1. WCN4_0036SR_C11 내부 구조

H. WCN4-XX36XX-C1X



1.2.2. WCN4_0036SR_C11 외부 구조

D. WCN4-XX36XX-A1X/C1X



2. I2C Device Address

as1115드라이버는 I2C Address를 Factory-set IC Address와 User-set IC Address모드 두가지 선택하여 사용가능합니다.

2.1. I2C Device Address Byte

Figure 20. I²C Device Address Byte

	MSB	6	5	4	3	2	1	LSB
predefined address:	0	0	0	0	0	0	0	R/W
	MSB	6	5	4	3	2	1	LSB
updated address:	0	0	0	0	0	A1	A0	R/W

Factory-set IC Address 모드 사용 시, Figure 20의 predefined address를 사용합니다.

User-set IC Address 모드 사용 시, Figure 20의 updated address를 사용합니다.

2.2. Self Addressing Register

Table 12. Self Addressing Register (Address (HEX) = 0x2D))

	D7	D6	D5	D4	D3	D2	D1	D0
Factory-set IC address	X	X	X	X	X	X	X	0
User-set IC address	X	X	X	X	X	X	X	1

2.3. void zF_CR7SEG_SelfAddressModeControl (uint8_t addressMode, struct cr7seg_struct cr7seg_setting)

>> Factory-set IC Address 모드, User-set IC Address 모드 선택

3. Shutdown Mode

아래와 같이 4가지 모드로 설정 할 수 있습니다.

Table 7. Shutdown Register Format (Address (HEX) = 0x0C))

Mode	HEX Code	Register Data							
		D7	D6	D5	D4	D3	D2	D1	D0
Shutdown Mode, Reset Feature Register to Default Settings	0x00	0	X	X	X	X	X	X	0
Shutdown Mode, Feature Register Unchanged	0x80	1	X	X	X	X	X	X	0
Normal Operation, Reset Feature Register to Default Settings	0x01	0	X	X	X	X	X	X	1
Normal Operation, Feature Register Unchanged	0x81	1	X	X	X	X	X	X	1

Normal Operation, Reset Feature Register to Default Settings 모드를 사용합니다.

3.1. zF_CR7SEG_ShutdownModeControl(uint8_t shutdownMode, struct cr7seg_struct cr7seg_setting)

>> 위의 4가지 모드를 선택

4. Basis Setting

as1115드라이버는 크게 3가지를 설정할 수 있습니다.

4.1. Decode Control

4.1.1. Decode Mode Examples

아래와 같이 예제가 있습니다.

Table 8. Decode Enable Register Format Examples

Decode Mode	HEX Code	Register Data							
		D7	D6	D5	D4	D3	D2	D1	D0
No decode for digits 7:0	0x00	0	0	0	0	0	0	0	0
Code-B/HEX decode for digit 0. No decode for digits 7:1	0x01	0	0	0	0	0	0	0	1
Code-B/HEX decode for digit 0:2. No decode for digits 7:3	0x07	0	0	0	0	0	1	1	1
Code-B/HEX decode for digits 0:5. No decode for digits 7:6	0x3F	0	0	1	1	1	1	1	1
Code-B/HEX decode for digits 0,2,5. No decode for digits 1, 3, 4, 6, 7	0x25	0	0	1	0	0	1	0	1


















모든 digits가 No decode mode일 경우, 0x00입니다. 원하는 위치의 값을 16진수로 변환하여 입력하면 됩니다.

4.1.2. Decode Mode

decode 모드에는 2가지 Font를 사용할 수 있습니다.

Code-B-Font와 Hex Font를 사용할 수 있습니다.

Code-B-Font

Char-acter	Register Data						Char-acter	Register Data						Char-acter	Register Data					
	D7	D6:D4	D3	D2	D1	D0		D7	D6: D4	D3	D2	D1	D0		D7	D6:D4	D3	D2	D1	D0
		X	0	0	0	0			X	0	1	1	0			X	1	1	0	0
		X	0	0	0	1			X	0	1	1	1			X	1	1	0	1
		X	0	0	1	0			X	1	0	0	0			X	1	1	1	0
		X	0	0	1	1			X	1	0	0	1			X	1	1	1	1
		X	0	1	0	0			X	1	0	1	0			1	X	X	X	X
		X	0	1	0	1			X	1	0	1	1							

Hex Font

Char-acter	Register Data						Char-acter	Register Data						Char-acter	Register Data					
	D7	D6:D4	D3	D2	D1	D0		D7	D6: D4	D3	D2	D1	D0		D7	D6:D4	D3	D2	D1	D0
		X	0	0	0	0			X	0	1	1	0			X	1	1	0	0
		X	0	0	0	1			X	0	1	1	1			X	1	1	0	1
		X	0	0	1	0			X	1	0	0	0			X	1	1	1	0
		X	0	0	1	1			X	1	0	0	1			X	1	1	1	1
		X	0	1	0	0			X	1	0	1	0			X	X	X	X	X
		X	0	1	0	1			X	1	0	1	1							

이 두가지 모드는 Feature Register(0xXE)의 D2레지스터로 선택할 수 있습니다.

D2	decode_sel	0	R/W	Selects display decoding for the selected digits (Table 8 on page 13). 0 = Enable Code-B decoding (see Table 9 on page 14). 1 = Enable HEX decoding (see Table 10 on page 14).
----	------------	---	-----	--

4.1.3. No Decode Mode

No Decode Mode 사용 시 아래의 레지스터 Map를 참고하여 사용합니다.

Register Address Map

Type	Register	Address						Page	
		D15:D13	D12	D11	D10	D9	D8		D7:D0
Keyscan/Diagnostic Register	Diagnostic Digit 0	000	1	0	1	0	0		N/A
	Diagnostic Digit 1	000	1	0	1	0	1		N/A
	Diagnostic Digit 2	000	1	0	1	1	0		N/A
	Diagnostic Digit 3	000	1	0	1	1	1		N/A
	Diagnostic Digit 4	000	1	1	0	0	0		N/A
	Diagnostic Digit 5	000	1	1	0	0	1		N/A
	Diagnostic Digit 6	000	1	1	0	1	0		N/A
	Diagnostic Digit 7	000	1	1	0	1	1		N/A
	KEYA	000	1	1	1	0	0		
	KEYB	000	1	1	1	0	1		

```
// Individual digit registers
```

```
#define AS1115_DIGIT0 (0x01)
```

```
#define AS1115_DIGIT1 (0x02)
```

```
#define AS1115_DIGIT2 (0x03)
```

```
#define AS1115_DIGIT3 (0x04)
```

```
#define AS1115_DIGIT4 (0x05)
```



```
#define AS1115_DIGIT5          (0x06)
#define AS1115_DIGIT6          (0x07)
#define AS1115_DIGIT7          (0x08)
```

No Decode Mode사용시 7Segment LED를 각각 제어 가능합니다.

Figure 25. Standard 7-Segment LED Intensity Control and Inter-Digit Blanking

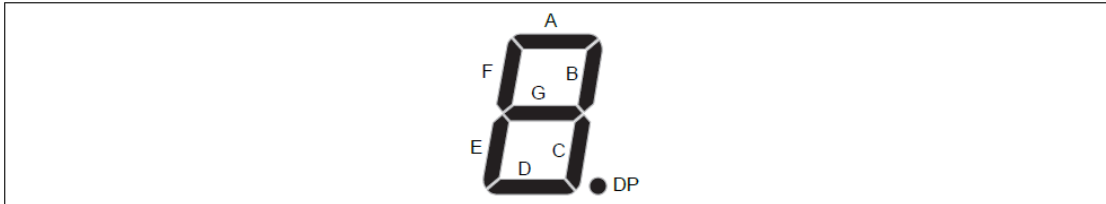


Table 11. No-Decode Mode Data Bits and Corresponding Segment Lines

	D7	D6	D5	D4	D3	D2	D1	D0
Corresponding Segment Line	DP	A	B	C	D	E	F	G

```
// No-Decode Mode Data Bits and Corresponding Segment Lines
#define AS1115_CORRESPONDING_SEGMENT_LINE_DP (0x80)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_DP
#define AS1115_CORRESPONDING_SEGMENT_LINE_A (0x40)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_A
#define AS1115_CORRESPONDING_SEGMENT_LINE_B (0x20)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_B
#define AS1115_CORRESPONDING_SEGMENT_LINE_C (0x10)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_C
#define AS1115_CORRESPONDING_SEGMENT_LINE_D (0x08)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_D
#define AS1115_CORRESPONDING_SEGMENT_LINE_E (0x04)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_E
#define AS1115_CORRESPONDING_SEGMENT_LINE_F (0x02)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_F
#define AS1115_CORRESPONDING_SEGMENT_LINE_G (0x01)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_G
#define AS1115_CORRESPONDING_SEGMENT_LINE_NO_PRINT (0x00)
////AS1115_NO_DECODEMODE_CORRESPONDING_SEGMENT_LINE_NO_PRINT
```

4.1.4. zF_CR7SEG_SetDecodeModeControl(uint8_t decodeSetting, struct cr7seg_struct cr7seg_setting)

Decode 모드를 설정하는 함수입니다.

```
// Decode Enable Register Format
#define AS1115_NO_DECODEMODE_ALL      (0x00) //No decode for digits 7:0
#define AS1115_DECODEMODE_0          (0x01) //Code-B/HEX decode for digit 0.
#define AS1115_DECODEMODE_1          (0x02) //Code-B/HEX decode for digit 1.
#define AS1115_DECODEMODE_2          (0x04) //Code-B/HEX decode for digit 2.
#define AS1115_DECODEMODE_3          (0x08) //Code-B/HEX decode for digit 3.
#define AS1115_DECODEMODE_4          (0x10) //Code-B/HEX decode for digit 4.
#define AS1115_DECODEMODE_5          (0x20) //Code-B/HEX decode for digit 5.
#define AS1115_DECODEMODE_6          (0x40) //Code-B/HEX decode for digit 6.
#define AS1115_DECODEMODE_7          (0x80) //Code-B/HEX decode for digit 7.
#define AS1115_DECODEMODE_ALL        (0xFF) //decode for digits 7:0
```

함수 호출 시 원하는 모드를 decodeSetting값에 입력하면 됩니다.

4.2. Scan_Limit Register

아래와 같이 레지스터 Format이 있습니다.

Table 19. Scan-Limit Register Format (Address (HEX) = 0x0B)

Scan Limit	HEX Code	Register Data				Scan Limit	HEX Code	Register Data			
		D7:D3	D2	D1	D0			D7:D3	D2	D1	D0
Display digit 0 only	0xX0	X	0	0	0	Display digits 0:4	0xX4	X	1	0	0
Display digits 0:1	0xX1	X	0	0	1	Display digits 0:5	0xX5	X	1	0	1
Display digits 0:2	0xX2	X	0	1	0	Display digits 0:6	0xX6	X	1	1	0
Display digits 0:3	0xX3	X	0	1	1	Display digits 0:7	0xX7	X	1	1	1

4.2.1. zF_CR7SEG_SetScanLimitControl(uint8_t scanLimitSetting, struct cr7seg_struct cr7seg_setting)

ScanLimit를 설정합니다.

모든 digit를 출력할 경우 Display digits 0:7 인 AS1115_Display_DIGIT0_7을 함수 호출시 사용합니다.

```
//Scan_Limit Registers
#define AS1115_Display_DIGIT0          (0x00) //Display digit 0 only
#define AS1115_Display_DIGIT0_1        (0x01) //Display digits 0:1
#define AS1115_Display_DIGIT0_2        (0x02) //Display digits 0:2
#define AS1115_Display_DIGIT0_3        (0x03) //Display digits 0:3
#define AS1115_Display_DIGIT0_4        (0x04) //Display digits 0:4
#define AS1115_Display_DIGIT0_5        (0x05) //Display digits 0:5
#define AS1115_Display_DIGIT0_6        (0x06) //Display digits 0:6
#define AS1115_Display_DIGIT0_7        (0x07) //Display digits 0:7
```

함수 호출 시 원하는 모드를 scanLimitSetting 값에 입력하면 됩니다.

4.3. Brightness Control

Duty Cycle을 조정하여 밝기 조정이 가능합니다.

4.3.1. Intensity Register Format

Duty Cycle	HEX Code	Register Data				Duty Cycle	HEX Code	Register Data			
		MSB	D2	D1	LSB			MSB	D2	D1	LSB
1/16 (min on)	0xX0	0	0	0	0	9/16	0xX8	1	0	0	0
2/16	0xX1	0	0	0	1	10/16	0xX9	1	0	0	1
3/16	0xX2	0	0	1	0	11/16	0xXA	1	0	1	0
4/16	0xX3	0	0	1	1	12/16	0xXB	1	0	1	1
5/16	0xX4	0	1	0	0	13/16	0xXC	1	1	0	0
6/16	0xX5	0	1	0	1	14/16	0xDD	1	1	0	1
7/16	0xX6	0	1	1	0	15/16	0xEE	1	1	1	0
8/16	0xX7	0	1	1	1	15/16 (max on)	0xFF	1	1	1	1

16개의 단계로 나뉘져 있습니다.

아래와 같이 정의하였습니다.

```
//Intensity Register Format
#define AS1115_Duty_Cycle_1           (0x00) //Duty Cycle 1/16(min on)
#define AS1115_Duty_Cycle_2           (0x01) //Duty Cycle 2/16
#define AS1115_Duty_Cycle_3           (0x02) //Duty Cycle 3/16
#define AS1115_Duty_Cycle_4           (0x03) //Duty Cycle 4/16
#define AS1115_Duty_Cycle_5           (0x04) //Duty Cycle 5/16
#define AS1115_Duty_Cycle_6           (0x05) //Duty Cycle 6/16
#define AS1115_Duty_Cycle_7           (0x06) //Duty Cycle 7/16
#define AS1115_Duty_Cycle_8           (0x07) //Duty Cycle 8/16
#define AS1115_Duty_Cycle_9           (0x08) //Duty Cycle 9/16
#define AS1115_Duty_Cycle_10          (0x09) //Duty Cycle 10/16
#define AS1115_Duty_Cycle_11          (0x0A) //Duty Cycle 11/16
#define AS1115_Duty_Cycle_12          (0x0B) //Duty Cycle 12/16
#define AS1115_Duty_Cycle_13          (0x0C) //Duty Cycle 13/16
#define AS1115_Duty_Cycle_14          (0x0D) //Duty Cycle 14/16
#define AS1115_Duty_Cycle_15          (0x0E) //Duty Cycle 15/16
#define AS1115_Duty_Cycle_16          (0x0F) //Duty Cycle 15/16(max on)
```

4.3.2. Intensity Register Address

Register HEX Address	Type	Register Data	
		D7:D4	D3:D0
0x0A	Global	X	Global Intensity
0x10	Digit	Digit 1 Intensity	Digit 0 Intensity
0x11	Digit	Digit 3 Intensity	Digit 2 Intensity
0x12	Digit	Digit 5 Intensity	Digit 4 Intensity
0x13	Digit	Digit 7 Intensity	Digit 6 Intensity

각각의 레지스터의 어드레스 입니다. 한 개의 register address에 두개의 digit를 설정합니다. 0x10 레지스터의 D7:D4는 Digit 1의 밝기를 D3:D0은 Digit 0의 밝기를 설정할 수 있습니다. 아래와 같이 정의되어 있습니다.

```
// Intensity Register Address
#define AS1115_DIGHT_Global_Intensity (0x0A) //| D7:D4 X | D3:D0 Global Intensity |
#define AS1115_DIGHT_0_1_Intensity (0x10) //| D7:D4 Digit 1 Instensity | D3:D0 Digit 0 Intensity |
#define AS1115_DIGHT_2_3_Intensity (0x11) //| D7:D4 Digit 3 Instensity | D3:D0 Digit 2 Intensity |
#define AS1115_DIGHT_4_5_Intensity (0x12) //| D7:D4 Digit 5 Instensity | D3:D0 Digit 4 Intensity |
#define AS1115_DIGHT_6_7_Intensity (0x13) //| D7:D4 Digit 7 Instensity | D3:D0 Digit 6 Intensity |
```

4.3.3. zF_CR7SEG_SetBrightnessControl(uint8_t brightnessSetting, cr7seg_setting);

함수 호출시 4.3.1 Intensity Register Format 목록에서 정의된 값을 brightnessSetting값에 입력합니다.

5. Function

4. Basis Setting 목차에서 기본 설정을 확인하였습니다.

보드 부팅 후 "A> zF_CR7SEG TEST(AS1115)"를 선택하면 zF_CR7SEG 모듈이 동작합니다.

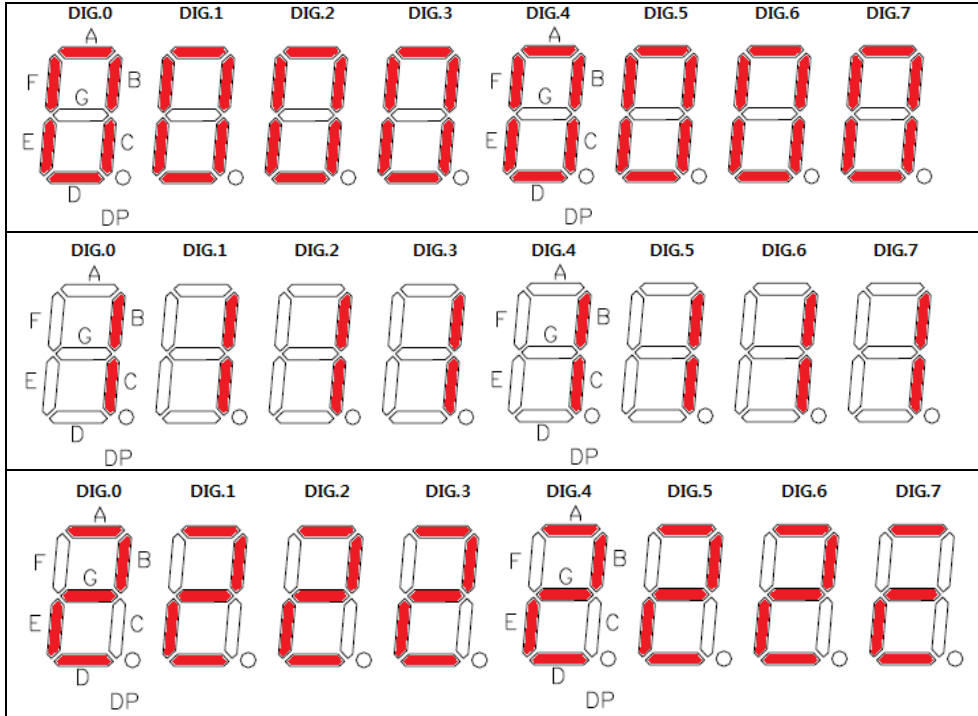
아래와 같이 출력됩니다.

```
-----
Press menu key
-----
0> System Information
1> zF_CR7SEG_Code_B_Font_Print
2> zF_CR7SEG_Test_Brightness_Print
3> zF_CR7SEG_Choice_Character_Print
4> zF_CR7SEG_Blink_Print
5> zF_CR7SEG_Scroll_Right
6> zF_CR7SEG_Scroll_Left
-----
x> quit
```

1번부터 6번까지 선택을 하면 각각의 함수를 호출합니다.

5.1. zF_CR7SEG_Code_B_Font_Print(struct cr7seg_struct cr7seg_setting)

함수 호출시 아래와 같이 Digit0~7까지 동시에 출력합니다.



출력은 아래의 "Code-B-Font"를 순차적으로 반복 출력합니다

Code-B-Font

Character	Register Data						Character	Register Data						Character	Register Data					
	D7	D6:D4	D3	D2	D1	D0		D7	D6: D4	D3	D2	D1	D0		D7	D6:D4	D3	D2	D1	D0
		X	0	0	0	0			X	0	1	1	0			X	1	1	0	0
		X	0	0	0	1			X	0	1	1	1			X	1	1	0	1
		X	0	0	1	0			X	1	0	0	0			X	1	1	1	0
		X	0	0	1	1			X	1	0	0	1			X	1	1	1	1
		X	0	1	0	0			X	1	0	1	0			X	X	X	X	X
		X	0	1	0	1			X	1	0	1	1							

5.2. zF_CR7SEG_Test_Brightness_Print(struct cr7seg_struct cr7seg_setting)

함수 호출시 아래와 같이 Digit0~7까지 순차적으로 밝게 출력합니다.



"AS1115_DIGHT_0_1_Intensity"레지스터는 아래와 같이 정의되어 있습니다.

```
#define AS1115_DIGHT_0_1_Intensity    (0x10)  //| D7:D4 Digit 1 Instensity | D3:D0 Digit 0 Intensity |
```

0x10 레지스터의 D7:D4는 Digit 1의 밝기를 D3:D0은 Digit 0의 밝기를 설정할 수 있습니다.

```
cr7seg_setting.AS1115_Register=AS1115_DIGHT_0_1_Intensity;  
dataVal=((AS1115_Duty_Cycle_3<<4)+AS1115_Duty_Cycle_1);  
printf("DIGHT_0_1_Intensity = %x\n",dataVal);  
AS1115_Write_Bytes_Data(cr7seg_setting,&dataVal);
```

소스를 보면 AS1115_DIGHT_0_1_Intensity 레지스터에 ((AS1115_Duty_Cycle_3<<4)+AS1115_Duty_Cycle_1)의 값을 write합니다.

Digit 0은 AS1115_Duty_Cycle_1의 밝기로 설정되고

Digit 1은 AS1115_Duty_Cycle_3의 밝기로 설정됩니다.

5.3. zF_CR7SEG_Choice_Character (struct cr7seg_struct cr7seg_setting)

함수 호출 시 아래와 같이 출력됩니다.

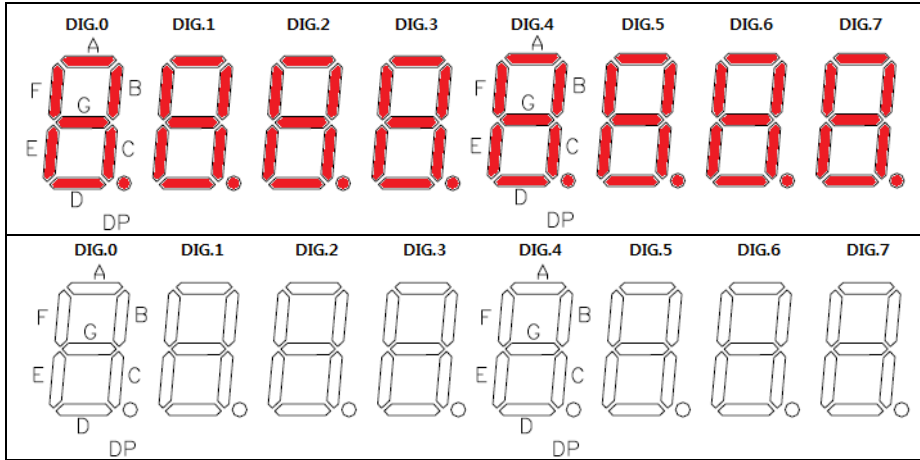
```
-----  
Press Select key (A~Z), (0~9), (.)  
-----  
x> quit
```

원하는 문자를 입력하면 Digit0~7까지 입력하는 순서대로 출력됩니다.

8번 입력 후 추가 입력할 경우 Digit0에 입력한 문자가 출력되고 Digit 1~7까지는 모두 off됩니다.

5.4. zF_CR7SEG_Blink_Print(struct cr7seg_struct cr7seg_setting)

함수 출력시 모든 엘이디가 On/Off를 반복합니다.



5.5. zF_CR7SEG_Scroll_Right(struct cr7seg_struct cr7seg_setting)

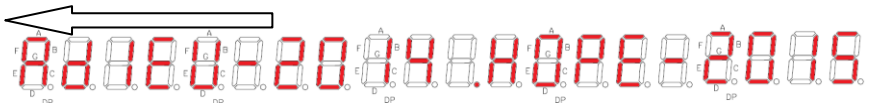
함수를 실행하면 지정한 메시지를 오른쪽으로 스크롤합니다.

메시지 지정 방법은 코드 안에 보면 "rightScrollMessage[]"배열이 있습니다.

알파벳의 경우 원하는 알파벳을 작성하면 되고, 숫자의 경우 앞에 "NUM_"을 같이 작성하면 됩니다.

```
rightScrollMessage[]={A,D,I,E,U,LINE_G,NUM_2,NUM_0,NUM_1,NUM_4,DP,H,O,P,E,NUM_G,NUM_2,NUM_0,NUM_1,NUM_5,CLEAN,CLEAN,CLEAN,CLEAN,CLEAN,CLEAN,CLEAN,CLEAN};
```

배열에 작성한 내용은 ADIEU-2014.HOPE-2015입니다.

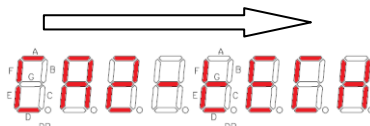


5.6. zF_CR7SEG_Scroll_Left(struct cr7seg_struct cr7seg_setting)

함수를 실행하면 지정한 메시지를 왼쪽으로 스크롤합니다.

메시지 지정방법은 코드 안에 보면 "leftScrollmessage []"배열이 있습니다.

알파벳의 경우 원하는 알파벳을 작성하면 되고, 숫자의 경우 앞에 "NUM_"을 같이 작성하면 됩니다.



5.7. zF_CR7SEG_SetBasisSetting(cr7seg_setting)

시작과 함께 함수를 호출하면 4.1, 4.2, 4.3 목록의 세팅을 Basis Setting으로 설정합니다.

5.8. zF_CR7SEG_SetClean(struct cr7seg_struct cr7seg_setting)

Digit 0 ~ Digit7까지 모든 LED를 Off합니다.