OLED DISPLAY MODULE

Application Notes

PRODUCT NUMBER	DD-160128FC-1A with EVK board

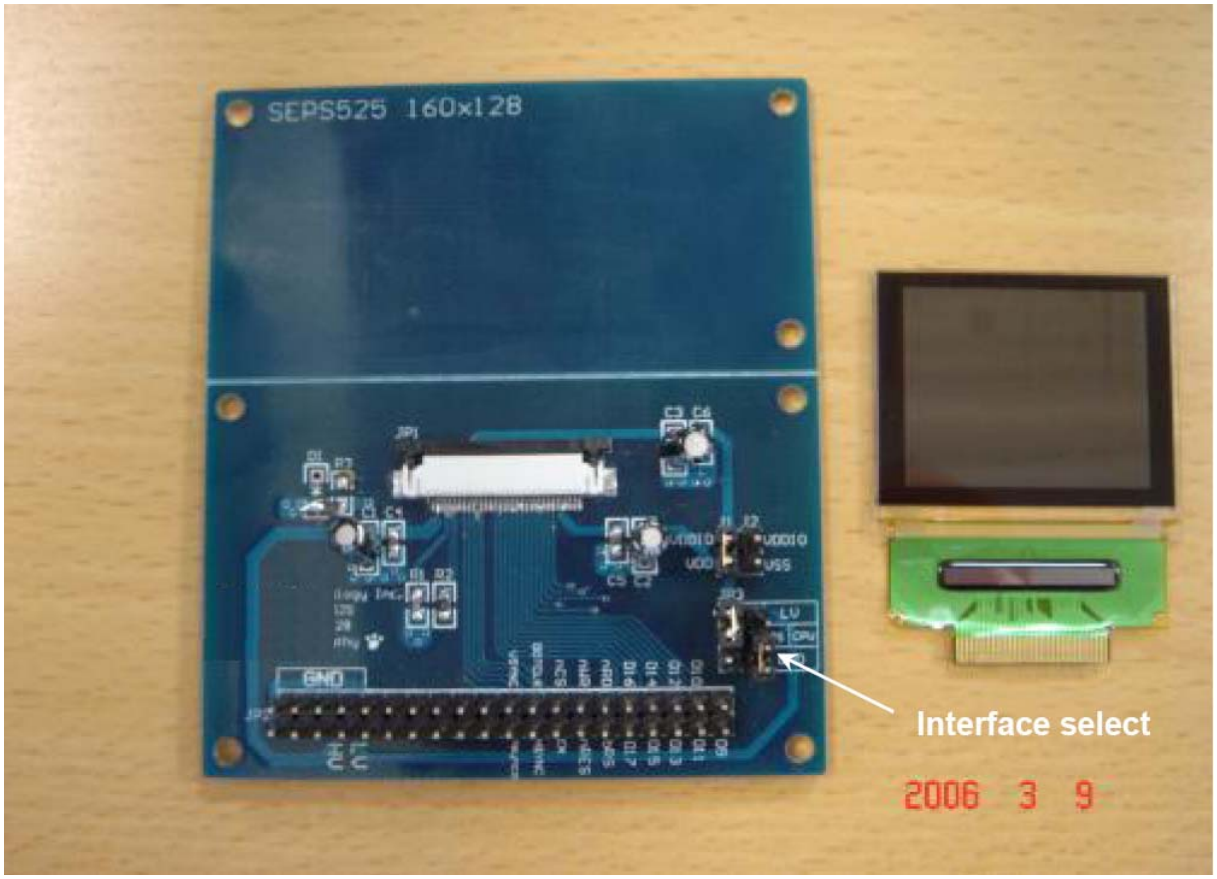
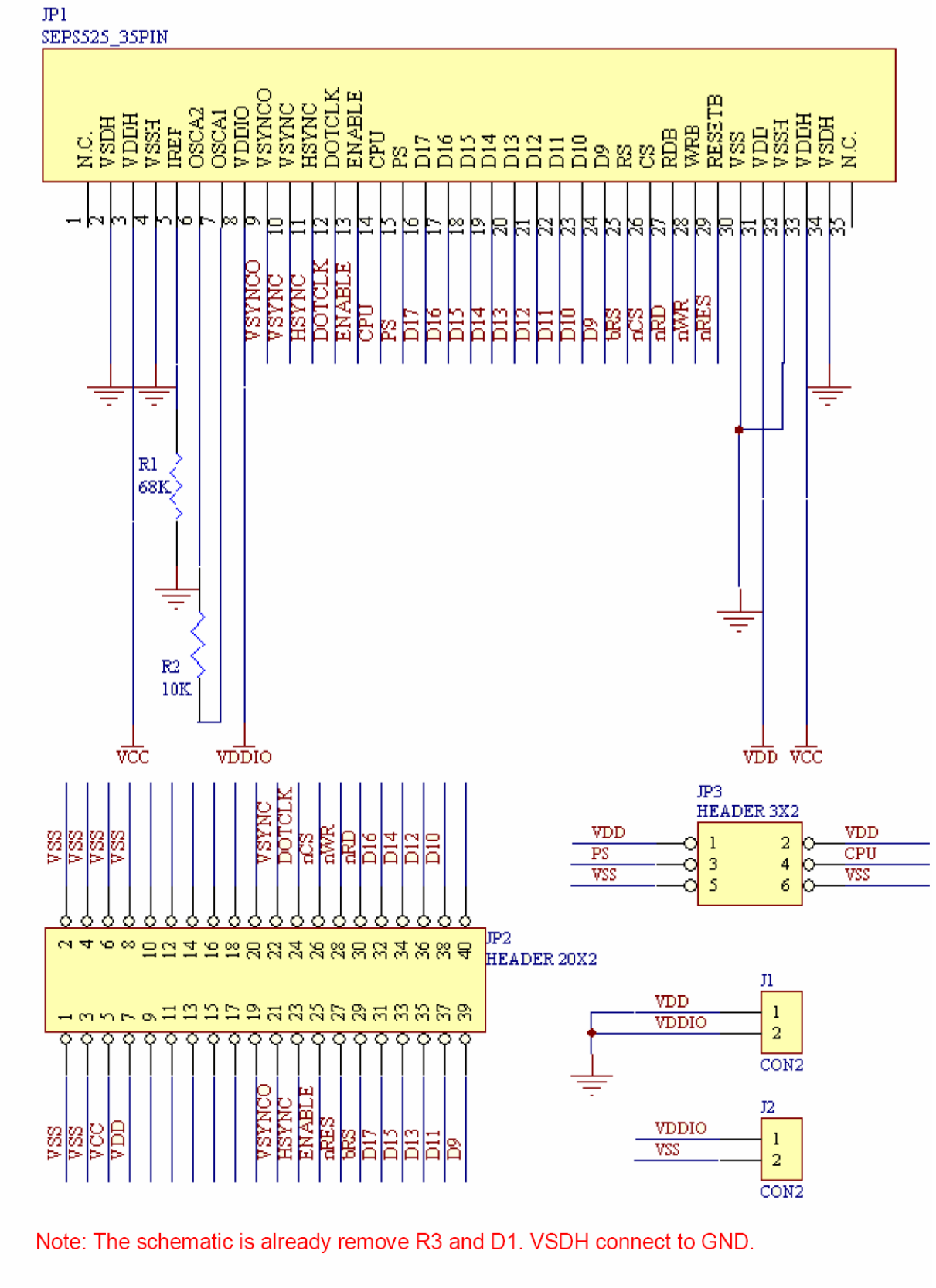


TABLE OF CONTENTS

REVISION RECORD

Rev.	Date	Page	Chapt.	Comment	ECR no.
A	30 Sep 06			First Issue	

1 EVK Schematic



Note: The schematic is already remove R3 and D1. VSDH connect to GND.

2 Symbol Definition

D17-D9：These pins are 9-bit bi-directional data bus to be connected to the MCU's data bus.

D17-D10： These pins are 8-bit bi-directional data bus to be connected to the MCU's data bus.

CSB：These pins are CSB pins for master and slave driver IC. This pin is the chip select input. The chip is enabled for MCU communication only when CSB is pulled low.

CPU： Selects the CPU type.

Low: 80-series CPU
High: 68-Series CPU

PS： Selects parallel/Serial interface type.

Low: serial
High: parallel.

RDB： For an 80-system bus interface, read strobe signal(active low).
For a 68-system bus interface, bus enable strobe (active high).
When using SPI, fix it to VDD or VSS level.

WRB：For an 80-system bus interface, write strobe signal (active low).
For a 68-system bus interface, read/write select.
Low: Write
High: Read.
When using SPI, fix it to VDD or VSS level.

RESB：Reset SEPS525F(active low).

HV：External Column Driving Power Supply.

LV：Logic power supply.

GND：Power supply ground.

Note1: Please ground the unused data pins
Note2: If you do not use RGB Interface, please ground VSYNC, HSYNC, Enable, DOTCLK and floating VSYNCO.
Note3: If you doe not use VDDIO, please connect it to LV (VDD).

3 Timing characteristics

80-Series MPU parallel Interface (write timing)

(VDD = 2.8V, Ta = 25˚C)

ITEM	SYMBOL	CONDITION	MIN	MAX	UNIT	PORT
Address hold timing	tAH8	-	5	-	ns	CSB
Address setup timing	tAS8		5		ns	RS
System cycle timing	tCYC8	-	100	-	ns	WRB
Write “L” pulse width	tWRLW8		45		ns	
Write “H” pulse width	tWRHW8		45		ns	
Data setup timing	tDS8	-	30	-	ns	DB[17:0]
Data hold timing	tDH8		10		ns	

All the timing reference is 10% and 90% of VDD

Table 1 80-Series MPU Parallel Interface Timing Characteristics (Write)

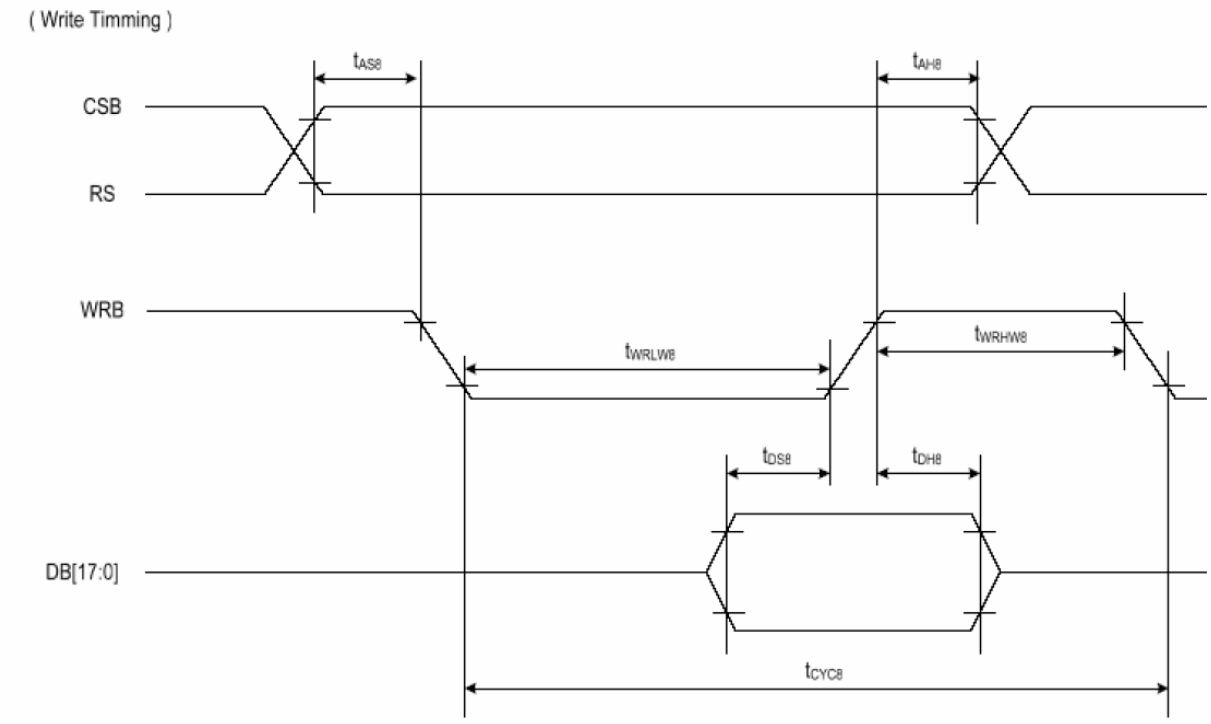


Figure 1 80-Series MPU parallel Interface Timing Diagram (Write)

80-Series MPU parallel Interface (Read timing)

(VDD = 2.8V, Ta = 25˚C)

ITEM	SYMBOL	CONDITION	MIN	MAX	UNIT	PORT
Address hold timing	tAH8	-	5	-	ns	CSB
Address setup timing	tAS8		5		ns	RS
System cycle timing	tCYC8	-	200	-	ns	RDB
Read “L” pulse width	tWRLW8		90		ns	
Read “H” pulse width	tWRHW8		90		ns	
Data setup timing	tDS8	CL = 15 pF	-	60	ns	DB[17:0]
Data hold timing	tDH8		0		ns	

All the timing reference is 10% and 90% of VDD

Table 2 80-Series MPU Parallel Interface Timing Characteristics (Read)

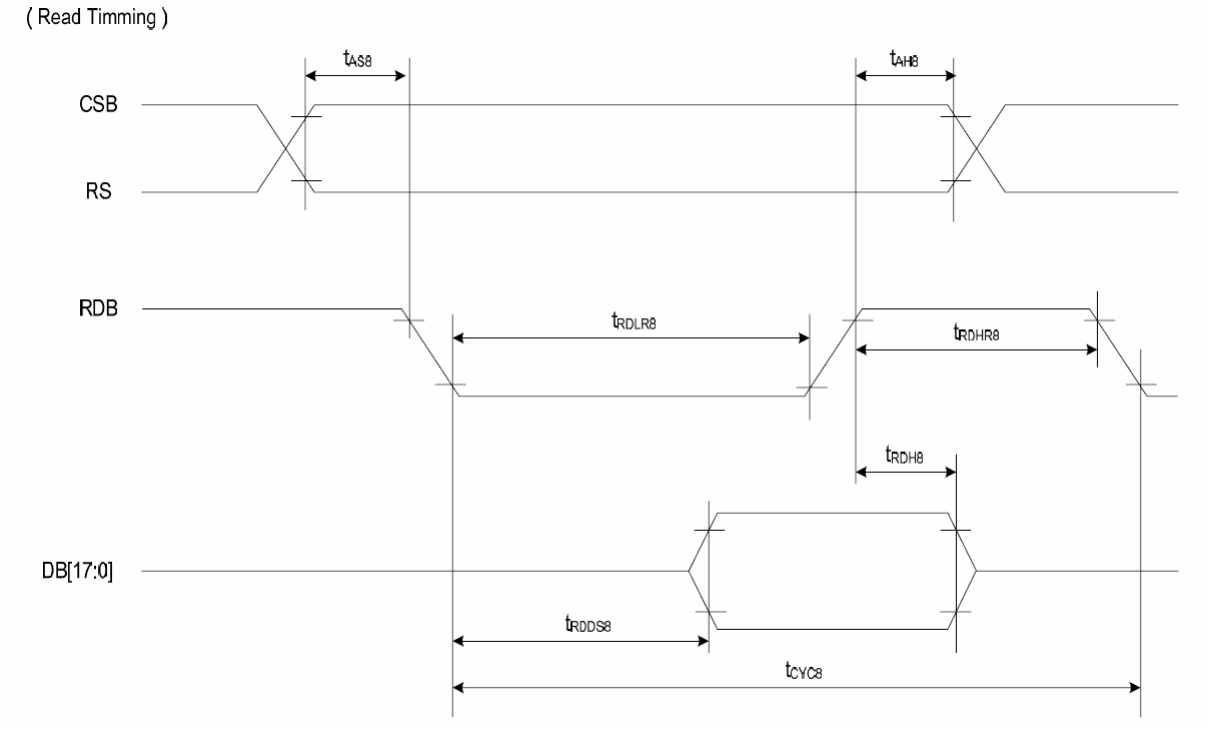


Figure 2 80-Series MPU parallel Interface Timing Diagram (Read)

6800-Series MPU parallel Interface (write timing)

(VDD = 2.8V, Ta = 25˚C)

ITEM	SYMBOL	CONDITION	MIN	MAX	UNIT	PORT
Address hold timing	tAH8	-	5	-	ns	CSB
Address setup timing	tAS8		5		ns	RS
System cycle timing	tCYC8	-	100	-	ns	E
Write “L” pulse width	tWRLW8		45		ns	
Write “H” pulse width	tWRHW8		45		ns	
Data setup timing	tDS8		40	60	ns	DB[17:0]
Data hold timing	tDH8		10		ns	

All the timing reference is 10% and 90% of VDD

Table 3 6800-Series MPU Parallel Interface Timing Characteristics (Write)

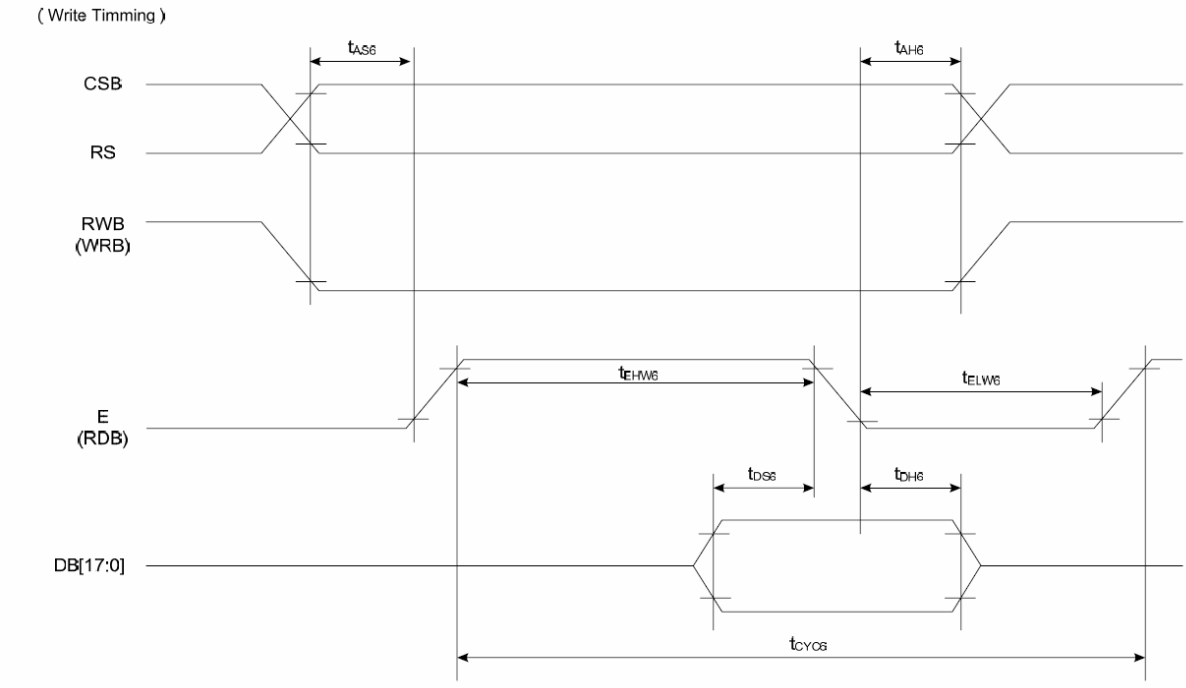


Figure 3 6800-Series MPU parallel Interface Timing Diagram (Write)

68-Series MPU parallel Interface (Read timing)

(VDD = 2.8V, Ta = 25˚C)

ITEM	SYMBOL	CONDITION	MIN	MAX	UNIT	PORT
Address hold timing	tAH8	-	10	-	ns	CSB
Address setup timing	tAS8		10		ns	RS
System cycle timing	tCYC8	-	200	-	ns	E
Read “L” pulse width	tWRLW8		90		ns	
Read “H” pulse width	tWRHW8		90		ns	
Data setup timing	tDS8	CL = 15 pF	0	70	ns	DB[17:0]
Data hold timing	tDH8				ns	

All the timing reference is 10% and 90% of VDD

Table 4 80-Series MPU Parallel Interface Timing Characteristics (Read)

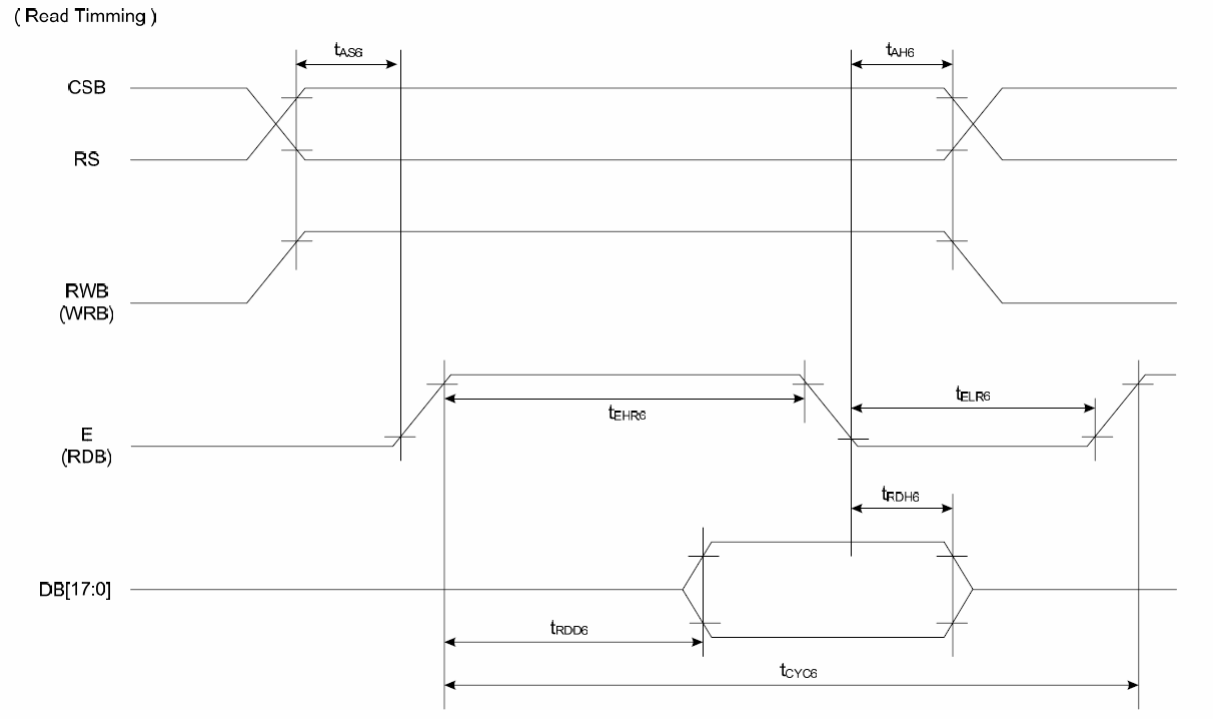


Figure 4 6800-Series MPU parallel Interface Timing Diagram (Read)

SPI Interface

(VDD = 2.8V, Ta = 25˚C)

ITEM	SYMBOL	CONDITION	MIN	MAX	UNIT	PORT
Serial clock cycle	tCYCS	-	60	-	ns	SCL
SCL “H” pulse width	tSHW		25		ns	
SCL “L” pulse width	tSLW		25		ns	
Data setup timing	tDSS	-	25	-	ns	SDI
Data hold timing	tDHS		25		ns	
CSB SCL timing	tCSS	-	25	-	ns	CSB
CSB hold timing	tCSH				ns	

All the timing reference is 10% and 90% of VDD

Table 5 SPI Interface Timing Characteristics

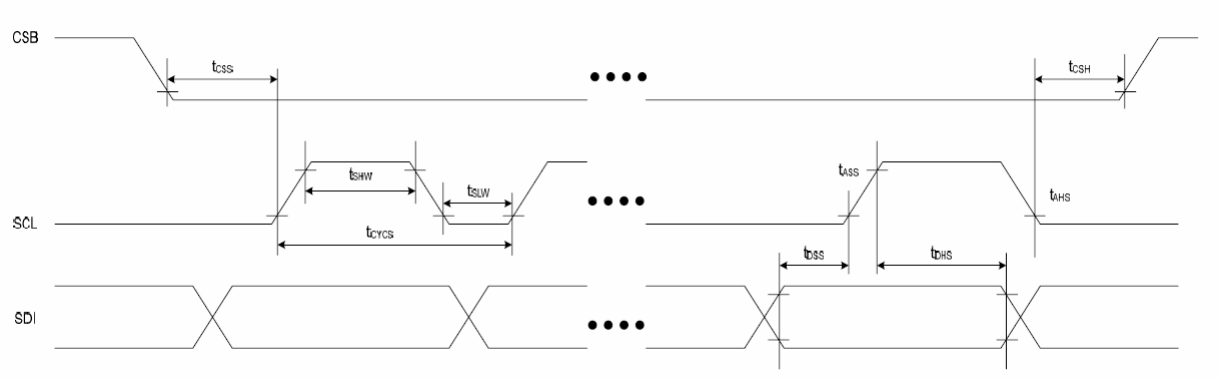


Figure 5 SPI Interface Timing Characteristics

4 Connection Between OLED and EVK

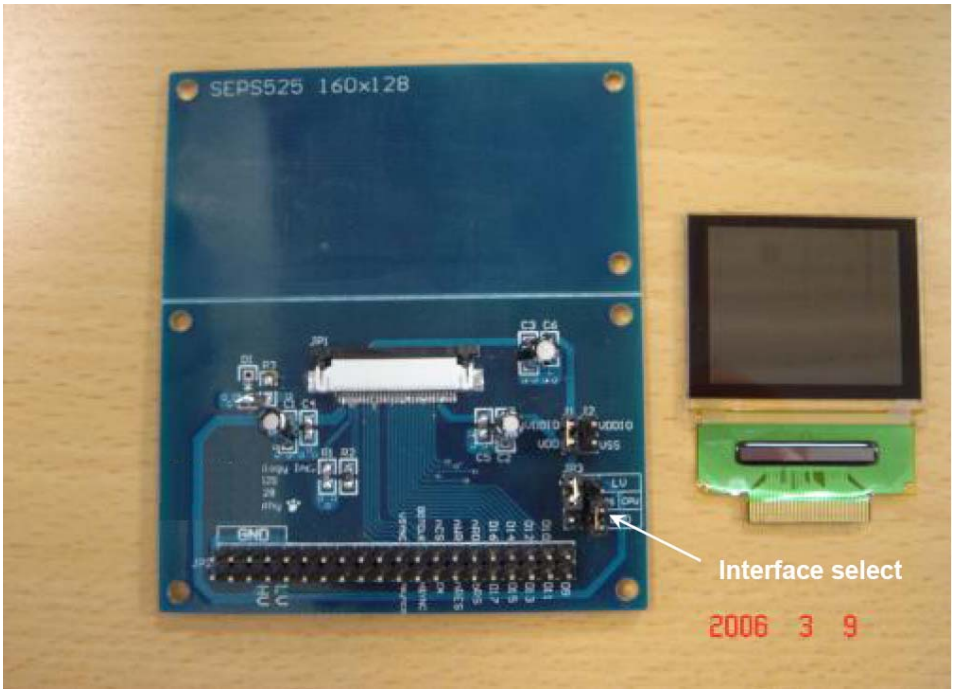


Figure 6 EVK PCB and DD-160128FC-1A Module

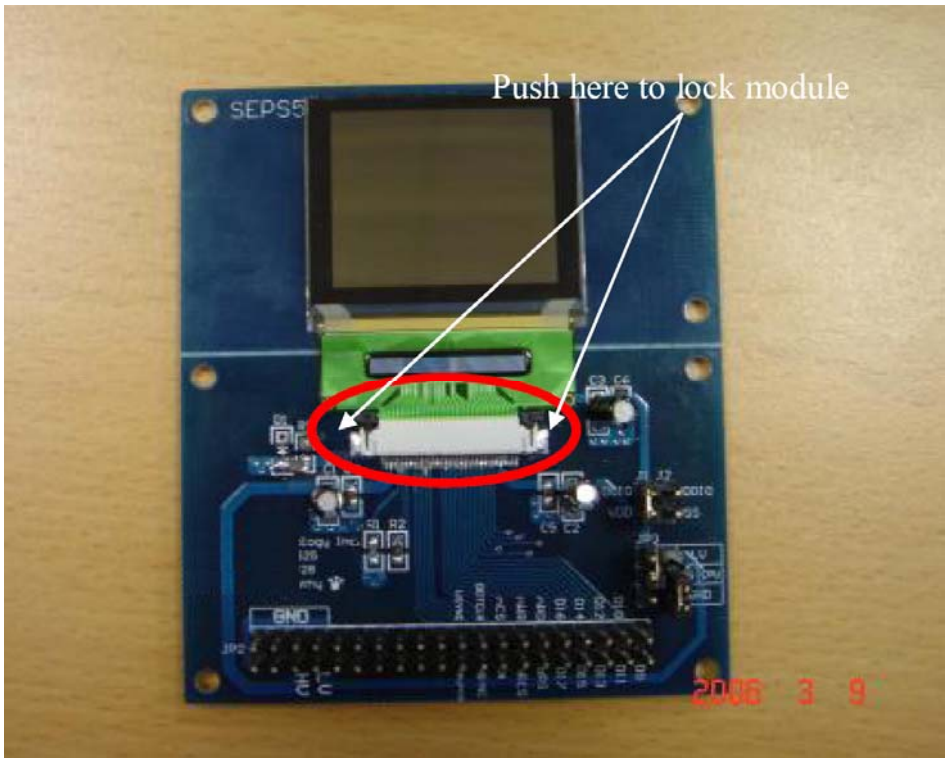


Figure 7 DD-160128FC-1A and EVK assembled (Top view)

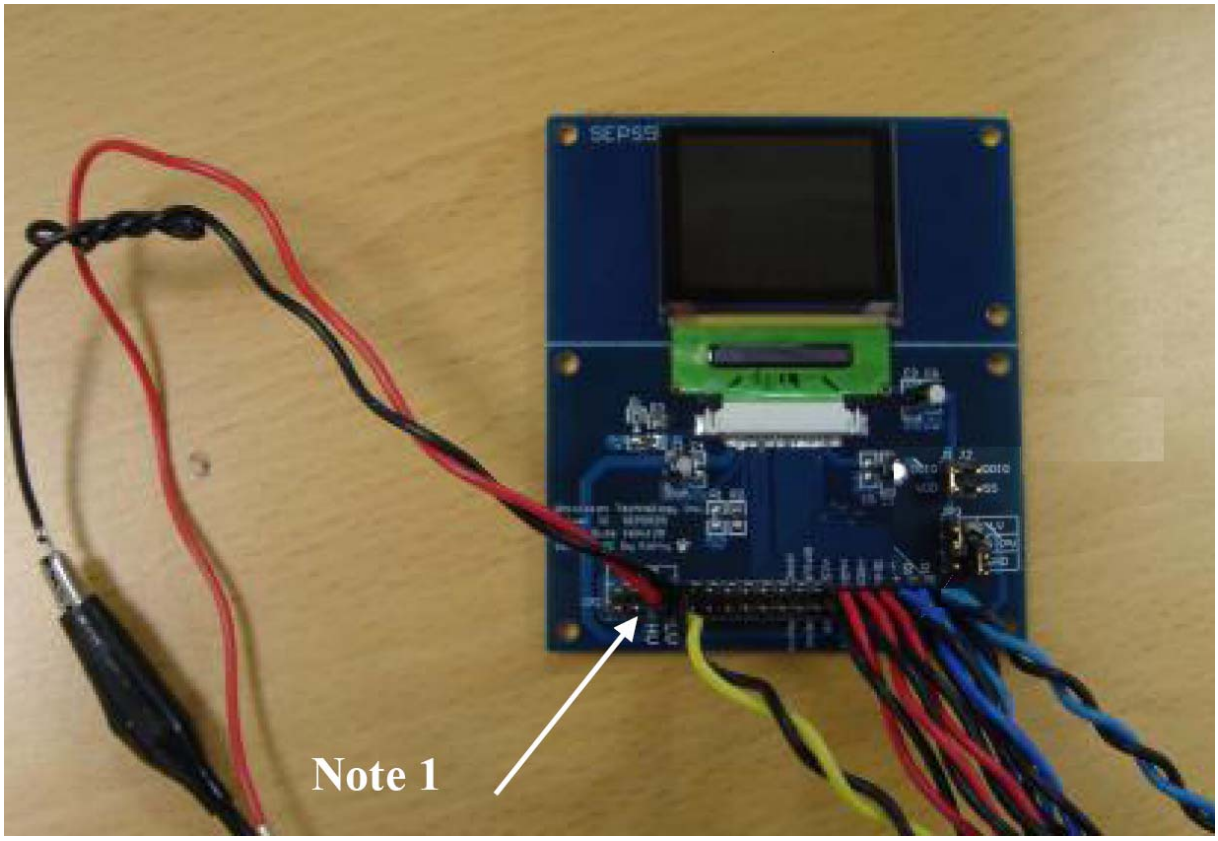
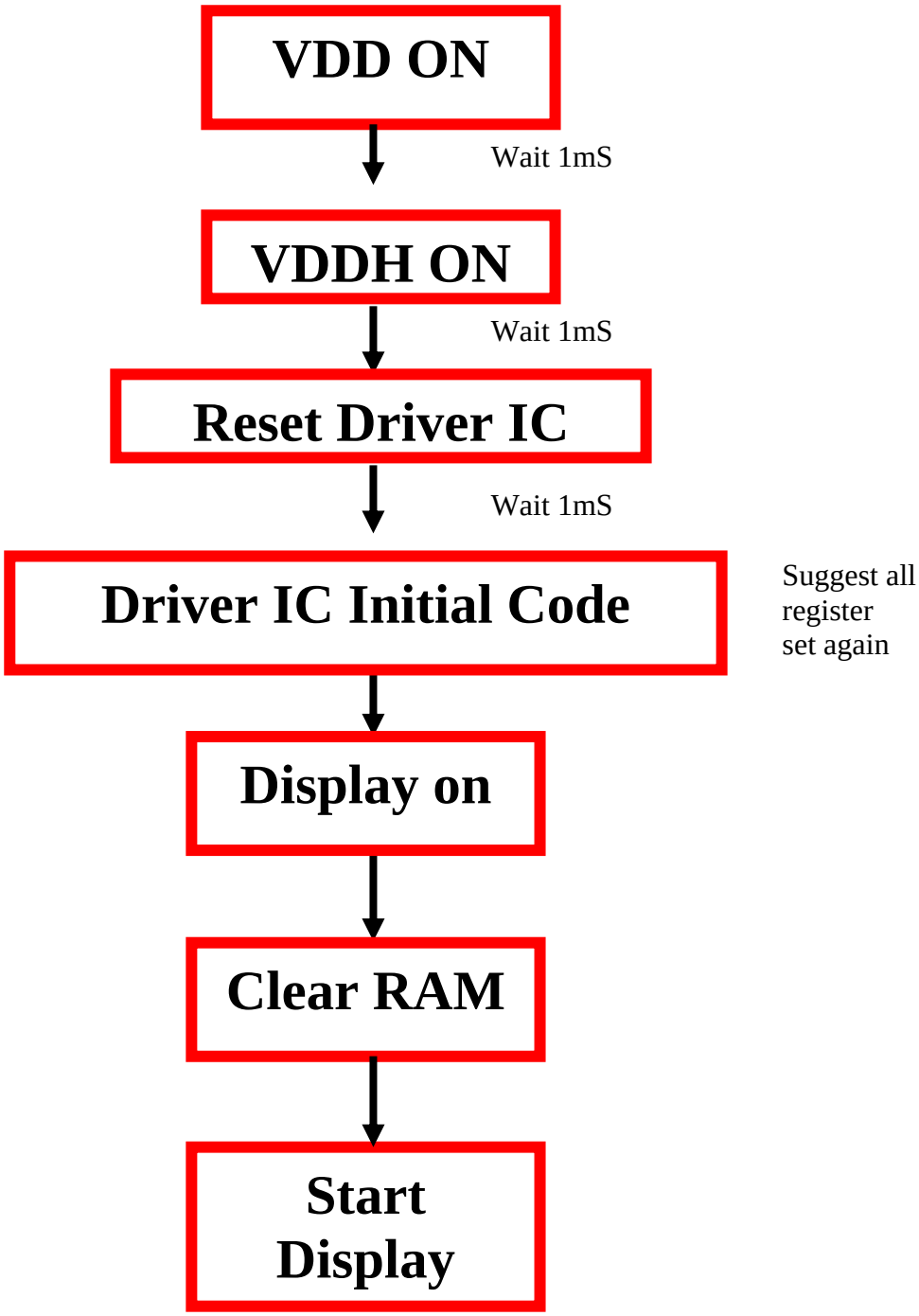


Figure 8 control MCU (not supplied) connected with EVK

Note 1：It is the external most positive voltage supply. In this sample is connected to power supply.

5 How to use the DD-160128FC-1A



5.1 Recommended Initial code for 80 interface

```
//Reg:04h Action:Normal current and PS ON ; Internal osc power off
Write_Command(rREDUCE_CURRENT,0x03);
T0_05sec();
//Reg:04h Action:Normal current and PS OFF
Write_Command(rREDUCE_CURRENT,0x00);
T0_05sec();
//Reg:3Bh Action:Screen Saver OFF
Write_Command(rSCREEN_SAVER_CONTEROL,0x00);
//Reg:02h Action:Export 0 /OSC with external resister/Internal OSC ON
Write_Command(rOSC_CTL,0x41);
//Reg:03h Action:FR=90Hz DIV=1
Write_Command(rCLOCK_DIV,0x30);
//Reg:80h Action:PDAC OFF,DDAC OFF/Reference Volt.control with external resister
Write_Command(rIREF,0x00);
//Reg:08h Action:set color R precharge time
Write_Command(rPRECHARGE_TIME_R,0x01);
//Reg:09h Action:set color G precharge time
Write_Command(rPRECHARGE_TIME_G,0x01);
//Reg:0Ah Action:set color B precharge tiem
Write_Command(rPRECHARGE_TIME_B,0x01);
//Reg:0Bh Action:set color R precharge current
Write_Command(rPRECHARGE_Current_R,0x0a);
//Reg:0Ch Action:set color G precharge current
Write_Command(rPRECHARGE_Current_G,0x0a);
//Reg:0Dh Action:set color B precharge current
Write_Command(rPRECHARGE_Current_B,0x0a);
//Reg:10h Action:set color R dot driving current
Write_Command(rDRIVING_CURRENT_R,0x46);
//Reg:11h Action:set color G dot driving current
Write_Command(rDRIVING_CURRENT_G,0x38);
//Reg:12h Action:set color B dot driving current
Write_Command(rDRIVING_CURRENT_B,0x3a);
//Reg:13h Action:Col D0 to D159/col normal display
Write_Command(rDISPLAY_MODE_SET,0x00);
//Reg:14h Action:MPU mode
Write_Command(rRGB_IF,0x31);
//Reg:16h Action:8btis dual transfer,65K support
Write_Command(rMEMORY_WRITE_MODE,0x66);
//Reg:17h Action:Memory addr.X start
Write_Command(rMX1_ADDR,0x00);
//Reg:18h Action:Memory addr.X end
Write_Command(rMX2_ADDR,0x9f);
//Reg:18h Action:Memory addr.Y start
Write_Command(rMY1_ADDR,0x00);
//Reg:1Ah Action:Memory addr.Y end
Write_Command(rMY2_ADDR,0x7f);
//Reg:20h Action:Memory X start addr.
```

```
Write_Command(rMEMORY_ACCESS_POINTER_X,0x00);
//Reg:21h Action:Memory Y start addr.
Write_Command(rMEMORY_ACCESS_POINTER_Y,0x00);
//Reg:28h Action:Display duty ratio
Write_Command(rDUTY,0x7f);
//Reg:29h Action:Display start line
Write_Command(rDSL,0x00);
//Reg:2Eh Action:Display First screen X start point
Write_Command(rD1_DDRAM_FAC,0x00);
//Reg:2Fh Action:Display First screen Y start point
Write_Command(rD1_DDRAM_FAR,0x00);
//Reg:31h Action:Display Second screen X start point
Write_Command(rD2_DDRAM_SAR,0x00);
//Reg:32h Action:Display Second screen Y start point
Write_Command(rD2_DDRAM_SAR,0x00);
//Reg:33h Action:Display size X start
Write_Command(rSCR1_FX1,0x00);
//Reg:34h Action:Display size X end
Write_Command(rSCR1_FX2,0x9f);
//Reg:35h Action:Display size Y start
Write_Command(rSCR1_FY1,0x00);
//Reg:36h Action:Display size Y end
Write_Command(rSCR1_FY2,0x7f);
//Reg:06h Action:Scan signal is high level at precharge period/Dispaly ON
Write_Command(rDISP_ON_OFF,0x01);
```

Sub Function for 80 Interface

```
void Write_Register(unsigned char data)
        {
                IOCLR = 0x0000000ff;//reset D0~D7
                IOCLR=bRS;
                IOCLR=nCS;
                IOCLR=nWR;
                IOSET=data;
                IOSET=nWR;
                IOSET=nCS;
                IOSET=bRS;
        }
void Write_Parameter(unsigned char data)
        {
                IOCLR = 0x0000000ff;//reset D0~D7
                IOSET=bRS;
                IOCLR=nCS;
                IOCLR=nWR;
                IOSET=data;
                IOSET=nWR;
                IOSET=nCS;
        }
void Write_Command(unsigned char Reg, unsigned char data)
        {
                Write_Register(Reg);
                Write_Parameter(data);
        }
```

Recommended Initial Code and Sub Function

Note：1.For 80 series CPU interface.

2. For 8bits DDRAM transfer.