Using the Renesas Graphics API to Create a User Interface
Renesas Technology & Solution Portfolio

Microcontrollers
No.1 Market Share Worldwide

Advanced and Proven Technologies
System LSIs

Extensive, High-quality Portfolio
Analog & Power

Enabling the Smart Society
Renesas Technology & Solution Portfolio
Microcontroller and Microprocessor Line-up

2010

- **1200 DMIPS, Superscalar**
  - Industrial & Automotive, 65nm
  - 600µA/MHz, 10µA standby

- **500 DMIPS, Low Power**
  - Industrial, 90nm
  - 190µA/MHz, 100µA standby

- **165 DMIPS, FPU, DSC**
  - Automotive, 40nm
  - 400µA/MHz, 35µA deep standby

- **25 DMIPS, Low Power**
  - Industrial, 90nm
  - 1mA/MHz, 100µA standby

- **10 DMIPS, Capacitive Touch**
  - Industrial & Automotive, 130nm
  - 350µA/MHz, 1µA standby

2012

- **1200 DMIPS, Performance**
  - Automotive, 40nm
  - 200µA/MHz, 0.3µA deep standby

- **1200 DMIPS, Superscalar**
  - Automotive, 40nm
  - 500µA/MHz, 35µA deep standby

- **1200 DMIPS, True Low Power**
  - Industrial & Automotive, 130nm
  - 144µA/MHz, 0.2µA standby
‘Enabling The Smart Society’

Challenge:
“As everyday products become more complex, the ability to control and configure them becomes more difficult.”

Solution:
“Enabling low cost, high quality, simple to design graphical interfaces allows for intuitive use of complex products by all users”
Agenda

- Introduction to TFT Framebuffer, GAPI and Framework
  - Lab 1: Explore Raster Frame and GAPI
  - Lab 2: Understanding the Framework

- GUI translation process
  - Lab 3: Create graphical resource files
  - Lab 4: Create new screen
  - Lab 5: Creating Interaction with the Screen
Introduction to GAPI and Framework
Lab 1: Explore Frame Buffer and GAPI
RGB pixel placement and sync pulses

- Vertical Sync (1 per frame)
  - L1,P1 means “Line 1, Pixel 1”
  - L2,P1 means “Line 2, Pixel 1”
  - 1 PIXEL (or DOT)
    - RGB 5 6 5
    - 16 BITS
    - THERE ARE 130,560 (= 480x272) OF THESE ON THIS PANEL

- Horizontal Sync (1 per line, 240 lines per frame)
Start Lab 1

- Please refer to the Lab Handout and let’s get started!

---

**Renesas**

**LAB PROCEDURE**

**5L08I-Graphics Lab: Using the Renesas Graphics API to Create a User Interface – Part 1**

**RX TFT-LCD Solution**

**Description:** This objective of this lab session is to introduce you the Renesas Graphics API (GAPI) and associated application framework. We will start with an introduction to frame based graphics and basic GAPI calls, then move to more advanced GAPI calls and framework details, and finally an example of a user interface screen design process. Starting with concept artwork, we will work through the components required to create a user interface, discusses details about behavior, and illustrate how the embedded graphics are generated. Finally, we will implement the user interface on a low-cost embedded system platform.

**Lab Objectives**

1. Introduce concepts of frame based graphics and basic GAPI operation.
2. Familiarity with the graphics framework

**Lab Materials**

- Please verify you have the following materials at your lab station
  - HEW v4
  - RX Standard Toolchain v1.2
  - E1 debugger
  - RX62N based Serious SIM206

**Skill Level**

1. Familiarity with the HEW IDE and debuggers

**Time to Complete Lab**

100 Minutes
The Frame Buffer

- Data (8/8/8, 24bpp)
- Data (6/6/6, 18bpp)
- Data (5/6/5, 16bpp)

Control:
- H-SYNC
- V-SYNC
- Pixel CLK

RAM

Frame Pointer

480 × 272

Frame Pointer

Frame Pointer
Lab 1 Questions:
1. Where are the bitmaps and fonts source files being stored within the MCU?
2. How are they being located there?
3. What types of issues are you seeing with these methods of writing to the frame buffer?
Introduction to GAPI and Framework
Lab 2: Understanding the Framework
What is the “Framework”?  

- Overview of demonstration code
Start Lab 2

- Please refer to the Lab Handout and let’s get started!

**Renesas**

**LAB PROCEDURE**

**5L081-Graphics Lab: Using the Renesas Graphics API to Create a User Interface – Part 1**

**RX TFT-LCD Solution**

**Description:** This objective of this lab session is to introduce you the Renesas Graphics API (GAPI) and associated application framework. We will start with an introduction to frame based graphics and basic GAPI calls, then move to more advanced GAPI calls and framework details, and finally an example of a user interface screen design process. Starting with concept artwork, we will work through the components required to create a user interface, discusses details about behavior, and illustrate how the embedded graphics are generated. Finally, we will implement the user interface on a low-cost embedded system platform.

**Lab Objectives**

1. Introduce concepts of frame based graphics and basic GAPI operation.
2. Familiarity with the graphics framework

**Lab Materials**

Please verify you have the following materials at your lab station:

- HEW v4
- RX Standard Toolchain v1.2
- E1 debugger
- RX62N based Serious SIM20G

**Skill Level**

1. Familiarity with the HEW IDE and debuggers

**Time to Complete Lab**

180 Minutes
Framework Flow

Event Sources
(Touchscreen, RTC)

Event Message Queue

Event Manager

Screen Manager

User Screen_t and ScreenObj_t

Screen Objects
Lab 2 Review/Questions

Lab 2 Questions:
1. How are we changing the color of the slider button?
2. Do you need to use Screen Object in the callback?
3. How are we handling the display corruption issues from Lab 1?
Lab 2 Questions:
1 - How are we changing the color of the slider button?

- Determines the visual behavior of an associated object
- The usage of the scheme is dependent on object:
  - In case of button handler
    - [0]: behavior in inactive state
    - [1]: behavior in active state
  - In case of slider handler
    - [0]: appearance of the slider bar
    - [1]: appearance of the slider button
2-Do you need to use screen objects in the callback?

- No
  - Screen objects are just a template for common behavior.
  - Completely custom behavior can be defined in the callback.
  - A combination of custom behavior and screen object is typical.
  - Creation of your own “Screen Objects” is encouraged.
  - In case of “Touched” callback, no screen object is used.
3-How are we handling the display corruption?

- Via frame buffer manipulation
GUI Translation Process:
Lab 3: Create Graphical Resource Files
Original Artwork

- Your Mission: Make it “Work”
Decomposing the Problem

Touch Spots Static Text Dynamic Text

TUE 12:34 PM 22 JAN 2010

85 HR [bpm]

Set Target

75

ALL

101 TEMP [°F]

98 SpO₂ [%]

148 SYS [mmHg]

81 DIA [mmHg]
Breaking out Resources

- Background Image
- Waveform Box
- Slider
- Font “Square 721 BT” in sizes 12, 18, 24, 42
- Icons, Buttons and Regions
Converting Resources

- Button and Box Conversions
Start Lab 3

- Please refer to the Lab Handout and let’s get started!

**LAB PROCEDURE**

5L081-Graphics Lab: Using the Renesas Graphics API to Create a User Interface – Part 2
RX TFT-LCD Solution

**Description:** This objective of this lab session is to introduce you the Renesas Graphics API (GAPI) and associated application framework. We will start with an introduction to frame based graphics and basic GAPI calls, then move to more advanced GAPI calls and framework details, and finally an example of a user interface screen design process. Starting with concept artwork, we will work through the components required to create a user interface, discusses details about behavior, and illustrate how the embedded graphics are generated. Finally, we will implement the user interface on a low-cost embedded system platform.

**Lab Objectives**
1. Learn a method to create GUI graphics
2. Use GAPI and application framework on an embedded platform to create a user interface.
3. Understand callback function interaction.

**Lab Materials**
- Please verify you have the following materials at your lab station.
  - HEW v4
  - RX Standard Toolchain v1.2
  - E1 debugger
  - Inkscape and ImageMagick software
  - RX62N based Serious SIM205

**Skill Level**
1. Familiarity with the HEWIDE and debuggers

**Time to Complete Lab**
160 Minutes
Lab 3 Review/Questions

Lab 3 Questions:
1. What are several advantages and disadvantages to building the screen at runtime from components (as opposed to pre-rendering the information into the bitmaps)?
GUI Translation Process:
Lab 4: Create new screen
Resource File

- Contains all resources
- Individual components accessed at runtime
Initial Screen Creation

- Build Background
- Add buttons

Background Frame

Display Frame

Resources.bin
Start Lab 4-Almost Done

- Please refer to the Lab Handout and let’s get started!

**LAB PROCEDURE**

5L08I-Graphics Lab: Using the **Renesas Graphics API** to Create a User Interface – Part 2
RX TFT-LCD Solution

**Description:** This objective of this session is to introduce you to the Renesas Graphics API (GAPI) and associated application framework. We will start with an introduction to frame based graphics and basic GAPI calls, then move to more advanced GAPI calls and framework details, and finally an example of a user interface screen design process. Starting with concept artwork, we will work through the components required to create a user interface, discuss details about behavior, and illustrate how the embedded graphics are generated. Finally, we will implement the user interface on a low-cost embedded system platform.

**Lab Objectives**
1. Learn a method to create GUI graphics
2. Use GAPI and application framework on an embedded platform to create a user interface.
3. Understand callback function interaction.

**Lab Materials**
- Please verify you have the following materials at your lab station.
  - HEW v4
  - RX Standard Toolchain v1.2
  - E1 debugger
  - Inkscape and ImageMagick software
  - RX62N based Serious SIM205

**Skill Level**
1. Familiarity with the HEW IDE and debuggers

**Time to Complete Lab**
160 Minutes
Lab 4 Review/Questions

Lab 4 Questions:
1. Where in the source code are the six “data boxes” being handled?
2. What would be another candidate for a “handler” looking at the source in this example?
GUI Translation Process:
Lab 5: Creating Interaction with the Screen
Using Events to Signal Updates

- Use Callback to control non-graphical elements
- Use Events to signal updates to Callback functions
  - Control Target Rate from Slider
  - Display simulated data values on graph and monitors
Start Lab 5 – Last One

- Please refer to the Lab Handout and let’s get started!

### 5L081-Graphics Lab: Using the Renesas Graphics API to Create a User Interface – Part 2

**RX TFT-LCD Solution**

**Description:** The objective of this lab session is to introduce you to the Renesas Graphics API (GAPI) and associated application framework. We will start with an introduction to frame-based graphics and basic GAPI calls, then move on to advanced GAPI calls and framework details, and finally an example of a user interface screen design process. Starting with concept artwork, we will work through the components required to create a user interface, discuss details about behavior, and illustrate how the embedded graphics are generated. Finally, we will implement the user interface on a low-cost embedded system platform.

<table>
<thead>
<tr>
<th>Lab Objectives</th>
<th>Lab Materials</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Learn a method to create GUI graphics</td>
<td></td>
</tr>
<tr>
<td>2. Use GAPI and application framework on an embedded platform to create a user interface.</td>
<td></td>
</tr>
<tr>
<td>3. Understand callback function interaction.</td>
<td></td>
</tr>
<tr>
<td>- HEW v4</td>
<td></td>
</tr>
<tr>
<td>- RX Standard Toolchain v1.2</td>
<td></td>
</tr>
<tr>
<td>- E1 debugger</td>
<td></td>
</tr>
<tr>
<td>- Inkscape and ImageMagick software</td>
<td></td>
</tr>
<tr>
<td>- RX62N based Serious SIM205</td>
<td></td>
</tr>
</tbody>
</table>

**Skill Level**
- Familiarity with the HEWIDE and debuggers

**Time to Complete Lab**
- 160 Minutes
Lab 5 Review/Questions

Lab 5 Questions:
1. What happens if you use the FileFind framework call to search for a resource and it is not located?
Questions?
‘Enabling The Smart Society’

**Challenge:**
“As everyday products become more complex, the ability to control and configure them becomes more difficult.”

**Solution:**
“Enabling low cost, high quality, simple to design graphical interfaces allows for intuitive use of complex products by all users”