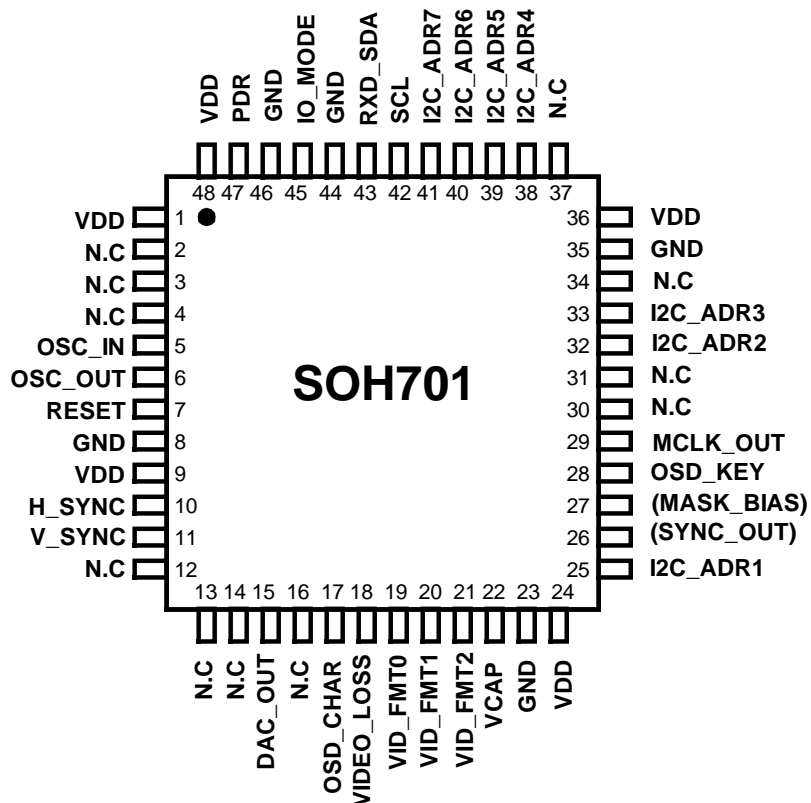


SOH701 / SOH702

Analog HD Video On-Screen Display

특징

- 아날로그 HD 비디오 신호(AHD, TVI, CVI) 호환
- 영어, 일본어, 러시아어 문자 표시
- 한글 문자 표시 (SOH702 전용)
- 16x18 픽셀 문자 크기
- Full HD 비디오에서 세로 15 줄 x 가로 24 문자 표시 가능
- HD 비디오에서 세로 16 줄 x 가로 24 문자 표시 가능
- 3M 픽셀 비디오에서 세로 16 줄 x 가로 24 문자 표시 가능
- SD 비디오(NTSC, PAL)에서 세로 12 줄 x 가로 24 문자 표시 가능
- 입력 비디오 없이도 내부동기 모드로 SD 비디오로 OSD 출력 가능
- UART 또는 I2C 외부 인터페이스
- 5V 허용(tolerant) 입출력
- -40 ~ 85°C 산업용 동작 온도



PIN 정의

PIN	이름	입출력	기능
1	VDD		3.3V
2	N.C		
3	N.C		
4	N.C		
5	OSC_IN		8MHz 크리스탈 오실레이터
6	OSC_OUT		8MHz 크리스탈 오실레이터
7	RESET	input	외부 수동 리셋 입력
8	GND		GND
9	VDD		3.3V
10	H_SYNC	input	비디오 수평동기 신호
11	V_SYNC	input	비디오 수직동기 신호
12	N.C		
13	N.C		
14	N.C		
15	DAC_OUT	output	아날로그 전압 출력
16	N.C		
17	OSD_CHAR	output	OSD 문자 신호
18	VIDEO_LOSS	output	Video loss 검출. 無: high, 有: low
19	VID_FMT0	output	비디오 형식 출력 (PAL: 000, NTSC: 001, HD25: 010, HD30: 011, FHD25: 100, FHD30: 101, 3M: 110, 4M: 111)
20	VID_FMT1	output	
21	VID_FMT2	output	
22	VCAP		4.7uF ceramic CAP
23	GND		GND
24	VDD		3.3V
25	I2C_ADR1	input	I2C 슬레이브 어드레스 비트 1
26	(SYNC_OUT)	output	내부동기 신호 (내부동기 모드 사용시)
27	(MASK_BIAS)	output	OSD 테두리 신호 레벨 (내부동기 모드 사용시)
28	OSD_KEY	output	OSD 스위치 신호
29	MCLK_OUT	output	8MHz 클럭 출력
30	N.C		
31	N.C		
32	I2C_ADR2	Input	I2C 슬레이브 어드레스 비트 2
33	I2C_ADR3	Input	I2C 슬레이브 어드레스 비트 3
34	N.C		
35	GND		GND
36	VDD		3.3V
37	N.C		
38	I2C_ADR4	input	I2C 슬레이브 어드레스 비트 4
39	I2C_ADR5	input	I2C 슬레이브 어드레스 비트 5
40	I2C_ADR6	input	I2C 슬레이브 어드레스 비트 6
41	I2C_ADR7	input	I2C 슬레이브 어드레스 비트 7
42	SCL	input	I2C SCL
43	RXD_SDA	input	UART 입력 또는 I2C SDA
44	GND		GND
45	IO_MODE	input	인터페이스 선택. UART: high, I2C: low (파워온 시)
46	GND		GND
47	PDR		3.3V
48	VDD		3.3V

제어 명령 형식

SOH7 시리즈의 명령은 다음과 같은 형식으로 구성되어 있습니다.

STX	CMD	N bytes data ...	ETX
-----	-----	------------------	-----

STX: 시작 바이트 (EDh)

CMD: 명령 바이트

데이터 바이트: 명령에 따라 가변적인 길이

EXT: 종료 바이트 (00h)

제어 명령

(1) 01h: LOCATE_POS

문자의 시작 위치를 지정합니다. 시작 위치는 시스템 리셋이나 **CLEAR_ALL** 명령시 좌측 최상단 (1,1) 으로 초기화 됩니다.

데이터 바이트 길이: 2 바이트

데이터 1	ROW 값 (1 ~ 12)
데이터 2	COLUMN 값 (1~24)

(2) 03h: CHAR_WRITE

화면에 출력될 문자를 지정합니다. 문자 코드는 각 언어별 기본 문자 코드와 동일하게 구성되어 있습니다. 한글을 예로 들면 완성형 한글 코드(KSC5601-1987)를 그대로 전달하면 글자가 출력되는 편리한 방식으로 되어 있습니다.

데이터 바이트 길이: 1 바이트 이상

데이터 1	<p>문자셋 페이지: 출력될 문자셋의 페이지를 지정합니다.</p> <p>01h: 영어 및 특수기호 (01h ~ FFh 까지 출력 가능)</p> <p>02h: 한글 완성형 코드 (01h ~ 7Fh 까지의 영문자도 함께 출력 가능)</p> <p>03h: 일본어 Shift JIS 코드 (01h ~ 7Fh 까지의 영문자도 함께 출력 가능)</p> <p>04h: 러시아어 (01h ~ BFh 까지의 영문자도 함께 출력 가능)</p> <p>05h: 몽골어 (01h ~ 21h 와 7Fh ~FFh 사이의 영문자도 함께 출력 가능)</p> <p>06h: 중국어 간체 (전용 모델 만 지원)</p> <p>08h: 커스텀 디자인 문자셋</p> <p>* 문자셋은 차후에 추가가 될 수 있습니다.</p>
데이터 2 ~ N	문자 데이터

(3) 05h: CLEAR_ALL

모든 문자를 지웁니다. 시작 위치는 좌측 최상단 (1,1) 으로 설정되며, 현재 설정된 문자속성과 문자크기도 초기화 됩니다. 약 1.4 msec 의 내부 처리 시간이 필요합니다.

데이터 바이트 길이: 없음

* 일부 글자만 지우고자 할 경우에는 공백(스페이스) 문자를 해당 위치에 덮어쓰면 됩니다.

(4) 07h: CHAR_ONOFF

전체 문자를 보이거나 숨깁니다.
데이터 바이트 길이: 1 바이트

데이터 1	01h: 문자를 보임 [기본값] 02h: 문자를 숨김
-------	---

(5) 0Ch: WORD_WRAP

문자 입력시 자동 줄넘김 기능을 On/Off 합니다.
데이터 바이트 길이: 1 바이트

데이터 1	01h: 자동 줄넘김 기능을 사용함 [기본값] 02h: 자동 줄넘김 기능을 사용하지 않음
-------	---

(6) 17h: DAC_OUTPUT

DAC_OUT 핀의 출력 전압값을 설정합니다.
데이터 바이트 길이: 1 바이트

데이터 1	00 ~ FFh (0 ~ 3.3V) 기본값: D1h (2.7V)
-------	--

문자출력 프로그램 예

```
#define SOH7_STX          0x0F          // start code
#define FOSD_ETX          0x00          // end code

#define FOSD_LOCATE_POS   0x01          // 문자 화면위치 지정
#define FOSD_CHAR_WRITE   0x03          // 문자열 쓰기
    #define PAGE_ENGLISH   1
    #define PAGE_KOREAN    2
    #define PAGE_JAPANESE  3
    #define PAGE_RUSSIAN    4
#define FOSD_CLEAR_ALL    0x05          // 모든 문자 지우기
#define FOSD_CHAR_ONOFF   0x07          // 문자 보이기/가리기
#define FOSD_WORD_WRAP    0x0C
#define FOSD_DACOUT        0x17
```

```
void uart_send(unsigned char data)
{
    // put your code here
}
```

```
void OsdLocate(unsigned char row, unsigned char col)
{
    uart_send(SOH7_STX);
    uart_send(FOSD_LOCATE_POS);
    uart_send(row);
}
```

```

        uart_send(col);
        uart_send(FOSD_ETX);
    }

void OsdSetAttribute(unsigned char attr)
{
    uart_send(SOH7_STX);
    uart_send(FOSD_SET_ATTRIB);
    uart_send(attr);
    uart_send(FOSD_ETX);
}

void OsdPutStr(unsigned char page, unsigned char *str)
{
    if(*str == 0) return;

    uart_send(SOH7_STX);
    uart_send(FOSD_CHAR_WRITE);
    uart_send(page);
    while(*str)
    {
        uart_send(*str);
        str++;
    }
    uart_send(FOSD_ETX);
}

void OsdPutChars(unsigned char charpage, unsigned char *str, unsigned
char length)
{
    unsigned char i;

    if(length == 0) return;

    uart_send(SOH7_STX);
    uart_send(FOSD_CHAR_WRITE);
    uart_send(charpage);
    for(i=0; i<length; i++)
    {
        uart_send(*str);
        str++;
    }
    uart_send(FOSD_ETX);
}

void OsdClear(void)
{
    uart_send(SOH7_STX);
    uart_send(FOSD_CLEAR_ALL);
    uart_send(FOSD_ETX);
}

int main(void)
{
    OsdClear();
    OsdLocate(1,1);

```

```

    OsdPutStr(PAGE_KOREAN, "아름다운 우리 강산");
    OsdLocate(2,1);
    OsdPutStr(PAGE_ENGLISH, "Best Korea");
    OsdLocate(3,1);
    OsdPutStr(PAGE_KOREAN, "한글영문 혼합 출력 Test!");

    return 0;
}

```

외부 인터페이스

(1) UART 인터페이스 모드

파워 온 시에 **IO_MODE** 핀이 **high** 인 경우에 **UART** 통신 모드로 동작합니다.

38400 BPS 속도를 사용합니다.

(Start bit: 1bit - Data bit: 8bits - Parity bit: none - Stop bit: 1bit)

(2) I2C 인터페이스 모드

파워 온 시에 **IO_MODE** 핀이 **low** 인 경우에 **I2C** 통신 모드로 동작합니다. 통신속도는 최대 **100kHz** 입니다. 슬레이브 어드레스는 **I2C_ADR7 ~ I2C_ADR1** 핀을 이용해 설정됩니다.

[슬레이브 어드레스 바이트]

비트 7	비트 6	비트 5	비트 4	비트 3	비트 2	비트 1	비트 0
I2C_ADR7	I2C_ADR6	I2C_ADR5	I2C_ADR4	I2C_ADR3	I2C_ADR2	I2C_ADR1	I2C READ / WRITE BIT

7비트이므로 최대 **127**개 까지 설정할 수 있습니다. 단, 모든 어드레스 비트가 0인 **00h** 어드레스는 브로드캐스트(공통) 어드레스로 설정돼 있으며 **I2C** 버스에 연결된 모든 **OSD IC**에 한꺼번에 동일한 명령을 보낼 경우에 사용됩니다. 모든 비디오 채널에 시간 등을 표시할 경우에 유용한 기능입니다.

I2C 로 통신을 하는 경우는 제어 명령을 **I2C** 쓰기명령 형식에 삽입되는 형태로 데이터를 실어 보냅니다.

start	slave address	write	OSD control command (N bytes)	stop
-------	---------------	-------	-------------------------------	------

예를들어 **LOCATE_POS** 명령의 경우 다음과 같이 적용될 수 있습니다.

```

void OsdLocate(unsigned char row, unsigned char col)
{
    i2c_start();
    i2c_write(FOSD_I2C_ADR);

    i2c_write(SOH7_STX);
    i2c_write(FOSD_LOCATE_POS);
    i2c_write(row);
    i2c_write(col);
    i2c_write(FOSD_ETX);

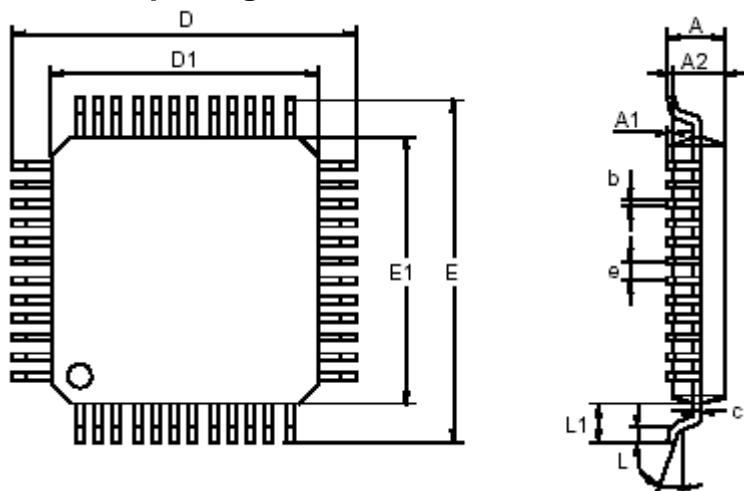
    i2c_stop();
}

```

}

데이터 수신시 내부적으로 **BUSY** 상태인 경우 **SCL** 라인을 **low** 로(**clock stretching**) 유지하여 **master** 장치와 **handshake** 를 실행하고 있으므로 **master** 장치는 **ACK** 신호 이후에 다음 클럭 신호를 출력할 때 **SCL** 라인을 검사하여 **SCL** 라인이 **low** 상태인지 확인하여 대기하는 기능이 필요합니다. 보통 **MCU**의 내장 **IC2** 컨트롤러는 이 기능이 자동으로 처리됩니다. 그러나 **GPIO**를 이용해 소프트웨어 적으로 제어하는 경우에는 주의가 필요합니다.

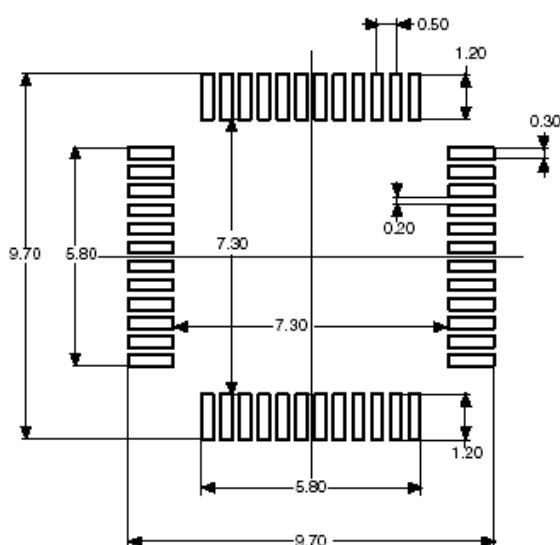
LQFP48 package outline



LQFP48 package mechanical data

Dimm.	mm		
	Min	Typ	Max
A			1.60
A1	0.05		0.15
A2	1.135	1.140	1.145
b	0.17	0.22	0.27
C	0.09		0.20
D		9.00	
D1		7.00	
E		9.00	
E1		7.00	
e		0.50	
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L1		1.00	

Recommended footprint



Dimensions are in millimeters.

English Font Table

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10	0x11
0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F	0x20	0x21	0x22	0x23
0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F	0x30	0x31	0x32	0x33	0x34	0x35
0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59
0x5A	0x5B	0x5C	0x5D	0x5E	0x5F	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B
0x6C	0x6D	0x6E	0x6F	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77	0x78	0x79	0x7A	0x7B	0x7C	0x7D
0x7E	0x7F	0x80	0x81	0x82	0x83	0x84	0x85	0x86	0x87	0x88	0x89	0x8A	0x8B	0x8C	0x8D	0x8E	0x8F
0x90	0x91	0x92	0x93	0x94	0x95	0x96	0x97	0x98	0x99	0x9A	0x9B	0x9C	0x9D	0x9E	0x9F	0xA0	0xA1
0xA2	0xA3	0xA4	0xA5	0xA6	0xA7	0xA8	0xA9	0xAA	0xAB	0xAC	0xAD	0xAE	0xAF	0xB0	0xB1	0xB2	0xB3
0xB4	0xB5	0xB6	0xB7	0xB8	0xB9	0xBA	0xBB	0xBC	0xBD	0xBE	0xBF	0xC0	0xC1	0xC2	0xC3	0xC4	0xC5

À	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	
0xC6	0xC7	0xC8	0xC9	0xCA	0xCB	0xCC	0xCD	0xCE	0xCF	0xD0	0xD1	0xD2	0xD3	0xD4	0xD5	0xD6	0xD7
Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â	ã	ä	å	æ	ç	è	é
0xD8	0xD9	0xDA	0xDB	0xDC	0xDD	0xDE	0xDF	0xE0	0xE1	0xE2	0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9
ê	ë	ì	í	î	ï	ð	ñ	ò	ó	ô	õ	ö			ø	ù	ú
0xEA	0xEB	0xEC	0xED	0xEE	0xEF	0xF0	0xF1	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7	0xF8	0xF9	0xFA	0xFB
ü	ý	þ	ÿ														
0xFC	0xFD	0xFE	0xFF														

Russian Font Table

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
0xC0	0xC1	0xC2	0xC3	0xC4	0xC5	0xC6	0xC7	0xC8	0xC9	0xCA	0xCB	0xCC	0xCD	0xCE	0xCF	0xD0	0xD1
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	а	б	в	г
0xD2	0xD3	0xD4	0xD5	0xD6	0xD7	0xD8	0xD9	0xDA	0xDB	0xDC	0xDD	0xDE	0xDF	0xE0	0xE1	0xE2	0xE3
д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х
0xE4	0xE5	0xE6	0xE7	0xE8	0xE9	0xEA	0xEB	0xEC	0xED	0xEE	0xEF	0xF0	0xF1	0xF2	0xF3	0xF4	0xF5
ц	ч	ш	щ	ъ	ы	ь	э	ю	я								
0xF6	0xF7	0xF8	0xF9	0xFA	0xFB	0xFC	0xFD	0xFE	0xFF								

Mongolian Font Table

П	”	/	:	?	п	()	.	Щ	ь	е	в	ю	0	1	2	3
0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F	0x30	0x31	0x32	0x33
4	5	6	7	8	9	Д	д	ь	щ	В	Ю	—	Й	М	Ё	Б	У
0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F	0x40	0x41	0x42	0x43	0x44	0x45
Ө	А	Х	Ш	Р	О	Л	Т	И	Ү	З	Ф	Ж	Ы	Э	Г	С	Ц
0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
Ч	Н	Я	К	—	ь	.	Е	=	й	м	ё	б	у	ө	а	х	ш
0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69
р	о	л	т	и	ү	з	ф	ж	ы	э	г	с	ц	ч	н	я	к
0x6A	0x6B	0x6C	0x6D	0x6E	0x6F	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77	0x78	0x79	0x7A	0x7B
І	Ь	+															
0x7C	0x7D	0x7E															

Revision History

버전 1.0 (2017/06/15)

- 최초 버전