Extreme Makeover with the RX600: Adding Touch/Graphics to Your Product

Terry West, CEO
Serious Integrated Inc.
Terry West, CEO

As CEO and co-founder, Terry West leads the team at Serious Integrated, Inc.

Previous Experience:

- As the first employee of RIM in 1988, Mr. West fueled the early growth of RIM while developing the technologies and customer relationships leading to the RIM BlackBerry.
- At Intel, he managed an XScale® (ARM) processor internal startup from first shipment to $85M run-rate in less than 3 years, led a $300M acquisition for the company, and developed and marketed numerous software+silicon+tools platform technologies for the embedded market.
- Recently at Microchip, Mr. West launched their flagship line of 32-bit microcontrollers and directed a comprehensive reengineering of their global customer-facing marketing presence.
Renesas Technology & Solution Portfolio
Purpose

- This workshop covers the major steps & decisions when developing a complete graphic/touch front panel on the Renesas RX MCU Family.
- RX MCUs: best in class for cost effective engine graphic/touch front panel applications
- Graphic/touch: hard engineering problems!
  - what software?
  - what OS?
  - what GUI development strategy?
  - will the HW support the SW?
  - will the GUI end up looking great?
Agenda

- An Agility & Change-Resiliency Systems Problem
- Hardware Design and Production
  - LCD
  - Processing & Memory
  - Teardown: Serious SIM205 Intelligent Module
- Software & GUI Design
  - Storyboarding and digital assets
  - SW Architecture and approach
  - Renesas Graphics API (GAPI)
  - Serious Human Interface™ Platform (SHIP)
- Lab: Build a GUI in minutes using SHIP
- Q&A
Your Multi-Faceted Challenge

Marketing

Manufacturing

Engineering

a Viable
Solution

Your system design must...

• Help marketing & sales w/fast time to prototype
• Empower marketing with GUI change agility
• Enable purchasing across supply line changes
• Deliver a great GUI at a viable cost
Hardware Design

- LCD
- Processing & Memory
- Teardown: Serious SIM205 Intelligent Module
- Serious Human Interface™ Platform(SHIP)
LCD Selection: Many Issues

- LCDs in the Embedded World
- Touch Technology
- Drive Technology
- Quality
- Brightness/Luminance
- Interoperability
- Operating Range and MTBF
LCDs in the Embedded World

- Few long-lifetime providers
- Some long-lifetime models, sizes
  - 4.3/7.0 ok – avoid 5/6/8! – careful on 3.x,10.x
- Must examine volume drivers
- Many companies in supply chain:
  - Chips, Glass, Assembly, Materials
- Chip change $\rightarrow$ SW change
- Glass change $\rightarrow$ PCB change

Plan for LCD Refreshes in HW, SW, and Purchasing
Touch Technology

- Resistive still cheapest, easiest
  - A2D w/SW or HW controllers

- Capacitive problematic
  - Very hard systems integration
  - Reliable controllers very expensive
  - Controller lifetime highly questionable
  - Patents, Patents, Patents

- No-touch with button/LED wraps still common for “harsh” environments
Drive Technology

- RAM Buffer On Glass + Parallel Port
  - Common on <4” LCDs
  - Need to check timing carefully
  - Don’t believe the data sheet

- Direct Drive to RGB
  - Great looking GUIs at very low cost
  - Common on WQVGA (4.3” typically)
  - Some limits on GUI capabilities

- Graphics Controller to RGB
  - High performance graphics
  - Many accelerator options (2D, Video, Camera overlay,...)
  - Harder to contain costs and SW design portability
MCU/MPU? 16/32?

- Go 32-bit or go home
  - Software is far more than you expect
  - You *will* want an RTOS
  - Graphic objects make 16-bit addressing/data problematic

- Think three times before jumping to an MPU
  - Much higher HW costs (power supplies, clocks, ....)
  - Higher power
  - Larger software inevitable

- For <= WQVGA, the RX600 is one of the most cost-effective and powerful choices
 MCU/MPU Considerations

- Choose the best for the resolution, GUI expectations, and “other” CPU requirements
- Code & memory bandwidth is important!
- I/O bus can make/break GUI performance
  - Match I/O bus to optimized LCD Drive
  - E.g. RX600 SRAM bus with OTM4001A RAM-on-LCD controller
  - E.g. RX600 SDRAM bus with Direct Drive
Memory Considerations

- RAM Bandwidth drives GUI performance
- You **can** build a complete GUI with only 128KB RAM + RAM-on-LCD but your software will be challenging
- More RAM $\rightarrow$ more performance, easier software, higher cost
- Use fast-read bulk FLASH storage for images, etc.
  - Micron N25Q series is impressive!
SIM205-A00 Teardown

- 100MHz Renesas RX621 MCU 512K Flash, 96K RAM
- LiPo Battery charger with Battery/Switch connector
- Battery Backed RTCC with EEPROM and SRAM
- USB device port (can power whole module!)
- Renesas E1-type standard debug port
- SPI, I2C, USB, CAN, UART, GPIO, DACs, ...
- 8MB (64Mbit) SDRAM
- 4.3" WQVGA 480×272 TFT Resistive Touch Display
- 8MB (64Mbit) Serial FLASH
- Audio w/0.7W Speaker
- Embedded USB Host Port
- System Temp Sensor

Handles Rich Graphic GUIs, including alpha blending. Lower cost/feature options & volume pricing available.
Serious Integrated Modules

- “Your new front panel. Done.”
  - Off-the-Shelf fast prototyping
  - Off-the-Shelf cost-effective production
- Balanced LCD, MCU, memory, peripherals
- Best in class MCU technology
- Easily connect existing OEM system
- Resiliency across LCD changes
Software & GUI Design

- Storyboarding and digital assets
- SW Architecture and approach
- Renesas Graphics API (GAPI)
- Serious Human Interface™ Platform (SHIP)
Storyboarding & Managing Expectations

- Marketing teams need to be level-set up front
  - It’s not an iPhone/iPad... not in BOM cost expectations
- Few marketing teams have designed GUIs
- PowerPoint/Adobe screen-by-screen storyboard: critical tool for agreement on GUI capabilities
  - Drives MCU, LCD,... needs and BOM cost
  - Drives software & GUI strategy
  - Drives cost and time to completion
- Rapid GUI tools can turn a storyboard into a functioning simulation very quickly
Merged or Partitioned Architecture?

- Merged HW & SW
- Partitioned HW: Front Panel + Control Unit (CU)
  - C-based GUI on Front Panel + Existing SW on CU
- Partitioned HW: Front Panel + Control Unit (CU)
  - High Level Rapid GUI + Existing CU SW
Why Partition The Problem?

- GUIs: big, complex, different than control SW
  - Most teams don’t have experience/capacity
- Control SW: your product/company specific
  - Complex low level code with legacy code/architectures
  - Networking & connectivity: many options
  - OEMs generally have a long history in their control SW
- Combining GUI + Control: big custom engineering problem – initially and for on-going maintenance
- Keeping these separate: focus on differentiation and leverage the Serious off-the-shelf GUI system

“Nothing is particularly hard if you divide it into small jobs” – Henry Ford
Avoid the Zone of Death!

Traditional GUI Development; Merged HW

Zone of Death

Serious Human Interface™ Platform

Unaware

Aware

Capable

• Knowledge
• Capacity
• Time
• Budget

Incapable
Merged C-Based Design: Resources

- AE Support
- Reference Code
- App Notes
- GAPI

- Pre-Ported OSs
  - FreeRTOS, Micrium, Segger,…
- No Cost Drivers
- Full BSPs
- GAPI

SW Partners
- Micrium uC/GUI
- Segger emWin

Best in Class Ecosystem for Getting Started
The Serious Human Interface™ Platform (SHIP)

Rapid GUI Development and Deployment System

GUI Authoring Tool
The SHIP Total IDE: SHIPTIDE

Target Hardware
Serious Integrated Modules

GUI data file created by SHIPTIDE
The “cargo”
SHIPTIDE: the SHIP Total IDE

- Manages all aspects of GUI creation
- Fast modify/upload/try cycles
- Powerful yet simple events, actions, layout capabilities
- Minimal coding
- Fosters collaboration between graphics artists and GUI engineer

WYSIWYG Layout

Resources (images, fonts, ...)

Events

Actions

Cargo Building

HW Upload/Run

PC MAC Linux
SHIP Deployment Example

- OEM had next-day customer demo
  - A comms management front panel
- 4 interactive screens with simulated functions
  - They supplied digital media assets (gfx, storyboard,...)
- *Serious* custom services team: demo GUI in 4 hrs
SHIP Runtime GUI Engine

- Portable & Scalable
  - Multiple platforms from 128KB – 64MB RAM
  - RAM-on-Glass, Direct Drive, and full Graphics Controllers

- Optimized
  - Tuned to each CPU/Graphics architecture for the best GUI performance

- Robust
  - Significant run-time code checking
  - SAIL virtual machine puts customer “code” in safe sandbox
SHIP Summary

- Extremely rapid GUIs in days, not years
- Modern looking front panels – minimal coding
- Available with on all Serious Integrated Modules
- Prototype ready, Production Worthy
Takeaways

- The RX600 MCU family is an outstanding engine for TFT graphic/touch front panels up to WQVGA
- LCD selection + change management is hard
- Understanding your GUI needs, SW architecture & development path is critical to your HW design
- Getting something demonstrable quickly is imperative to validate your assumptions
- Developing a full-custom C-based GUI is the right choice for some designs

- Want fast TT-Prototypes & Production, with Marketing/Purchasing friendly solutions? Get Serious!
Need More Info?

- Visit our corporate website: seriousintegrated.com
- Visit our community website: mySerious.com
- Contact your local Arrow Electronics FAE
- Contact your local Serious Manufacturer’s Rep
  - see website for list
- Contact us directly: support@mySerious.com
Backup Slides
LCD Quality

- All assemblies are not equal
- Infant mortality, MTBF, etc. highly dependent on quality assembly
- Wide variation in quality processes
- Some LCD vendors are primarily brokers

Know Your Manufacturers’ Supply Chains!
Brightness/Luminance

- Measured in cd/m² aka **NITS**
- Lose up to 80 NITS with a touch layer
- Your eyes aren’t linear: small screens need smaller NITS to look similarly bright

<table>
<thead>
<tr>
<th>NITS</th>
<th>Usage Model</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;100</td>
<td>Night, Dark Area</td>
<td>$</td>
</tr>
<tr>
<td>100-300</td>
<td>Indoor</td>
<td>$</td>
</tr>
<tr>
<td>300-500</td>
<td>Bright Lit Indoor</td>
<td>$$</td>
</tr>
<tr>
<td></td>
<td>Indirect Outdoor</td>
<td></td>
</tr>
<tr>
<td>600-800</td>
<td>Normal Outdoor</td>
<td>$$$</td>
</tr>
<tr>
<td>800-1000</td>
<td>Strong sunlight readable</td>
<td>$$$$$</td>
</tr>
</tbody>
</table>
Interoperability

- Some LCD vendors have compatible models
- Some supply line resilience

HOWEVER...

- Often same base glass supplier
- Same chip supplier
- Sometimes even same actual module supplier

Plan for LCD Refreshes in HW, SW, and Purchasing
Operating Range and MTBF

- 0-60C/70C common
- -20 to +60/70C on some models
- Below -20C is problematic
  - LCD’s crack, layers separate, ...
  - Need a warming device
  - Many (most) heaters are unreliable
  - Look for field-proven solutions

- LED Backlight solutions vary widely
  - Check the backlight typical hours of operation
  - Use backlight-off/PWM to extend lifetime
Lab Workbook

- Step 1: Basic Operations – 20 minutes
- Step 2: Adding Actions and Audio – 20 minutes
- Step 3: Advanced Features – 20 minutes
Open a Project
SHIPTIDE Basic Elements
Setting the <page> Background Color
Background Color Now Set
Adding A Logo (part 1 – adding a <box>)}
Adding a Logo (Part 2 – attaching the image)
Centering the Logo
Adding a Settings Button
Positioning the Button
Adding the New Settings Page
Ensuring the Right Pages are Visible at Boot
Adding Text to the Page

- Select the 'text' option from the 'Add New' menu.
- Insert the text 'To Be Implemented' into the page.
- Adjust the text properties as needed, such as font, size, and alignment.
Making the Button Listen to the Touch Screen

![Diagram showing the setup for making a button listen to a touch screen.]

- **Type**: listener
- **Name**: settingsListener
- **object**: <null value>
- **xmlLine**: <listener name="settingsListener" condition="TOUCH0.event.e == TOUCH.DN" listeningto="TOUCH0.event.e" />
- **iid**: No Value Set
- **xname**: enabled
- **enabled**: auto(true)

The diagram illustrates the setup process for enabling the button to respond to touch events.
Adding the Action to Change Pages

1. pHome.visible = false;
2. pSettings.visible = true;
Exporting the Cargo File
Connecting to the Hardware in SHIPTIDE
Uploading the Cargo to the SIM205 hardware
The Results..

To Be Implemented
Lab Workbook

- Step 1: Basic Operations – 20 minutes
- Step 2: Adding Actions and Audio – 20 minutes
- Step 3: Advanced Features – 20 minutes
Loading Step 2
Finding and Fixing the Error
Copying Pages
Creating the Button Bar for Enhanced Layout
Making the Button Bar with Layout Rules
Moving the Buttons into the Button Bar
Finishing the Button Bar
Adding Back Buttons to the Pages
Replacing the Image
Relocating the Button: More Layout Fun
Making the Back Button Work
Replicating it to other <page>s
Lab Workbook

- Step 1: Basic Operations – 20 minutes
- Step 2: Adding Actions and Audio – 20 minutes
- Step 3: Advanced Features – 20 minutes
Step 3: Even More
The shiplaunch Variable and Audio Clips
Pushbutton → LED Control Example

```java
platform.GPIO.LEDRO.value = false;
```
Backlight Kick and Global Touch Events

1. This "kicks" the glass backlight timer so that the glass does not go dark.
2. A touch listener at the very top level under the display gets woken on every touch but does not "capture" the touch.
3. `platform.GLASS0.kick = !platform.GLASS0.kick;`
Constants, Variables, and Auto-

```cpp
1 // This sets the nearest-parent-page's visible property
2 // based on the state variable being for this page.
3 //
4 // So if currentPage.state is my page, my page gets visible=true.
5 // If currentPage.state is not my page, my page goes invisible.
6 //
7 // That way anyone in the GUI can say
8 // currentPage.state = currentPage.<page constant>;
9 //
10 // And instantly the correct page will appear and any other page
11 // will go invisible.
12 //
13 // page.visible = (currentPage.state == currentPage.WEATHER);
```
The shiplanguage Built-in Variable

dynamic text follows

shiplanguage = "enUS";
Multi-Language Text Resources
Multi-Language Text Resources

Insufficient power to SIM205 for Audio. See Users Manual for more information.
Adding (yet another) Button
Starting an Audio Clip

```javascript
1
2 platform.AUDIOOUT.clip = aHeartMonitor;
3 platform.AUDIOOUT.playing = true;
4
```

Insufficient
See Users Manual
Serious Value Proposition

- Cost effective hardware for prototyping & production
  - Off the shelf Serious Integrated Modules (SIMs) 3” to 10”
  - Cost effective for low/medium volume production

- Best traditional C programming starting point
  - Pre-ported OSs, Drivers, GAPI, ...

- Serious Human Interface Platform™ (SHIP) v4
  - Ultra fast GUI prototyping and development
  - PC-based tools & runtime embedded engine
  - Pre-installed binary on every SIM
  - Licensable for customer hardware
  - Beta release now
Summary

- The RX600 MCU family is an outstanding engine for TFT graphic/touch front panel designs up to WQVGA.
- Understanding your GUI needs, software architecture and development strategy is critical to solidifying your hardware design.
- Getting something demonstrable quickly is imperative to validate your assumptions.
- Partitioning the architecture can deliver more products, faster.
- Many designs can get completed 10x+ faster/cheaper using high level GUI development systems like SHIP™.
Questions?
Please Provide Your Feedback...

- Please utilize the ‘Guidebook’ application to leave feedback

or

- Ask me for the paper feedback form for you to use...