ENABLING LOW-POWER, SMARTPHONE-LIKE GRAPHICAL UIS FOR RISC-V SOCS

5th RISC-V Workshop, Mountain View, CA
Michael Gielda, mgielda@antmicro.com
• 20+ people
• innovations in technology
• main business in services
• own platforms & tools
• IoT, CPU/GPU, HMI, FPGA
AXIOM OPEN SOURCE 4K CAMERA
GUIs
PREMISE

- industrial/embedded GUIs mostly look bad
- terrible user experience
- designed by engineers, hard to improve
- for better UIs have to jump to Android/Linux, no middle ground
- people care about what they can see and touch
- what if we could have needs drive relevant HW?
TODAY

As much a graphics design problem as it is a user experience problem.
ORIGINS

- GUI produced in close cooperation with designers familiar with Web/mobile development
- needed possibility to remotely load GUI via WiFi from base station
- mobile-like experience was extremely important
- end result (on STM32F4) is really good, current work focuses on issues outside the GUI
STARTING POINT

• get best of both worlds (good looks as well as better security, lower power, boot speed etc.)
• we do not need to reinvent the wheel - we have the example of mobile UIs
• with the right approach and tools, embedded GUIs can be beautiful too
LESS IS MORE

"I would have written a shorter letter, but I did not have the time."

-- Blaise Pascal
SMALL IS BEAUTIFUL

• smaller system (RTOS rather than Linux/Android) is easier to verify and understand
• boots immediately
• less prone to bugs
• easier to update and maintain
OUR INSPIRATIONS
ANTMICRO GUI LIBRARY - MAIN FEATURES

• mobile-like look & feel on MCUs (using minimal HW infrastructure)
• object-oriented
• standardised widgets
• ease or programming / design via e.g. XML parsed GUIs
• possibility of remote updates
POTENTIAL APPLICATIONS

• smart home control devices
• smartwatches
• coffee makers
• HVAC control displays
• ticket/validation/parking machine
• smart appliances
• 3D printers
• smart intercoms
TECHNICALITIES

• written in C++
• support for layers, formats
• own font engine with kerning, anti-aliasing
• main event loop is outside library
• full control over drawing
• PNG support & manipulation (e.g. format conversion) using libpng
• JPEG support using libjpeg
PROGRAMMATIC GUI CREATION

Manager* DispMan;
DispMan = &Manager::GetInstance();
...
Manager::Initialize(LCD_WIDTH, LCD_HEIGHT, 2, 2, false);
DispMan->AddImageContentToContainer(
   "icon", new ImageContent(
      ImageContent::FromPNG("/rom/images/icon.png")
   )
);
DispMan->ShowPopup("Hello world!", "Welcome to the GUI library", 1000);
XML PARSING

- build a GUI from XML and quickly rearrange things
- can load XML from outside e.g. via WiFi on startup
- facilitates work with graphics designers
- can express the essence of the GUI in understandable form
XML PARSING

```xml
<screen id="sleepScreen" onSlideToRight="ChangeScreen(start,ANIMATE_RIGHT)" >
  <button x="10" y="10" height="60" width="340" class="sleepButton"/>
  ...
</screen>
```
'PARETO' CSS

button { icoFont: symbol }
button.sleepButton {
    backgroundColor: #FFFFFF
    frameColor: #00000000
    icoColor: #FF808080
    icoChar: #E054
}
#sleepScreen { backgroundColor: #DDDDDDFF }
CALLBACKS

- there are some built-in callbacks like ChangeScreen
- new ones can be coded in C++ and then used in XML
- could embed a JS engine in future to make it fully dynamic

onSlideToRight="ChangeScreen(start, ANIMATE_RIGHT)"
INTERACTIONS & ANIMATIONS

• presses
• slides
• long-presses for additional action on element
• slides between screens are animated
• scrolling lists (showing list end)
• -> modelled after smartphone UIs
PLATFORMS

• initially developed for eCos RTOS
• no hard dependency, but requires some infrastructure e.g. threads, libraries
• initial port for FreeRTOS
• libc/POSIX-compatible - runs 1:1 on Linux/Mac (e.g. for the "PC simulator")
SMART DISPLAY

Home
- TV
- Light
- Socket #1
- Socket #2

12 devices

Light
- 50% dimmed
- 20°C temperature
MATERIAL DESIGN
EMUL8

• we can also run the GUI in our emulation framework, Emul8
• support for STM32F4/7 boards
XILINX ZYNQ

• no GPU, not even a framebuffer
• 'proper' GPU takes up lots of valuable FPGA fabric
• ARM CPU makes it ideal platform to start prototyping
• we would have liked to prototype on the SiFive Freedom ;)

XILINX ZYNQ

• no GPU, not even a framebuffer
• 'proper' GPU takes up lots of valuable FPGA fabric
• ARM CPU makes it ideal platform to start prototyping
• we would have liked to prototype on the SiFive Freedom ;)

XILINX ZYNQ

• no GPU, not even a framebuffer
• 'proper' GPU takes up lots of valuable FPGA fabric
• ARM CPU makes it ideal platform to start prototyping
• we would have liked to prototype on the SiFive Freedom ;)

XILINX ZYNQ

• no GPU, not even a framebuffer
• 'proper' GPU takes up lots of valuable FPGA fabric
• ARM CPU makes it ideal platform to start prototyping
• we would have liked to prototype on the SiFive Freedom ;)
Zynq

PS

memory controller

display controller

PL FPGA

DMA

MicroBlender

memory

MICRO BLENDER
CHISEL

• easier to parametrize
• much easier communication between HW-SW team
• better understanding of what is going on for SW people
RISC-V

• RISC-V ultimate target platform
• encourages SW-driven silicon, reuse, co-design
• graphics will be a tough milestone
• our work could enable modern UIs for small RISC-V based platforms
SOFTWARE-DRIVEN IP

• that's how we always work with FPGA IP
• RISC-V makes this feasible for ASICs
• co-design of HW and SW
SUMMARY

• with minimal graphics infrastructure, we can get great looks and smooth user experience on graphics-less MCU-class platforms
• we can make GUI development simpler at the same time
• software-driven IP (silicon) is possible (and advisable!)
THANK YOU FOR YOUR ATTENTION!

contact@antmicro.com
twitter: @antmicro