

User Interfaces and Industrial IoT, Hands-On

Class 4: Beginning Graphics Interface, Hands-on (Part 2)

March 1, 2018

Charles J. Lord, PE
President, Consultant, Trainer
Blue Ridge Advanced Design and Automation

This Week's Agenda

2/26 An Overview of IIoT Applications and Interface Needs

2/27 Simple Interfaces that Aren't So Simple

2/28 Beginning Graphics Interface, Hands-on (Part 1)

3/1 Beginning Graphics Interface, Hands-on (Part 2)

3/2 Advanced Human Interface design for the IIoT

This Week's Agenda

2/26 An Overview of IIoT Applications and Interface Needs

2/27 Simple Interfaces that Aren't So Simple

2/28 Beginning Graphics Interface, Hands-on (Part 1)

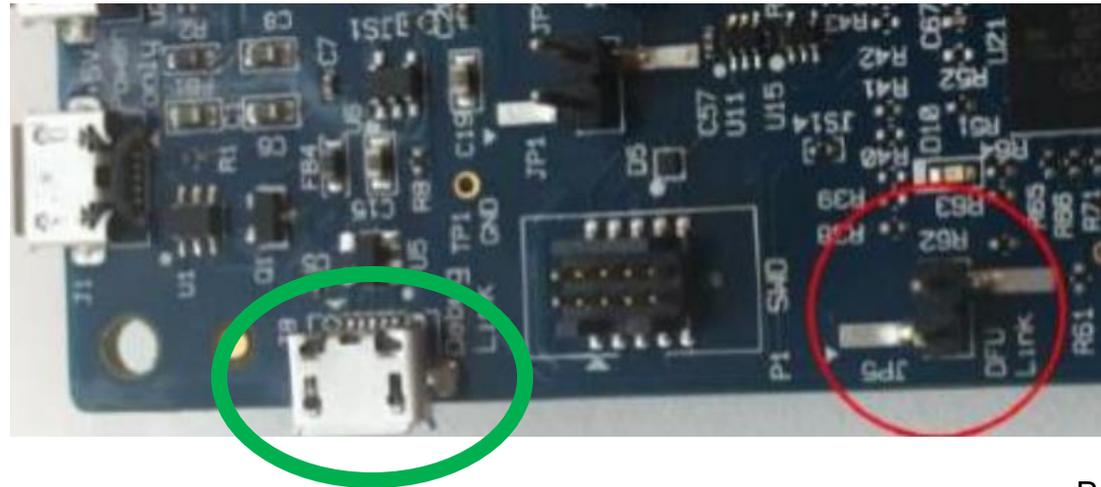
3/1 **Beginning Graphics Interface, Hands-on (Part 2)**

3/2 Advanced Human Interface design for the IIoT

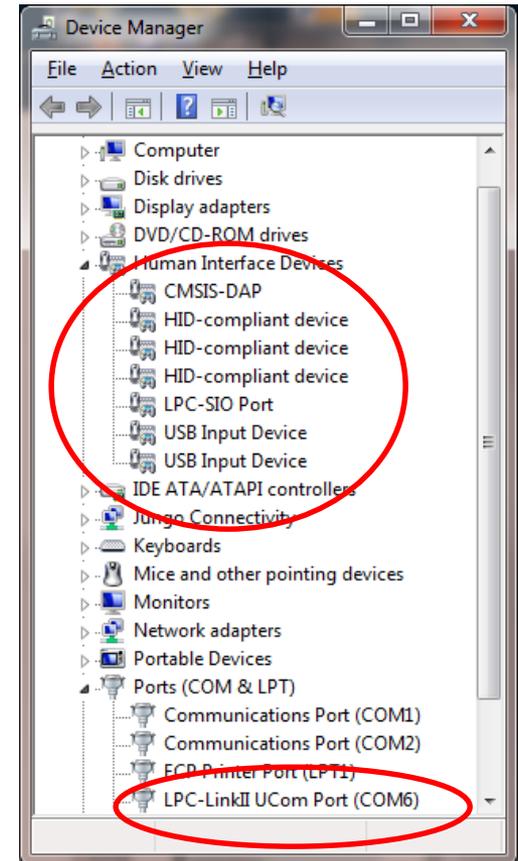
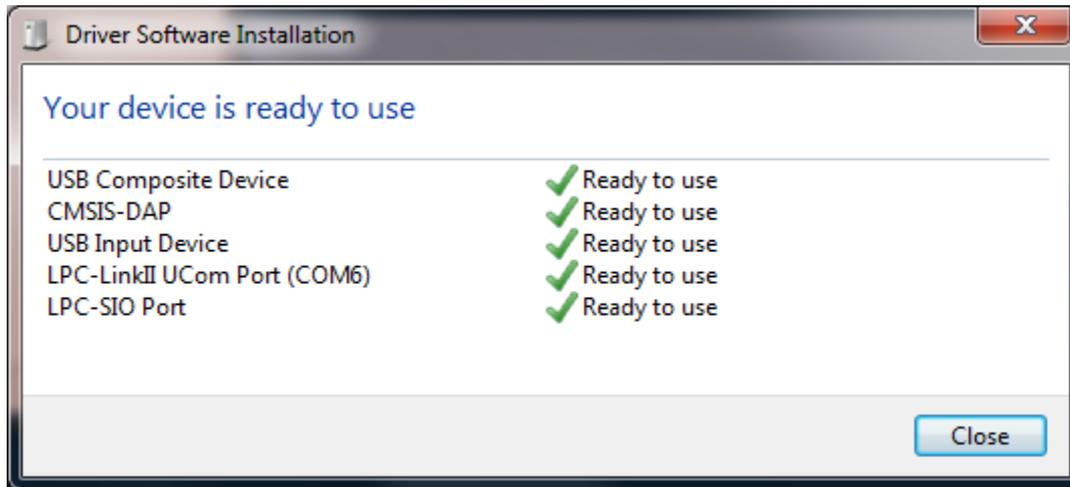
Prepare to Program the NXP Board

Before connecting your LPCXpresso54608 board to your host computer:

- Download and install LPCScript (www.nxp.com/lpcscript).
- Remove JP5
- Plug in board



Drivers will install



Process to Develop for the NXP

- Download SDK from NXP
- Develop UI in TouchGFX Designer – generate code
- Unzip SDK in proper directory
- Open Project in your IDE (IAR or Keil)
- Merge your code for non-GUI functions
- Compile, Flash, and Debug

Question 1 – What IDE(s) are you using for ARM?

https://mcuxpresso.nxp.com/en/builder

The screenshot shows the NXP MCUXpresso SDK Builder interface. A modal dialog titled "Select Software Components" is open, displaying a list of optional components. The dialog includes a search bar, "Select All" and "Deselect All" buttons, and a list of components with checkboxes. The "Add software component" button in the background is circled in red. A white text box at the bottom of the screenshot contains the text: "Choose 'open in Windows Explorer'".

Select Software Components

Select optional components from the list below to be added to your SDK.

4 items selected

Select All Deselect All

Middleware

- CMSIS DSP Library
- FatFS
- NTAG 12C
- QCA400x WiFi
- USB Type-C PD stack
- USB stack
- emWin
- lwIP
- mbedtls
- wolfssl

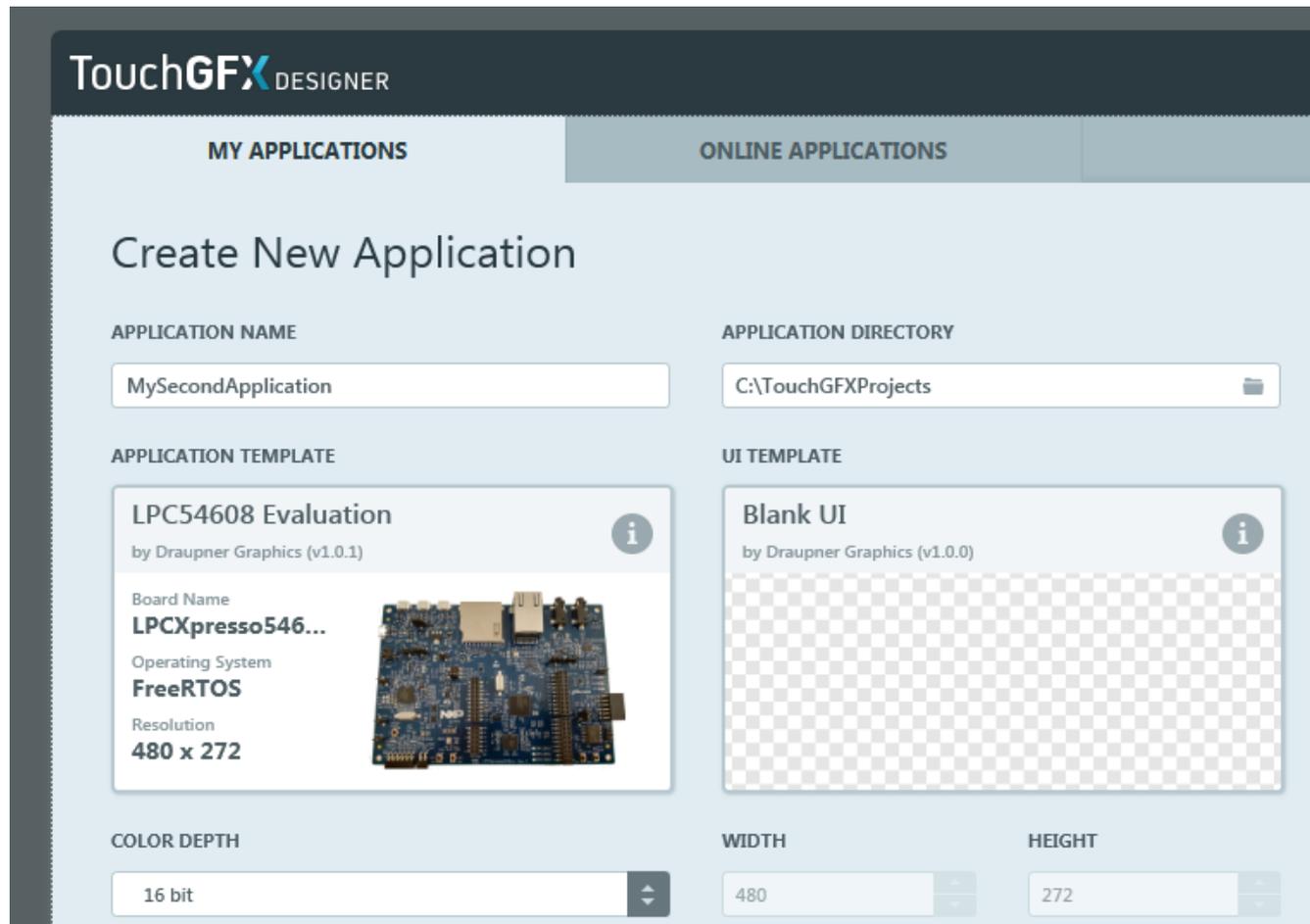
Operating systems

- FreeRTOS

Cancel Save changes

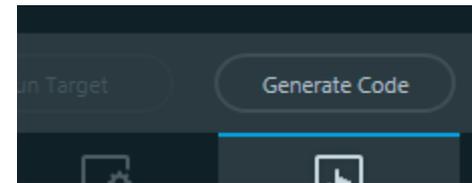
Choose "open in Windows Explorer"

Develop your GUI for the LPC54608



Remaining Steps

- When you have developed and tested your GUI with the simulator, you are ready to generate your target code



- Your code is now at

C:\TouchGFXProjects\{your application name}

- We will use MyFirstApplication in the following example

i.e. C:\TouchGFXProjects\MyFirstApplication

Adding the SDK

- Code generated by TouchGFX references all of the SDK code starting at the directory

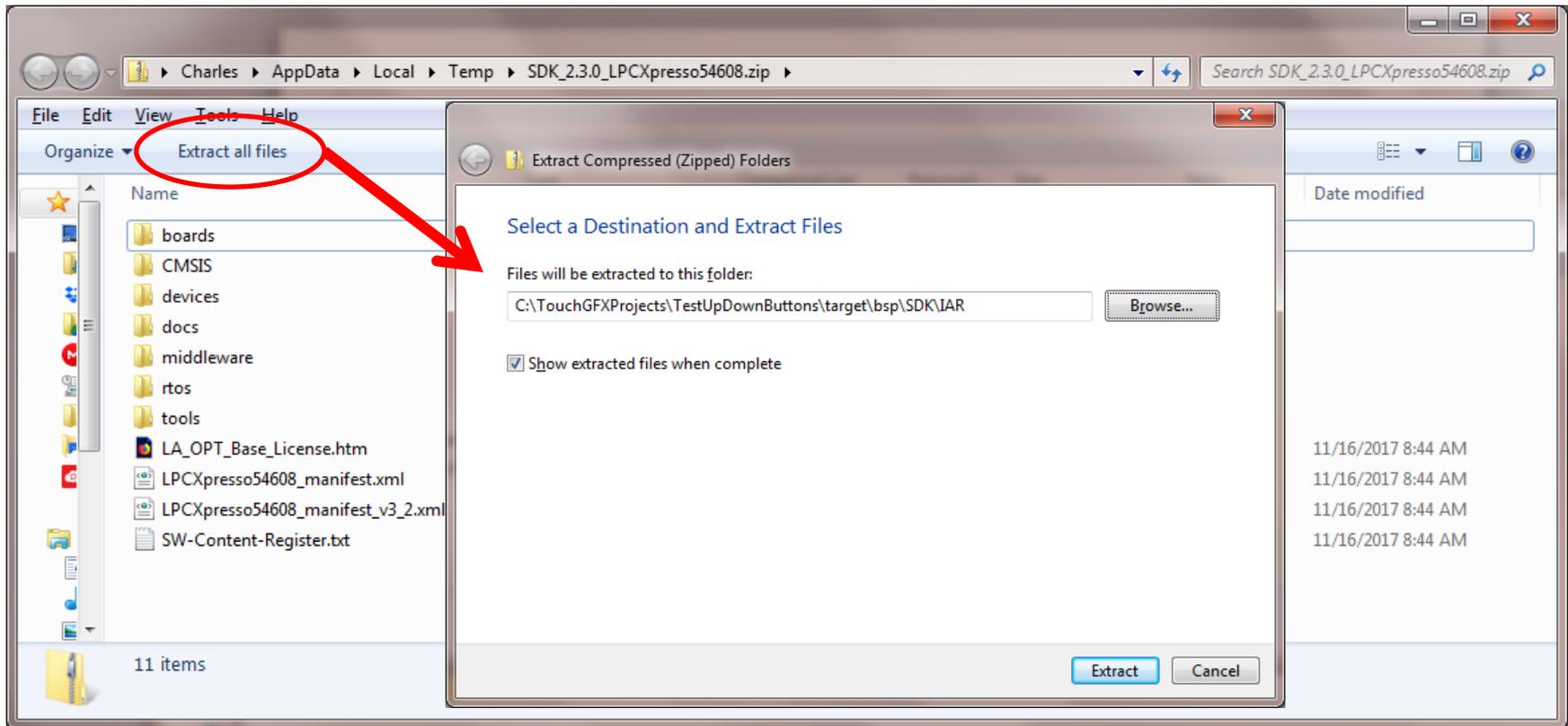
`C:\TouchGFXProjects\{your application name}\target\bsp\SDK\{IAR/KEIL}\`

Or in our example

`C:\TouchGFXProjects\MyFirstApplication\target\bsp\SDK\IAR`

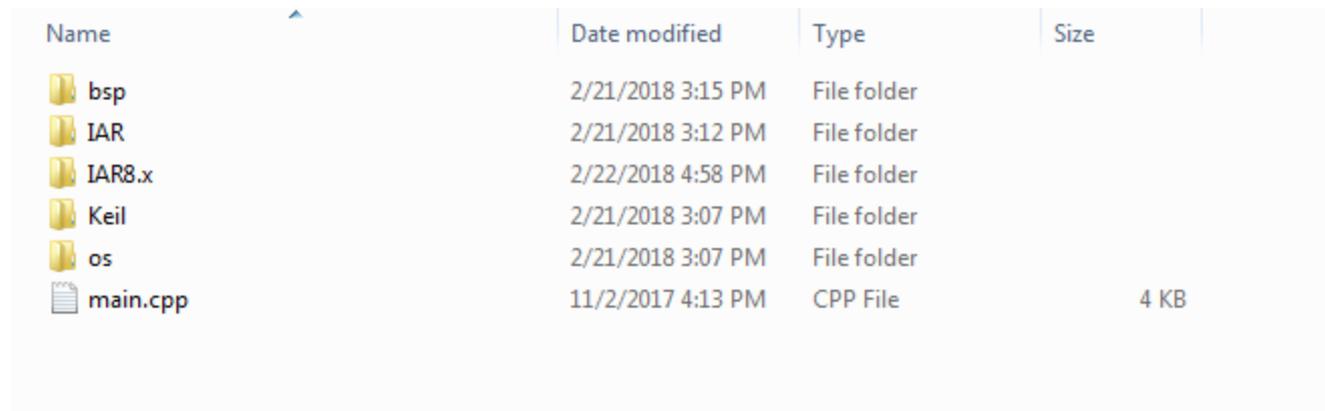
Note that the `\SDK\IAR` directory does not exist and will need to be created

Now Unzip to the proper directory



We now run IAR

- Navigate to the project's target directory



Name	Date modified	Type	Size
bsp	2/21/2018 3:15 PM	File folder	
IAR	2/21/2018 3:12 PM	File folder	
IAR8.x	2/22/2018 4:58 PM	File folder	
Keil	2/21/2018 3:07 PM	File folder	
os	2/21/2018 3:07 PM	File folder	
main.cpp	11/2/2017 4:13 PM	CPP File	4 KB

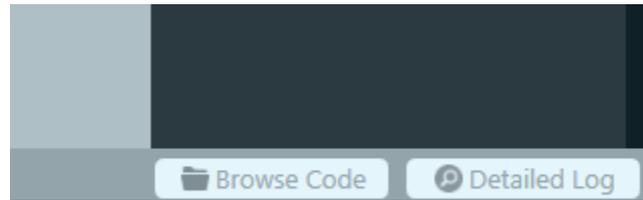
- You will see target workspace files for IAR 7.8x, IAR 8.x, and Keil

Click on the workspace file

Name	Date modified	Type	Size
debug	2/22/2018 4:58 PM	File folder	
settings	2/22/2018 6:30 PM	File folder	
application.custom_argvars	11/7/2017 11:31 AM	CUSTOM_ARGVA...	1 KB
application.dep	2/22/2018 6:30 PM	DEP File	112 KB
application.ewd	2/22/2018 4:59 PM	EWD File	100 KB
application.ewp	2/22/2018 4:59 PM	EWP File	91 KB
application.ewt	2/22/2018 4:59 PM	EWT File	186 KB
application.eww	11/7/2017 11:31 AM	IAR IDE Workspace	1 KB
Backup of application.ewp	2/21/2018 3:08 PM	EWP File	68 KB
LPC54608J512_flash.icf	11/7/2017 11:31 AM	ICF File	7 KB
SPIFIMAP_LPC546xx.mac	11/7/2017 11:31 AM	MacPaint Image	7 KB

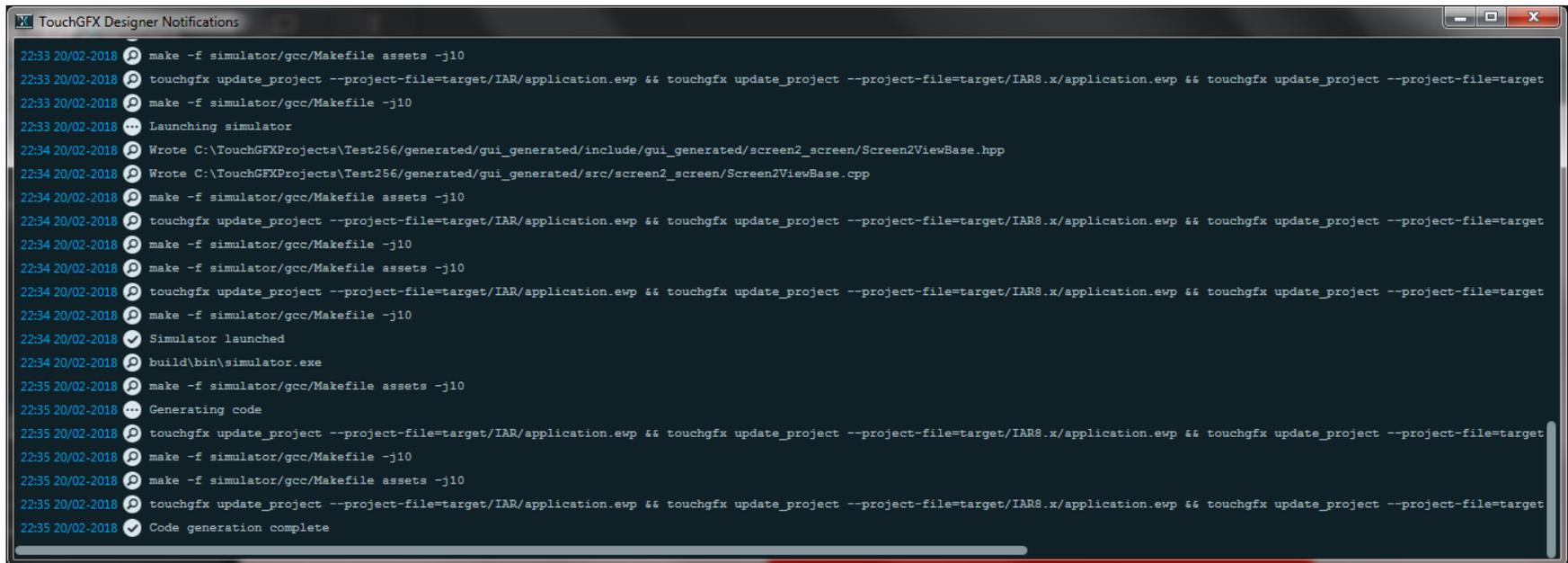
- As long as IAR 8.2x is your default install, this will bring up the project in IAR EWARM
- You should be able to build and run

Two Useful Buttons



- At the bottom right, there are two useful buttons for bringing up the project directory in Windows Explorer and for viewing the internal log for Designer

Great when errors occur or to check progress



```
TouchGFX Designer Notifications
22:33 20/02-2018 [i] make -f simulator/gcc/Makefile assets -j10
22:33 20/02-2018 [i] touchgfx update_project --project-file=target/IAR/application.ewp && touchgfx update_project --project-file=target/IAR8.x/application.ewp && touchgfx update_project --project-file=target
22:33 20/02-2018 [i] make -f simulator/gcc/Makefile -j10
22:33 20/02-2018 [i] Launching simulator
22:34 20/02-2018 [i] Wrote C:\TouchGFXProjects\Test256/generated/gui_generated/include/gui_generated/screen2_screen/Screen2ViewBase.hpp
22:34 20/02-2018 [i] Wrote C:\TouchGFXProjects\Test256/generated/gui_generated/src/screen2_screen/Screen2ViewBase.cpp
22:34 20/02-2018 [i] make -f simulator/gcc/Makefile assets -j10
22:34 20/02-2018 [i] touchgfx update_project --project-file=target/IAR/application.ewp && touchgfx update_project --project-file=target/IAR8.x/application.ewp && touchgfx update_project --project-file=target
22:34 20/02-2018 [i] make -f simulator/gcc/Makefile -j10
22:34 20/02-2018 [i] make -f simulator/gcc/Makefile assets -j10
22:34 20/02-2018 [i] touchgfx update_project --project-file=target/IAR/application.ewp && touchgfx update_project --project-file=target/IAR8.x/application.ewp && touchgfx update_project --project-file=target
22:34 20/02-2018 [i] make -f simulator/gcc/Makefile -j10
22:34 20/02-2018 [i] Simulator launched
22:34 20/02-2018 [i] build\bin\simulator.exe
22:35 20/02-2018 [i] make -f simulator/gcc/Makefile assets -j10
22:35 20/02-2018 [i] Generating code
22:35 20/02-2018 [i] touchgfx update_project --project-file=target/IAR/application.ewp && touchgfx update_project --project-file=target/IAR8.x/application.ewp && touchgfx update_project --project-file=target
22:35 20/02-2018 [i] make -f simulator/gcc/Makefile -j10
22:35 20/02-2018 [i] make -f simulator/gcc/Makefile assets -j10
22:35 20/02-2018 [i] touchgfx update_project --project-file=target/IAR/application.ewp && touchgfx update_project --project-file=target/IAR8.x/application.ewp && touchgfx update_project --project-file=target
22:35 20/02-2018 [i] Code generation complete
```

Adding a professional touch

- Generic buttons and plain color backgrounds are nice but don't give a professional feel
- With good graphics tools you can design your own screens, buttons, and other widgets
- There are a lot of royalty-free images and textures available on the web
- Buttons are simple. Note the size of button you want to use (pixels) and create a pair of images (pressed / not pressed)

Creating a Splash Screen



SplashDelay
When screen is entered delay

Trigger
Screen is entered

Action
Wait for

Duration
ms 5000

Can trigger another interaction

Interaction Name
SplashDelay

GoToMain
When SplashDelay completed
change screen to Screen2

Trigger
Another interaction is done

Choose interaction
SplashDelay

Action
Change screen

Choose screen
Screen2

Transition
ScreenTransitionNone

Can trigger another interaction

Interaction Name
GoToMain

Question 2 – What are the tradeoffs of a screen transition?

Talking to your application

- <https://touchgfx.zendesk.com/hc/en-us/articles/205074561-Connecting-the-UI-to-your-system>
- The GUI uses the model class to provide communications to and from your application
- DO NOT add your own C++ code in the widgets – if you ever re-generate (or modify) the GUI your changes are lost.

Advanced Widgets

- There are numerous widgets available for various inputs and outputs beyond the few that are in the designer
- Once you are comfortable with the basic design of screens and transitions and familiarize yourself with the code, it becomes much easier to integrate new widgets

<https://github.com/draupnergraphics/touchgfx-open-repository>

<https://touchgfx.zendesk.com/hc/en-us/categories/200529261-Knowledge-Base>

Using TouchGFX with other IDEs

- It is possible to develop with GCC based IDEs such as MCUXpresso and TrueStudio, but there is no automatic makefile and linking mechanism like there is with IAR and Keil
- <https://touchgfx.zendesk.com/hc/en-us/articles/206116381-Using-other-IDEs-Eclipse-LPCXPRESSO-with-TouchGFX>
- I am working on a script for MCUXpresso – stay tuned!

The Best Way to Learn

- GUI generation is a complex beast. In complex OS, you have powerful tools in the background to take care of a lot of the details
- With a smaller microcontroller and no graphics-based OS, you have to provide all functionality
- The best way to learn – or evaluate a potential OS – is to work through the examples and get familiar with the structure

Free GUI Tools?

- There are actually a number of free or open source GUI libraries out there for Cortex-M processors
- All require a good understanding of proper structure and for the most part don't have the nice WYSIWYG features we are using in this class
- Learning what is 'under the hood' using tools like this are a good start!

Tomorrow

- Some last words on TouchGFX
- Advanced topics in IIoT UI
 - Talking to existing systems
 - Integrating in other OS
 - Apps!

Question 3 – Experience with other (free) GUI apps?

This Week's Agenda

2/26 An Overview of IIoT Applications and Interface Needs

2/27 Simple Interfaces that Aren't So Simple

2/28 Beginning Graphics Interface, Hands-on (Part 1)

3/1 Beginning Graphics Interface, Hands-on (Part 2)

3/2 Advanced Human Interface design for the IIoT

Please stick around as I answer your questions!

- Please give me a moment to scroll back through the chat window to find your questions
- I will stay on chat as long as it takes to answer!
- I am available to answer simple questions or to consult (or offer in-house training for your company)

c.j.lord@ieee.org

<http://www.blueridgetechnc.com>

<http://www.linkedin.com/in/charleslord>

Twitter: @charleslord

<https://www.github.com/bradatrainning>