Freescale Semiconductor
Application Note

Document Number: AN3415
Rev. 0, 03/2007

OLED Display Driver for the HCS08 Family

OLED Driver Demonstration for 4 Bits-per-Pixel Displays

by: Wolfgang Bihlmayr
Systems Engineering
Munich, Germany

This application note describes how to interface an HCS08 device to an OLED display. Organic light-emitting diode (OLED) displays can provide a graphical user interface to various applications. They can be found in television screens, MP3 players, portable meters (electrical, medical), home automation equipment (thermostats, alarm controls), mobile phones, vending machines, coffee machines, white goods, etc.

The code was written for MCUs with limited RAM and flash memory, with a minimum of I/Os used to drive an OSRAM Pictiva™ 128x64 OLED Display, Elegance Yellow with 16 gray scales (Part No. OS128064PK27MY0B00). The code is provided as a zip file, AN3415SW.zip, and can be downloaded from the Freescale website, www.freescale.com. The example code can be easily modified for other OLED displays.

A hardware daughter card is also described. The card layout allows either the Freescale DEMO9S08AW60 board or the DEMO9S08QG8 board to to interface the

Contents

OLED. The OLED daughter card contains the interface, a DC/DC converter to provide the OLED voltage (12 V), and the actual OLED display itself.

The OLED displays are a self-emissive technology that typically require less power than LCDs with backlights. Only an “on” pixel consumes power. This feature makes OLED displays suitable for battery-powered devices.

Advantages of OLED displays:

- Less power than LCDs — Better for portable/handheld applications
- Self-emissive technology
- Vivid color — High brightness and contrast
- Video capability
- Wide viewing angle
- Thin form factor — No backlight required
- Long-lasting lifetime — Up to 55,000 hours
- Monochrome (yellow, orange, green, white, blue) with one- or four-bit gray-scale capability
- Full color capability

1 OLED Display Description

The OLED display used in this application note offers a parallel- or serial-interface-connection option. The BS1 and BS2 pins are used to select the interface to be used. In this example only the serial interface is used (Figure 1).

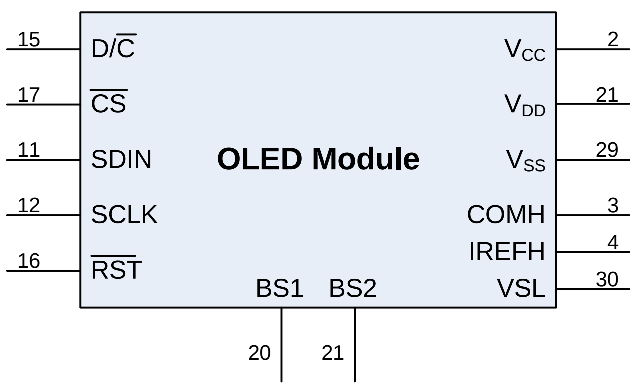


Figure 1. OLED Module Interface (Serial Mode)

The main advantage of using the serial interface rather than the parallel interface is the number of I/Os required to drive the display. For the serial interface, only 5 I/Os (D/$\overline{C}$, $\overline{CS}$, SDIN, SCLK, $\overline{RST}$) are required, compared with 13 I/Os (D/$\overline{C}$, $\overline{CS}$, R/$\overline{W}$, E/$\overline{RD}$, D0:D7, $\overline{RST}$) for the parallel interface.

Table 1 briefly describes the OLED pin functions for the serial interface.

Table 1. OLED Pin Description

Pin	Pin Name	Description
15	$D/\overline{C}$	Data/Command
17	$\overline{CS}$	Chip-select control
11	SDIN	Serial-data input
12	SCLK	Serial-clock input
16	$\overline{RST}$	Reset
20, 21	BS1, BS2	Interface selection BS1 = 0; BS2 = 0 for serial
2	V_{CC}	OLED power-supply voltage (12 V)
21	V_{DD}	Logic-supply voltage (3.3 V)
29	V_{SS}	Ground
3	COMH	Common (row) high voltage — A capacitor should be connected between this pin and V_{SS}.
4	IREFH	Segment (column) current reference — A resistor should be connected between this pin and V_{SS}.
30	VSL	Voltage segment low — A capacitor should be connected between this pin and V_{SS}.

1.1 Basic Operation

The OLED display contains a chip-on-glass (COG) controller (SSD0323 128 x 80, 16 Gray Scale Dot Matrix OLED/PLED Segment/Common Driver with Controller from Solomon Systech) with display RAM (graphic display data RAM — GDDRAM) that holds the data for the pattern/images to be displayed.

Four bits of data are required for each pixel (4 bpp = bits per pixel) allowing 16 possible gray values. One byte is used to store the information for two pixels. In total, 4 Kbytes (4096 bytes) are required to hold the complete display content for the 128_x_64 pixels. The MCU needs to transfer data only if the display content needs updated. Cyclic refresh of the display is not necessary .

The data transfer is done byte-wise, first sending the most significant bit (MSB) (Figure 2). The data is applied with the rising edge of the SCLK signal and is sampled at the falling edge of the SCLK signal. The data/command signal $D/\overline{C}$ is used to distinguish between graphic display data access and control commands sent to the display. Table 3 outlines the available display-control commands.

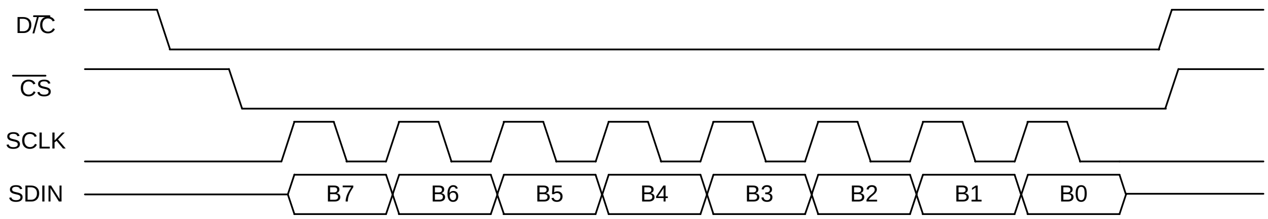


Figure 2. Serial Data Transfer (Command)

The display data is sent byte-by-byte. Each byte automatically increments the address pointer to the next position. The display supports different GDDRAM to pixel mappings for mounting flexibility (regarding display orientation). You can rotate the display output by 180 degrees. Figure 3 and Table 2 show the default pixel mapping the software uses.

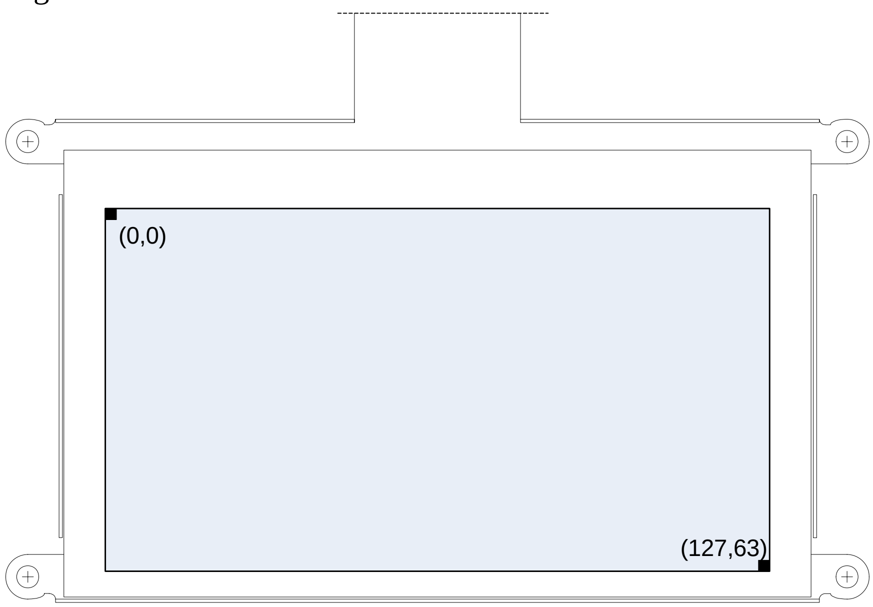


Figure 3. Pixel Mapping

Table 2. GDDRAM Address Mapping (Data-Byte Sequence D0, D1, ..., D5119)

		Column							
		0		1		...	63		Address
		0	1	2	3	...	126	127	Pixel
Row	0	D0[7:4]	D0[3:0]	D1[7:4]	D1[3:0]	...	D63[7:4]	D63[3:0]	
	1	D64[7:4]	D64[3:0]	D65[7:4]	D65[3:0]	....	D127[7:4]	D127[3:0]	
	. . .	. . .	. . .	. . .	. . .	...	. . .	. . .	
	63	D5056[7:4]	D5056[3:0]	D5057[7:4]	D5057[3:0]	...	D5119[7:4]	D5119[3:0]	
	Address								

To update a portion of the display partially, you can use the set-column-address and set-row-address commands. They allow defining a portion (canvas) of the display to be updated without updating the whole 128 x 64 pixels. Also, each transferred data byte increases the address pointer to the next (two) pixel within the defined area. Basically, this mechanism allows you to individually update any (two) pixel (respectively, any byte in the GDDRAM).

One constraint of using the serial interface is that it supports only unidirectional data transfer (write-only). Basically, in the serial mode you cannot read back the GDDRAM content; you cannot update only a single pixel of two pixel in a byte by using a read-modify-write one byte method.

1.2 Command Description

Table 3 briefly explains the available commands (for the serial interface) and their corresponding parameters. For a more detailed description, consult the OSRAM Pictiva or SSD0323 documentation (Section 8, “References,” on page 32). Most commands are used to parametrize the COG for a specific display under specific ambient conditions. Recommendations for setup for various displays are provided by the display manufacturers (for OSRAM OLED see Table 4).

Table 3. Command Table (Sheet 1 of 5)

D/$\overline{C}$	Hex	Command	Length	Description[1]
0 0 0	0x15 A[5:0] B[5:0]	Set Column Address	3 bytes	This command specifies column start address and end address of the display data RAM. It also sets the column address pointer to column start address. Second byte A[5:0] sets the column start address from 0–63. POR = 0. Third byte B[5:0] sets the column end address from 0–63, POR = 63
0 0 0	0x75 A[5:0] B[5:0]	Set Row Address	3 bytes	This command specifies row start address and end address of the display data RAM. It also sets the row address pointer to row start address. Second byte A[5:0] sets the row start address from 0–79, POR= 0. Third byte B[5:0] sets the row end address from 0–79, POR = 79
0 0	0x81 A[6:0]	Set Contrast Control Register	2 bytes	This command is to set contrast setting of the display. The COG chip has 128 contrast steps from 0x00 to 0x7F. The segment output current increases linearly with the increase of contrast step. Second byte A[6:0] sets the contrast in 128 steps. Contrast increases as level increases. POR = 0x40.
0	0x84–0x86	Set Current Range	1 byte	This command is used to select quarter range or half range or full range current mode. With the same contrast level, quarter range mode gives a quarter of the current output of the full range mode. Similar to half-range current mode, it gives a half of the current output of the full-range mode. 0x84 = Quarter-current range (POR) 0x85 = Half-current range 0x86 = Full-current range

Table 3. Command Table (Sheet 2 of 5)

D/$\overline{C}$	Hex	Command	Length	Description[1]
0 0	0xA0 A[6:0]	Set Re-map	2 bytes	This command changes the mapping between the display data column address and segment driver, row address and common driver. It allows flexibility in layout during OLED module assembly. Column Address Re-map If column address re-map is set, columns 0–63 map to SEG127–0, regardless of star-column and end-column commands. Nibble Re-map If nibble re-map is set, the two nibbles of the data bus for RAM access are re-mapped, such that B7, B6, B5, B4, B3, B2, B1, B0 acts like B3, B2, B1, B0, B7, B6, B5, B4. This feature working with column address re-map produces an effect of flipping outputs SEG0–127 to SEG127–SEG0. Address Increment Mode If horizontal increment mode is set, the column address pointer advances after each RAM access. If vertical increment mode is set, the row-address pointer advances after each RAM access. COM Re-map If COM re-map is set, ROW 0–79 maps to COM79–0, regardless of start and end row commands. A[0]=0, Disable Column Address Re-map (POR) A[0]=1, Enable Column Address Re-map A[1]=0, Disable Nibble Re-map (POR) A[1]=1, Enable Nibble Re-map A[2]=0, Horizontal Address Increment (POR) A[2]=1, Vertical Address Increment A[4]=0, Disable COM Re-map disable (POR) A[4]=1, Enable COM Re-map A[5]=0, Reserved (POR) A[5]=1, Reserved A[6]=0, Disable COM Split Odd Even (POR) A[6]=1, Enable COM Split Odd Even
0 0	0xA1 A[6:0]	Set Display Start Line	2 bytes	This command sets the display-start-line register, determining the display RAM address that can be shown by selecting a value from 0 to 79. Second byte A[6:0] sets the display RAM start-line register from 0 to 79. POR = 0.

Table 3. Command Table (Sheet 3 of 5)

D/$\overline{C}$	Hex	Command	Length	Description[1]
0 0	0xA2 A[6:0]	Set Display Offset	2 bytes	This command is to set the display-offset register to determine the mapping of the display start line to one of COM0–79. (It is assumed that COM0 is the display start line, display start line register equals 0.) For example, to move the COMX toward the COM0 direction for L lines, the 7-bit data in the second command should be given by L. In other words, to move the COMX towards the COM79 direction for L lines, the 7-bit data in the second command should be given by 80–L. Second byte A[6:0] sets the vertical scroll by COM from 0–79. POR = 0.
0	0xA4–0xA7	Set Display Mode	1 byte	This command is used to set normal display, entire display on, entire display off, and inverse display. Normal Display turns the data to ON at the corresponding gray levels GS0–GS15 (see Set Gray Scale Table). Entire Display On forces the entire display to be at gray level GS15, regardless of the contents of the display data RAM. Entire Display Off forces the entire display to be at gray level GS0 regardless of the contents of the display data RAM. Inverse Display uses the gray scale table in reverse order gray level 0–15 uses GS15–GS0, and so on. 0xA4 = Normal Display (POR) 0xA5 = Entire Display On (GS15) 0xA6 = Entire Display Off (GS0) 0xA7 = Inverse Display
0 0	0xA8 A[6:0]	Set Multiplex Ratio	2 bytes	This command sets multiplex ratio N from 16 to 80. Second byte A[6:0] determines multiplex ration N. POR = 0x4F (80).
0 0	0xAD A[1:0]	Set Master Configuration	2 bytes	This command is used to enable or disable the internal DC–DC voltage converter. This command is executed when display is on. A[0] = 0, Disable DC–DC converter A[0] = 1, Enable DC–DC converter (POR) A[1] = 0, Disable internal VCOMH A[1] = 1, Enable internal VCOMH (POR)
0	0xAE–0xAF	Set Display On/Off	1 byte	This command turns the display on or off. When the display is off, the segment and common output are in high impedance state. 0xAE = display off (sleep mode) (POR) 0xAF = display on
0 0	0xB0 A[5:0]	Set Pre-charge Compensation Enable	2 bytes	This command enables the pre-charge voltage. A[5:0] = 0x08 (POR) A[5:0] = 0x28, enable pre-charge compensation
0 0	0xB4 A[2:0]	Set Pre-charge Compensation Level	2 bytes	This command sets the pre-charge voltage level. A[2:0] = 0 (POR) A[2:0] = 3, recommended level

Table 3. Command Table (Sheet 4 of 5)

D/$\overline{C}$	Hex	Command	Length	Description[1]
0 0	0xBF A[3:0]	Set Segment Low Voltage (VSL)	2 bytes	This command is used to set segment low voltage (VSL). The value of VSL is the same for display all on, display all off pattern with internal or external DC–DC voltage converter. The second byte A[3:0] sets the VSL voltage as follow: 1000–1110 A[3:0] = 0010 connects to V_{SS} A[3:0] = 1110 (POR) (When V_{DD} > 2.5 V)
0 0	0xBE A[5:0]	Set VCOMH Voltage	2 bytes	This command is used to set VCOMH voltage level. The second byte A[5:0] specifies the VCOMH voltage level 000000–011111 A[5:0] = 1xxxxx = 1.0*VREF A[5:0] = 010001(POR)
0 0	0xBC A[7:0]	Set Pre-charge Voltage	2 bytes	This command is used to set the pre-charge voltage level. The second byte A[7:0] sets the pre-charge voltage level 00000000–00011111 A[7:0] = 1xxxxxxx connects to VCOMH A[7:0] = 001xxxxx equals 1.0*VREF A[7:0] = 00011000(POR)
0 0	0xB1 A[7:0]	Set Phase Length	2 bytes	This command sets the phase length. The lower nibble of the second byte selects phase 1 period (no pre-charge and current drive) from 1 to 16 DCLKs. The higher nibble of the second byte is used to select phase 2 period (pre-charge) from 1 to 16 DCLKs. A[3:0] = P1, phase 1 period of 1–15 DCLK clocks, POR = 3DCLKS = 3 A[7:4] = P2, phase 2 period of 1–15 DCLK clocks, POR = 5DCLKS = 5
0 0	0xB2 A[7:0]	Set Row Period	2 bytes	This command is used to set the row period. It is defined by multiplying the internal display clock period by the number of DCLKSs per row (value from 2–158). The larger the value, the more precise tuning of each gray scale level. See set gray scale table command for details. Also, it is used to define the frame frequency with the display clock divide ratio command. Row period equal to the sum of phase 1, 2 periods and the pulse width of GS15. Second byte sets the number of DCLKs, K, per row between 2–158DCLKS, POR = 37DCLKS = 0x25 The K value should be set as K = P1+P2+GS15 pulse width (POR: 3+5+29DCLKS)

Table 3. Command Table (Sheet 5 of 5)

D/$\overline{C}$	Hex	Command	Length	Description[1]
0 0	0xB3 A[7:0]	Set Display Clock Divide Ratio/Oscillator Frequency	2 bytes	This command is used to set the frequency of the internal display clocks, DCLKs. It is defined by dividing the oscillator frequency by the divide ratio (Value from 1 to 16). Frame frequency is determined by divide ratio, number of display clocks per row, MUX ratio and oscillator frequency. The lower nibble of the second byte is used to select the oscillator frequency. The lower nibble of the next byte sets the divide ratio of the display clocks: Divide ratio = 1–16, POR = 2 The higher nibble of the next byte sets the Oscillator Frequency. Oscillator Frequency increases with the value of A[7:4] and vice versa. POR=0
0 0 0 0 0 0 0 0 0	B8 A[2:0] B[6:4][2:0] C[6:4][2:0] D[6:4][2:0] E[6:4][2:0] F[6:4][2:0] G[6:4][2:0] H[6:4][2:0]	Set Gray Scale Table	9 bytes	This command is used to set the gray scale table for the display. The next 8 bytes set the gray scale level of GS1–15 as below: A[2:0] = L1, POR=1 B[2:0] = L2, POR=1 B[6:4] = L3, POR=1 C[2:0] = L4 POR=1 C[6:4] = L5, POR=1 D[2:0] = L6, POR=1 D[6:4] = L7, POR=1 E[2:0] = L8, POR=1 E[6:4] = L9, POR=1 F[2:0] = L10, POR=1 F[6:4] = L11, POR=1 G[2:0] = L12, POR=1 G[6:4] = L13, POR=1 H[2:0] = L14, POR=1 H[6:4] = L15, POR=1
0 0	0xCF A[7:4]	Set Biasing Current for DC–DC converter	1 byte	F0H = High (POR) 70H = Low
0	0xE3	NOP	1 byte	Command for no operation

[1] POR indicates the default value after power on reset

1.3 Power Up / Power Down Sequence

To protect the OLED display and to extend the display life time the power up/down sequence should be controlled, specifically, the display voltage V_{CC} (typ. 12–15 V).

1.3.1 Power-Up Sequence

To power-up the OLED display follow the steps below:

1. Power-up V_{DD} (3.3 V) and wait until stable
2. OLED display hardware reset
3. Send display-off command

4. Power-up V_{CC} (12 V) and wait until stable
5. Delay 100 ms
6. Send display-on command

1.3.2 Power-Down Sequence

To power-down the OLED display follow the steps below:

1. Send display-off command
2. Power down V_{CC} (12 V)
3. Delay 100 ms (when V_{CC} has reached 0 V and panel is completely discharged)
4. Power down V_{DD} (3.3 V)

1.4 Display Initialization

Table 4 shows the manufacturers (OSRAM) recommended initialization sequence after power-up for the Pictiva 2.7-inch 128x64 OLED Display, Elegance Yellow with the following parameters:

- V_{DD} = 3.3 V
- V_{CC} = 12 V
- Frame frequency = 100 Hz

CAUTION

Consult the manufacturer documentation for the latest information and for different displays (for example, spring green instead of elegance yellow) and/or parameters (for example, different V_{CC} voltages).

Table 4. Initialization Sequence After Power-Up

Command	Code	POR Default Value	Initialization (Dual Voltage Supply)
Set column address	0x15	0x00 0x3F	0x00 0x3F
Set row address	0x75	0x00 0x4F	0x00 0x3F
Set contrast control	0x81	0x40	0x66[1]
Set current range	0x84–0x86	quarter (0x84)	full (0x86)
Set re-map	0xA0	0x00	0x41
Set display start line	0xA1	00	default
Set display offset	0xA2	0x00	0x44
Set multiplexer ratio	0xA8	0x4F	0x3F
Set display ON/OFF		0xAE (OFF)	0xAF (ON)
Set display mode		0xA4	default

Table 4. Initialization Sequence After Power-Up (continued)

Command	Code	POR Default Value	Initialization (Dual Voltage Supply)
Set DC-DC converter	0xAD	0x02	0x02 (disabled)
Set DC-DC bias current	0xCF	0xF0	default
Set row period	0xB2	0x25	0x46
Set pre-charge compensation enable	0xB0	0x08	0x28
Set pre-charge compensation level	0xB4	0x00	0x07
Set clock divide	0xB3	0x02	0xF1
Set phase length	0xB1	P1=3, P2=5	0x22 (P1=2, P2=2)
Set VSL	0xBF	0x0E	0x0D
Set VCOMH	0xBE	0x11	0x02[1]
Set Vprecharge	0xBC	0x18	0x04
Set gray scale table	0xB8	all 1	see Table 5

[1] This setting represents maximum luminance for proper operation of the display. A lower setting can be used for dimming. A higher setting adversely affects the operating lifetime, as defined in this specification.

Table 5 shows the corresponding recommended gray-scale settings.

Table 5. Gray-Scale Settings

GS Level	Phase 1	Phase 2	S/W Set	GS Pulse	Total DCLK
L0	2	2	0	0	4
L1	2	2	1	1	5
L2	2	2	1	3	7
L3	2	2	1	5	9
L4	2	2	2	8	12
L5	2	2	2	11	15
L6	2	2	2	14	18
L7	2	2	3	18	22
L8	2	2	3	22	26
L9	2	2	4	27	31
L10	2	2	4	32	36
L11	2	2	5	38	42
L12	2	2	5	44	48
L13	2	2	6	51	55
L14	2	2	6	58	62
L15	2	2	7	66	70

2 Hardware

The system is set up using the standard DEMO9S08AW60 and DEMO9S08QG8 boards with a second pcb board called an OLED daughter card (Figure 4).

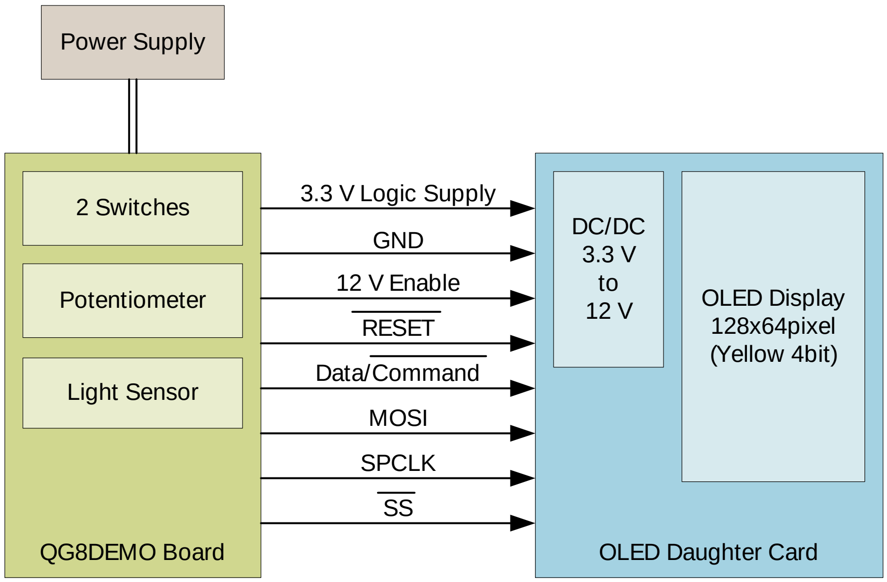


Figure 4. System Setup Example

2.1 OLED Daughter Card

The OLED daughter card allows the OLED display to connect to the DEMO9S08QG8 or DEMO9S08AW60 demo boards. The OLED daughter card contains the socket for the flat flex cable (FFC) of the OLED display, a DC/DC converter (12 V, 120 mA boost converter) to supply the OLED display with the necessary 12 V derived from the 3.3 V supply of the DEMO9S08AW60 and DEMO9S08QG8 boards. It has two separate 0.1-inch pinheader connectors, J1 and J2, that can be directly connected to the DEMO9S08AW60 and DEMO9S08QG8 boards, respectively.

Further details of the OLED Daughter Card, like schematics, layout and bill of materials (BOM), can be found in Section Appendix A, "OLED Daughter Card."

2.1.1 Interface

- OLED daughter-card interface — Requires six signals and two power-supply connections (Table 6, Table 7).
- MCU serial peripheral interface — Used to transfer data and commands to the OLED display. Three SPI pins—slave select ($\overline{SS}$), serial clock (SPSCK), and master-out slave-in (MOSI)—are used for the unidirectional data flow from the MCU to the OLED display. The master-in slave-out (MISO) pin is available for other use.
- 12VEN signal — Controls DC/DC converter output voltage. Allows a controlled power up/down of the OLED power voltage V_{CC}.

- $\overline{\text{RST}}$ signal — Connected to an MCU general-purpose input output (GPIO) pin to control the OLED display hardware reset by software.

Table 6. OLED Daughter-Card Connector J1 (DEMO9S08QG8)

Pin	Signal	Description
1	3.3 V	3.3 V power supply input
3	GND	Ground
17	SDIN	Serial data input
19	12VEN	12 V enable control input
21	SCLK	Serial clock input
23	$\overline{\text{CS}}$	Chip Select input
27	D/$\overline{\text{C}}$	Data/command input
29	$\overline{\text{RST}}$	Reset input
all other pins		Unused — not connected

Table 7. OLED Daughter-Card Connector J2 (DEMO9S08AW60)

Pin	Signal	Description
1	3.3 V	3.3 V power supply input
3	GND	Ground
13	D/$\overline{\text{C}}$	Data/command input
15	$\overline{\text{RST}}$	Reset input
17	SDIN	Serial data input
19	12VEN	12 V enable control input
21	SCLK	Serial clock input
23	$\overline{\text{CS}}$	Chip Select input
All other pins		Unused — not connected

2.2 Setup with DEMO9S08QG8 Board

Figure 4 indicates the OLED driver setup using the DEMO9S08QG8 board.

To protect the system and for proper operation, set up the DEMO9S08QG8 board before the system powers up.

Use the documentation provided with the board for more details about the DEM09S08QG8 board.

2.2.1 Jumpers

Jumper VX_EN must be closed to provide the OLED daughter card with power (3.3 V).

The Jumper PWR_SEL must be in position V_{DD}. This selects the external power supply as the power source.

The Jumper USER_EN should be closed for full demo functionality.

Figure 5 summarizes the jumper settings.

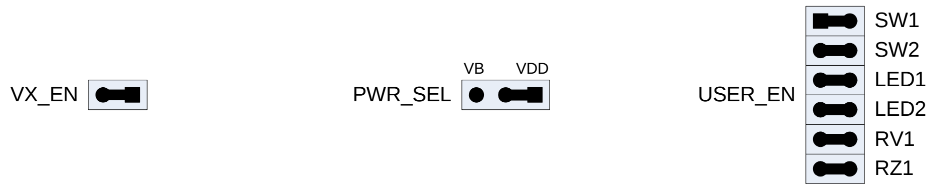


Figure 5. DEMO9S08QG8 Jumper Settings

2.3 Setup with DEMO9S08AW60 Board

Figure 13 indicates the OLED Driver setup using the DEMO9S08AW60 board.

To protect the system and for proper operation, set up the DEMO9S08AW60 board before the system powers up.

WARNING

Set up the DEMO9S08AW60 board correctly to avoid system damage or malfunction! The board must be configured for 3 V (see Section 2.3.1, "Jumpers").

Use the documentation provided with the board for more details about the DEM09S08AW60 board.

2.3.1 Jumpers

- Jumper W1 — V_{DD}_SEL must be in position 3 V. This selects the 3 V operation.
- Jumper W2 — 5V_SEL must be in position EXT. This selects the external power supply as the power source.
- Jumper W3 — P_IO_5V must be closed to provide the OLED daughter card with power (3.3 V).
- Jumpers from J3 to J31 — Should be closed for full demo functionality.

Figure 6 summarizes jumper settings.

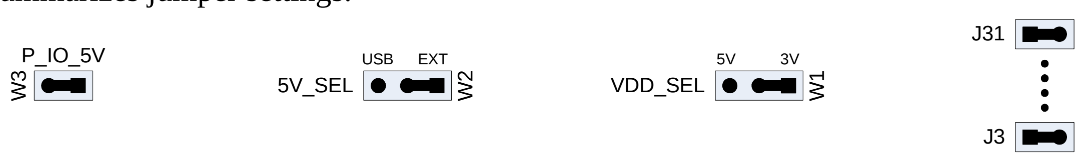


Figure 6. DEMO9S08AW60 Jumper Settings

3 Software and CodeWarrior™ Development Tool

The application software was developed for the DEMO9S08QG8 and the DEMO9S08AW60 boards driving an OSRAM Pictiva OSRAM Pictiva 2.7-inch 128x64 OLED Display, Elegance Yellow with 16 gray scales (Part.No. OS128064PK27MY0B00). It is written in a way which should make it easy to adopt for different MCUs or different OLED displays.

The software is also tested to be used with the OSRAM Pictiva 1.6-inch 128x64 OLED display, elegance yellow with 16 gray scales (Part.No. OS128064PK16MY0A01).

The driver project was developed with the CodeWarrior for HC08 V5.1 — Special Edition development tool.

Application features:

- Serial interface to OLED display using SPI module (low number of I/Os used)
- Low-level routines:
  - Send data byte
  - Send command byte
  - OLED power voltage ON/OFF
  - OLED hardware reset
- High-level routines:
  - Display initialization
  - Set drawing area
  - Fill drawing area
  - Draw text using bitmap font
  - Draw bitmap graphic
- Two bitmap type fonts 5x7 and 8x15 pixel size are included
- Adaptable for other OLED displays or MCUs

To open the OLED driver demo project, open file OLEDDemo.mcp in the CodeWarrior integrated development environment (IDE). Figure 7 shows the project view.

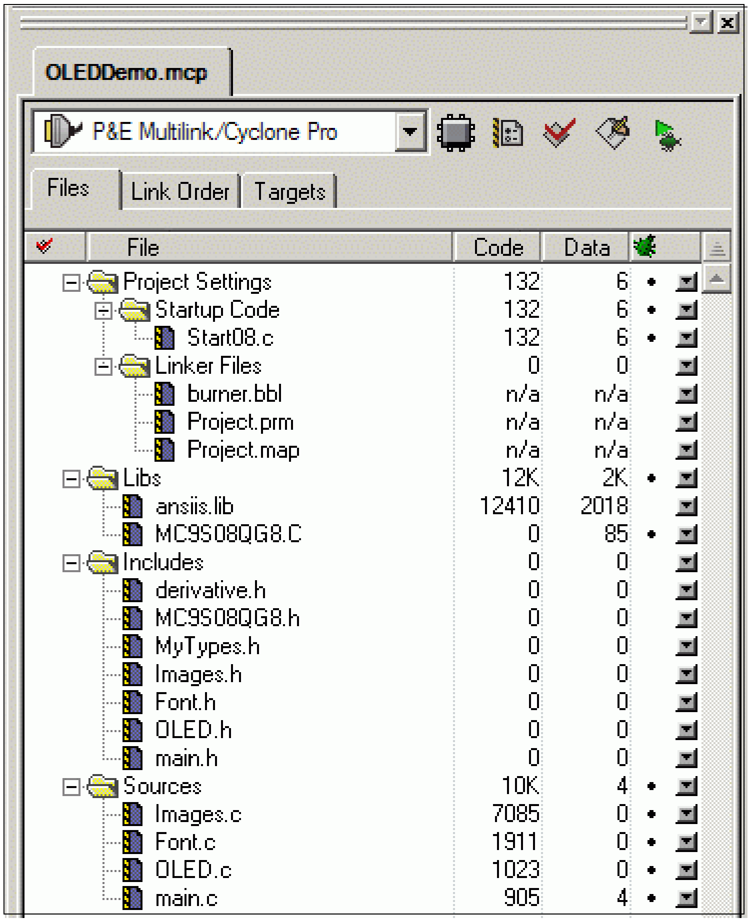


Figure 7. OLED Driver Demo Project Tree

3.1 OLED Driver Software Description

The OLED driver project contains the following files:

Table 8. OLED Driver Files

Files	Description
OLED.H	OLED driver header file — Constants (defines) for OLED commands — Preparation for different OLED displays — Definition of MCU hardware interface
OLED.C	OLED driver c file — OLED initialization sequence (array) — OLED functions (Table 9)
FONT.H	Font include file
FONT.C	Font c file — Definition of two fonts 5x7pixel and 8x16pixel
IMAGES.H	Image include file
IMAGES.C	Image c file — Example images
DERIVATIVE.H	Header file to include the derivative specific header file — This and the MCU derivative-specific files (for example, MC9S08QG8.H and MC9S08QG8.C) are controlled by the CodeWarrior IDE (Section 3.1.1, “Driver Configuration”).
MYTYPES.H	Type definitions
MAIN.C	Shows an example implementation of the OLED driver

3.1.1 Driver Configuration

The OLED.H header file can adapt the driver for different microcontrollers. Currently, it supports the MC9S08QG8 and the MC9S08AW60 MCUs.

To change the microcontroller, click on the “Change MCU/Connection ...” button and select the MCU derivative and the debugger connection you want to use (Figure 8 and Figure 9).

NOTE

You must select the correct MCU derivative and debugger interface to download the software into the microcontroller flash memory.

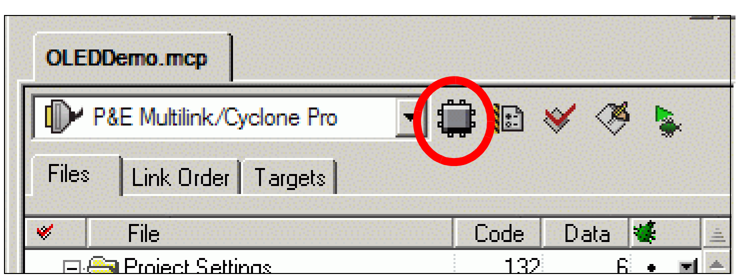


Figure 8. Change MCU

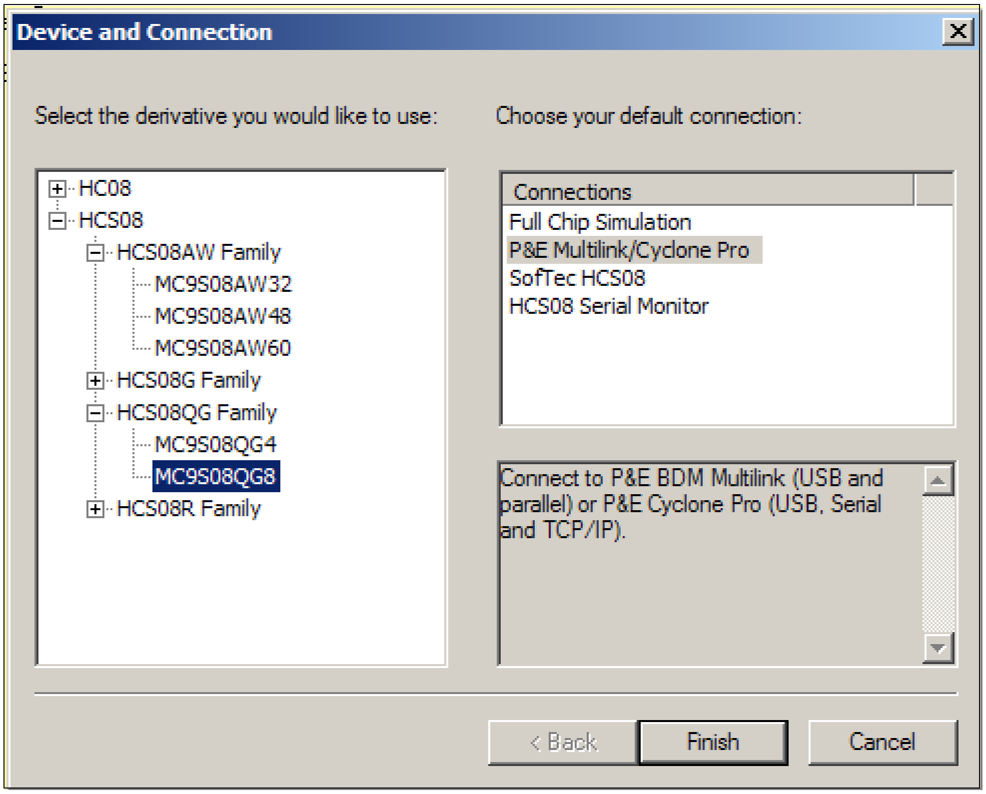


Figure 9. Select MCU Derivative and Debugger Connection

The CodeWarrior IDE automatically modifies the DERIVATIVE.H file and inclues the right MCU derivative files.

For example, after changing to the MC9S08QG8 MCU the DERIVATIVE.H file looks like this:

```
/*
 * Note: This file is recreated by the project wizard whenever the MCU is
 *       changed and should not be edited by hand
 */

/* Include the derivative-specific header file */
#include <MC9S08QG8.h>
```

And now, the project contains the MC9S08QG.C and MC9S08QG8.H files:

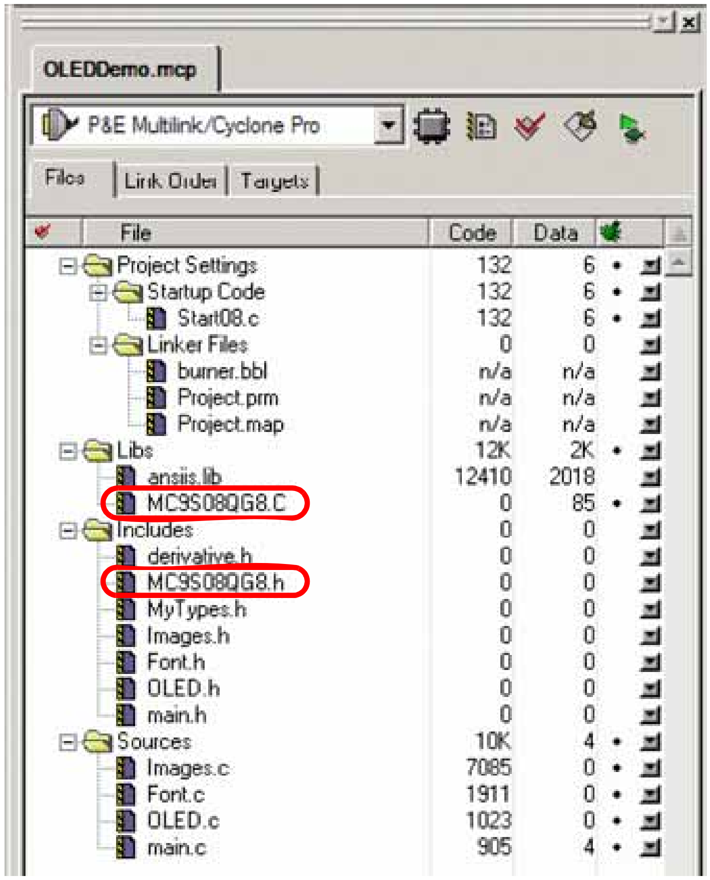


Figure 10. IDE Updates MCU Derivative Files

The MC9S08QG8.H file defines a macro with the derivative name, for example, _MC9S08QG8_H, which is used to implement the hardware interface-specific code.

```
// ---------------------------------------------------------------------------
// macros for hardware interface
// ---------------------------------------------------------------------------

// ------------------------------
// pinout for DEMO9S08QG8 board:
#ifdef _MC9S08QG8_H

// use PTB4/MISO pin to enable 12V for OLED display
#define  _12V_Init()        PTBD_PTBD4 = 0; PTBDD_PTBDD4 = 1
#define  _12V_Enable()      (PTBD_PTBD4 = 1)
#define  _12V_Disable()     (PTBD_PTBD4 = 0)

// use PTB7 pin to drive reset signal OLED display
#define  _RST_Init()        PTBD_PTBD7 = 1; PTBDD_PTBDD7 = 1
#define  _RST_Assert()      (PTBD_PTBD3 = 0)
#define  _RST_Release()     (PTBD_PTBD3 = 1)

// use PTB6 pin to drive D/#C (Data/Command) signal to OLED display
#define  _DC_Init()         {PTBD_PTBD6 = 1; PTBDD_PTBDD6 = 1;}
#define  _DC_Set()          (PTBD_PTBD6 = 1)
#define  _DC_Clr()          (PTBD_PTBD6 = 0)

// SPI init
#define  _SPI_Init()        { \
  SPIC1 = SPIC1_SPE_MASK|SPIC1_MSTR_MASK|SPIC1_SSOE_MASK; \
  SPIC2 = SPIC2_MODFEN_MASK|SPIC2_BIDIROE_MASK|SPIC2_SPC0_MASK; \
  SPIBR = 0; \
}

#define  _SPI_Send(v)       {while(!SPIS_SPTEF) {;} SPID = (v);}

#endif
```

To adapt the driver software for a different OLED display, modify the initialization routine. This is currently limited to only 4 bpp displays. Different color resolutions require further adaptations.

In the OLED.H file, a define, based on the manufacturer part number (for example, OS128064PK27MY0B00), is used to select which display is connected. A second definition _UP_SIDE_DOWN allows to select the display orientation.

```
// ---------------------------------------------------------------------------
// macros for display selection and orientation
// ---------------------------------------------------------------------------
// select which display to use, if necessary add new display and init sequence
//#define OS128064PK27MY0B00
#define OS128064PK16MY0A01

// select if display is mounted up side down
//#define _UP_SIDE_DOWN                  // define to rotate output 180 degrees
```

In the OLED.C file, modify the actual initialization sequence.

```
//-------------------------------------------------------------------------------
// OLED Init Sequence for Pictiva 128x64 2.7" Yellow   OS128064PK27MY0B00
// is used by OLED_Display_Init()
//-------------------------------------------------------------------------------
#if defined(OS128064PK27MY0B00)

const UINT8 _Display_Init_Seq[] = {
  OLED_SETCONTRAST,0x5D,                              // for yellow
  OLED_SETCURRENTRANGE_100,

#if defined (_UP_SIDE_DOWN)
  OLED_SETREMAP,0x52,
  OLED_SETDISPLAYOFFSET,0x4C,                         // mapping of RAM to display
#else
  OLED_SETREMAP,0x41,
  OLED_SETDISPLAYOFFSET,0x44,                         // mapping of RAM to display
#endif

  OLED_SETDISPLAYSTARTLINE,0,                         // TOP
  OLED_SETMULTIPLEXRATIO,63,                          // 64 MUX
  OLED_SETDISPLAYMODE_NORM,
  OLED_SETPHASELENGTH,0x22,
  OLED_SETROWPERIOD,0x46,
  OLED_SETDISPLAYCLOCKDIVIDE,0x41,
  OLED_SETSEGMENTLOWVOLT,0x0D,
  OLED_SETVCOMH,0x00,
  OLED_SETPRECHAGEVOLT,0x10,
  OLED_SETGREYSCALETABLE,0x01,0x11,0x22,0x32,0x43,0x54,0x65,0x76,
  OLED_SETMASTERCONFIG,0x02,                         // DC-DC 0x02 disabled, 0x03 enabled
};

#endif
```

3.1.2 OLED Driver API

Table 9 summarizes the services provides by the OLED driver:

Table 9. OLED Driver Services

Function Name	Type	Parameters	Description
OLED_Interface_Init	void	void	Initialization of MCU <> OLED hardware-interface (I/O ports, SPI)
OLED_VCC_On	void	void	Turn ON V_{CC} OLED power voltage (12 V)
OLED_VCC_Off	void	void	Turn OFF V_{CC} OLED power voltage (12 V)
OLED_Reset	void	void	Applies hardware reset ($\overline{RST}$) to OLED display
OLED_WriteCmd	void	UINT8 cmd	Transfers one byte of command data (D/$\overline{C}$ = 0)
OLED_WriteData	void	UINT8 data	Transfers one byte of display data (D/$\overline{C}$ = 1)
OLED_Display_Init	void	void	Initialization sequence for OLED display. The actual command sequence is specified as a const array for easy adaptation, for example, for a different display type.

Table 9. OLED Driver Services (continued)

Function Name	Type	Parameters	Description
OLED_Fill	void	UINT8 left UINT8 top UINT8 width UINT8 height UINT8 fill	Fills a specified area with a color: — Left/top are the x/y coordinates of the top/left corner of the area — Width/height of the area — Fill is the pattern=color to fill the area (high/low nibble specifying the color) (4bpp) Attention: 4bpp is the supported color depth. An even number must be set for left and width.
OLED_SetCanvas	void	UINT8 left UINT8 top UINT8 width UINT8 height	Sets a canvas (an sub-set of the display area) to be active — Left/top are the x/y coordinates of the top/left corner — Width/height of the area Attention: 4bpp is the supported color depth. An even number must be set for left and width.
OLED_DrawImage	void	UINT8 left UINT8 top UINT8 mask UINT8* image	draws a image (bitmap) to the specified position — Left/top are the x/y coordinates of the top/left corner — Mask allows to modify the color of the image to be modified. Darkens an onscreen button to indicate it has been pressed (see demo). — *Image is a pointer to the image data Attention: for 4bpp left and the width of the image must be an even number.
OLED_WriteString5x7	void	UINT8 left UINT8 top UINT8 cc char* text	Writes the text with the Font 5x7 to the specified position — Left/top are the x/y coordinates of the top/left corner — cc is the color of the font (high/low nibble specifying the color) (4bpp) Attention: for 4bpp left must be an even number.
OLED_WriteString8x15	void	UINT8 left UINT8 top UINT8 cc char* text	Writes the text with the Font 8x15 to the specified position — Left/top are the x/y coordinates of the top/left corner — cc is the color of the font (high/low nibble specifying the color) (4bpp) Attention: On the left side for 4bpp, set an even number.

4 Demo Functionality

The demo functionality varies slightly between the DEMO9S08QG8 and the DEMO9S08AW60 setups.

Figure 11 depicts the basic program flow and the OLED display outputs for DEMO9S08QG8 example.

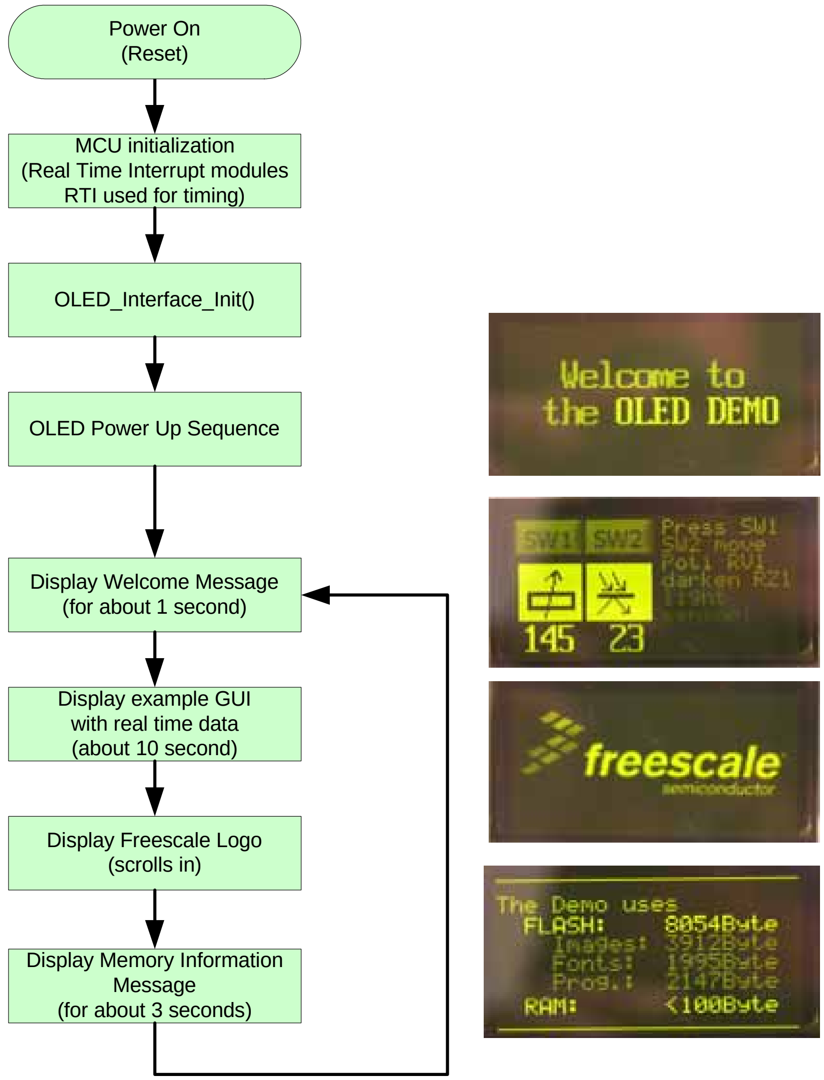


Figure 11. Basic Program Flow

4.1 MC9S08QG8 Example Demo

The demo software demonstrates the OLED display drive and displays some real-time data (see Figure 4):

- Status of two switches
- Analog-voltage reading set by potentiometer
- Analog-voltage reading of ambient-light sensor

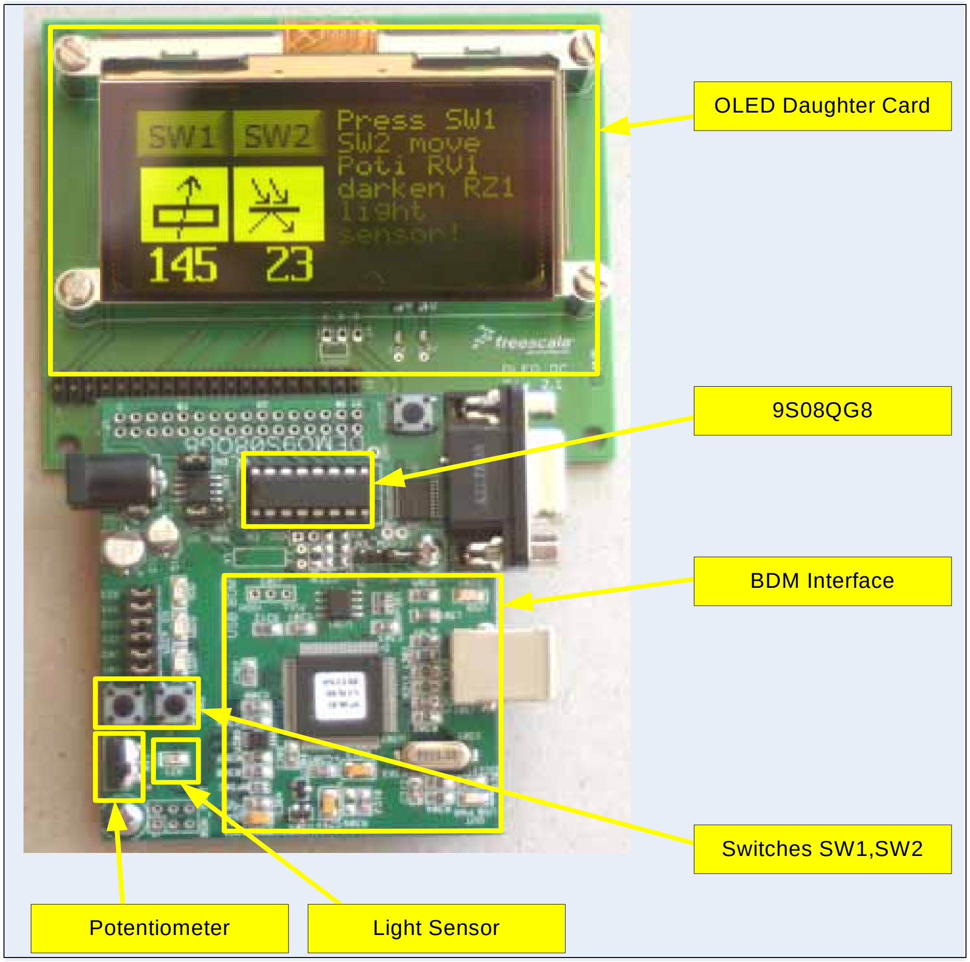


Figure 12. Demo Setup with DEMO9S08QG8 (Photo)

4.2 9S08AW60 Example Demo

The demo software demonstrates the drive of the OLED display and displays some real-time data:

- Status of four switches
- Analog-voltage reading set by potentiometer
- Analog-voltage reading of ambient light sensor
- Two analog-voltage readings of a X/Y accelerometer

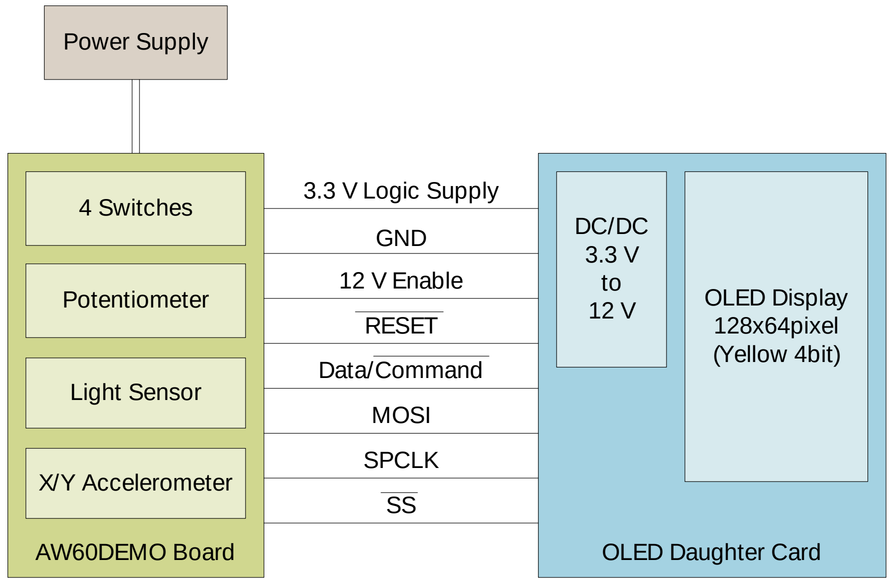


Figure 13. Demo Setup with DEMO9S08AW60 Board

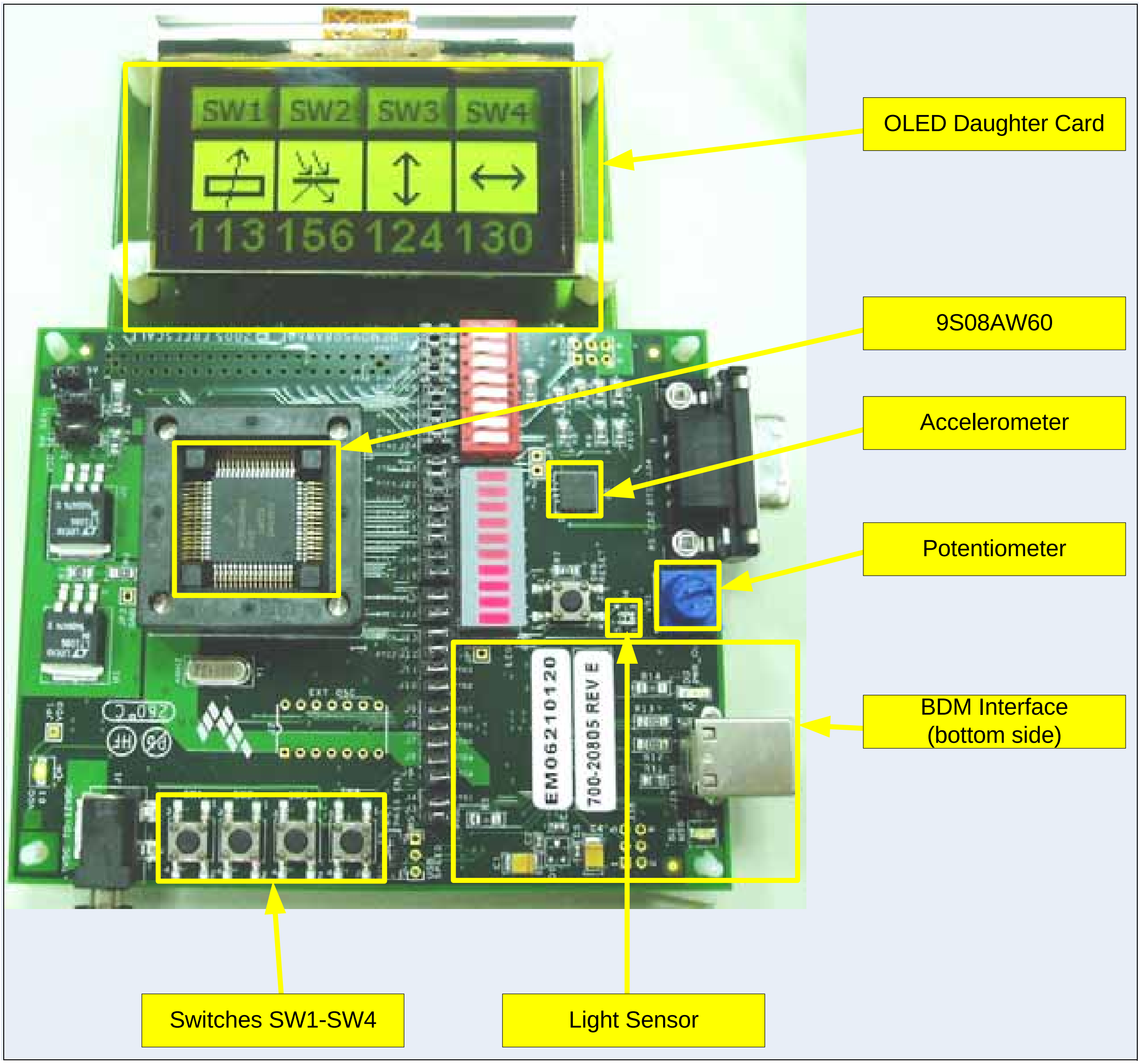


Figure 14. Demo Setup with DEMO9S08AW60 (Photo)

5 Image format and BMP2C.EXE Utility

The BMP2C.EXE utility (freeware from Freescale) is a command-line tool to convert a Windows bitmap file (*.bmp) to a C-language structure used with the OLED driver software. To convert a bitmap file, open a command window and run the BMP2C.EXE, as described below:

```
Usage:
BMP2C.exe  bmpfile outfile
    bmpfile  bitmap file to be converted
    outfile  output "C" file created
```

The tool converts bitmap pixel from the RGB format to the 4 bpp format using the following equation:

$$Pixel = \frac{R + G + B}{3 \cdot 16}$$

R, G, B are the 8bit values representing the red, green, and blue portion of the pixel.

All converted pixel are packed into a C-language structure (byte array).

```
const char Image[] = {
128,40,                              // first two byte are size in pixel (x * y)
0x00, 0x00, .........};              // pixel data
```

Each byte contains the information for two pixels.

NOTE

The software and the BMP2C.EXE requires that the source bitmap file horizontal resolution is an even number.

This structure can be included into the IMAGE.C source file (for example, in the CodeWarrior IDE).

6 Font Format

Bitmap type fonts are stored in a format where one bit represents one pixel of information — monochrome font. Each letter can then be displayed in one of 16 gray levels.

To explain how the font data is represented, the character 1 is used as an example.

For the 5x7 font, five bytes are used to store each character:

```
const char FONT5X7 [][5] =  {
 ....................
{ 0x00, 0x42, 0x7F, 0x40, 0x00 }, // 1
....................
}
```

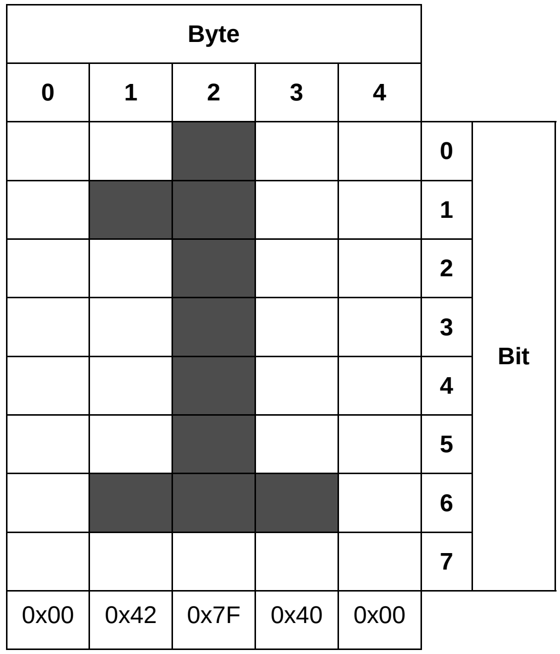


Figure 15. Number "1" in FONT 5x7 Format

For the 8x15 font, 8 times 2 = 16 bytes are used to store each character:

```
const char FONT8X15 [][8][2] =  {
.....................
{ {0x00, 0x00}, {0x04, 0x08}, {0x06, 0x08}, {0xFF, 0x0F}, {0xFF, 0x0F}, {0x00, 0x08},
{0x00, 0x08}, {0x00, 0x00} } , // 1
.....................
}
```

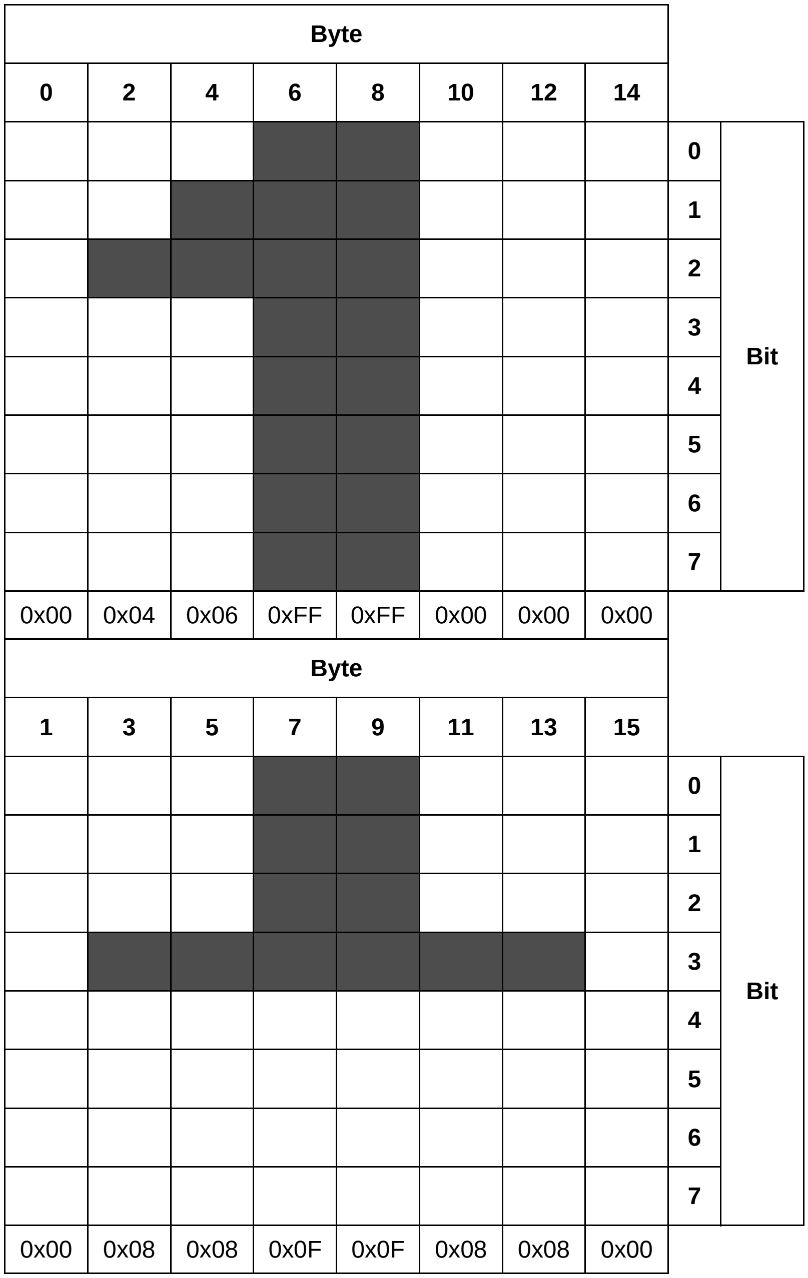


Figure 16. Number “1” in FONT8x15 Format

7 Conclusion

The provided documentation and software show how to control an OLED display with an HCS08 microcontroller, even with a limited amount of memory or GPIOs. The software can be used as the basis for other projects and further enhancements, such as:

- Support for monochrome (1bpp) OLED displays
- Support for full-color OLED displays
- Support for software SPI or parallel interface
- Scalable fonts
- Use of power-down sequence (controlled shut down)
- Implementation of a screen saver
- Power-consumption optimization
- Implementation of drawing functions, for example, lines, rectangulars, etc.(requires sufficient amount of RAM memory)
- Use of compressed storage of data to enhance memory efficiency
- Utility to generate fonts

More information on how to use an OLED in a real-world application can be found on the OSRAM Pictiva website (see Section 8, “References”).

For more information on CodeWarrior tools, see Section 8, “References”.

To get to a small memory footprint, some specific actions were incorporated (the storage of the Freescale Logo image). Only the logo itself was stored and not the surrounding background. Like this the size of the bitmap could be reduced by 37.5% from 4096 bytes (128 x 64 x 4 Bit) to 2560 bytes (128 x 40 x 4 Bit).

A second method used for the push buttons (Figure 11) is the ability to “darken” a bitmap during runtime when it is drawn (OLED_DrawImage function — see Table 9). Like this, the storage of two images—button released and button pressed—can be reduced to only one image. This saves 258 bytes (50% reduction).

Table 10 summarizes the memory consumption for the example applications. This differences are mainly caused by demonstration functionalities.

Table 10. Memory Consumption Overview

	MC9S08QG8	MC9S08AW60
Images	3912 bytes	5260 bytes
Fonts	1995 bytes	1995 bytes
Text	187 bytes	134 bytes
Program	1951 bytes	1987 bytes
Flash total	8045 bytes	9376 bytes
RAM	<100 bytes	<100 bytes

8 References

Further details can be found in the following documentation

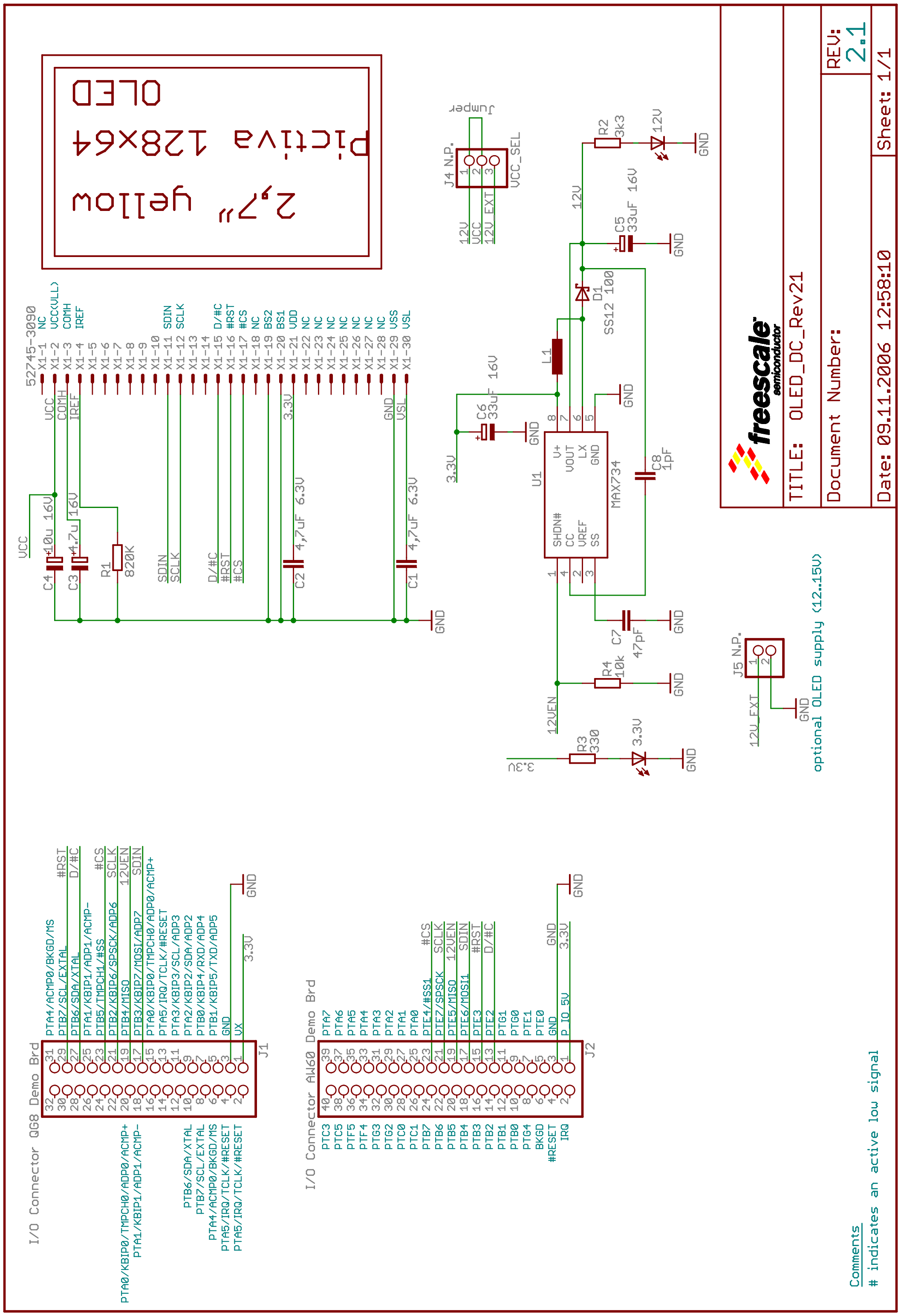
- MC9S08QG8 Data Sheet
- MC9S08AW60 Advanced Information Data Sheet
- SSD0323 Data sheet (128x80, 16 gray scale dot matrix controller)

Useful web sites:

- Freescale website http://www.freescale.com
- OSRAM Pictiva website http://www.pictiva.com
- Wikipedia website http://en.wikipedia.org/wiki/Oled
- OLED Information website http://www.oled-info.com

Appendix A OLED Daughter Card

A.1 Schematics



A.2 Layout

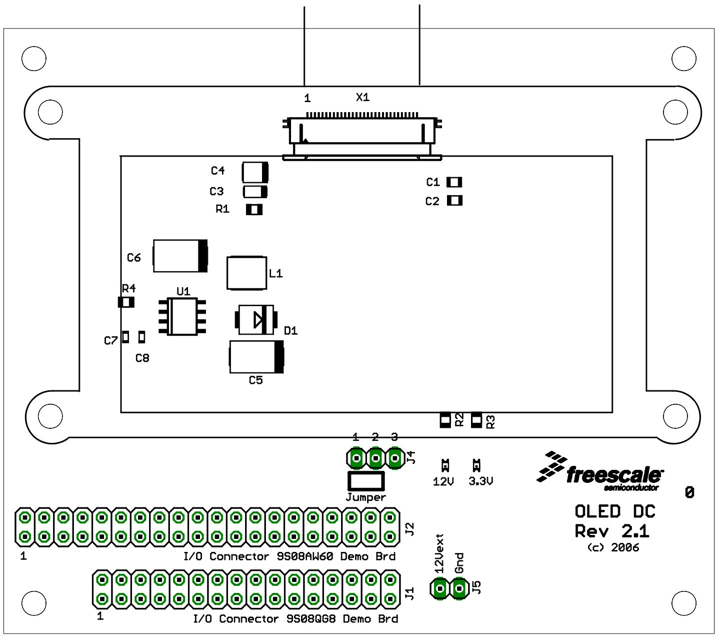


Figure 17. Placement

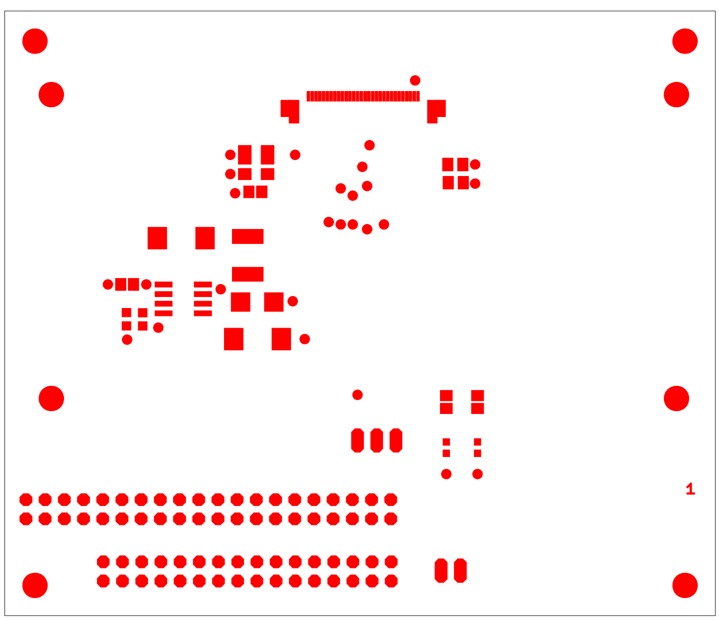


Figure 18. Solderstop Mask (Top)

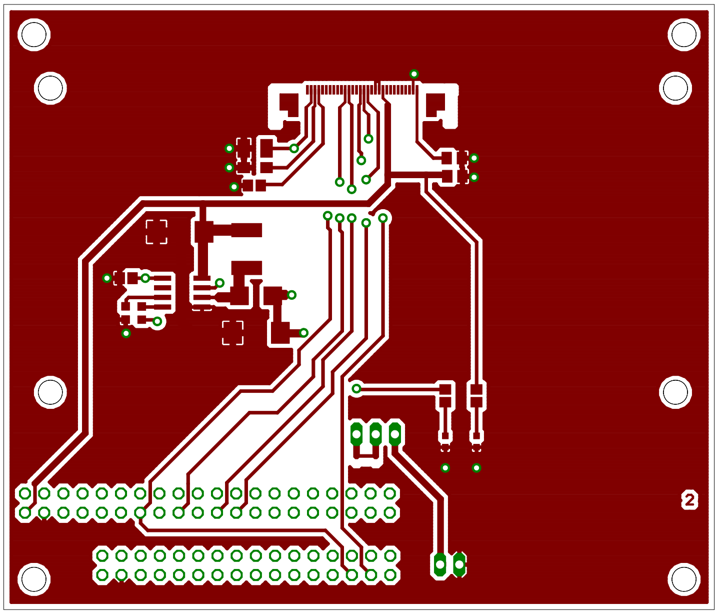


Figure 19. Copper (Top)

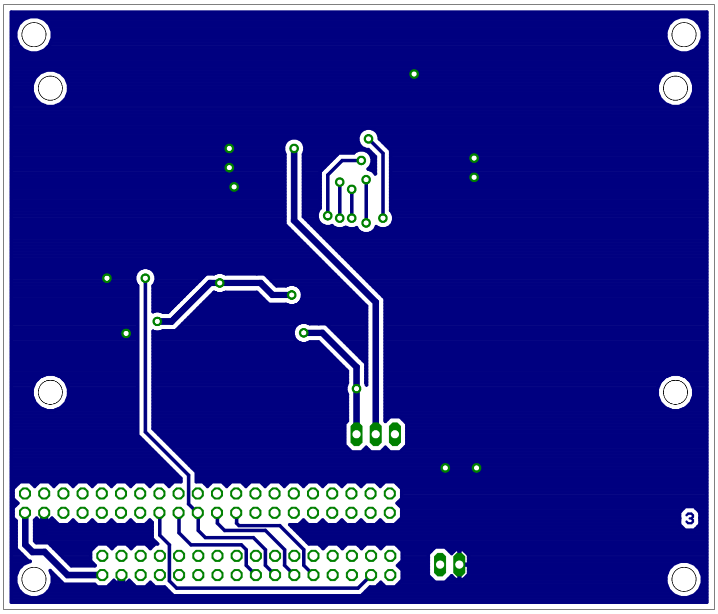


Figure 20. Copper (Bottom)

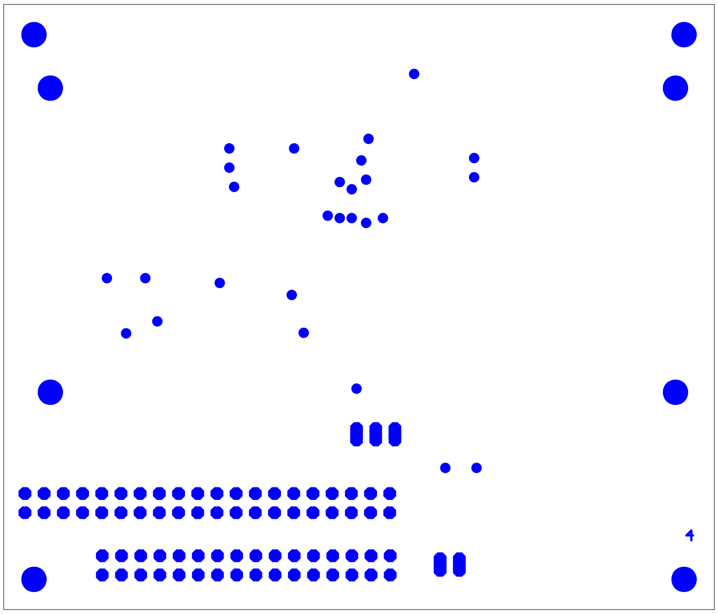


Figure 21. Solderstop Mask (Bottom)

A.3 OLED Daughter Card BOM (Bill of Materials)

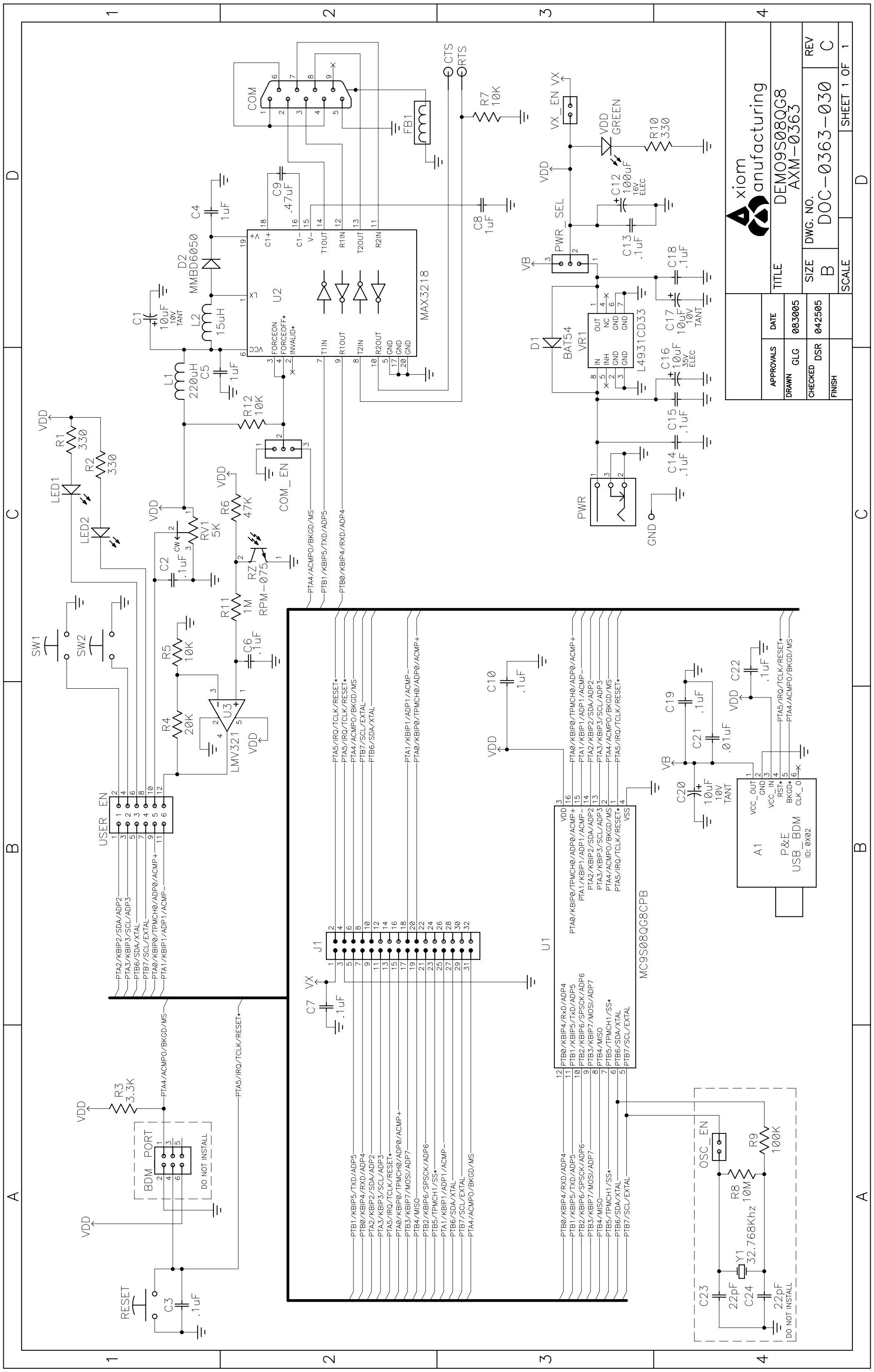
Table 11. OLED Daughter Card Bill of Materials

Quantity	Reference	Description	Supplier PartNo.	Supplier
1	U1	TPS7634ID (Texas Instruments) IC, DC/DC 120 mA, 12 V SMD-SOIC8	8458111	Farnell
2	C1, C2	Ceramic Capacitor 4.7 µF 6.3 V 20% SMD–0805	9227857	Farnell
1	C3	Tantal Capacitor 4.7 µF 16 V 20% SMC–A	197269	Farnell
1	C4	Tantal Capacitor 10 µF 16 V 20% SMC–B	498737	Farnell
2	C5, C6	Tantal Capacitor 33 µF 16 V 10% SMC–D	757858	Farnell
1	C7	Ceramic Capacitor 47pF 50 V 10% SMD–0603	722042	Farnell
1	C8	Ceramic Capacitor 1 pF 50 V 10% SMD–0603	721840	Farnell
1	D1	Schottky Diode MBRA130LT3 ONSEMI SMD–SMA (DO–214AC)	3004120	Farnell
1	J1	PinHeader 2x16 0.1 inch pitch	9729070	Farnell
1	J2	PinHeader 2x20 0.1 inch pitch	9729070	Farnell
1	J3	FCB Flex 30 pin socket 0.5 mm pitch	SFV30R1–ST	FCI
1	L1	Inductor 15 µH, max. 850 mA, 0.15 Ω Panasonic ELL6RH150M	3837403	Farnell

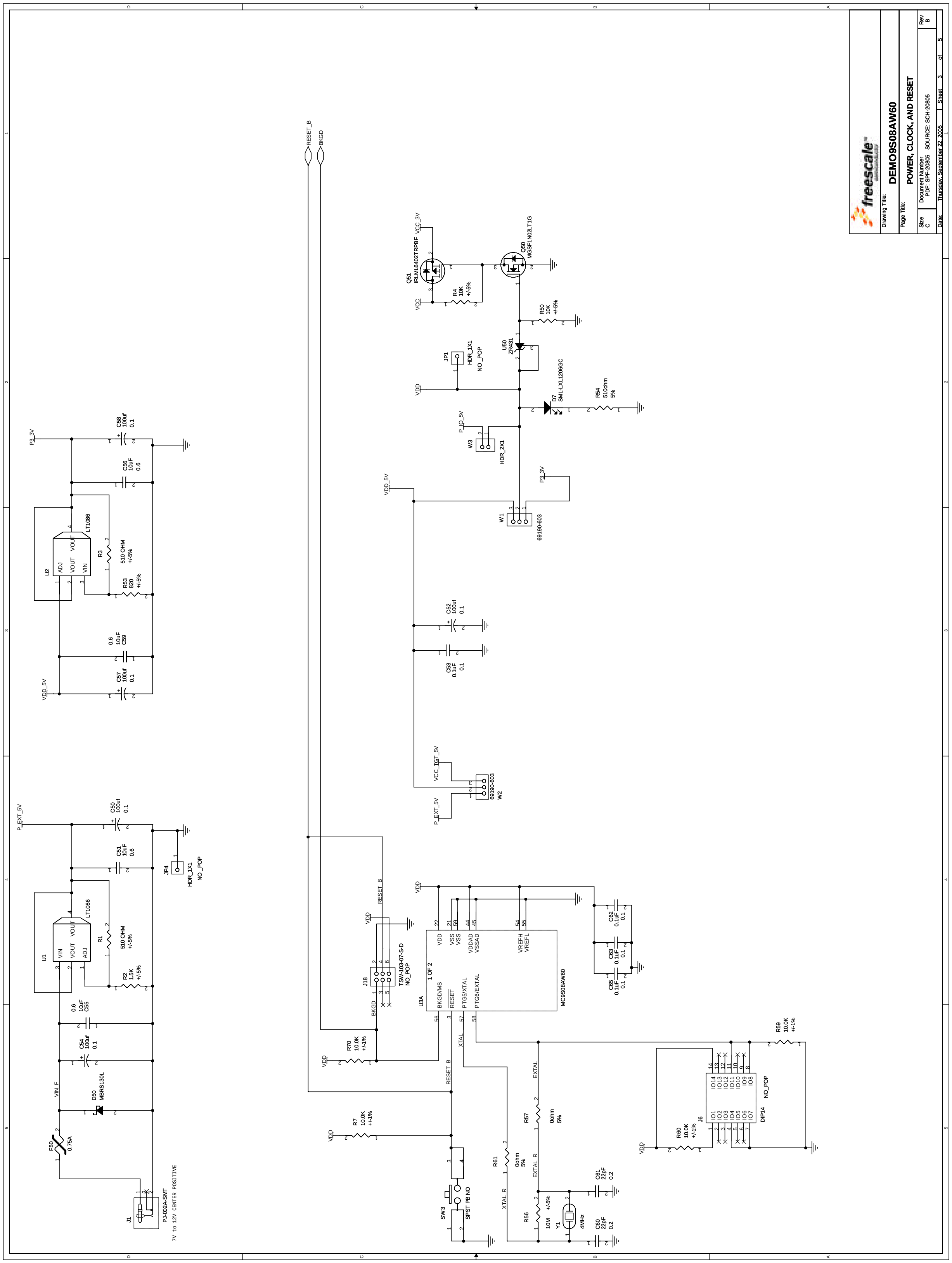
Table 11. OLED Daughter Card Bill of Materials (continued)

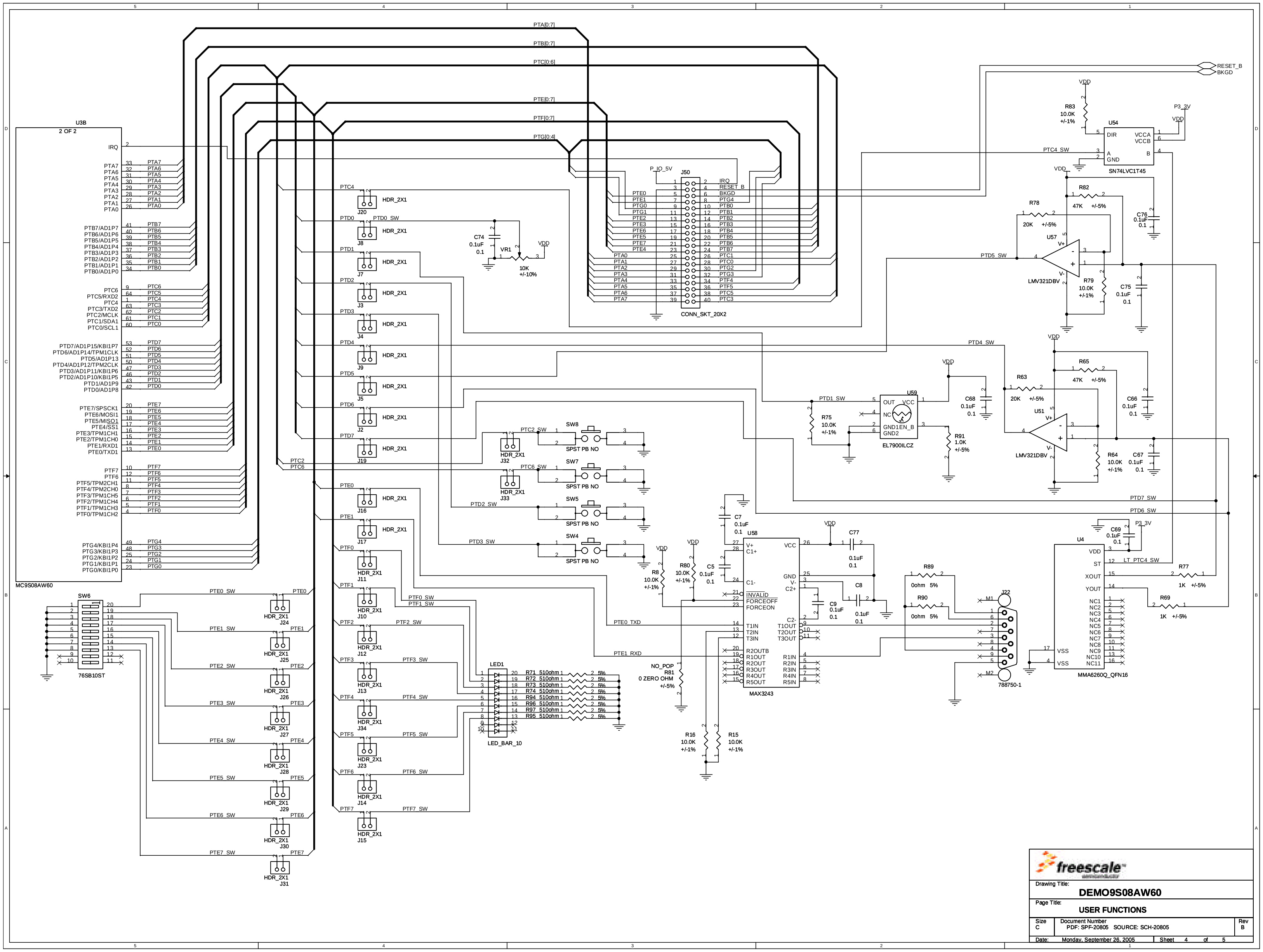
Quantity	Reference	Description	Supplier PartNo.	Supplier
1	LED1, LED2	LED green low power SMD–ChipLED 0603	8554609	Farnell
1	R1	Resistor 820 kΩ, 5%, 0.1 W SMD–0805	9334920	Farnell
1	R2	Resistor 3 kΩ, 5%, 0.1 W SMD–0805	9334319	Farnell
1	R3	Resistor 330 Ω, 5%, 0.1 W SMD–0805	9334351	Farnell

A.4 DEMO9S08QG8 Board



A.5 DEMO9S08AW60 Board





Drawing Title: DEMO9S08AW60
Page Title: USER FUNCTIONS
Size C
Document Number PDF: SPF-20805 SOURCE: SCH-20805
Rev B
Date: Monday, September 26, 2005
Sheet 4 of 5

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3415
Rev. 0
03/2007