Freescale Semiconductor, Inc.
Application Note

Document Number: AN5069
Rev. 0, 01/2015

Integrate TWR-EPD Software with MQX RTOS Based on the TWR-K21F120M Platform

1 Introduction

This application note describes how to integrate TWR-EPD software with MQX™ real-time operating system (RTOS) based on the TWR-K21F120M platform. A sample code is also provided as a reference to drive Electronic Paper Display (EPD) with MQX RTOS.

Contents

1.1 TWR-EPD

Freescale collaborated with Pervasive Displays Inc (PDI), the designer and manufacturer of electronic paper modules to develop TWR-EPD. The TWR-EPD display module mounts an ePaper display to the Freescale Tower System via the expansion port with the TWR-ELEV module. This allows the evaluation of Freescale microcontrollers and adds ePaper functionality to embedded designs. The onboard circuit supports driving PDI's 1.44 inch, 2 inch and 2.7 inch EPD panels via SPI interface.

1.2 Electronic Paper Display (EPD)

Electronic paper is a display technology that mimics the appearance of ordinary ink on paper. Unlike conventional backlit flat panel displays the emitting light; electronic paper displays the reflecting light like paper. This makes them more comfortable to read, and provide a wider viewing angle than most light-emitting display. Electronic paper display can hold static text and images indefinitely without electricity. Therefore, EPDs are commonly used in eReaders, industrial signal and electronic shelf labels.

1.3 TWR-EPD design documents and software

The most relevant design documents and software of TWR-EPD module are available on PDI's website at http://www.pervasivedisplays.com/kits/twr_epd. Engineers may refer to the "TWR-EPD User Guide" and "COG Driver Interface Timing document" before integrating EPD software solution with MQX RTOS. A bare-metal sample code of TWR-EPD module for CodeWarrior was also provided in PDI website. This application note will describe how to integrate this bare-metal sample code into MQX RTOS for multi-tasks application development.

2 Integrate TWR-EPD software with MQX RTOS

2.1 File structure of TWR-EPD software

The sample code of TWR-EPD module released on the PDI website is a bare-metal example compatible with TWR-KL25Z48M. The file structure is shown in Figure 1.

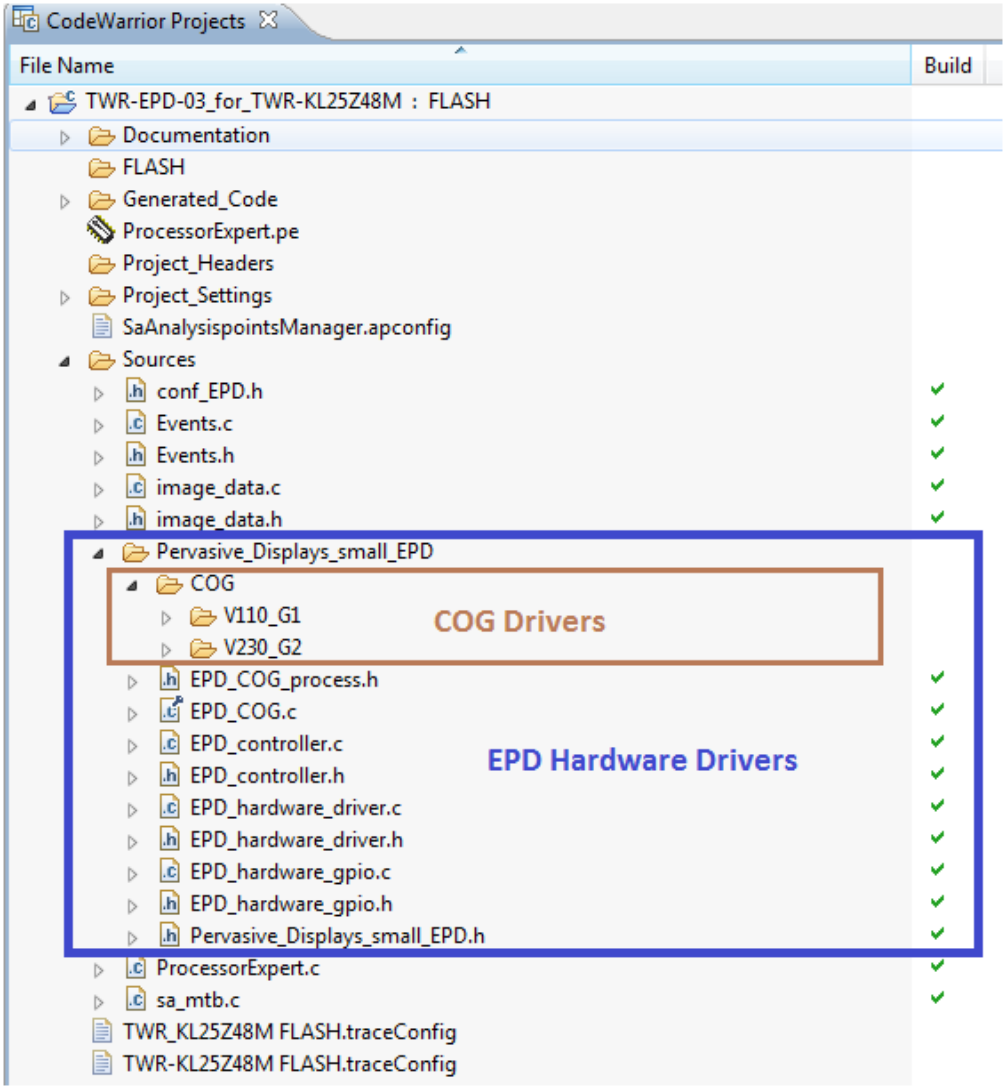

Figure 1. The file structure of TWR-EPD software

The source code in "Pervasive_Displays_small_EPD" folder can be cataloged into COG drivers and EPD hardware drivers. COG (chip-on-glass) is the driver IC for display construction where the row and column drivers are mounted directly to the glass substrate to drive TFT. PDI provided two versions of COG which are G1 and G2. The TWR-EPD display module supports the EPD panel that embedded FPL material (Front Plane Laminate that was provided by E-Ink) are version of V110 and V230. The G1 COG is combined with V110 FPL and the G2 COG is combined with V230 FPL. COG drivers provided waveform driving processes and update stage of EPD. EPD hardware drivers included the most of the hardware initialization and configuration which are SPI, PWM, GPIO, temperature ADC and EPD initialization.

2.2 Software architecture of TWR-EPD

The command interfaces were defined in EPD_controller.* files for applications to display image on EPD panel. The COG Interface and Driving Process are compatible with different platforms and systems. Therefore, the bare-metal sample code can be executed on MQX RTOS when hardware drivers were

ported as MQX RTOS I/O device drivers accordingly. The next section describes how to port these hardware drivers with MQX RTOS and Figure 2 shows the software architecture of TWR-EPD module.

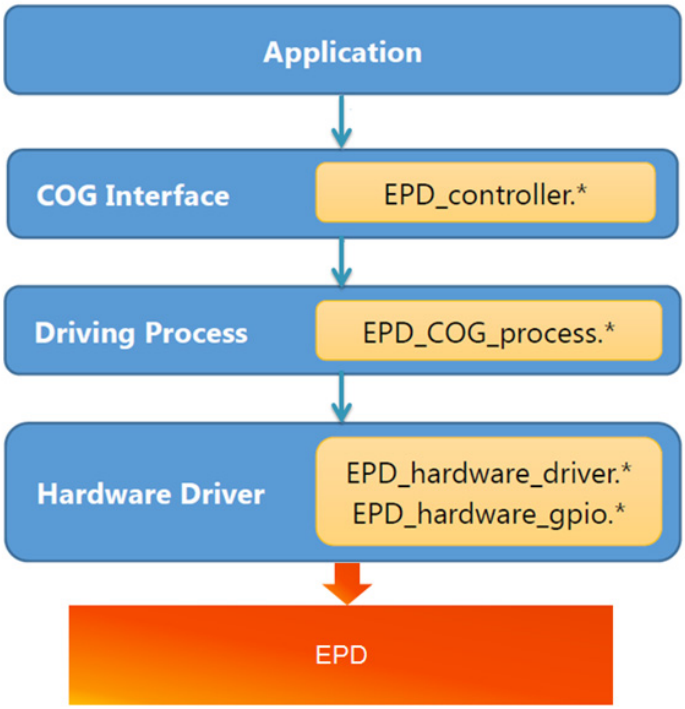

Figure 2. Software architecture of TWR-EPD module

2.3 Porting hardware drivers with MQX RTOS

Four hardware interfaces SPI, PWM, ADC and GPIO of Kinetis MCU used to control and communicate with TWR-EPD. The descriptions of TWR-EPD I/O pin function are shown in Table 1 and the according hardware components of MCU I/O function were created as shown in Figure 3. The hardware components were initialized and configured in EPD_hardware_driver.* and EPD_hardware_gpio.* files which are the key hardware driver files for EPD software porting with MQX RTOS.

Table 1. I/O pin function of TWR-EPD module

Function	Description
Vcc	Supply voltage 3.3V
SPI_SCLK	Clock for SPI
SPI_MISO	Serial output from EPD to host MCU
SPI_MOSI	Serial input from host MCU to EPD
PWM	Pulse width modulation. Square wave when EPD power on (PWM)
TEMPERATURE	Temperature sensor (ADC interface)
BUSY	COG busy pin (GPIO)
/RESET	Reset signal. Low enable (GPIO)

Table 1. I/O pin function of TWR-EPD module

Function	Description
PANEL_ON	COG driver power control pin (GPIO)
DISCHARGE	EPD discharge when EPD power off (GPIO)
BORDER_CONTROL	Border control pin (GPIO)
FLASH_CS	On board Flash chip select (GPIO)
/EPD_CS	EPD chip select (GPIO)
GND	Ground

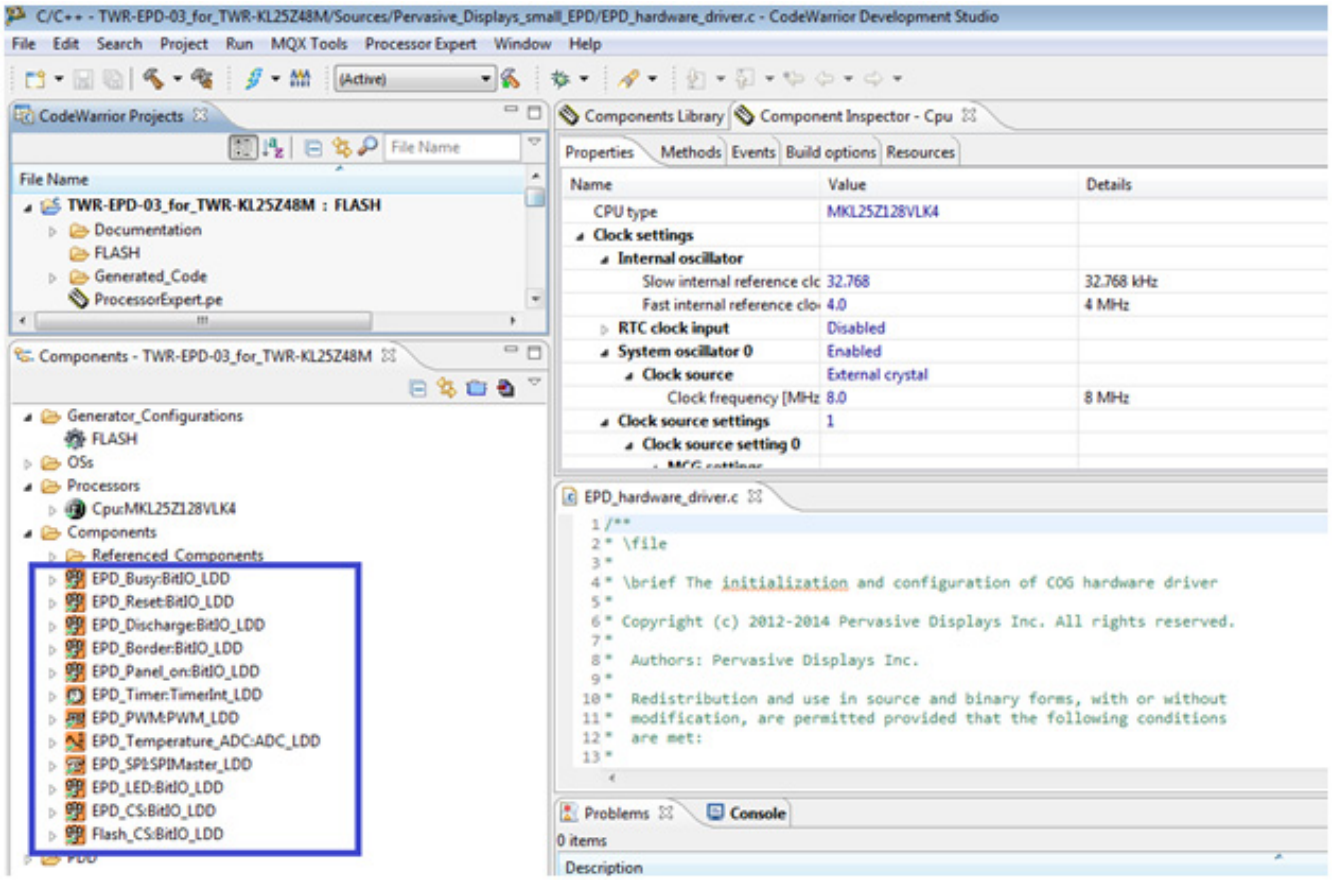

Figure 3. Hardware components of MCU I/O function

There are two methods for hardware components porting to integrate this bare-metal sample into MQX RTOS. One is to use MQX RTOS I/O device drivers instead of hardware components of Processor Expert as shown in Figure 4. It is a porting example of SPI driver which used fread() and fwrite() APIs of MQX RTOS I/O device driver to replace the API of Processor Expert to receive and transfer data via SPI interface.

Figure 4. Using MQX RTOS I/O device drivers instead of hardware component API of Processor Expert

The other method is to copy the hardware components into MQX RTOS BSP as shown in Figure 5. In the sample code of this application note, SPI and GPIO used MQX RTOS I/O device drivers; PWM and ADC used hardware components of Processor Expert for the reference.

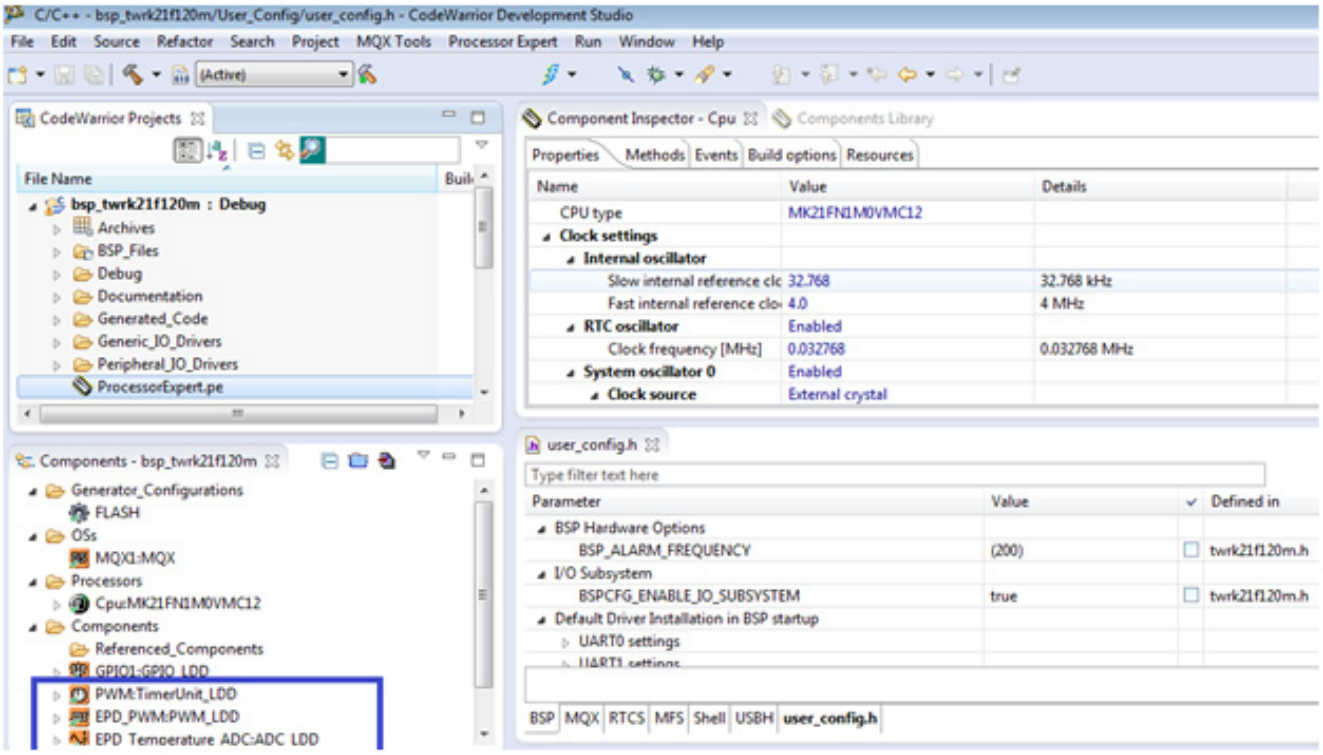

Figure 5. Copy the hardware components into MQX RTOS BSP

3 Sequence of image update on EPD panel

This section introduces the sequence of image update on EPD panel in which the related source codes are located in EPD_controller.* and EPD_COG_process_* files.

3.1 Power on and initialize COG driver

The MCU controls GPIOs and PWM for COG power on sequence, and sends SPI command to initialize COG driver. The developers must take care the timing of signal control, especially in multi-task environments.

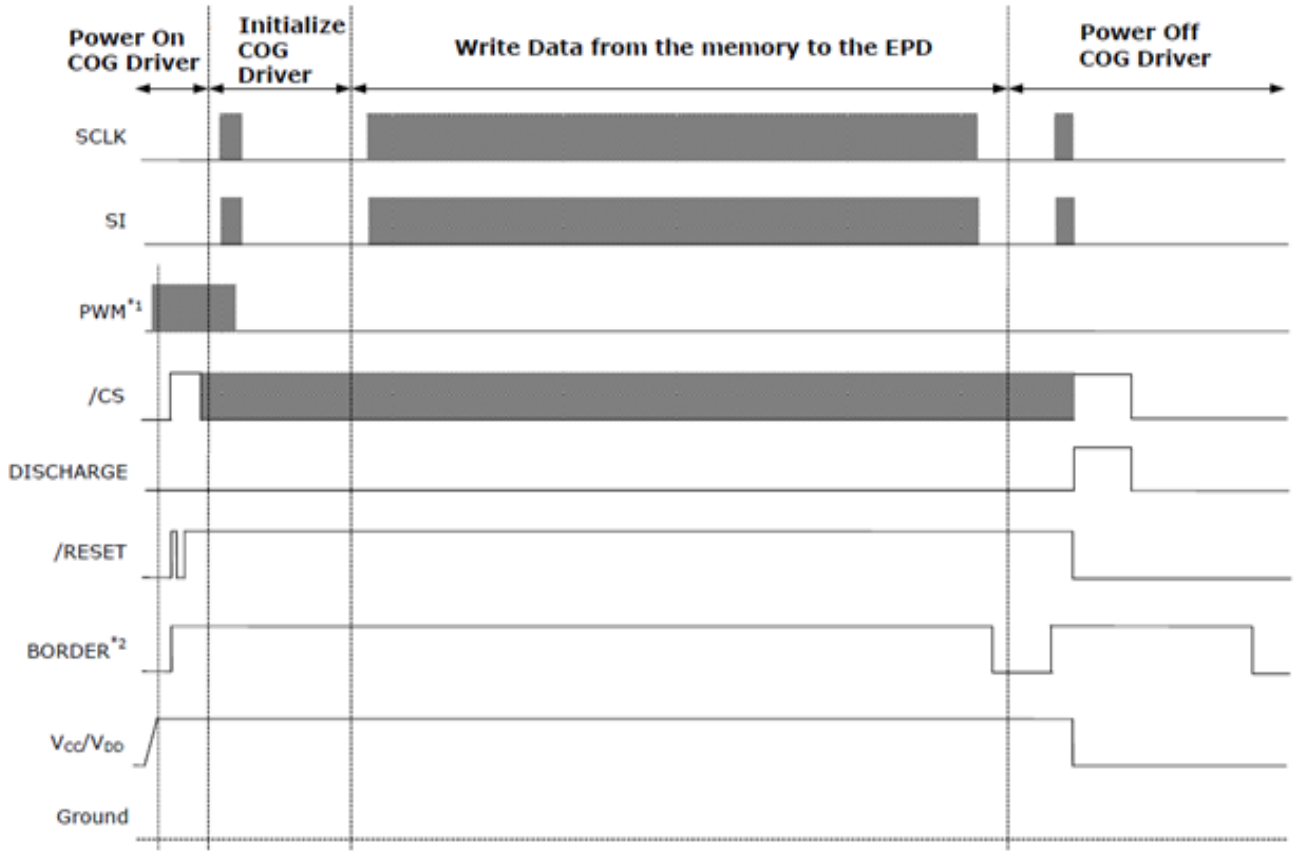

Figure 6. Signal control overview during an EPD update cycle

3.2 Write image data from MCU memory to EPD panel

The MCU must store two frame buffers in memory. One is the new image which will be displayed and the other is the old image currently visible on the panel. There are four stages for EPD panel to update the display from the previous to the new pattern:

- Inverse the previous image
- The entire panel is drawn white
- Inverse the new image
- The new image is drawn

For each stage, a frame will be written multiple times to display. PDI's design writes 6 frames of data per stage, and then 4 stages for 2" and 1.44" EPD panel to update the display from the previous to the new pattern. 2.7" EPD needs 3 frames of data per stage.

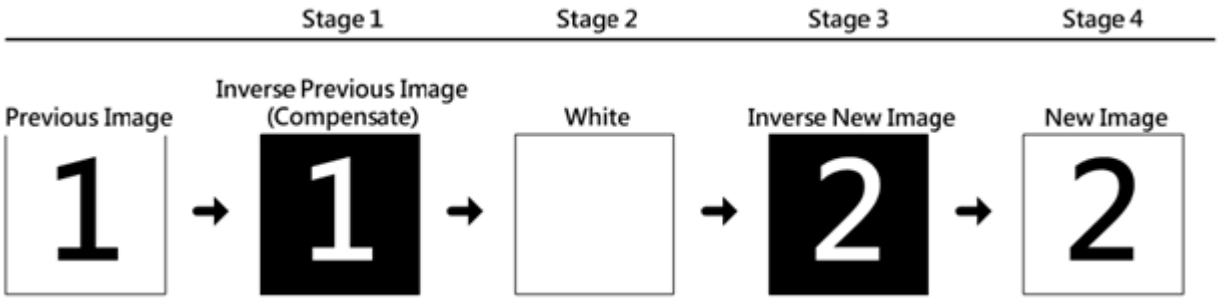

Figure 7. Four stages of EPD display update

3.3 Power off COG driver

Once the image has been updated on EPD panel, the COG driver should be powered down. After the specific power off sequence and SPI commands, EPD panel was powered off and the image could be kept on the panel without electricity.

4 Conclusion

In the end, reminding some matters needing attention when migrating this bare-metal sample to MQX RTOS.

1. The bare-metal sample code provided by PDI was based on TWR-KL25Z48M. If engineers want to port this sample code to other platforms, it should check the schematic to make sure the connection of hardware interface are correct. Then, initialize and configure the corresponding hardware components in EPD_hardware_driver.* and EPD_hardware_gpio.* files.
2. PWM is not supported in MQX RTOS I/O drivers, so it must use Processor Expert components. In the sample code of this application note, SPI and GPIO used MQX RTOS I/O device drivers; PWM and ADC used hardware components of Processor Expert for the reference. Remember to enable the corresponding MQX RTOS I/O drivers in user_config.h of BSP.
3. This reference code demonstrates EPD display update with pre-build images. If developers want to implement a graphical user interface on EPD panel, they can further integrate Freescale Embedded GUI with MQX RTOS.
4. The timing of signal control for PED image update is critical. In the multi-task environment, the task handling EPD display should have higher priority to avoid being interrupted by other tasks.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN5069
Rev. 0
01/2015