

Rio OLED Module Application Instruction

Application Note

AN016

For any discrepancies between this document and the product specification, the product specifications take precedence.

Introduction

The objective of the application note is to serve as a design and application guideline for RIO OLED display module with STV8105 driver. The information presented covers device electrical description and operation, performance, programming parameters and interfacing to a module controller. Reference to this document is the STV8105 OLED driver datasheet.

Electrical & Mechanical Description

A single chip CMOS OLED driver with controller provide 16 level of grayscale operation to the Rio display, which consists of 256 x 64 pixels. The built-in controller also allows for software operation of the display module with its command set and data transfer through a parallel or serial communication interface. Each of 256 column drivers is capable of providing the maximum source current of 800 μ A to each column. The display is passive matrix and operated at a duty cycle of 1/64. Each row driver is capable of sinking 110mA. For details please refer to STV8105 datasheet.

Figure 1 is a block diagram of Rio module. It consists of a glass cell and a COF flex which is attached with Driver IC STV8105 within it. A 24 pins contact pad is provided at the end of the COF circuit for interfacing with the controller. The module is configured to use internal oscillator and an external OLED driver power (VPP, VCOL and VROW).

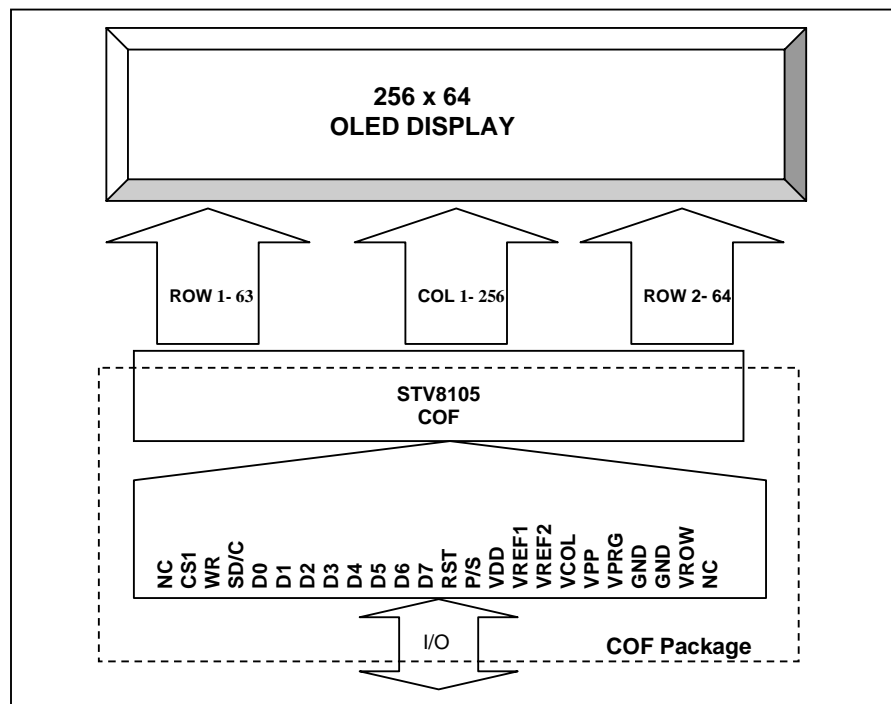


Figure 1: Rio Block Diagram

Basic Specifications

Display Specifications

- 1) Display Mode : Passive Matrix
- 2) Display Color : Monochrome
- 3) Drive Duty : 1/64 Duty

Mechanical Specifications

- 1) Number of Pixels : 256 x 64
- 2) Panel Size : 91.04 x 29.24 x 2.20
- 3) Active Area : 79.33 x 19.81
- 4) Pixel Pitch : 0.31 x 0.31
- 5) Pixel Size : 0.28 x 0.28

Driver Connection Diagram

Recommended Component				
Part name	Part number	Description	Case	Qty
C1, C2, C3, C4	F921C105MBA	Chip Cap 10uF 16V	1210	1
R1	RMC1/16-682FTP	Chip Resistor 6.8K ohm	0603	1

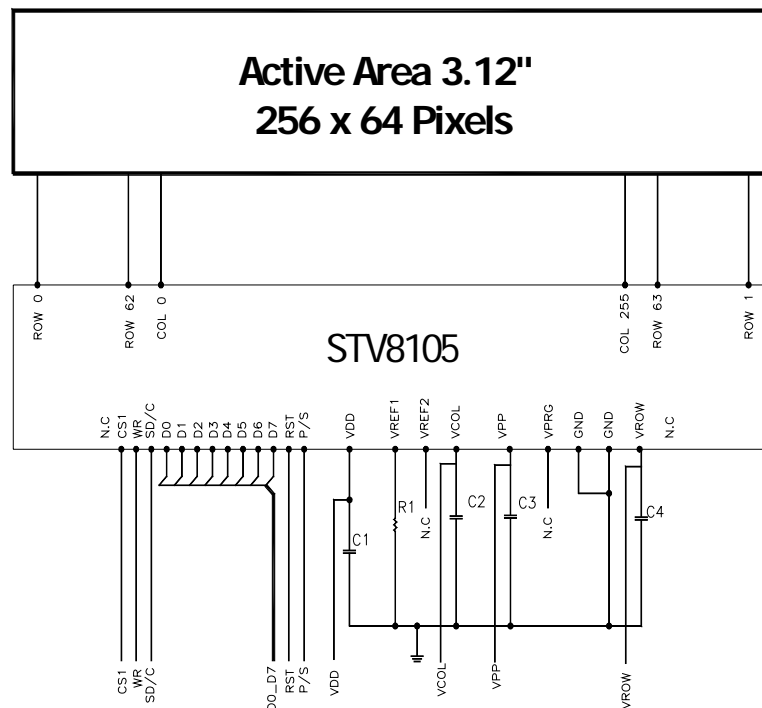


Figure 2: Driver Connection Diagram

In order to understand the way to connect STV8105, the input and output pins are classified according to their functionalities.

- Power supply pins
 - GND, VDD, VPP, VCOL, VROW,
- MCU interface pins
 - D0 – D7, CS1, SD/C, RST, WR, P/S, VREF1
- Row and column output pins
 - Row 0 – Row 63, Column 0 – Column 255

Recommended Components				
Part Name	Part Number	Description	Case	Qty
C1,C2,C3,C4,C5,C6	F92IC106MBA	Chip Cap 10uF 16V	1210	6
C7,C8,C9,C10,C11,C12,C13	EMK107BJ104KA-T	Chip Cap 0.1uF	0603	7
C14,C15	F498-55	Chip Cap 33pF	0603	2
R1,R2	RC1608J103CS	Chip Resistor 10K ohm	0603	2
R3	RMC 1/16-5903FTP	Chip Resistor 590K ohm	0603	1
R4,R5	RC1608J914CS	Chip Resistor 910K ohm	0603	2
R6	RC1608F1503CS	Chip Resistor 150K ohm	0603	1
R7	RC1608J102CS	Chip Resistor 1K ohm	0603	1
R8,R9,R10	RC1608J472CS	Chip Resistor 4.7K ohm	0603	3
R11	RC1608J683CS	Chip Resistor 68K ohm	0603	1
R12	RC1608J563CS	Chip Resistor 56K ohm	0603	1
L1,L2,L3,L4	A918CY-220M=P3	Inductor 22 uH		4
D1,D2,D3,D4	MBRM12LT10SCT_ND	Schottky Diode		4
U1	LT1944EMS#PBF	Voltage Regulator IC		1
U2	LT3471EDD#PBF	Voltage Regulator IC		1

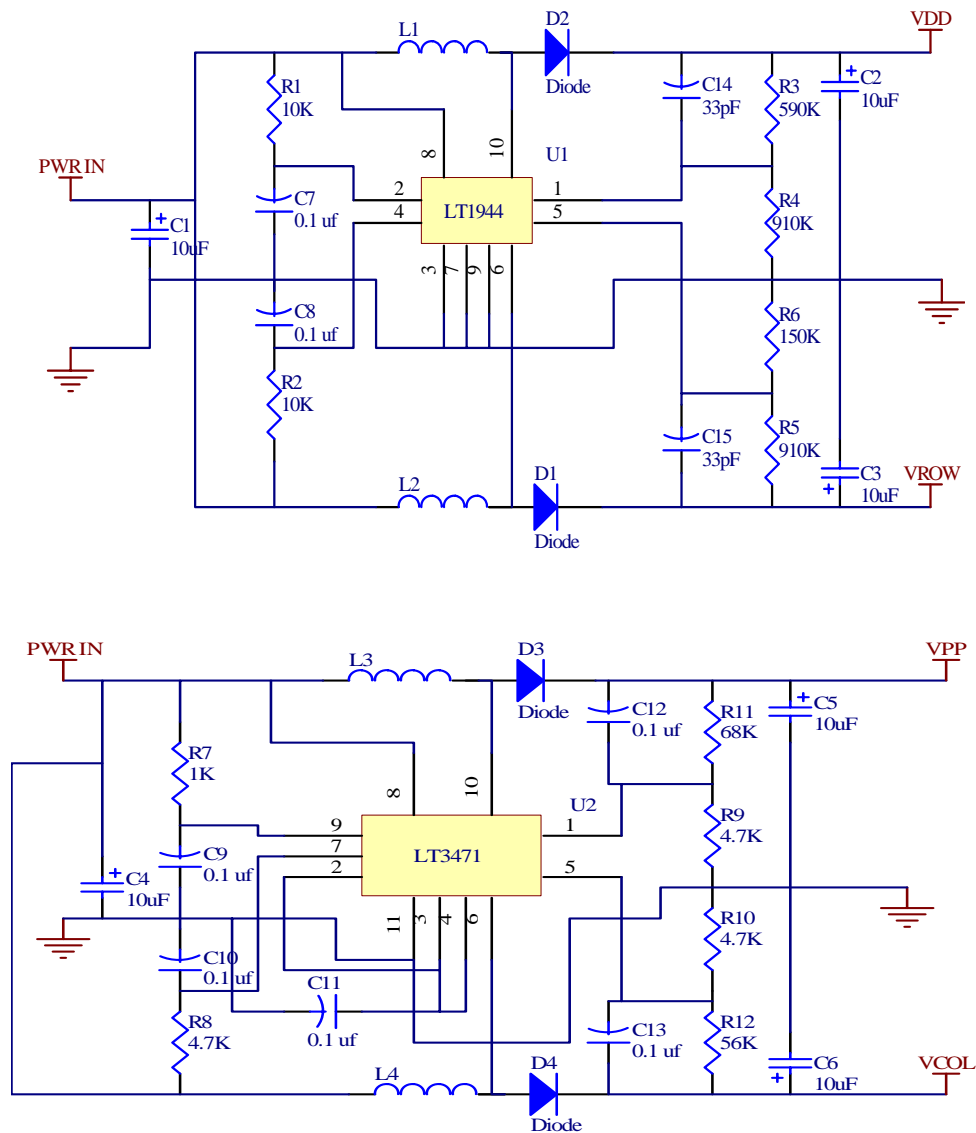


Figure 3: Application Interface Schematic for Voltage Supply (VDD, VROW, VPP & VCOL)

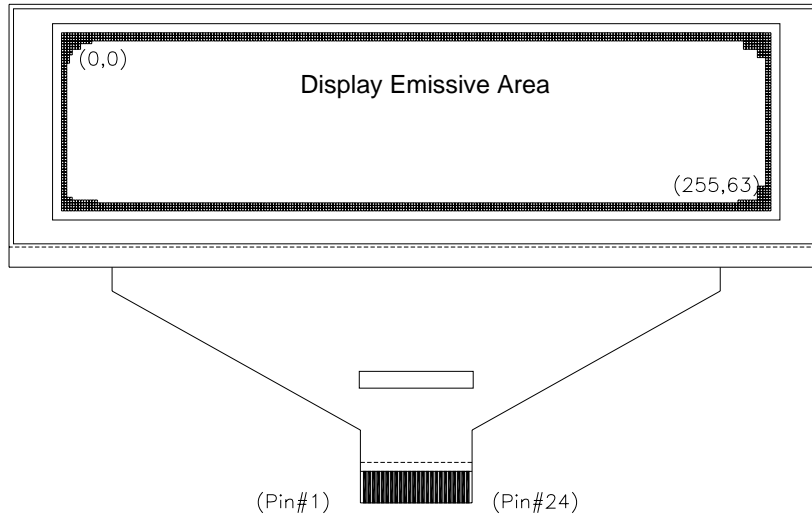


Figure 4: Connector Pad Orientation

The interleave configuration is used in the design of Rio glass cell. Sixty-four common drivers are physically tied to the OLED glass cell leaving 8 un-connected drivers. **Figure 4** depicts the orientation of interface connector pad. **Figure 5** shows the driver connection scheme of Rio module.

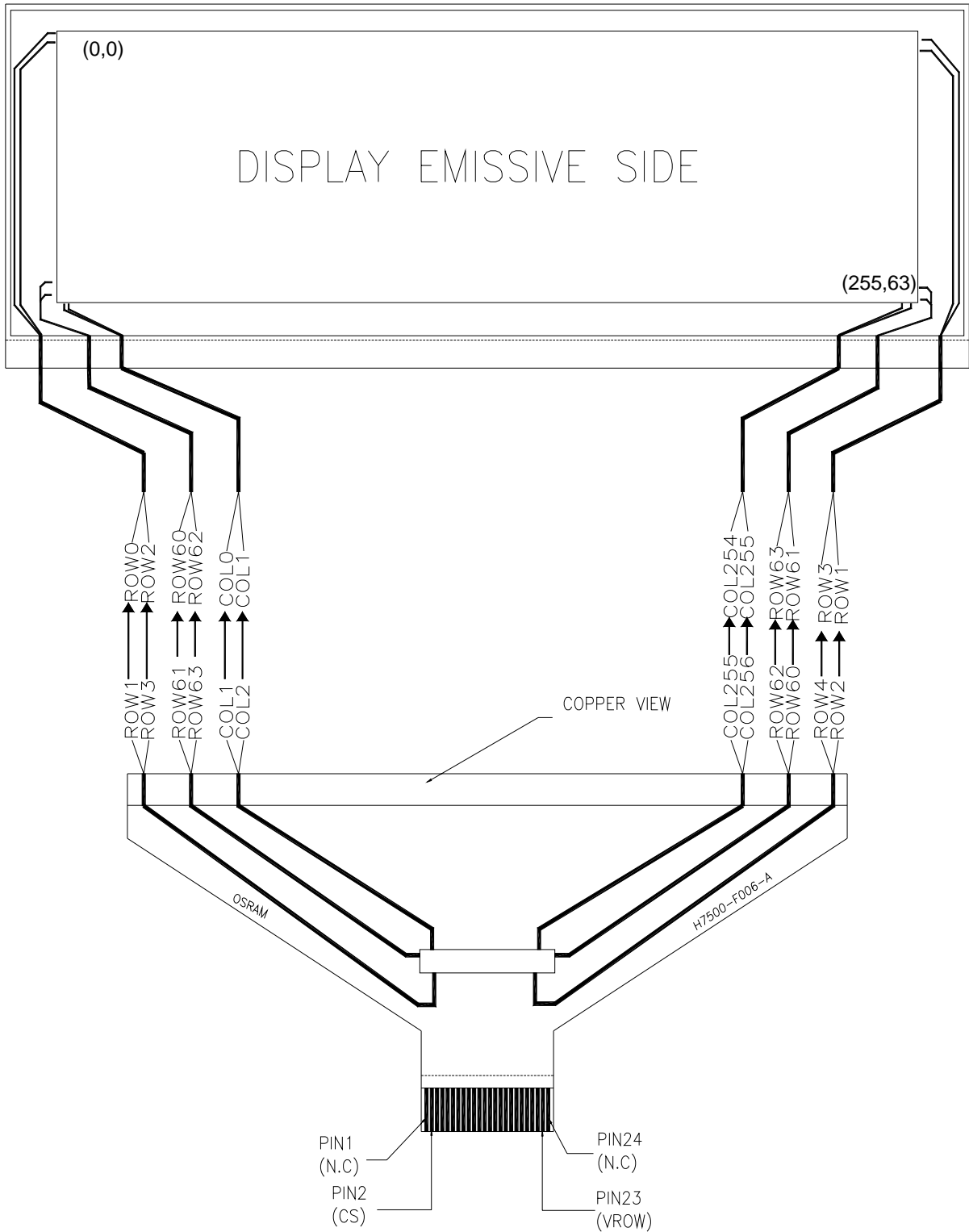


Figure 5: Rio Module Driver Connection

Communications

The detailed pin out of the 24 pin pad is shown in **Table 1**. Rio module is configured for an 8-bit parallel or SPI interface. The user can choose parallel or SPI interface through the P/S pin (pin #14). **Table 2** showed the available interface setting for Rio module. Detailed communication timing diagram is available in STV8105 data sheet.

Pin Number	Symbol	I/O	Description
1, 24	N.C	-	<i>Reserved Pin</i> These pins are reserved for compatible and flexible design
2	CS1	I	<i>Chip Select 1 Input</i> This pin is Master Device Chip Select. The chip is enabled for MCU communication only when CS1 is pulled low.
3	WR	I	<i>Write</i> This pin should pulled low to enable write mode.
4	SD/C	I	<i>Data/ Command Control</i> This pin is Data/Command Control pin. When this pin is pulled HIGH, the input at D0~D7 is treated as Display Data. When this pin is pulled LOW, the input at D0~D7 will be transferred to the command register.
5~12	D0-D7	I	<i>Host Data Input/Output Bus</i> These pins are 8-bit bi-directional data bus to be connected to the microprocessor's data bus. When serial mode is selected, D6 will be the serial clock input (SCLK) and D7 will be the serial data input (SDATA).
13	RST	I	<i>System Reset Input</i> This pin reset signal input. When the pin is pulled low, initialization of the chip is executed.
14	P/S	I	<i>Communicating Protocol Select</i> When this pin is pulled high, Parallel Interface is selected. When this pin is pulled low, Serial Interface is selected.
15	VDD	Supply	<i>Power Supply For Logic Circuit</i> This is Analog/Digital low-voltage controller supply. It must be connected to external source.
16~17	VREF1 – VREF2	I/O	<i>Reference Voltage</i> These pins are reference voltage pins. A resistor should be connected to each of the pins to Gnd.
18	VCOL	Supply	<i>Column Supply Voltage</i> This is odd and even column supply voltage. It must be connected to external source.
19	VPP	Supply	<i>Column Driver Power Supply</i> . This is odd and even column power supply voltage. It must be connected to external source.
20	VPRG	Supply	<i>OTP Power Supply</i> This is non-volatile OTP memory program power supply. It must be connected to external source.
21~22	GND	Supply	<i>Analog/Digital Ground</i> . This is ground pin. It also acts as a reference for the logic pin, the OEL driving voltage and analog circuit. It must be connected to external source.
23	VROW	Supply	<i>Row Driver Supply</i> This is odd and even row supply voltage. It must be connected to external source.

Table 1: Pin Description

	Parallel Interface	Serial Interface
P/S	1	0

Table 2: Communication Configuration

Power Connections

There are two settings to operate Rio module.

Option 1

The logic power, VDD is 3.0 volts, the OLED driver power, VPP is 12 volts, the column supply, VCOL is 10 volts, the row supply, VROW is 7 volts and the ground is common for both logic and analog. With this setup, *Rio Elegance Yellow* will get 75 cd/m² and *Rio Spring Green* will get 100 cd/m².

Option 2

The logic power, VDD is 3.0 volts, the OLED driver power, VPP is 15 volts, the column supply, VCOL is 12 volts, the row supply, VROW is 9 volts and the ground is common for both logic and analog. With this setup, *Rio Elegance Yellow* will get 100 cd/m² and *Rio Spring Green* will get 150 cd/m².

Power On and Power Off Sequence

To protect the OLED panel and extend the panel life time, the driver IC power up and down routine should include a delay period between high voltage and low voltage power sources turn on/off.

Power On Sequence

- a. Power On VDD
- b. Send Display Off command
- c. Power On VPP
- d. Power On VCOL
- e. Power On VROW
- f. Delay 30ms (when VPP, VCOL, VROW are stable)
- g. Send Display On command

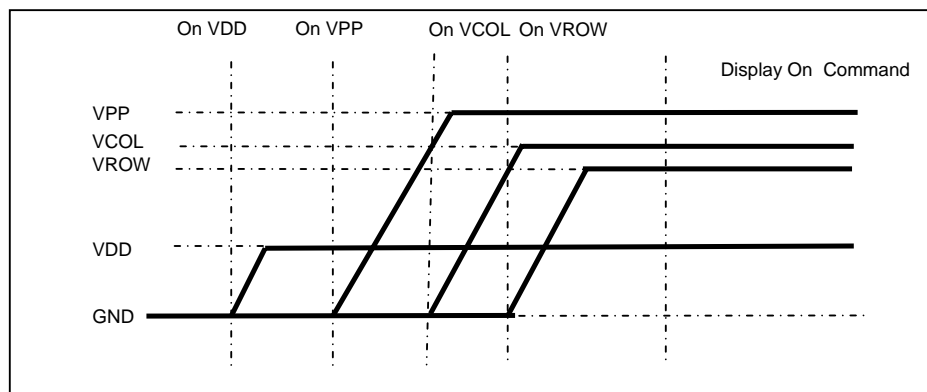


Figure 6: Power On Diagram

Power Off Sequence

- a. Send Display Off command
- b. Power Off VROW
- c. Power Off VCOL
- d. Power Off VPP
- e. Delay 30ms (when VPP, VCOL, VROW are reach 0 and the panel is completely discharges)
- f. Power Off the VDD

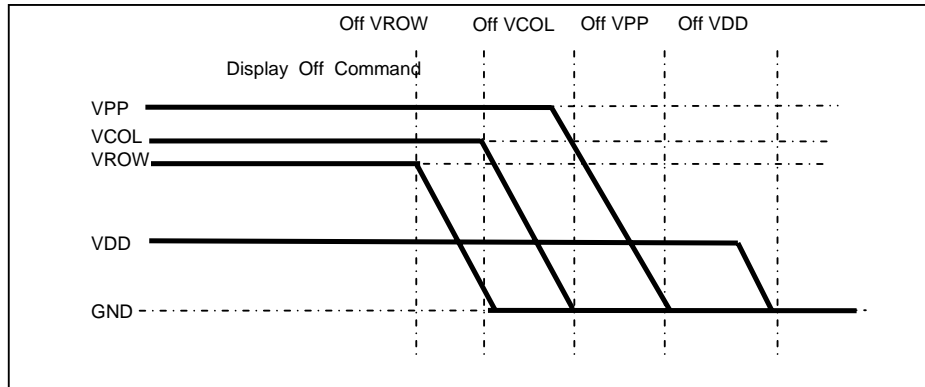


Figure 7: Power Off Diagram

Packaging

The driver IC is mounted on the COF flex using ACF process. The COF flex is attached to the glass cell using the ACF process. Sealer is applied to the edge of the glass and COF to protect against peeling force due to folding of the flex.

Programming

Refer to the section "List Of Commands Ordered By Command Code" of the STV8105 data sheet for more information.

Initialization

Rio module requires certain commands to be executed upon a power up for its proper operation. Failure to execute these commands may lead to shortened display lifetime, poor image quality, and incorrect image display. **Figure 8** is an example of initialization software flow of Rio module. Under different condition and application, the values of the register may not be the same as shown in **Table 3**. A sample C program is included for reference in the last page.

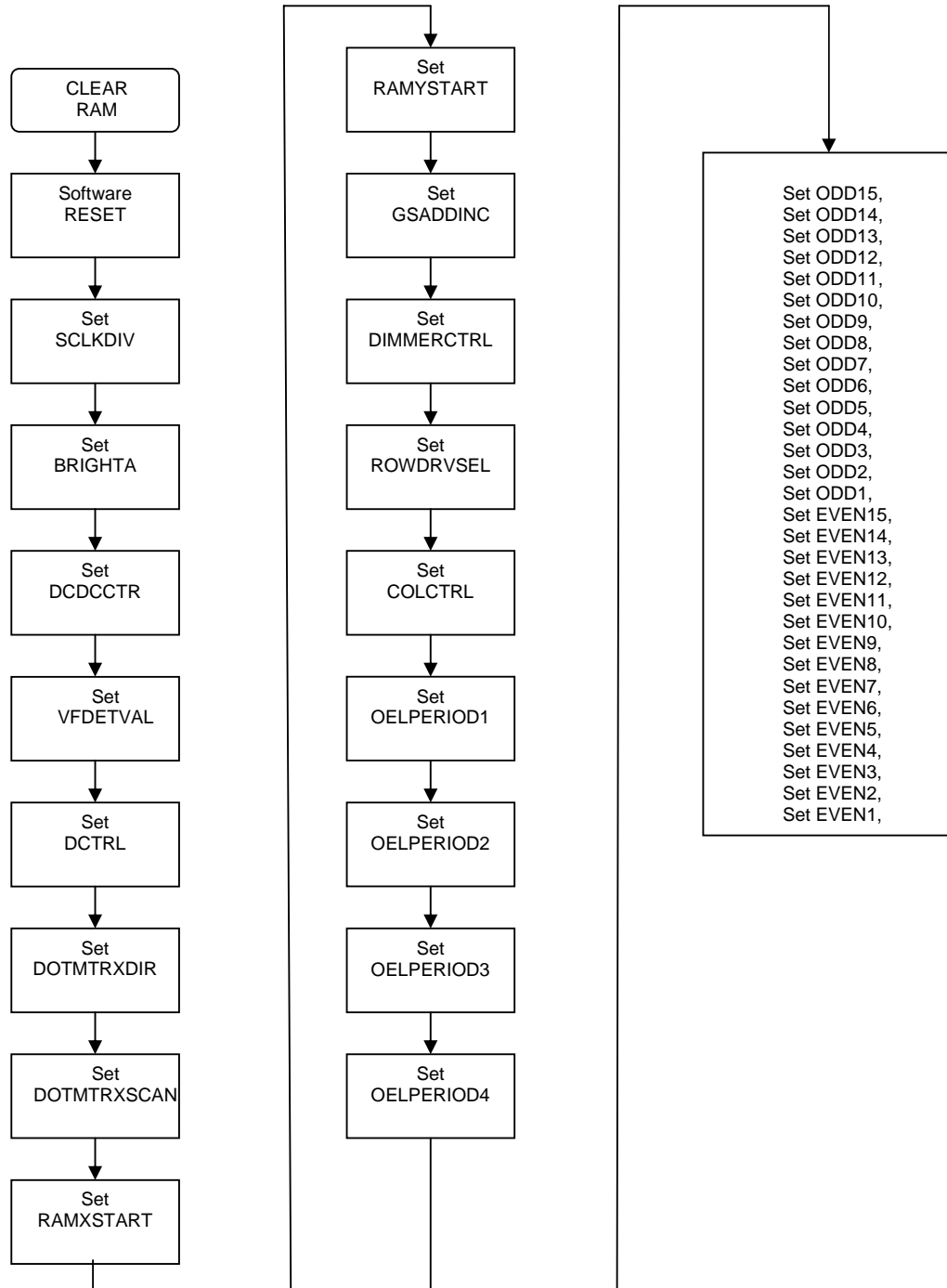


Figure 8: Software Initialization Flow Chart

Table 3: Recommended Initialization Command

Command	Code (H)	Power On Default	Option 1 Initialization (16 Grayscale)	Option 2 Initialization (16 Grayscale)	Option 1 Initialization (Monochrome)	Option 2 Initialization (Monochrome)
Software Reset		F2	F2	F2	F2	F2
SCLK Clock Divide Ratio	00	00	00	00	00	00
Initial Brightness Adj. A	01	00	A0	C9	A0	C9
Initial Brightness Adj. B	02	00	00	00	00	00
DC/DC Converter Control	03	00	00	00	00	00
Selection Of Voltage To Add To VF To Produce VH	06	00	00	00	00	00
Dot Matrix Display Control	10	00	01	01	01	01
Dot Matrix Display Direction	11	00	00	00	00	00
Dot Matrix Scanning Line	12	47	3F	3F	3F	3F
Display RAM X Start Address	13	00	00	00	00	00
Display RAM Y Start Address	14	00	00	00	00	00
Gray Scale & Increment Mode Set	15	00	03	03	83	83
Dimmer Control	16	0F	0B	0A	0B	0A
Row Driver Mode Select	17	02	02	02	02	02
Column Output Control	1A	00	00	00	00	00
Setup Period 1	1B	0F	0F	0F	0F	0F
Setup Period 2	1C	00	81	81	81	81
Setup Period 3	1D	00	81	81	81	81
Setup Period 4	1E	00	89	89	89	89
Odd 15 Level Of Grayscale	1F	FF	Default	Default	-	-
Odd 14 Level Of Grayscale	20	AF	Default	Default	-	-
Odd 13 Level Of Grayscale	21	79	Default	Default	-	-
Odd 12 Level Of Grayscale	22	53	Default	Default	-	-
Odd 11 Level Of Grayscale	23	39	Default	Default	-	-
Odd 10 Level Of Grayscale	24	27	Default	Default	-	-
Odd 9 Level Of Grayscale	25	1A	Default	Default	-	-
Odd 8 Level Of Grayscale	26	12	Default	Default	-	-
Odd 7 Level Of Grayscale	27	0C	Default	Default	-	-
Odd 6 Level Of Grayscale	28	08	Default	Default	-	-
Odd 5 Level Of Grayscale	29	05	Default	Default	-	-
Odd 4 Level Of Grayscale	2A	03	Default	Default	-	-
Odd 3 Level Of Grayscale	2B	02	Default	Default	-	-
Odd 2 Level Of Grayscale	2C	01	Default	Default	-	-
Odd 1 Level Of Grayscale	2D	00	Default	Default	FF	FF
Even 15 Level Of Grayscale	2E	FF	Default	Default	-	-
Even 14 Level Of Grayscale	2F	AF	Default	Default	-	-
Even 13 Level Of Grayscale	30	79	Default	Default	-	-
Even 12 Level Of Grayscale	31	53	Default	Default	-	-
Even 11 Level Of Grayscale	32	39	Default	Default	-	-
Even 10 Level Of Grayscale	33	27	Default	Default	-	-
Even 9 Level Of Grayscale	34	1A	Default	Default	-	-
Even 8 Level Of Grayscale	35	12	Default	Default	-	-
Even 7 Level Of Grayscale	36	0C	Default	Default	-	-
Even 6 Level Of Grayscale	37	08	Default	Default	-	-
Even 5 Level Of Grayscale	38	05	Default	Default	-	-
Even 4 Level Of Grayscale	39	03	Default	Default	-	-
Even 3 Level Of Grayscale	3A	02	Default	Default	-	-
Even 2 Level Of Grayscale	3B	01	Default	Default	-	-
Even 1 Level Of Grayscale	3C	00	Default	Default	-	-

Horizontal and Vertical Image Flip

The display image can be flipped horizontally and vertically by software command.

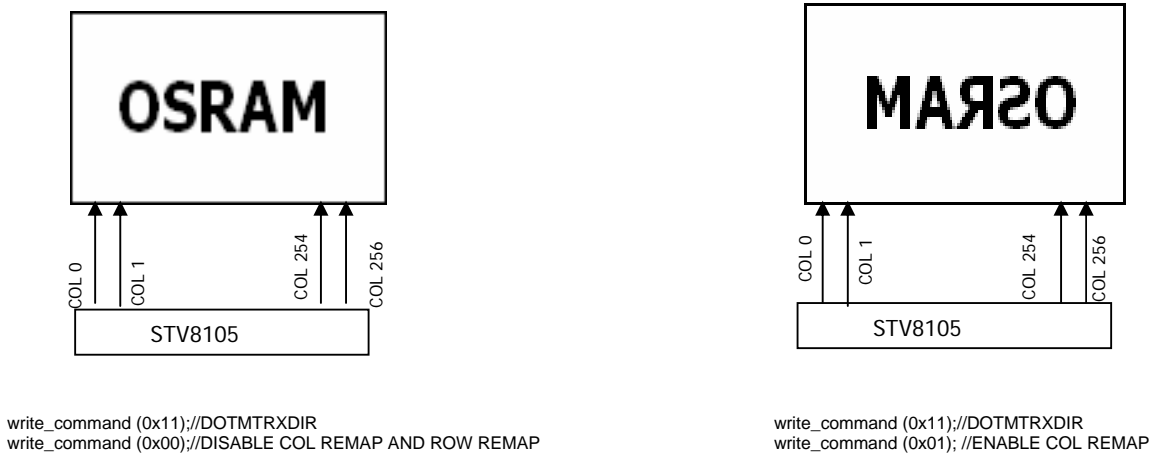


Figure 9. Flipping Image Horizontally

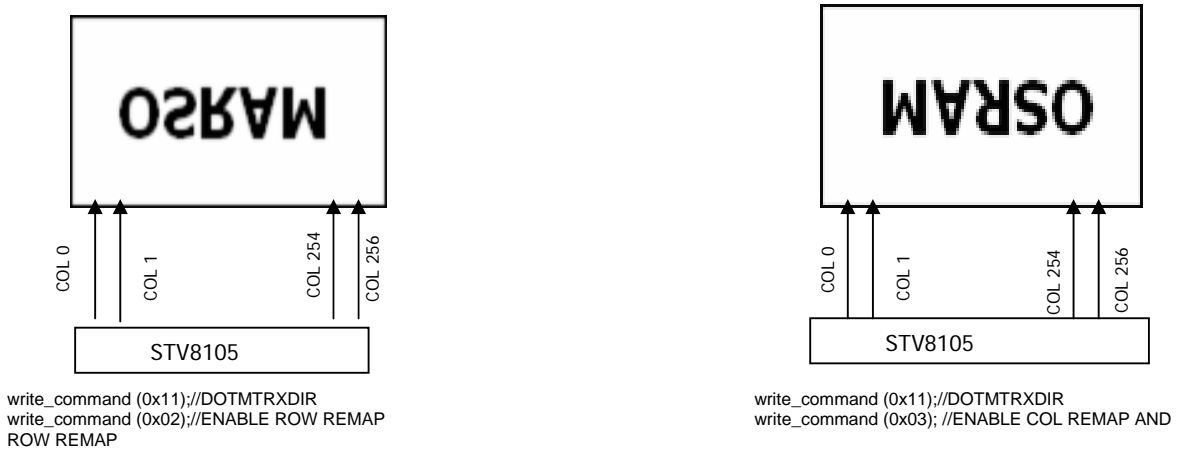


Figure 10: Flipping Image Vertically

Frame Frequency

Frame frequency is set at the maximum rate of 120 Hz. If a lower rate is desired, it can be adjusted to the desirable rate by varying the SCLK Clock Divider Ratio Select (SCLKDIV) command during initialization. Default value of SCLKDIV upon power up is 0 and its range is plotted in blue in **Figure 11**. An operation of the frame frequency below 60 Hz is not recommended because flickering of image becomes visible. A frame frequency can be calculated using the following equation.

$$\text{Frame Frequency} = \frac{2.4\text{MHz} * \text{SCLKDIV}}{\left[1 + \frac{\text{DRIVE PERIOD}}{\text{PERIOD}} + \frac{\text{OEL PERIOD1}}{\text{PERIOD1}} + \frac{\text{OEL PERIOD2}}{\text{PERIOD2}} + \frac{\text{OEL PERIOD3}}{\text{PERIOD3}} + \frac{\text{OEL PERIOD4}}{\text{PERIOD4}} \right] * \left[\frac{\text{DOTMTR XSCAN}}{\text{DIR}} + \text{DOTMTRX DIR} \right]}$$

Example :-

OEL Period1= 0FH = 16 SCLK
 OEL Period2= 81H = 2 SCLK
 OEL Period3= 81H = 2 SCLK
 OEL Period4 = 89H = 10 SCLK
 SCLKDIV = 00H = 1/1
 DOTMTRSCAN = 3FH = 64 ROWS
 DOTMMTRXDIR = 00H = 1 DUMMYLINE

Hence:

$$\text{Frame Frequency} = \frac{2.4\text{MHz} * (1/1)}{(1+256+16+2+3+10)*(64+1)} = 129 \text{ Hz}$$

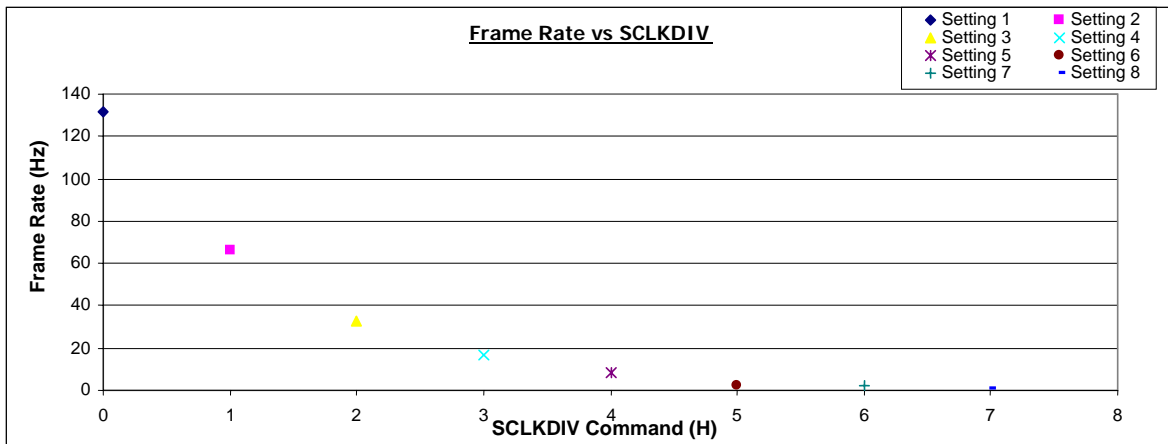


Figure 11: Frame Frequency

Luminance Control and Power Consumption

The luminance of an OLED is proportional to its power consumption. Brightness DAC A (BrightA) and DimmerControl are the command that control the luminance of the display. It controls the amplitude of current drive pulse rather than the pulse width. **Figure 12** showed that the brightness is not increased when BrightA is set from 00H to 7FH. This is because the input data is from the OTP (One Time Program) memory. The brightness only will increase when Brightness DAC A is enabled, starting from 80H until FFH.

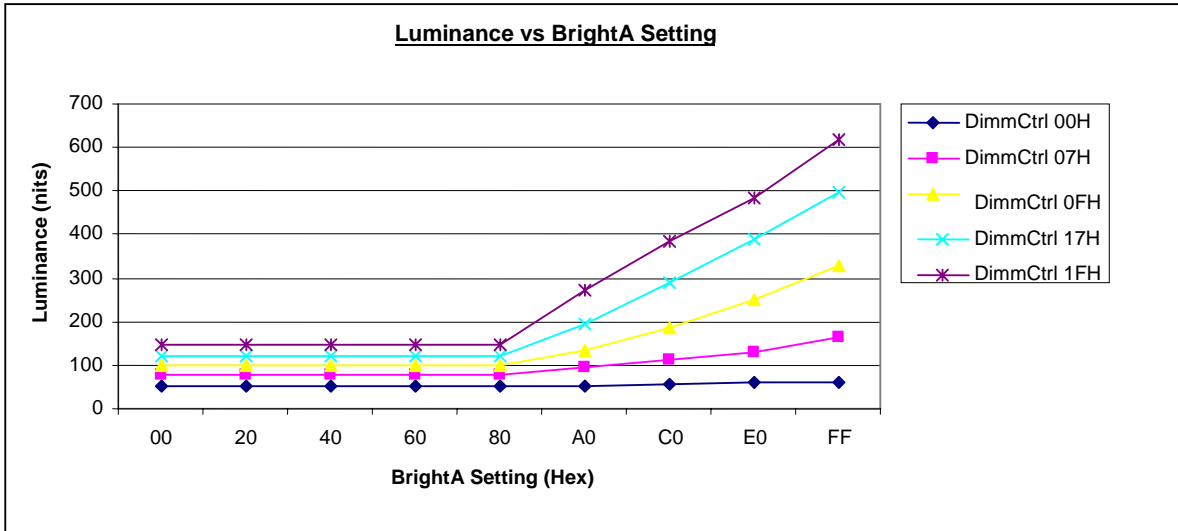


Figure 12: Luminance vs BrightA Setting

Figure 13 will show a linear relationship of VREF1 and Brightness DAC A setting with different voltage supply for VDD.

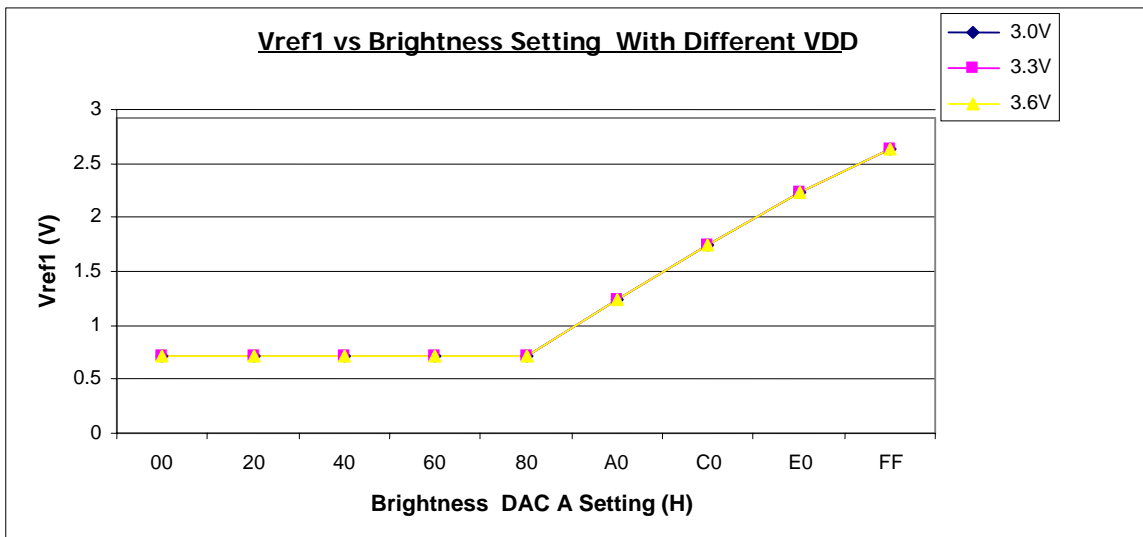


Figure 13: VREF1 vs Brightness Setting With Different VDD

Table 4 below represents dimming setting for Rio Elegance Yellow from 10 cd/m² to 60 cd/m².

Luminance (cd/m ²)	VPP (V)	VCOL (V)	VROW (V)	Software Setting	
				BRIGHTA (H)	DIMMERCTRL (H)
10	9	6	3.5	80	01
20	9	7	4.0	80	03
30	9	7	4.0	80	08
40	12	8	5.0	80	08
50	12	8	5.0	90	09
60	12	9	6.0	90	0A

Table 4: Rio Elegance Yellow Dimming Setting

Table 5 below represents dimming setting for Rio Spring Green from 10 cd/m² to 60 cd/m².

Luminance (cd/m ²)	VPP (V)	VCOL (V)	VROW (V)	Software Setting	
				BRIGHTA (H)	DIMMERCTRL (H)
10	8	5	3.5	80	01
20	9	6	3.5	80	01
30	9	7	4.0	80	02
40	12	7	4.5	80	05
50	12	8	4.5	80	03
60	12	9	5.0	80	06

Table 5: Rio Spring Green Dimming Setting

The power consumption also has a linear relationship with the luminance and number of pixel illuminated. Both of the charts shown below, **Figure 14** and **Figure 15** show linearly increasing power usages for higher brightness. The higher the number of illuminated pixels are on and brightness, the higher the power consumption. Note that power consumption is not zero when no pixels are turned on due to the power used by the logic portion of the IC.

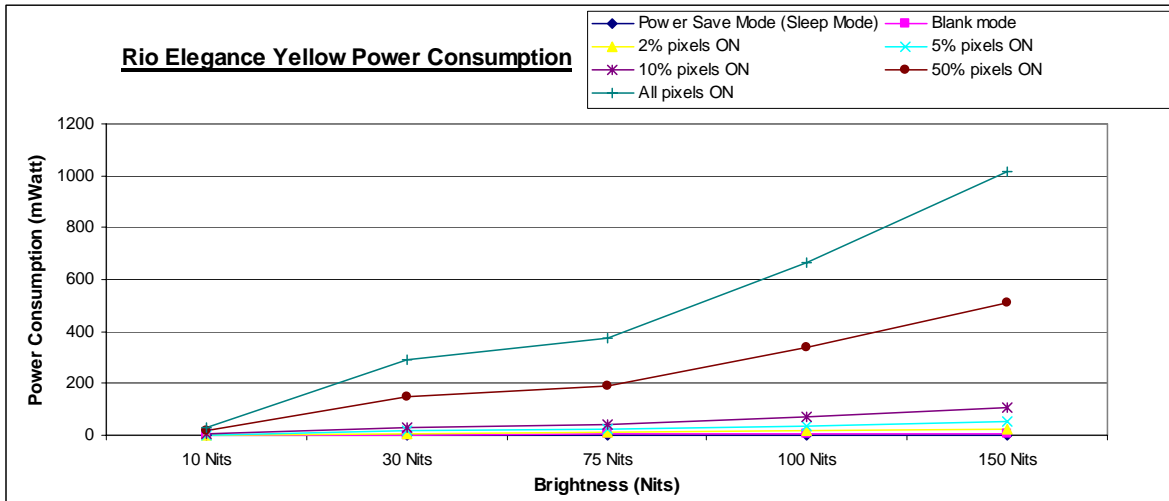


Figure 14: Rio Elegance Yellow Power Consumption vs Brightness

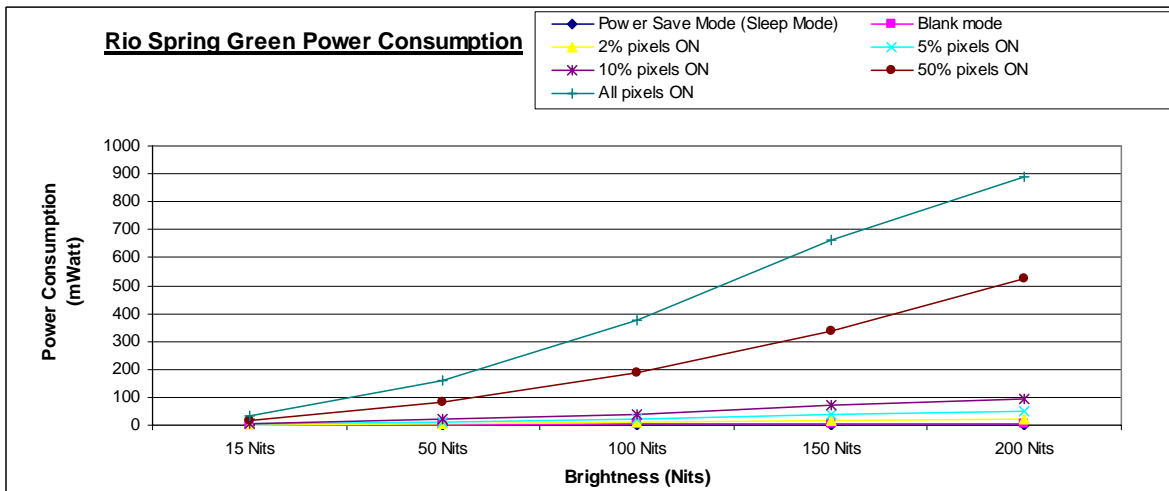


Figure 15: Rio Spring Green Power Consumption vs Brightness

```
////////////////////////////////////////////////////////////////////////////////////////////////////
//                               RIO SPRING GREEN STV8105 PARALLEL MODE (4Bits, 16 GRAYSCALE)
//
//WRITTEN BY: MARDIANA BINTI KHALID
//DATE: 31 JULY 2006
////////////////////////////////////////////////////////////////////////////////////////////////////

#include <reg52.h>
#include <intrins.h>

#define uint unsigned int
#define uchar unsigned char

const char code bmp1[128*64];

/*****PIN CONFIGURATION*****/
// control signal
sbit CST  = P1^0;
sbit DC   = P1^2;
sbit WRT  = P1^3;
sbit RST  = P1^6;
sbit Tog  = P3^4;          //Freeze Button

/*****FUNCTION DECLARE*****/
void main          (void);
void write_data    (void);
void write_command (void);
void sub_comm      (void);
void reset         (void);
void init_OLED     (void);
void write_PATTERN1 (void);
void write_RAM     (void);
void data_out      (unsigned char);
void data_out1    (char i);
void comm_out      (unsigned char i);
void freeze(void); //interrupt by Tog

/******/
void main(void)
{
    reset();
    write_RAM();
    init_OLED();
    while(1)
    {
        write_PATTERN1();
        freeze();
    }
}

void reset(void)
{
    RST = 1;
    RST = 0;          //Reset Driver IC
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    RST = 1;
}
}
```



```

void init_OLED(void)
{
//////////COMMAND SETTING//////////
    comm_out(0xF2); //SOFTWARE RESET

    comm_out(0x00); //SCLKDIV :- SCLK CLOCK DIVIDER RATIO SELECT
    comm_out(0x00); //120 Hz FRAME FREQUENCY

    comm_out(0x01); //BRIGHTA :- INITIAL BRIGHTNESS ADJUSTMENT A
    comm_out(0xC9); //REGISTER A

    comm_out(0x02); //BRIGHTB :- INITIAL BRIGHTNESS ADJUSTMENT B
    comm_out(0x00); //REGISTER B

    comm_out(0x03); //DCDF0TRL :- DC/DC STEP UP CONVERTER CONTROL
    comm_out(0x00); //DISABLE INTERNAL DC TO DC

    comm_out(0x06); //VFDETVAL :-SELECTION OF VOLTAGE TO A0F AS ADJUSTMENT TO VH
    comm_out(0x00); //ONLY USED WHEN INTERNAL DC TO DC ENABLE

    comm_out(0x10); //DCTRL :- DOT MATRIX DISPLAY CONTROL
    comm_out(0x01); //DISPLAY ON & NORMAL DISPLAY

    comm_out(0x15); //GSADDINC :- GRAYSCALE MODE SEL.& DISP. RAM ADDR. INCREMENT
    comm_out(0x03); //X & Y INCREMENT ON,16 LEVEL GRAYSCALE MODE

    comm_out(0x16); //DIMMERCTRL :- DIMMER CONTROL
    comm_out(0x0A); //11/16

    comm_out(0x17); //ROWDRVSEL :- ROW DRIVER MODE SELECTION
    comm_out(0x02); //INTERNAL ROW DRIVER, SINGLE SCANNING, NO ROW REMAP

    comm_out(0x1A); //COLUMN OUTPUT CONTROL
    comm_out(0x00); //COMMAND DISABLED

    comm_out(0x1B); //OELPERIOD1 :- SETUP PERIOD 1 COMMAND
    comm_out(0x0F); //COLUMN OUTPUTS AT GNDL, 16 SCLK

    comm_out(0x1C); //OELPERIOD2 :- SETUP PERIOD 2 COMMAND
    comm_out(0x81); //OUTPUT CONNECTED TO VCOL, 2 SCLK

    comm_out(0x1D); //OELPERIOD3 :- SETUP PERIOD 3 COMMAND
    comm_out(0x81); //OUTPUT CONNECTED TO VCOL, 2 SCLK

    comm_out(0x1E); //OELPERIOD4 :- SETUP PERIOD 4 COMMAND
    comm_out(0x89); //OUTPUT CONNECTED TO VCOL, 10 SCLK

    comm_out(0x1F); //ODD15 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0xFF); //256 SCLK

    comm_out(0x20); //ODD14 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0xAF); //176 SCLK

    comm_out(0x21); //ODD13 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0x79); //122 SCLK

    comm_out(0x22); //ODD12 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0x53); //84 SCLK

    comm_out(0x23); //ODD11 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0x39); //58 SCLK

    comm_out(0x24); //ODD10 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0x27); //40 SCLK

    comm_out(0x25); //ODD09 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0x1A); //27 SCLK

    comm_out(0x26); //ODD08 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0x12); //19 SCLK

    comm_out(0x27); //ODD07 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0x0C); //13 SCLK

```

```

comm_out(0x28); //ODD06 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x08); //9 SCLK

comm_out(0x29); //ODD05 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x05); //6 SCLK

comm_out(0x2A); //ODD04 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x03); //4 SCLK

comm_out(0x2B); //ODD03 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x02); //3 SCLK

comm_out(0x2C); //ODD02 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x01); //2 SCLK

comm_out(0x2D); //ODD01:- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x00); //1 SCLK

comm_out(0x2E); //EVEN15 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0xFF); //256 SCLK

comm_out(0x2F); //EVEN14 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0xAF); //176 SCLK

comm_out(0x30); //EVEN13 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x79); //122 SCLK

comm_out(0x31); //EVEN12 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x53); //84 SCLK

comm_out(0x32); //EVEN11 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x39); //58 SCLK

comm_out(0x33); //EVEN10 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x27); //40 SCLK

comm_out(0x34); //EVEN09 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x1A); //27 SCLK

comm_out(0x35); //EVEN08 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x12); //19 SCLK

comm_out(0x36); //EVEN07 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x0C); //13 SCLK

comm_out(0x37); //EVEN06 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x08); //9 SCLK

comm_out(0x38); //EVEN5 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x05); //6 SCLK

comm_out(0x39); //EVEN04 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x03); //4 SCLK

comm_out(0x3A); //EVEN03 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x02); //3 SCLK

comm_out(0x3B); //EVEN02 :- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x01); //2 SCLK

comm_out(0x3C); //EVEN01:- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
comm_out(0x00); //1 SCLK
}

void sub_comm (void)
{
    comm_out(0x11); //DOTMTRXDIR :- DOT MATRIX DISPLAY DIRECTION
    comm_out(0x00); //DISNABLE COL REMAP & ROW REMAP

    comm_out(0x12); //DOTMTRXSCAN :- DOT MATRIX SCAN LINE SELECT
    comm_out(0x3F); //LINE 64 SELECTED AS SCAN LINE

    comm_out(0x13); //RAMSXTART :- DISPLAY RAM X STARTING ADDRESS
    comm_out(0x00); //00 STARTING ADDRESS
    comm_out(0x14); //RAMYSTART :- DISPLAY RAM Y STARTING ADDRESS
}

```

```

        comm_out(0x00); //00 STARTING ADDRESS
    }

void comm_out(unsigned char i) //i is 8-bit command value
{
    DC          = 0; //SET LOW FOR COMMAND
    CST         = 0; //CHIP SELECT SET TO LOW
    WRT         = 0; //LOW FOR WRITE
    P2          = i;
    WRT         = 1; //NO WRITE
    CST         = 1; //CHIP SELECT SET TO HIGH
    DC          = 1; //CHIP SELECT SET TO LOW
}

void data_out(unsigned char i) //i is 8-bit image data value
{
    DC          = 1; //SET LOW FOR COMMAND
    CST         = 0; //CHIP SELECT SET TO LOW
    WRT         = 0; //LOW FOR WRITE
    P2          = i;
    WRT         = 1; //NO WRITE
    CST         = 1; //CHIP SELECT SET TO HIGH
    DC          = 0; //CHIP SELECT SET TO LOW
}

void data_out1(char i) //i is 8-bit image data value
{
    P2          = i; // Send data value to data pins D[7:0]
    write_data();
}

void write_PATTERN1(void)
{
    int i,j;
    sub_comm();
    for (i=0;i<64;i++)
    {
        comm_out(0x13); //RAMSXTART :- DISPLAY RAM X STARTING ADDRESS
        comm_out(0x00); //00 STARTING ADDRESS

        comm_out(0x14); //RAMYSTART :- DISPLAY RAM Y STARTING ADDRESS
        comm_out(0x00+i); //00 STARTING A0FRESS
        for (j=0;j<128;j++)
        {
            data_out(bmp1[(63-i)*128+j]);
        }
    }
}

void write_RAM(void)
{
    int i,j;
    sub_comm();
    for (i=0;i<72;i++)
    {
        for (j=0;j<256;j++)
        {
            data_out(0X00);
        }
    }
}

void write_data(void)
{
    DC          = 1; //SET LOW FOR COMMAND
    CST         = 0; //CHIP SELECT SET TO LOW
    WRT         = 0; //LOW FOR WRITE
    _nop_(); //NO OPERATION
    WRT         = 1; //NO WRITE
    CST         = 1; //CHIP SELECT SET TO HIGH
    DC          = 0; //CHIP SELECT SET TO LOW
}

```

```

void write_command(void)
{
    DC          = 0;    //SET LOW FOR COMMAND
    CST         = 0;    //CHIP SELECT SET TO LOW
    WRT        = 0;    //LOW FOR WRITE
    _nop_(); //NO OPERATION
    WRT        = 1;    //NO WRITE
    CST         = 1;    //CHIP SELECT SET TO HIGH
    DC          = 1;    //CHIP SELECT SET TO LOW
}

void freeze (void)
{
    int i,j;
    for(i=0;i<500;i++)
    {
        for(j=0;j<500;j++)
        {
            while(Tog)
            {}
        }
    }
}

```

```

////////////////////////////////////////////////////////////////////////////////
//                RIO SPRING GREEN STV8105 PARALLEL MODE (1BIT, MONOCHROME)
//
//WRITTEN BY: MARDIANA BINTI KHALID
//DATE: 31 JULY 2006
////////////////////////////////////////////////////////////////////////////////

#include <reg52.h>
#include <intrins.h>

#define uint unsigned int
#define uchar unsigned char

const char code bmp1[128*64];

/*****PIN CONFIGURATION*****/
// control signal
sbit CST = P1^0;
sbit DC = P1^2;
sbit WRT = P1^3;
sbit RST = P1^6;
sbit Tog = P3^4; //Freeze Button

/*****FUNCTION DECLARE*****/
void main (void);
void write_data (void);
void write_command (void);
void sub_comm (void);
void reset (void);
void init_OLED (void);
void write_PATTERN1 (void);
void write_RAM (void);
void data_out (unsigned char);
void data_out1 (char i);
void comm_out (unsigned char i);
void freeze(void); //interrupt by Tog

/*****/
void main(void)
{
    reset();
    write_RAM();
    init_OLED();
    while(1)
    {
        write_PATTERN1();
        freeze();
    }
}

void reset(void)
{
    RST = 1;
    RST = 0; //Reset Driver IC
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    RST = 1;
}

```

```

void init_OLED(void)
{
//////////COMMAND SETTING//////////
    comm_out(0xF2); //SOFTWARE RESET

    comm_out(0x00); //SCLKDIV :- SCLK CLOCK DIVIDER RATIO SELECT
    comm_out(0x00); //120 Hz FRAME FREQUENCY

    comm_out(0x01); //BRIGHTA :- INITIAL BRIGHTNESS ADJUSTMENT A
    comm_out(0xC9); //REGISTER A

    comm_out(0x02); //BRIGHTB :- INITIAL BRIGHTNESS ADJUSTMENT B
    comm_out(0x00); //REGISTER B

    comm_out(0x03); //DCDF0TRL :- DC/DC STEP UP CONVERTER CONTROL
    comm_out(0x00); //DISABLE INTERNAL DC TO DC

    comm_out(0x06); //VFDETVAL :-SELECTION OF VOLTAGE TO A0F AS ADJUSTMENT TO VH
    comm_out(0x00); //ONLY USED WHEN INTERNAL DC TO DC ENABLE

    comm_out(0x10); //DCTRL :- DOT MATRIX DISPLAY CONTROL
    comm_out(0x01); //DISPLAY ON & NORMAL DISPLAY

    comm_out(0x15); //GSADDINC :- GRAYSCALE MODE SEL.& DISP. RAM ADDR. INCREMENT
    comm_out(0x83); //X & Y INCREMENT ON, MONOCHROME MODE

    comm_out(0x16); //DIMMERCTRL :- DIMMER CONTROL
    comm_out(0x0A); //11/16

    comm_out(0x17); //ROWDRVSEL :- ROW DRIVER MODE SELECTION
    comm_out(0x02); //INTERNAL ROW DRIVER, SINGLE SCANNING, NO ROW REMAP

    comm_out(0x1A); //COLUMN OUTPUT CONTROL
    comm_out(0x00); //COMMAND DISABLED

    comm_out(0x1B); //OELPERIOD1 :- SETUP PERIOD 1 COMMAND
    comm_out(0x0F); //COLUMN OUTPUTS AT GNDL, 16 SCLK

    comm_out(0x1C); //OELPERIOD2 :- SETUP PERIOD 2 COMMAND
    comm_out(0x81); //OUTPUT CONNECTED TO VCOL, 2 SCLK

    comm_out(0x1D); //OELPERIOD3 :- SETUP PERIOD 3 COMMAND
    comm_out(0x81); //OUTPUT CONNECTED TO VCOL, 2 SCLK

    comm_out(0x1E); //OELPERIOD4 :- SETUP PERIOD 4 COMMAND
    comm_out(0x89); //OUTPUT CONNECTED TO VCOL, 10 SCLK

    comm_out(0x2D); //ODD01:- LOADING BYTE OF THE ODD GRAYSCALE LOOKUP TABLE
    comm_out(0xFF); //MONOCHROME MODE
}

void sub_comm (void)
{
    comm_out(0x11); //DOTMTRXDIR :- DOT MATRIX DISPLAY DIRECTION
    comm_out(0x00); //DISABLE COL REMAP & ROW REMAP

    comm_out(0x12); //DOTMTRXSCAN :- DOT MATRIX SCAN LINE SELECT
    comm_out(0x3F); //LINE 64 SELECTED AS SCAN LINE

    comm_out(0x13); //RAMSXTART :- DISPLAY RAM X STARTING ADDRESS
    comm_out(0x00); //00 STARTING ADDRESS
    comm_out(0x14); //RAMYSTART :- DISPLAY RAM Y STARTING ADDRESS
    comm_out(0x00); //00 STARTING ADDRESS
}

void comm_out(unsigned char i) //"i" is 8-bit command value
{
    DC          = 0;    //SET LOW FOR COMMAND
    CST         = 0;    //CHIP SELECT SET TO LOW
    WRT         = 0;    //LOW FOR WRITE
    P2         = i;
    WRT         = 1;    //NO WRITE
    CST         = 1;    //CHIP SELECT SET TO HIGH
    DC          = 1;    //CHIP SELECT SET TO LOW
}

```

```

}
void data_out(unsigned char i) //i is 8-bit image data value
{
    DC          = 1;    //SET LOW FOR COMMAND
    CST         = 0;    //CHIP SELECT SET TO LOW
    WRT         = 0;    //LOW FOR WRITE
    P2          = i;
    WRT         = 1;    //NO WRITE
    CST         = 1;    //CHIP SELECT SET TO HIGH
    DC          = 0;    //CHIP SELECT SET TO LOW
}

void data_out1(char i) //i is 8-bit image data value
{
    P2          = i;    // Send data value to data pins D[7:0]
    write_data();
}

void write_PATTERN1(void)
{
    int i,j;
    sub_comm();
    for (i=0;i<64;i++)
    {
        comm_out(0x13); //RAMSXTART :- DISPLAY RAM X STARTING ADDRESS
        comm_out(0x00); //00 STARTING ADDRESS

        comm_out(0x14); //RAMYSTART :- DISPLAY RAM Y STARTING ADDRESS
        comm_out(0x00+i); //00 STARTING A0FRESS
        for (j=0;j<128;j++)
        {
            data_out bmp1[(63-i)*128+j]);
        }
    }
}

void write_RAM(void)
{
    int i,j;
    sub_comm();
    for (i=0;i<72;i++)
    {
        for (j=0;j<256;j++)
        {
            data_out(0X00);
        }
    }
}

void write_data(void)
{
    DC          = 1;    //SET LOW FOR COMMAND
    CST         = 0;    //CHIP SELECT SET TO LOW
    WRT         = 0;    //LOW FOR WRITE
    _nop_(); //NO OPERATION
    WRT         = 1;    //NO WRITE
    CST         = 1;    //CHIP SELECT SET TO HIGH
    DC          = 0;    //CHIP SELECT SET TO LOW
}

void write_command(void)
{
    DC          = 0;    //SET LOW FOR COMMAND
    CST         = 0;    //CHIP SELECT SET TO LOW
    WRT         = 0;    //LOW FOR WRITE
    _nop_(); //NO OPERATION
    WRT         = 1;    //NO WRITE
    CST         = 1;    //CHIP SELECT SET TO HIGH
    DC          = 1;    //CHIP SELECT SET TO LOW
}

```

```
void freeze (void)
{
    int i,j;
    for(i=0;i<500;i++)
    {
        for(j=0;j<500;j++)
        {
            while(Tog)
            {}
        }
    }
}
```


Created By: Mardiana Binti Khalid
OLED PRODUCT DEVELOPMENT
PENANG MALAYSIA

About Osram Opto Semiconductors

Osram Opto Semiconductors GmbH, Regensburg, is a wholly owned subsidiary of Osram GmbH, one of the world's three largest lamp manufacturers, and offers its customers a range of solutions based on semiconductor technology for lighting, sensor and visualisation applications. The company operates facilities in Regensburg (Germany), San José (USA) and Penang (Malaysia). Further information is available at www.osram-os.com.

All information contained in this document has been checked with the greatest care. OSRAM Opto Semiconductors GmbH can however, not be made liable for any damage that occurs in connection with the use of these contents.