June 24, 2010

Rapid Embedded Application Development Using a New CodeWarrior Development Studio with Processor Expert

FTF-ENT-F0517

Erich Styger
DMTS & Architect
Session Introduction

- Cost Pressure, Time to Market
- Shorter Product Life Cycles
- Increased focus on SW
  - Maximum re-use of IP
  - Rapid Application Development
- Research Project (HSLU T&A)
  - Reusable Embedded Software Components
  - Cross Platform
  - Set of standard components
  - Proof of Concept
Session Objectives

After completing this session you will learn about:

• Aspects of reusability and productivity with Processor Expert Embedded Components

• How embedded real time applications can be built with Processor Expert Embedded Components
Agenda

► Concepts

► Embedded Application Components
  • BitIO
  • LED
  • LED Sensor
  • LED Communication
  • Display

► Example Applications

► Session Review and Wrap-up
Embedded Applications
Embedded Components University Research Project

► How to make embedded software components portable and re-usable across architectures?

► University of Lucerne Research Project
  • Using Processor Expert Technology
  • >60 Embedded Components developed
  • Funded by University
  • Used in student work (industry projects, partner projects)
  • Applied in CC (Competence Center) projects
  • Used and extended in Student Diploma work
How to Solve the Problem

Re-Usability

- Properties/Interfaces/Events
- Cross-Platform
- Reusing existing IP
- Availability/Deployment

Productivity

- Ease of Use
- Graphical Configuration
- Consistency Checks
- Automation/Documentation
CodeWarrior with Processor Expert

- Eclipse based
- Rapid Application Generation Tool
- Embedded Software Components
- Source Code Driver
  - Properties
  - Methods
  - Events
  - Inheritance
Embedded Components

- **Component**
  - Building block of an application
  - Functionality separated into small objects
  - Components have interface
    (similar way classes have in object-oriented programming)

- **Methods**
  - Procedures that can be executed
  - Function calls

- **Events**
  - Indication of state changing
  - Usually implementation of ISRs

- **Properties**
  - Modify/Customize object behavior
  - Set during design-time
Component Model Development Flow
Component Library
Bit I/O

Component

Properties

Methods

Events

Bit1:BitIO

Methods

Properties

Events

BitIO

Component Inspector

Bit I/O
Generic Bit I/O

**Component**

**Properties**
- Component name
- Direction
- Data
  - Declaration
  - By Bit Name
  - By Port Name
  - By Address
- Bit Number
- Direction
  - Declaration
  - By Bit Name
  - By Port Name
  - By Address
- Initialization
  - Value
  - Direction

**Methods**
- ClrVal
- GetDir
- GetVal
- Init
- NegVal
- PutVal
- SetDir
- SetInput
- SetOutput
- SetVal

**Events**
LEDs are Easy
LEDs are Easy

![Diagram of LED connection to microcontroller (μC) through an I/O pin](image)
LEDs are Easy

Anode? Cathode?

BitIO

???

BitIO
Component Wizard lets you create/modify your own components
UI to develop driver
Deploying Components to sub-teams as single file for project integration
Component Model: Properties

Component Wizard Editor - "File LED"

```c
#define %ModuleName%.%On() %LEDPin% %ModuleName%.%SetVal()
#define %ModuleName%.%On() %LEDPin% %ModuleName%.%ClrVal()

void %ModuleName%.%On(void);
```
Component Model: Methods

```c
#define LED1_Put(val) ((val) ? LED1_On() : LED1_Off())
```

Method : LED1_Put (component LED)
Component Model: Events

Component Wizard Editor - "Driver Key.drv for Language ANSI C and Component Model"

```c
435 ifdef OnKeyPressed
436 ifdef IMPLEMENTATION OnKeyPressed
437 define! Parkeys
438 include Common\\KeyOnKeyPressed.Inc
439 void %OnKeyPressed(byte keys)
440 {
441     /* Write your code here. A bit in 'keys' indicates key pressed.
442         */
443     }
444 endif %- OnKeyPressed
445 %-METHOD_END OnKeyPressed
446 %-METHOD_BEGIN OnKeyReleased
```

Freescale, the Freescale logo, AliVac, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Component Model: Inheritance

```c
238  byte %ModuleName%.% Get(void)
239  {
240  %if AnodeOnPortSide = 'yes'
241  return inherited.LEDPin.GetVal();
242  %else
243  return !(inherited.LEDPin.GetVal());
244  %endif
245  }
246  %endif /* INLINE_ME */
```

```c
#define LED3_Get() LEDPin3_GetVal()
```

**Method**: LED3_Get (component LED)

**Description**: This returns logical 1 in case the LED is on, 0 otherwise.

**Parameters**: None

**Returns**: --- Status of the LED (on or off)
Generic Access to Bit I/O

- Data & Direction
- 'By Bit Name'
- 'By Port Name'
- 'By Address'

Useful if you want to create code outside of Processor Expert System or for extended portability

Replacement of BitIO

BitIO ↔ GenericBitIO
► Configurable Pin assignment (BitIO or GenericBitIO)
► Configurable wiring/port side
LED's are Photo Diodes

Idea: "Very Low-Cost Sensing and Communication Using Bidirectional LEDs", Paul Dietz, William Yerazunis, Darren Leigh
Bidirectional LED

- **Emit**
  - HIGH
  - μC
  - LOW
  - Emit

- **Reverse Bias**
  - LOW
  - μC
  - HIGH
  - Reverse Bias

- **Discharge**
  - LOW
  - μC
  - Input
  - Discharge

Diagram shows the different states and their corresponding voltages and times:

- **Input Voltage (V_{Input})**
- **Output Voltage (V_{cc})**
- **Time (t)**
- **t_{Rev}**
- **t_{Dis}**

The diagram illustrates the behavior of a bidirectional LED under different conditions, including emission, reverse bias, and discharge.
Bidirectional LED

![Diagram of bidirectional LED](image)

- **HIGH**: Emit
- **LOW**: Reverse Bias
- **Input**: Discharge

**Graph**:
- Logic 1: Vcc
- Logic 0: 0

**Equations**:
- Logic 1: \( V \text{Input} = Vcc \)
- Logic 0: \( V \text{Input} = 0 \)

**Timeline**:
- \( t_{rev} \)
- \( t_{dis} \)

**Terms**:
- **Bidirectional LED**: A type of LED that can be controlled in two directions with a microcontroller (µC).
- **Emit**: The state when the LED is activated.
- **Reverse Bias**: The state when the LED is not activated.
- **Discharge**: The process of discharging the input.

**Notes**:
- The diagram illustrates the operation of a bidirectional LED with a microcontroller (µC) controlling its state.
- The graph shows the input voltage (VInput) for both logic states (Logic 1 and Logic 0) with corresponding time intervals (tRev and tDis).

**References**:
- Freescale Semiconductor, Inc.
LEDReverseBias

Component Inspector

Properties

Name | Value
---|---
Component name | LED7
LED on with initialization | no

HW Interface

- Anode Pin | AnodePin
- Cathode Pin | BitIO

LED7:LEDReverseBias[LEDSensor\RevBiasLED]
- Anode2:BitIO[LEDReverseBias\AnodePin]
- Cathode3:BitIO
- Get
- GetCathodeVal
- Neg
- Off
- On
- Put
- StartReverseBias
- StopReverseBias
LEDSensor

Component Inspector

Properties

- Name: Components name
- Value: LEDs
- Details: D

- Number of calibration sample: 2
- Rx Threshold: 2300
- Threshold (us): 2300
- Ambient Light Threshold: 62200
- Threshold Step (us): 100

- HW Interface
  - Reverse Bias LED: LED7

- System
  - Wait: WAIT1

LEDSensor

LEDReverseBias

BitIO  BitIO
Graphical Font: GFont

► Open Source BDF Fonts
► Mono-space and proportional fonts
► Multiple type faces (bold, normal)

```c
typedef struct {
    signed char dwWidth;
    byte height;
    signed char offsetX;
    signed char offsetY;
    PConstByte CharBMP;
} GFONT_CharInfo;

/* A descriptor describing the font. */
/* position of next char, relative to the current */
/* width of character in pixels */
/* height of character in pixels */
/* horizontal offset to left edge of bounding box */
/* vertical offset to bottom edge of bounding box */
/* pointer to character bitmap defined above */
```
Graphical Font Writer

Multiple Fonts (GFont)
Graphical Display: GDisplay

- Clear
- CirPixel
- Draw256BitmapHigh
- Draw256BitmapLow
- Draw65kBitmap
- DrawBarChart
- DrawBox
- DrawCircle
- DrawColorBitmap
- DrawFilledBox
- DrawFilledCircle
- DrawLine
- DrawMonoBitmap
- DrawOutline
- GetDisplay
- GetDisplayOrientation
- GetHeight
- GetLongerSide
- GetPixel
- GetShorterSide
- GetWidth
- GveDisplay
- Init
- NegPixel
- PutPixel
- SetDisplayOrientation
- SetPixel
- UpdateFull
- UpdateRegion
- CnError
Encapsulation of User Interface Widgets
Ambient Light Sensing Application

- Nokia Mobile LCD
- LED Sensor
  - LCD Backlight LED
- Sensing Ambient Light Level
  - Low Power LCD
  - Automatic Backlight

```c
void Backlight(void) {
    if (BackLight_GetLightLevel() < 1) {
        /* it is pretty dark here...
           turn backlight on */
        BackLight_On();
    } else {
        /* no backlight needed */
        BackLight_Off();
    } /* if */
} /* Backlight */
```
Communication Protocol

Sync

Idle

1S – 8D

Sync1

Sync2

Idle

Start

0

1

Sync

Sync1

Sync2

I1

S0

D1

Sync

Sync1

Sync2

I1

S0

D0

D1

Freescale, the Freescale logo, AllVec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
LightComm

Component Inspector

Properties | Methods | Events
---|---|---
Name | Value | Det
Component name | LC1 | D
Communication attempts | 20 | D
Tx Start Marks | 3 | D
Tx End Marks | 10 | D
Tx Post idle Marks | 5 | D

LED
- LED5
- WAIT1
- no

LED Revers Bias

Event module name
- OnStatus
don't generate...
- OnRxByte
don't generate...
- OnTxByte
don't generate...
- OnData
generate code

Event procedure name
- LC1_OnData
void Communicate(void) {
    for(;;) {
        /* read from terminal: */
        Term_ReadLine(tx.data, sizeof(tx.data));

        /* number of data bytes: */
        tx.nofBytes = (byte)strlen(tx.data)+1;

        /* send data: */
        LC_EnableRequest(&tx, &rx, 1);
    } /* for */
} /* Communicate */
LED Bidirectional Communication

Freescale MC13213 (HCS08, 8bit)
Freescale JM128 Badge Board

Freescale MCF51JM128 (ColdFire, 32bit)
Application Component Sharing

- Components shared between applications
- Low Level HAL, Drivers, middleware and application code
- Maximum reuse

<table>
<thead>
<tr>
<th>Component</th>
<th>CodeWarrior Processor Expert HAL Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEDMatrix</td>
<td>LightComm</td>
</tr>
<tr>
<td>MC34673</td>
<td>I2C</td>
</tr>
<tr>
<td>MMA7260Q</td>
<td>ADC</td>
</tr>
<tr>
<td>MPR08x</td>
<td>LEDRow</td>
</tr>
<tr>
<td>PPG</td>
<td>LightComm</td>
</tr>
<tr>
<td>GFont</td>
<td>TimerInt</td>
</tr>
<tr>
<td>GDisplay</td>
<td>uCOS_II</td>
</tr>
<tr>
<td>LEDsensor</td>
<td>BitIO</td>
</tr>
<tr>
<td>LEDRevBias</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>PDC8544</td>
</tr>
<tr>
<td>PPG</td>
<td>GenericSWSPI</td>
</tr>
<tr>
<td>uCOS_II</td>
<td></td>
</tr>
<tr>
<td>MC34673</td>
<td></td>
</tr>
<tr>
<td>MMA7260Q</td>
<td></td>
</tr>
<tr>
<td>MCF51JM128</td>
<td></td>
</tr>
<tr>
<td>MPR08x</td>
<td></td>
</tr>
<tr>
<td>PPG</td>
<td></td>
</tr>
<tr>
<td>PDC8544</td>
<td></td>
</tr>
<tr>
<td>MC13213</td>
<td></td>
</tr>
</tbody>
</table>

Freescale, the Freescale logo, AliVec, C-5, Code/TEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MXC, Platform in a Package, Processor Expert, CorIQ, QUICC Engine, SMARTMOS, TurboLink and VoIIOs are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Sensing and Door Lock Application

http://www.youtube.com/watch?v=UocPU_ctL68
Selection of Embedded Software Components

- App_SMAC_Hello
- AT25HP512
- DCF77
- FontDisplay
- FreeRTOS
- KentChLCD
- Key
- LED
- LEDByte
- LEDMatrix
- LEDReverseBias
- LEDSensor
- LightComm
- LM61B
- MaxonF2140
- MC13192
- MC34673
- MMA7260Q
- MPR08x
- OLED

- PDC8544
- PID_Int
- QuadCounter
- S65Sharp
- SimpleEvents
- SMAC
- SPHY
- SPIHCS08
- SSEC
- SWSPI
- Tacho
- Trigger
- TouchSensor
- TouchScreen
- uCOS_II
- USB_CMX
- USB_Config_CMX
- USB_HID_CMX
- Utility
- Wait
► Reusable Embedded Software Components with CodeWarrior and Processor Expert
  • Properties
  • Methods
  • Events
  • Inheritance

► CodeWarrior comes already with set of HAL components

► Cross-Platform
  • HCS08, HCS12, ColdFire, …

► Applicable for Real-Time Embedded Design
Freescale Product Longevity Program

► The embedded market needs **long-term product support**

► Freescale has a longstanding track record of providing long-term production support for our products

► Freescale offers a **formal product longevity program** for the market segments we serve

  • For the automotive and medical segments, Freescale will make a broad range of program devices available for a minimum of **15 years**

  • For all other market segments in which Freescale participates, Freescale will make a broad range of devices available for a minimum of **10 years**

  • **Life cycles** begin at the time of launch

► A list of participating **Freescale products** is available at: www.freescale.com/productlongevity
For Further Information

► Go to the Freescale Web site
  • Processor Expert - http://www.processorexpert.com
  • CodeWarrior - http://www.codewarrior.com
  • Freescale forums – search CodeWarrior http://forums.freescale.com

► Send us an e-mail
  • support@freescale.com