OLED DISPLAY MODULE

Application Notes

PRODUCT NUMBER	DD-2864BY-2A with EVK board

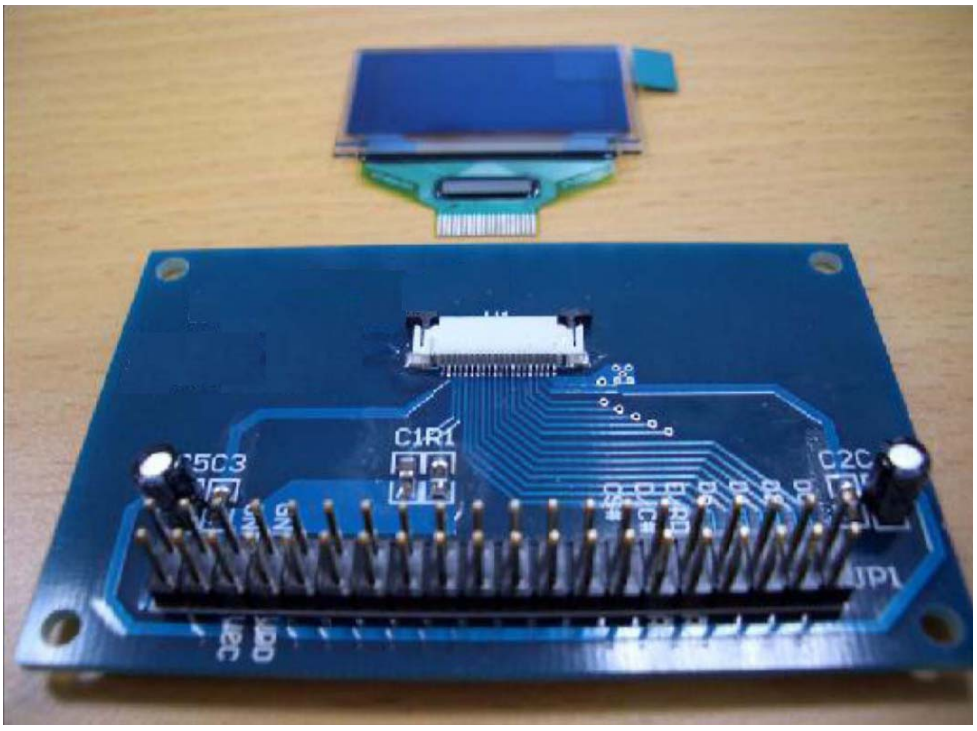


TABLE OF CONTENTS

1 EVK SCHEMATIC 4

2 SYMBOL DEFINITION 5

3 TIMING CHARACTERISTICS 6

4 CONNECTION BETWEEN OLED AND EVK 7

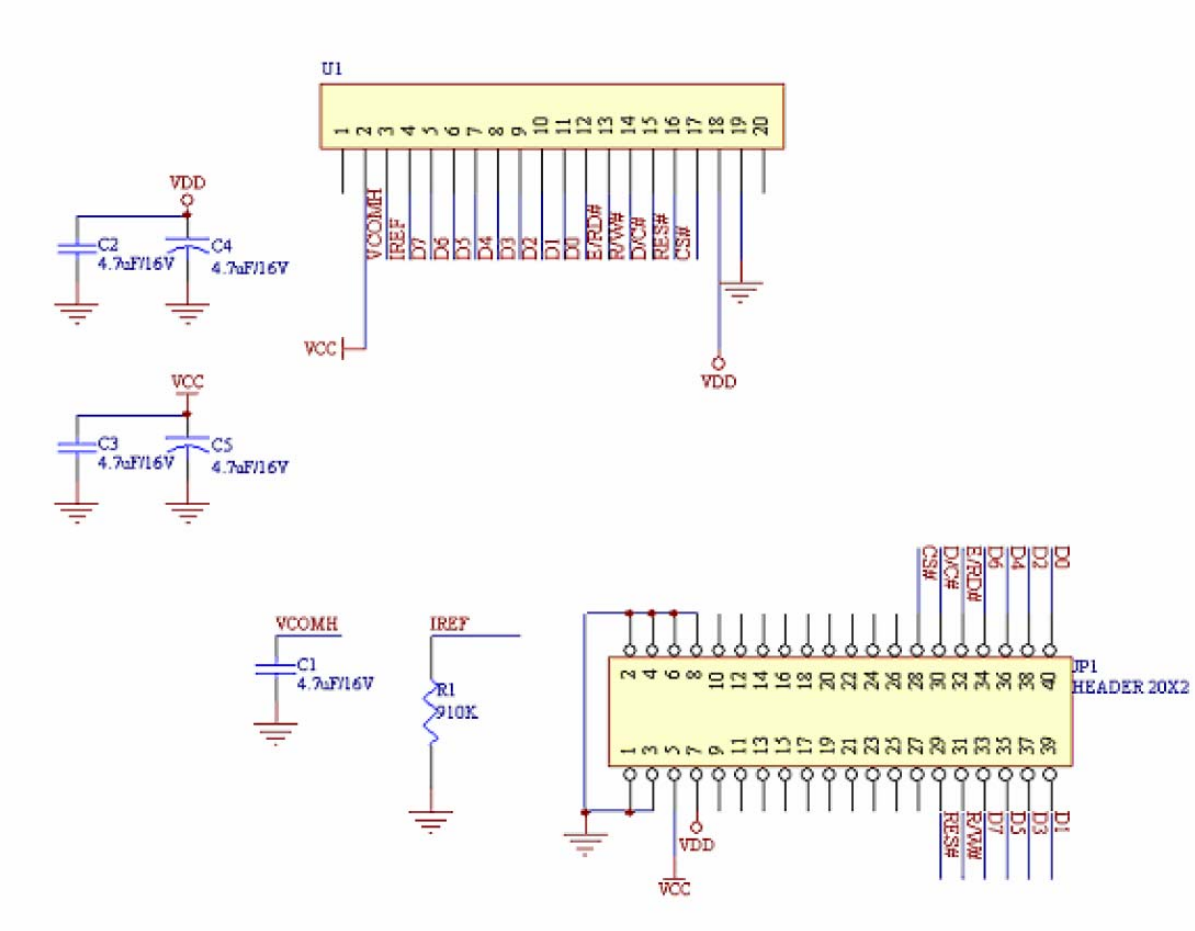
5 HOW TO USE THE DD-2864BY-2A 9

5.1 RECOMMENDED INITIAL CODE 10

REVISION RECORD

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

1 EVK Schematic



2 Symbol Definition

Note: The EVK is hard wired to 8080 Parallel interface.

D0-D7：These pins are 8-bit bi-directional data bus to be connected to the MCU's data bus.

RD#：This pin is MCU interface input. When connecting to an 8080-microprocessor, this pin receives the Read (RD#) signal. Data read operation is initiated when this pin is pulled low and the chip is selected.

WR#：This pin is MCU interface input. When 8080 interface mode is selected, this pin will be the Write (WR#) input. Data write operation is initiated when this pin is pulled low and the chip is selected.

D/C#：This pin is Data/Command control pin. When the pin is pulled high, the data at D7-D0 is treated as display data. When the pin is pulled low, the data at D7-D0 will be transferred to the command register. For detail relationship to MCU interface signals, please refer to the timing characteristics diagrams.

RES#：This pin is reset signal input. When the pin is low, initialization of the chip is executed.

CS#：This pin is the chip select input. The chip is enabled for MCU communication only when CS is pulled low.

VCC：This is the most positive voltage supply pin of the chip.

VDD：Power Supply pin for logic operation of the driver.

VSS：This is a ground pin. It also acts as a reference for the logic pins and the OLED driving voltages. It must be connected to external ground.

VCOMH：This is an input pin for the voltage output high level for COM signals. A capacitor should be connected between this pin and VSS.

NC：Dummy pad. Do not group or short NC pins together.

3 Timing characteristics

VDD = 2.4 to 3.5V, TA= -40to 85°C

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	300	-	-	ns
t_{AS}	Address Setup Time	0	-	-	ns
t_{AH}	Address Hold Time	0	-	-	ns
t_{DSW}	Write Data Setup Time	40	-	-	ns
t_{DHW}	Write Data Hold Time	15	-	-	ns
t_{DHR}	Read Data Hold Time	20	-	-	ns
t_{OH}	Output Disable Time	-	-	70	ns
t_{ACC}	Access Time	-	-	140	ns
PW_{CSL}	Chip Select Low Pulse Width (read) Chip Select Low Pulse Width (write)	120 60	-	-	ns
PW_{CSH}	Chip Select High Pulse Width (read) Chip Select High Pulse Width (write)	60 60	-	-	ns
t_R	Rise Time	-	-	15	ns
t_F	Fall Time	-	-	15	ns

Table 2 8080-Series MPU Parallel Interface Timing Characteristics

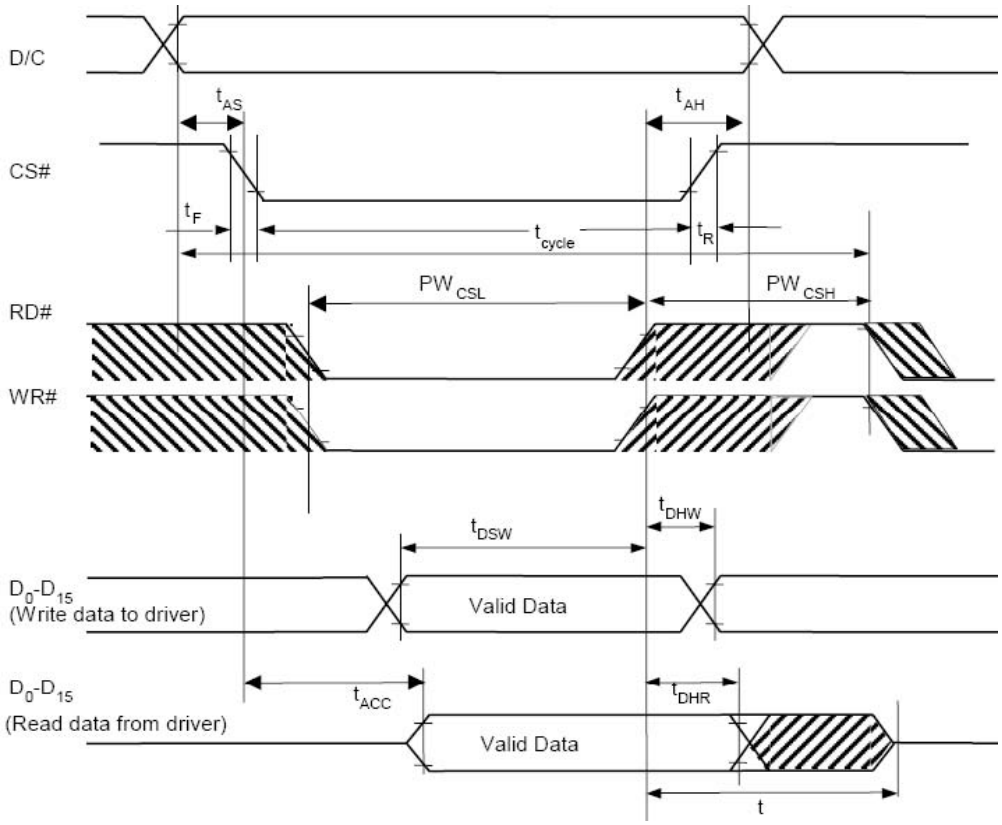


Figure 1 8080-series MPU parallel interface characteristics

4 Connection Between OLED and EVK

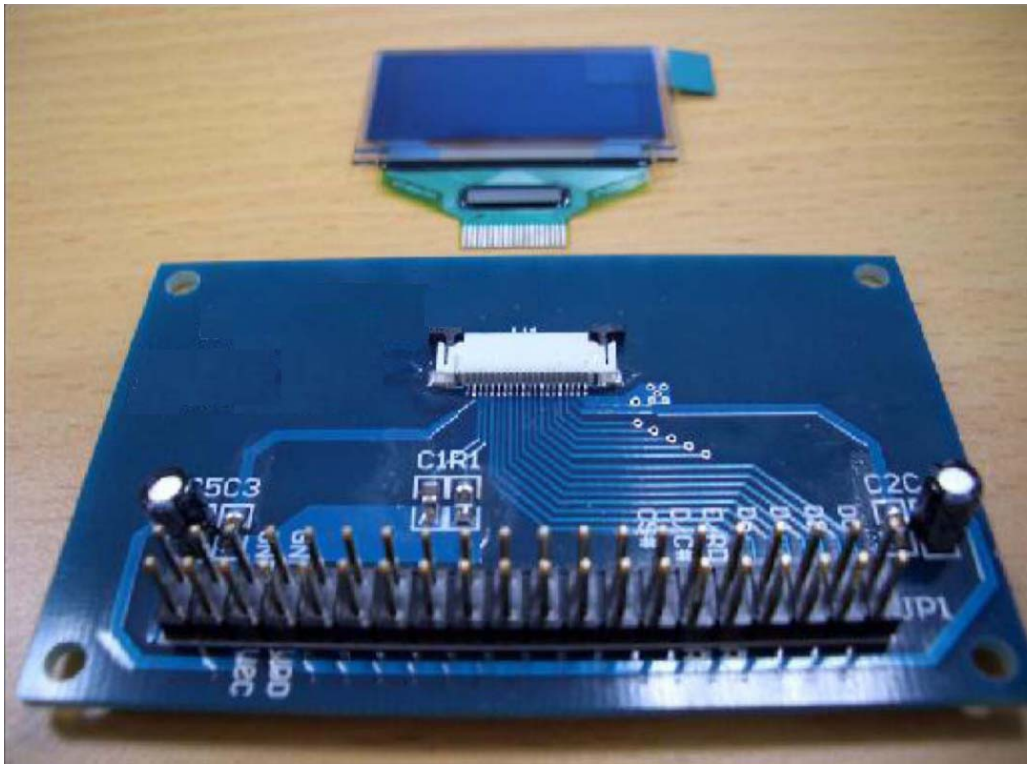


Figure 2 EVK PCB and DD-2864BY-2A Module

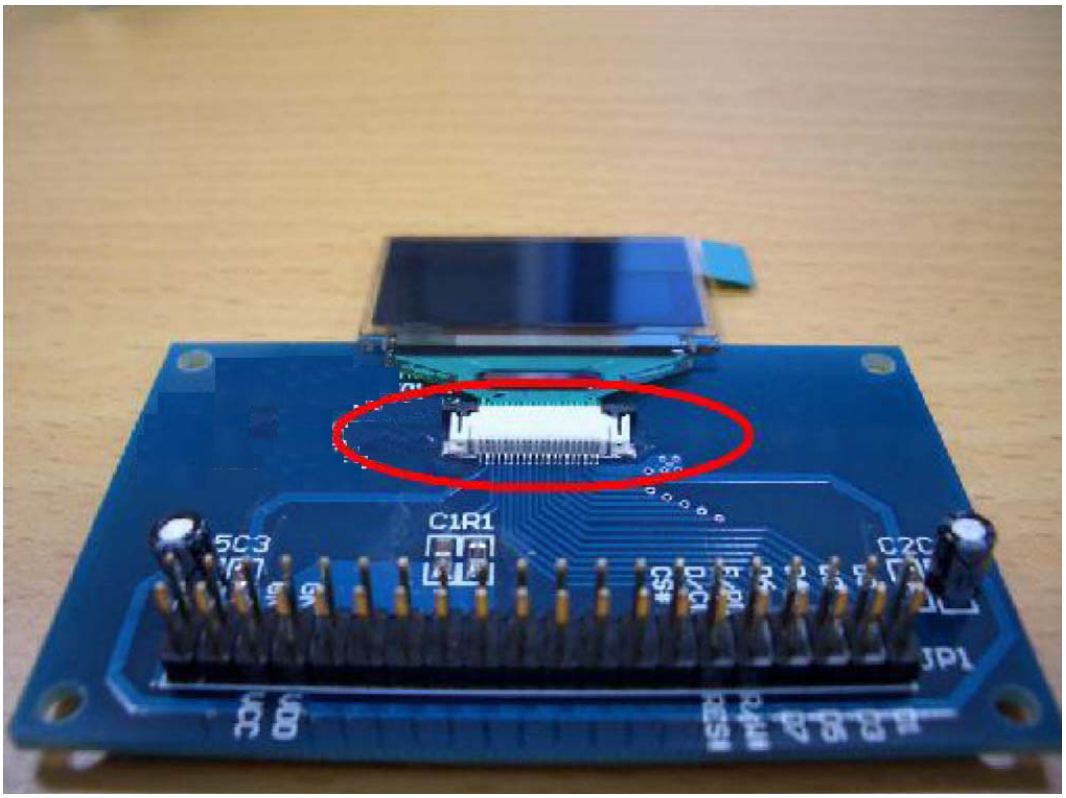


Figure 3 the DD-2864BY-2A and EVK assembled (Top view)

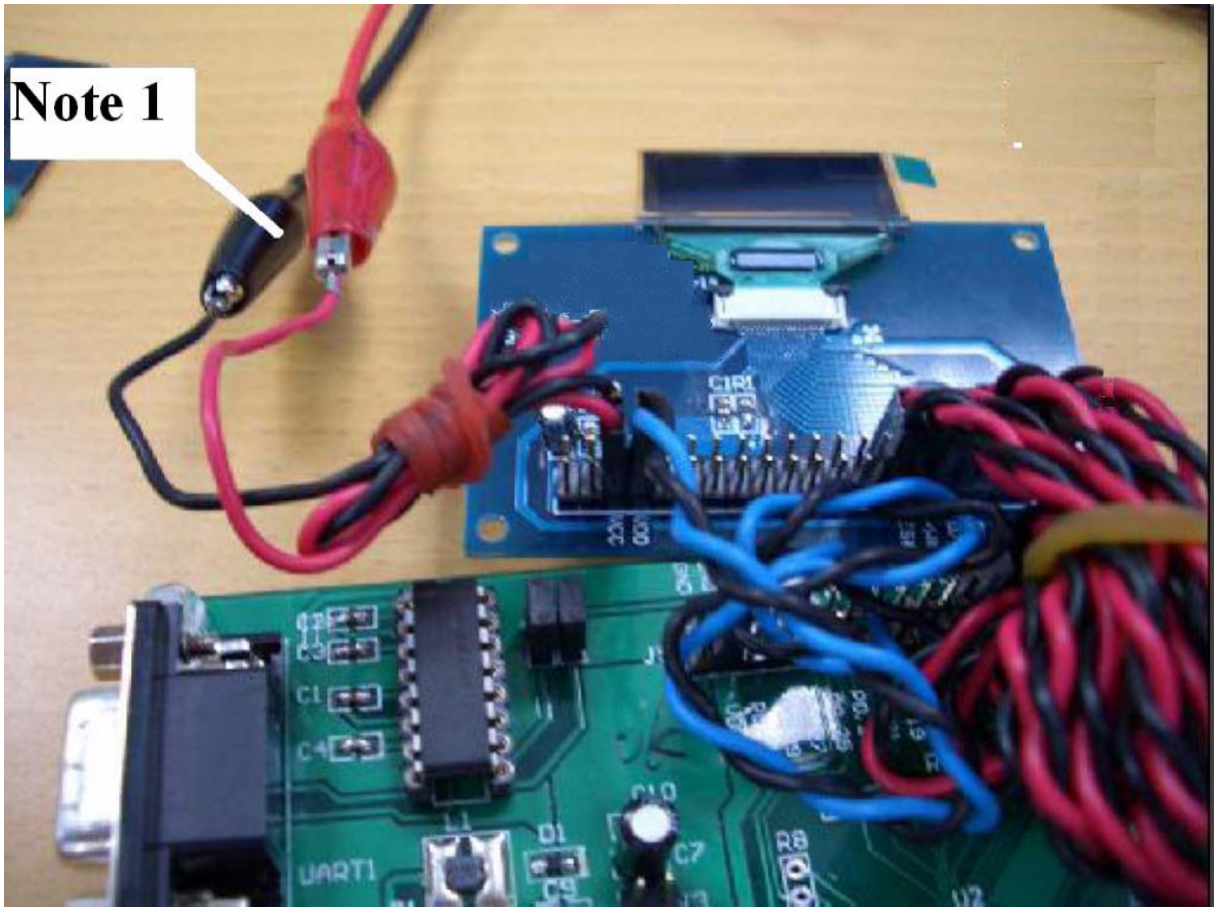
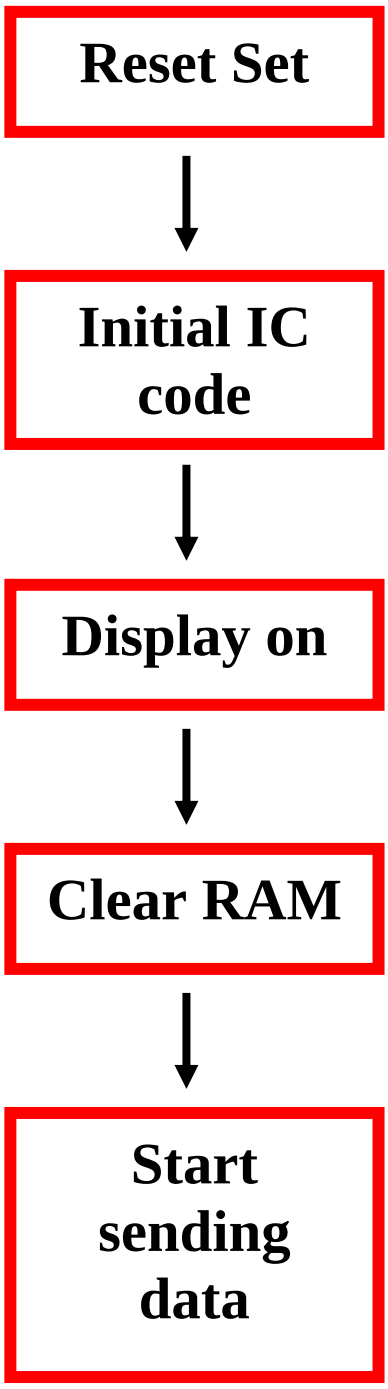


Figure 4 control MCU (not supplied) connected with EVK

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

5 How to use the DD-2864BY-2A



5.1 Recommended Initial code

```
void initial(void)
{
BS1=1;                  // use 8080 interface
BS2=1;
DC=0;
WR=0;
RD=0;
CS=0;
RES=0;
delay(100);
RES=1;
write_c(0xAE);          //display off
write_c(0x81);          //set contrast
write_c(0xff);          //max current
write_c(0xa8);          //set duty
write_c(0x3F);          //duty 63
write_c(0xA0);          //Set Segment Re-map
write_c(0xd3);          //display offset
write_c(0x00);          //set 00
write_c(0x40);          //Start line
write_c(0xC8);          //Set COM Output Scan Direction
write_c(0xda);          //Set COM pins hardware configuration
write_c(0x12);          //Set COM pins hardware configuration
write_c(0xD9);          //Set precharge
write_c(0xf1);          //precharge=fh , discharge=1h
write_c(0xDB);          //Set VcomH
write_c(0x49);          //VcomH=73
write_c(0xA4);          //Normal Mode
write_c(0xA6);          //No Inverse
write_c(0xAF);          //display on
}
void write_c(unsigned char ins_c)
{
DC=0;
CS=0;
RD=1;                   /*tell system only write*/
WR=0;
d_bus=ins_c;
WR=1;
CS=1;
DC=1;
}
```

```
void write_d(unsigned char ins_d)
{
DC=1;
CS=0;
RD=1;                                   /*tell system only write*/
WR=0;
d_bus=ins_d;
CS=1;
WR=1;
DC=1;
}
void delay(int count)
{
int i,j;
for(i=0;i<=count;i++)
for(j=0;j<=1000;j++)
;
}
```

* write_c= Write Command , write_d= Write Data