悠景科技股份有限公司

UG-2896GDEAF01

Application note

Evaluation Kit User Guide

Engineer: Danny. Chen

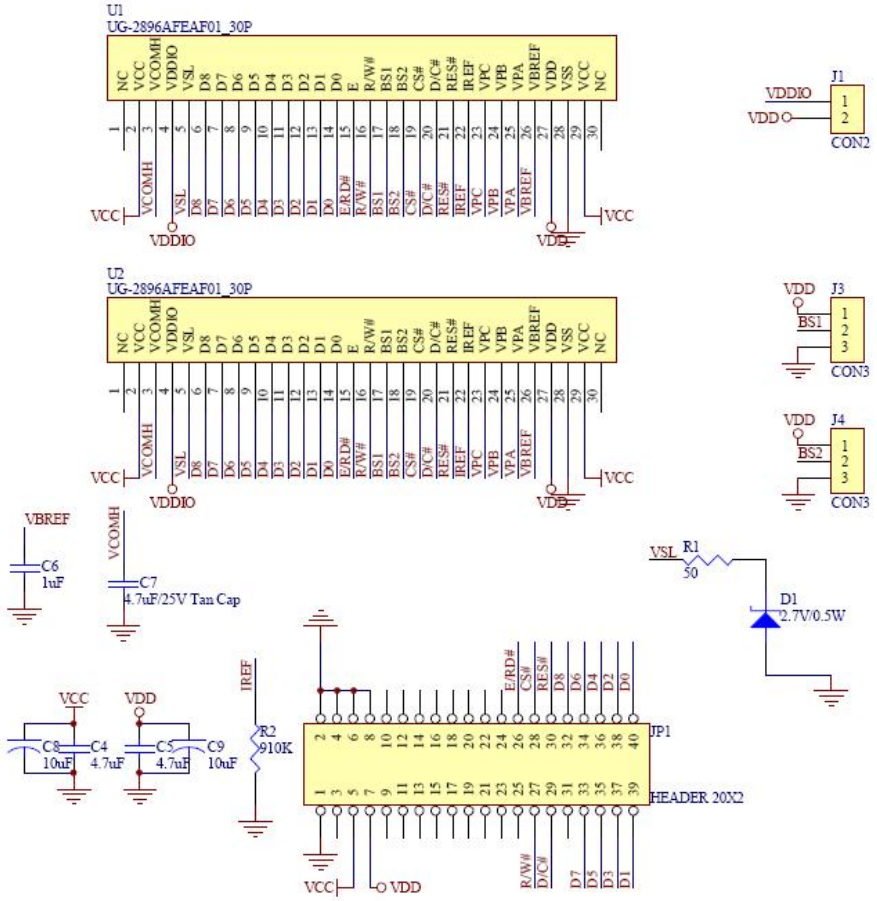
Email: Danny_chen@univision.com.tw

Version: V1.0

REVISION HISTORY

Date	Page	Contents	Version
2006/05/08		Preliminary	Preliminary
2006/5/25		Update power on sequence	V1.0

EVK Schematic



※C1、C2 and C3 please remove.

Symbol define

D0-D8：These pins are 9-bit bi-directional data bus to be connected to the MCU's data bus.

BS1,BS2,BS3：These input pins are used to configure MCU interface selection by appropriate logic setting, which is described in the following table. User can fixed these pins by jump (J3, J4). Unlike BS0~2 are can control by hardware, **BS3 is control by software command 0xA0 only**.**BS0 has already fixed to GND.**

	6800-parallel interface (8 bit)	8080-parallel interface (8 bit)	6800-parallel interface (16 bit)	8080-parallel interface (16 bit)	Serial interface
BS0	0	0	1	1	0
BS1	0	1	0	1	0
BS2	1	1	1	1	0
BS3	0	0	0	0	0

	6800-parallel interface (9 bit)	8080-parallel interface (9 bit)	6800-parallel interface (18 bit)	8080-parallel interface (18 bit)
BS0	0	0	1	1
BS1	0	1	0	1
BS2	1	1	1	1
BS3	1	1	1	1

Table 1 – MCU Interface Selection Setting

E/RD#：This pin is MCU interface input. When interfacing to a 6800-series microprocessor, this pin will be used as the Enable (E) signal. Read/write operation is initiated when this pin is pulled high and the chip is selected.

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. When serial interface is selected, this pin E(RD) must be connected to VSS.

R/W#：This pin is MCU interface input. When interfacing to a 6800-series microprocessor, this pin will be used as Read/Write (R/W) selection input. Read mode will be carried out when this pin is pulled high and write mode when low.

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. When serial interface is selected, this pin R/W must be connected to VSS.

D/C#：This pin is Data/Command control pin. When the pin is pulled high, the data at D0-D8 is treated as display data. When the pin is pulled low, the data at D0-D8 will be transferred to the command register. For detail relationship to MCU interface signals, please refer to the timing characteristics diagrams at following pages and datasheet.

RES#：This pin is reset signal input. When the pin is low, initialization of the chip is executed. The timing char please see the figure 4.

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.

GND：Power supply ground.

VDDIO：This pin is a power supply pin of I/O buffer. It should be connected to V_{DD} or external source. All I/O signal should have VIH reference to VDDIO. When I/O signal pins (BS01, 2, M/S, CLS, D0-D17, control signals.) pull high, they should be connected to VDDIO. In the EVK PCB board which is use a jump (J1) connect to VDD (Default). Customer can put out the jump and input the 1.8V through this jump.

VP A,B,C：These pins are the driving voltages for OLED driving segment pins SA0-SA131, SB0-SB131 and SC0-SC131 respectively. They can be supplied externally or internally generated by VP circuit. When internal VP is used, V_{PA}, V_{PB}, V_{PC} pins should be left open. But for application use to avoid noise from system, that we suggest add a capacitor (4.7uF) to GND.

VSL：This is segment voltage reference pin. For reduce power consumption, we suggest add a resistor and Zenor diode between this pin and GND.

V_{COMH}：This pin is the input pin for the voltage output high level for COM signals. It can be supplied externally or internally. When V_{COMH} is generated internally, a capacitor should be connected between this pin and GND.

VB_{REF}：This pin is the internal voltage reference of booster circuit. A stabilization capacitor, typ. 1uF, should be connected to GND. We suggest use 4.7uF will be more stable.

VDD = 2.4 to 3.5V, TA = -40 to 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 6800-Series MPU Parallel Interface Timing Characteristics

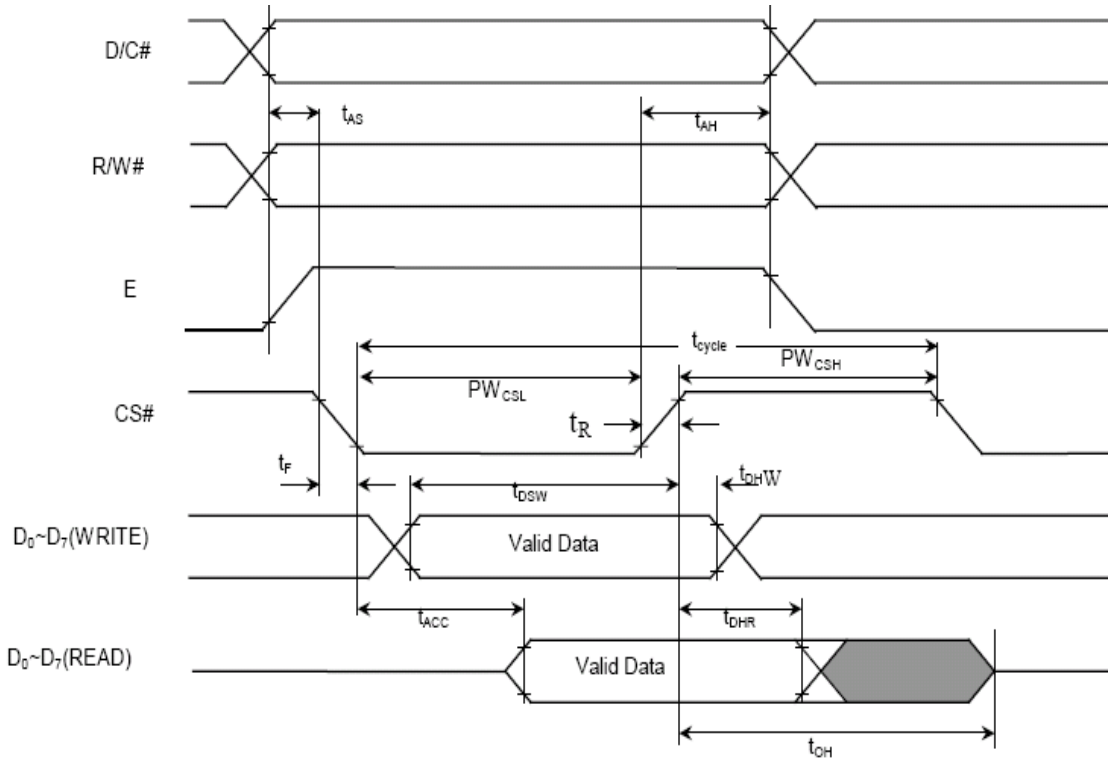


Figure 1 6800-series MPU parallel interface characteristics

Note：When 9 bit used: D_0 ~ D_8 instead.

VDD = 2.4 to 3.5V, TA = -40 to 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 3 8080-Series MPU Parallel Interface Timing Characteristics

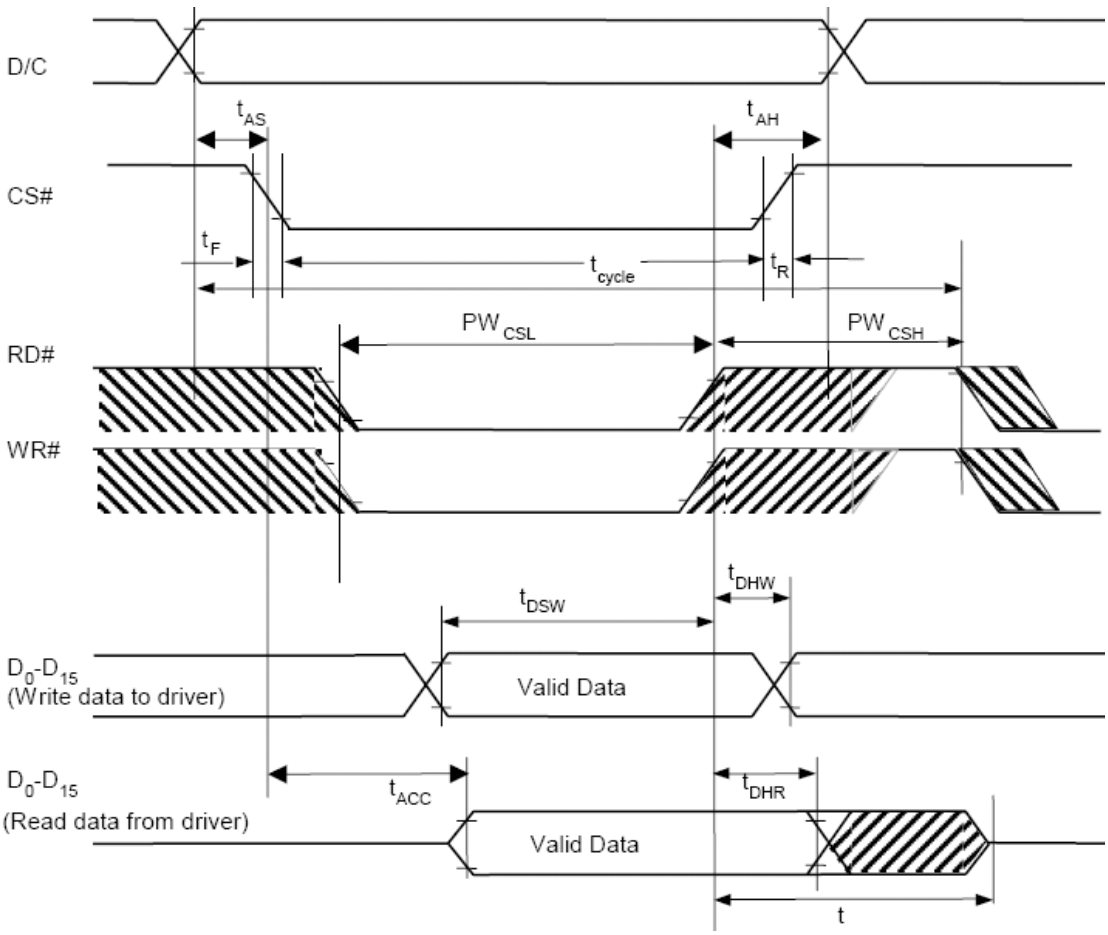


Figure 2 8080-series MPU parallel interface characteristics

Note：When 9 bit used: D_0 ~ D_8 instead.

VDD = 2.4 to 3.5V, TA = -40 to 85°C

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	250	-	-	ns
t_{AS}	Address Setup Time	150	-	-	ns
t_{AH}	Address Hold Time	150	-	-	ns
t_{CSS}	Chip Select Setup Time	120	-	-	ns
t_{CSH}	Chip Select Hold Time	60	-	-	ns
t_{DSW}	Write Data Setup Time	100	-	-	ns
t_{DHW}	Write Data Hold Time	100	-	-	ns
t_{CLKL}	Clock Low Time	100	-	-	ns
t_{CLKH}	Clock High Time	100	-	-	ns
t_R	Rise Time	-	-	15	ns
t_F	Fall Time	-	-	15	ns

Table 4 Serial Interface Timing Characteristics

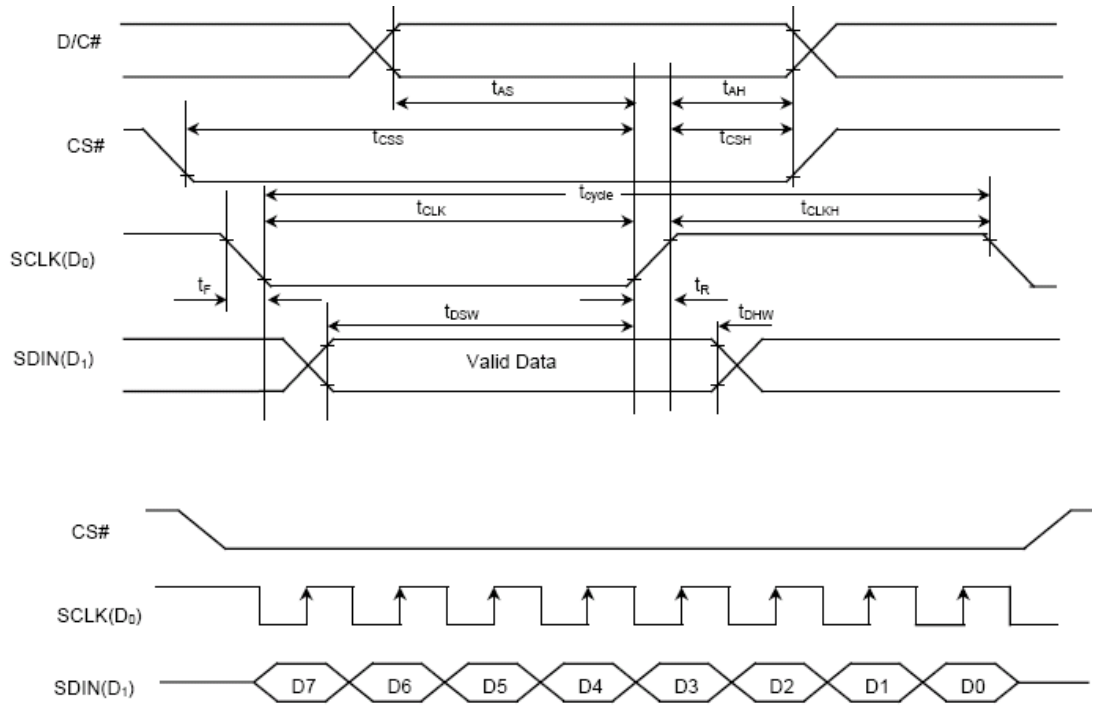


Figure 3 Serial interface characteristics

Note： More detail description about serial interface, please reference the datasheet.

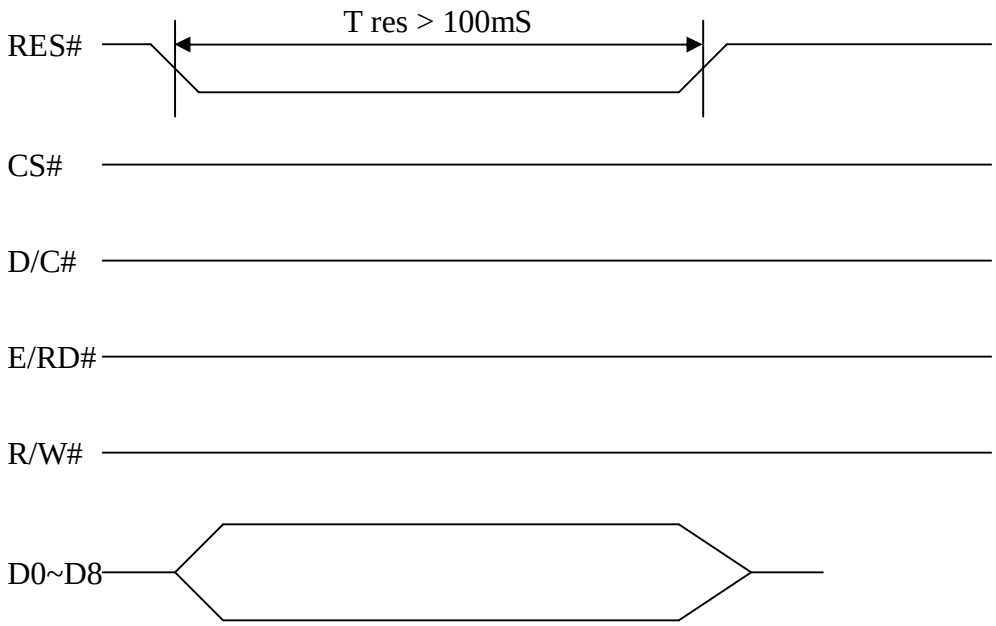


Figure 4 Reset timing char

To keep system stabile, we suggest the reset time should be longer than 100mS.

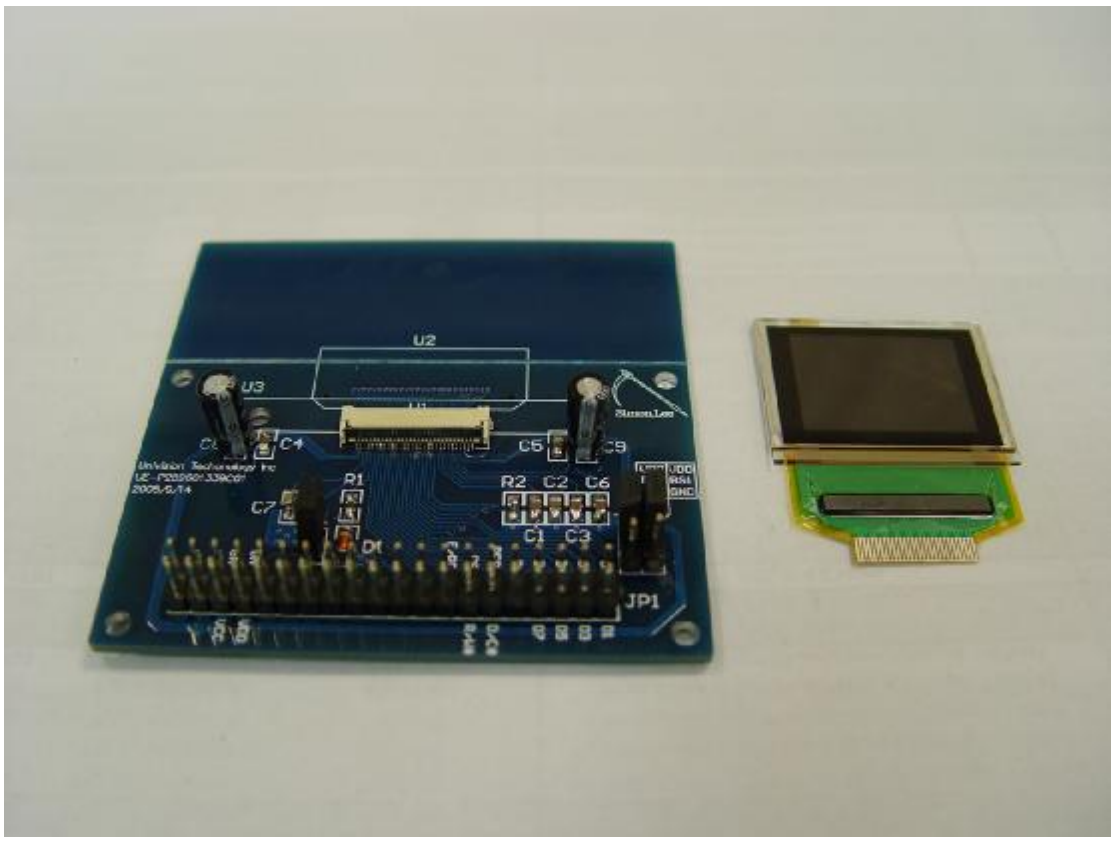


Figure 5 EVK PCB and OLED Module

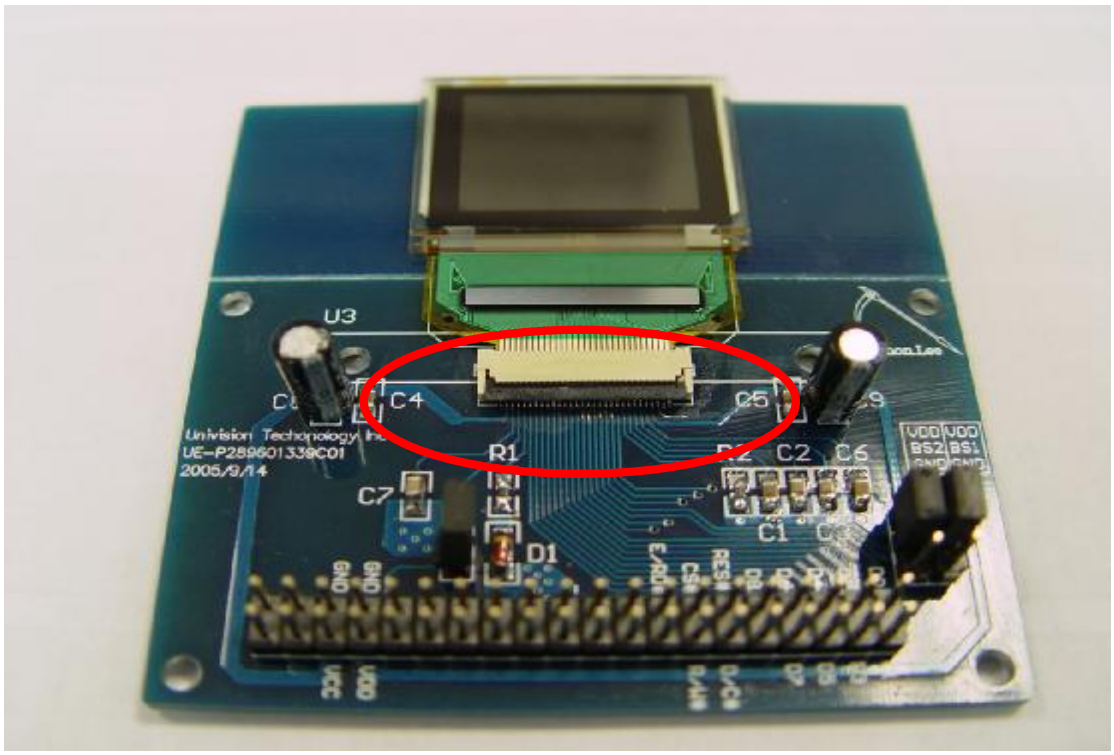


Figure 6 the module and EVK assembled (Top view)

Because the package of UG- 2896GDEAF01 is COF, that the connect pads are on the top of the module, and the connector which on the EVK PCB board is double size connect type. So when assemble the module with EVK. The module must face up first and plug into the connector. When finished assembled the module and EVK, then push the locking pad to lock the module. See the figure 6.

When finished assembled the module and EVK. User can use leading wire to connect EVK with customer's system. The example shows as figure 7.

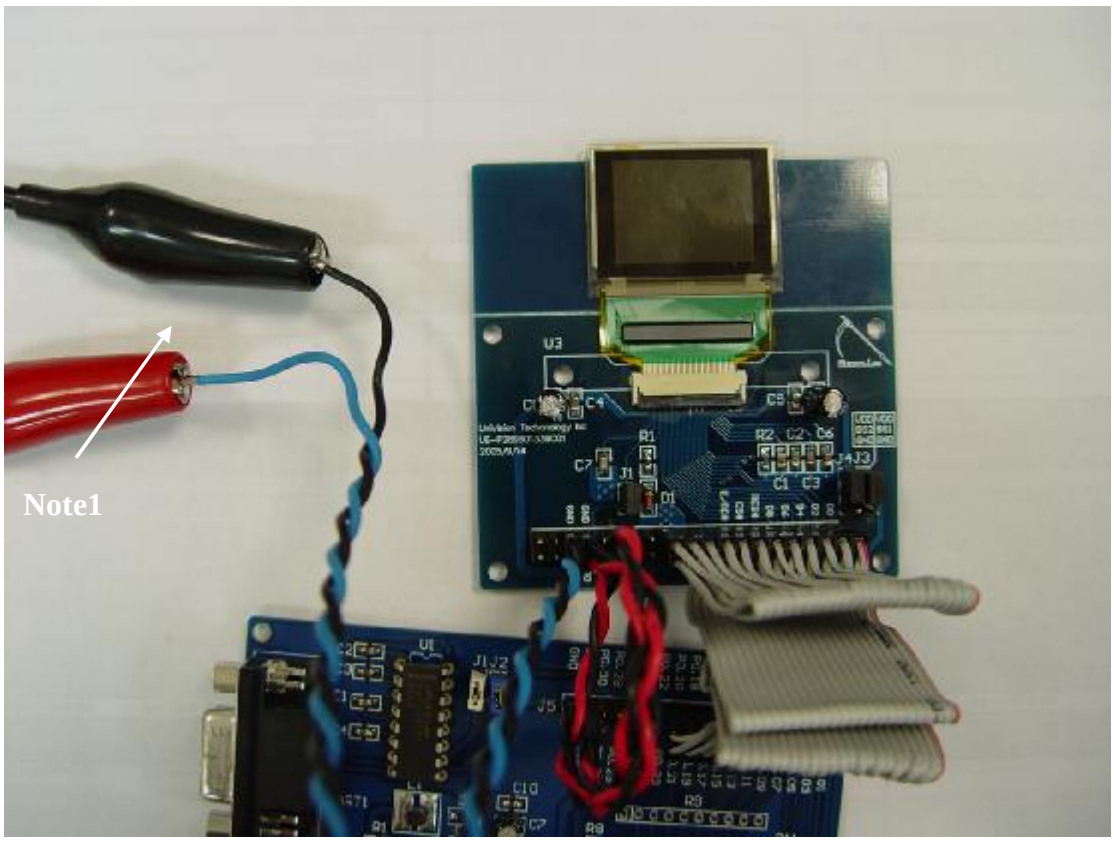


Figure 7 control MCU connect with EVK

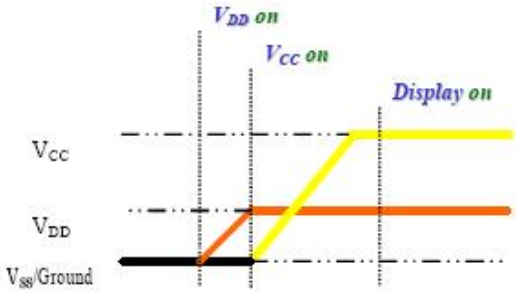
Note 1：It is the external most positive voltage supply. In this sample is connected to power supply.

Module power on sequence：

To protect OLED panel and extend the panel life time, the driver IC power up/down routine should include a delay period between high voltage and low voltage power sources during turn on/off. Such that panel has enough time to charge up or discharge before/after operation.

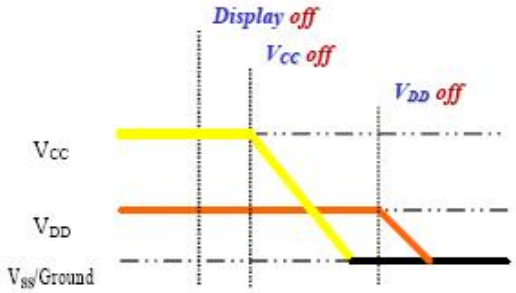
4.2.1 Power up Sequence:

1. Power up V_{DD}
2. Send Display off command
3. Driver IC Initial Setting
4. Clear Screen
5. Power up V_{DDH}
6. Delay 100ms
   (when V_{DD} is stable)
7. Send Display on command

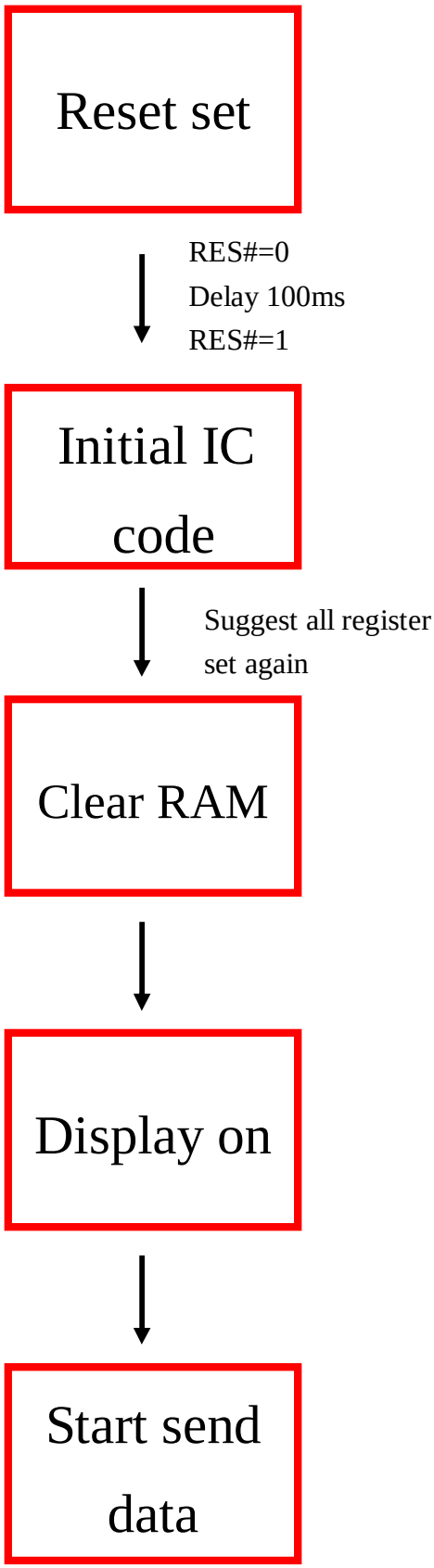


4.2.2 Power down Sequence:

1. Send Display off command
2. Power down V_{DDH}
3. Delay 100ms
   (when V_{DDH} is reach 0 and panel is completely discharges)
4. Power down V_{DD}



Module initial set sequence：



RD recommends Initial Code:

```
void Initial_ic(void)
{
      IOCLR=0xffffffff;                 //data=0
      IOSET=bE_RD;
      IOCLR=bD_C|bR_W|bCS;
      Reset_SSD1339();

      write_c(0xa0);          // Set Re-map / Color Depth
      write_d(0xb4);          // 262K 8bit      R->G->B
      write_c(0xa1);          // Set display start line
      write_d(0x00);          // 00h start
      write_c(0xa2);          // Set display offset
      write_d(0x60);          // 60h start
      write_c(0xa6);          // Normal display
      write_c(0xad);          // Set Master Configuration
      write_d(0x8e);          // DC-DC off & external VcomH voltage & external    pre-charge voltage
      write_c(0xb0);          // Power saving mode
      write_d(0x05);
      write_c(0xb1);          // Set pre & dis_charge
      write_d(0x2c);          // pre=1h dis=1h
      write_c(0xb3);          // clock & frequency
      write_d(0xf0);          // clock=Divser+1    frequency=fh
      write_c(0xbb);          // Set pre-charge voltage of color A B C
      write_d(0x1c);          // color A
      write_d(0x1c);          // color B
      write_d(0x1c);          // color C
      write_c(0xbe);          // Set VcomH
      write_d(0x3f);          //
      write_c(0xc1);          // Set contrast current for A B C
      write_d(0xa0);          // Color A
      write_d(0x96);          // Color B
      write_d(0xb4);          // Color C
      write_c(0xc7);          // Set master contrast
      write_d(0x09);          // no change
      write_c(0xca);          // Duty
      write_d(0x5f);          // 95+1
      write_c (0xb8)         //Set gamma table
      write_d (1);            //PW1
      write_f (3);           //PW3
      write_d (5);            //PW5
      write_d (7);            //PW7
      write_d (11);           //PW9
      write_d (16);           //PW11
      write_d (22);           //PW13
      write_d (26);           //PW15
      write_d (31);           //PW17
      write_d (36);           //PW19
      write_d (41);           //PW21
      write_d (46);           //PW23
      write_d (51);           //PW25
      write_d (56);           //PW27
      write_d (61);           //PW29
      write_d (66);           //PW31
      write_d (71);           //PW33
      write_d (76);           //PW35
      write_d (81);           //PW37
      write_d (86);           //PW39
```

```
        write_d (90);           //PW41
        write_d (94);           //PW43
        write_d (98);           //PW45
        write_d (101);          //PW47
        write_d (104);          //PW49
        write_d (107);          //PW51
        write_d (110);          //PW53
        write_d (113);          //PW55
        write_d (116);          //PW57
        write_d (119);          //PW59
        write_d (122);          //PW61
        write_d (125);          //PW63
        write_c(0xaf);          // Display on

}

void Reset_SSD1339(void)
{
        IOCLR=bRES;
        Delay_1ms(100);
        IOSET=bRES;
}

void write_c(unsigned char out_command)
{
        IOCLR=bD_C;
        IOCLR=bCS;
        IOCLR=bR_W;
        IOCLR=0x000000ff;
        IOSET=out_command;
        IOSET=bR_W;
        IOSET=bCS;
        IOSET=bD_C;
}


void write_d(unsigned char out_data)
{
        IOSET=bD_C;
        IOCLR=bCS;
        IOCLR=bR_W;
        IOCLR=0x000000ff;
        IOSET=out_data;
        IOSET=bR_W;
        IOSET=bCS;

}

void Delay_1ms(int Cycle)
{
unsigned int i,k;
        for (i=0 ;i<Cycle;i++)
                for(k=0;k<0x5fff;k++);
}
```

＊write_c= Write Command , write_d= Write Data