INTEL EMBEDDED SYSTEMS COMPETITION 2016

More info about software and hardware compatible with Intel® Galileo Gen 2

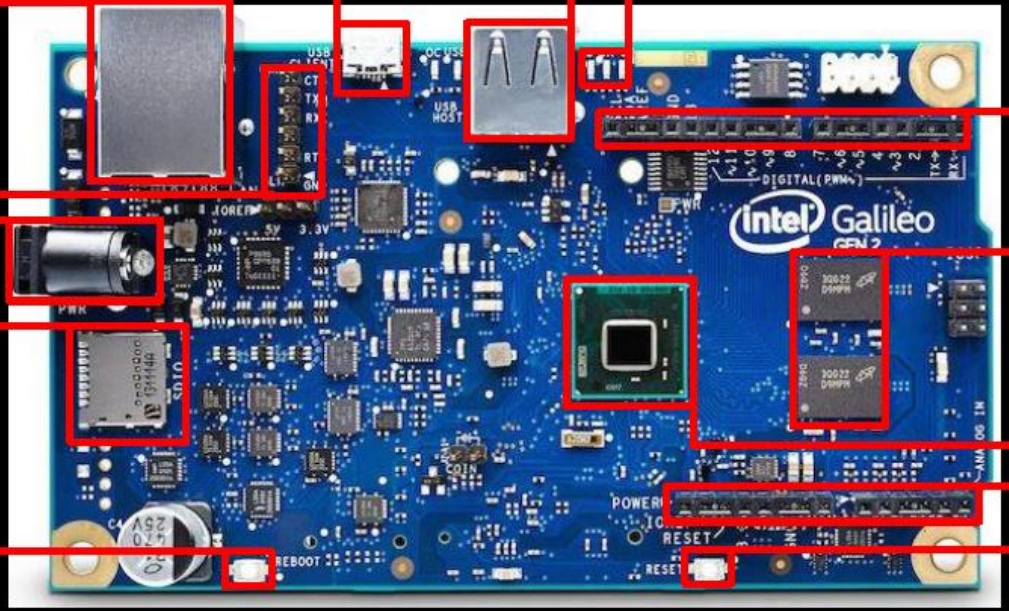
In this webinar

- Getting started
  - Hardware revision
  - Communicating to the board
  - Firmware update
- Operating Systems Highlights
  - Yocto project based image
  - Debian
  - FreeRTOS
  - Zephyr
  - Wind River* Rocket* / Linux
  - Ostro

In this webinar

- Building a custom kernel
  - Yocto Project
  - Ostro
- Software Highlights
  - Intel XDK IoT Edition
  - Intel System Studio IoT Edition
- General Information
- Q&A

Intel® Galileo Gen 2 – Hardware Revision



Communicating to the board

- USB – through Arduino IDE
  - Compile sketches and run Linux commands using the exclusive system() function (e.g. *system("ifconfig > /dev/ttyGS0“);*)
- SSH – through Ethernet cable or WiFi connection and PuTTY
  - Full access to terminal
  - Require SSH enabled
- SERIAL – through FTDI/USB cable
  - Full access to terminal
  - Great for testing and debugging an image
  - Do not require ssh connection

System( ) – why you should be careful

- Since sketch task is initiated at boot is important to notice the outcome of shell command parsed as a string on System( ) function
- One **really bad** use is System("shutdown –h now"); Please don't do it ☺
- In case you did this or something similar, in which you can not rewrite /sketch/sketch.elf file with a new sketch, follow these instructions.
    - Remove the sdcard and plug it in to a Linux host (VM is ok)
    - Find the sketch folder and remove sketch.elf file
    - Eject sd card and insert back in Galileo
    - You now should be able to boot properly

This section is only applicable when using distros that support Arduino and start the sketch during boot (e.g. image provided by Intel)

Firmware Update – option 1

- Update from 1.0.2 to 1.0.4 using the provided tool by Intel
- Download software and tutorial https://downloadcenter.intel.com/download/24748/Intel-Galileo-Firmware-and-Drivers-1-0-4
- Useful tips:
  - Don't run the program with the sdCard connected
  - Run as administrator (sudo for Linux users)
  - Wait for the board fully boot and be recognized before running the software
  - Make sure you selected the right port

Firmware Update – option 2

- Update from 1.0.2 to 1.0.4 using a **specific** version of Arduino IDE
- Download software
  https://downloadcenter.intel.com/download/24782/Intel-Arduino-1-5-3
- Open IDE and go to Help -> Galileo Firmware Update
- Useful tips:
  - Unzip into C:/ directory using 7-zip tool
  - Don't run the program with the sdCard connected
  - Wait for the board fully boot and be recognized
  - Make sure you selected the right board and port
  - If Arduino IDE does not open please consider the following solution
    http://forum.arduino.cc/index.php?topic=234307.0

Yocto Project – prebuilt image

- Download EGLIBC image
  https://software.intel.com/en-us/iot/hardware/galileo/downloads

  http://downloadmirror.intel.com/25384/eng/iot-devkit-201510010757-mmcblkp0-galileo.direct.xz

- Unzip with 7zip
- Burn .img or .direct file to micro sdCard (Win32DiskImager or using dd command)

Login: root **no password required**

Resources:

  - Arduino* IDE support
  - Development tools C/C++, Python*, Node.js* and OpenJDK 1.8

Debian

- Download image https://sourceforge.net/projects/galileodebian/files/SD%20card%20Image/
- Unzip with 7zip
- Burn .img file to micro sdCard

Login: root Password: root

Resources:

- Familiar Linux environment
- Access to Debian package repository and software updates
- Access to preconfigure Debian packages (e.g. Nano)

Ostro

- Getting started guide
  https://ostroproject.org/documentation/quick_start/quick_start.html
- Pre-built images
  https://download.ostroproject.org/

Resources:

  - OS tailored for IoT smart devices and built with security in mind
  - Base OS image can be used as-is or rebuilt (similar structure to Yocto Project)
  - Support for Node.js* , Python* 2.7, C/C++ and Java* (preconfigured in ostro-image-swupd-dev-intel-quark)

FreeRTOS

- Download image and full tutorial
  http://www.freertos.org/RTOS_Intel_Quark_Galileo_GCC.html
- Prebuilt examples

Login: root Password: intel

Resources:

  - Provides a predictable (deterministic) execution pattern
  - Allows user to assign a priority to each thread of execution (task)
  - Provides the core real time scheduling functionality, inter-task communication and timing

Zephyr

- Getting started and building demo for Galileo
  - Getting started https://www.zephyrproject.org/doc/getting_started/getting_started.html#getting-started
  - Linux install https://www.zephyrproject.org/doc/getting_started/installation_linux.html
  - Galileo + Zephyr https://www.zephyrproject.org/doc/board/galileo.html
  - Application Development https://www.zephyrproject.org/doc/application/apps_dev_process.html
- Prebuilt examples

Resources:

- Real-Time Operating System (RTOS) for IoT
- Small, scalable and modular
- Developed with security in mind
- Offers a microkernel and a nanokernel

Wind River* Rocket*

- Getting started guide https://software.intel.com/sites/default/files/managed/b0/51/Wind_River_Rocket_GETTING_STARTED_GUIDE.pdf
- Free embedded RTOS for IoT

Resources:

  - Kernel based on Zephyr microkernel
  - Code and debug applications from any browser
  - Cloud-based development environment
  - Development in C
  - Arduino* API
  - Require serial connection (FTDI cable)

Wind River* Linux*

- Initial setup https://software.intel.com/sites/default/files/managed/b0/51/Wind_River_Rocket_GETTING_STARTED_GUIDE.pdf
- Access your account on Wind River®Helix™App Cloud, select New Device -> Create a new device from the supported SDK -> follow the provided instructions

Resources:

  - Code and debug applications from any browser
  - Cloud-based development environment
  - Development in C/C++ and Node.js*
  - Require internet access and be on the same network with cloud workspace

Building a custom kernel

Why one might want to compile a custom kernel?

- ✓ Gain more control over the embedded application
- ✓ Performance – compile only what's necessary
- ✓ Better use of resources – reduce overhead
- ✓ Knowledge - Learn more about the kernel

Building a custom kernel – Yocto Project (1)

- BSP 1.2.1
- Tutorial and needed files https://downloadcenter.intel.com/download/23197/Intel-Quark-BSP?product=79084
- Offers:
  - Prebuilt Python* 2.7
  - Easy connection to wireless networks with connmanctl
  - Kernel version 3.14.28
  - Opkg package manager
- Built and validated on Debian 7 and 8

Building a custom kernel – Yocto Project (2)

- Devkit Daisy 1.6.1
- Tutorial http://www.embarcados.com.br/galileo-yocto/
- Offers:
  - Support to Python* 2.7, Node.js* and Arduino* IDE
  - MRAA and UPM libraries
  - Easy connection to wireless networks with connmanctl
  - Kernel version 3.8.7

- Built and validated on Ubuntu 12.04 and Debian 7 and 8

Building a custom kernel – Ostro

- Tutorial
  https://ostroproject.org/documentation/howtos/building-images.html#building-images
- Based on Yocto Project
- Offers:
  - Support to GCC, Python* 2.7, Node.js* and OpenJDK 1.8
  - Easy connection to wireless networks with connmanctl
  - Kernel version 4.4.9
- Built and validated on Debian 7 and 8

"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil."

—Donald Knuth

Making Embedded Systems by Elecia White – O'Reilly, 2011

Intel® System Studio IoT Edition

- Plugin for Eclipse* that allows to connect to, update, and program IoT projects on a compatible board
- C/C++ and Java

Tips:

- For Windows* users it may help have installed MinGW (with all basic tools)
- Requires Java* JDK 1.8+, if Eclipse* does not automatically find it, please consider the following: Window -> Preferences -> Java -> Installed JREs -> Add -> Standard VM -> JRE Home (set path to jdk1.8_x) -> Finish -> unselect jre8 -> Ok

Intel® XDK IoT Edition

- IDE for JavaScript* and Node.js* programming
- User guide https://software.intel.com/en-us/getting-started-with-the-intel-xdk-iot-edition

Resources:

  - Enables easy on-board app development and deployment
  - Deploy, run and debug in the same place
  - Provides quick start templates and samples
  - Integrates with cloud, web services, and sensors through JavaScript APIs
  - HTML5 app creation

Intel® Galileo (Gen 2) – Network Connectivity

- While the Galileo board doesn't come with Wi-Fi connectivity, you can add to it.
- Any Linux-supported Wi-Fi card should work.
- Both wired and wireless connectivity settings can also be managed through the `connmanctl` tool.

- Link for Intel Centrino drivers https://wireless.wiki.kernel.org/en/users/Drivers/iwlwifi

Configuring Package Repository – Intel Galileo

- OPKG is the package manager of Yocto images (usage e.g. opkg install nodejs-npm)
- To update the paths, please consider the following guide:

  In **/etc/opkg** we are going to edit *iotdk.conf* and *mraa-upm.conf*

For iotdk.conf:

src iotdk-i586 http://iotdk.intel.com/repos/3.0/iotdk/i586/
src iotdk-intel-core-2-32 http://iotdk.intel.com/repos/3.0/iotdk/intel_core2_32/
src iotdk-quark http://iotdk.intel.com/repos/3.0/iotdk/quark/
src iotdk-x86 http://iotdk.intel.com/repos/3.0/iotdk/x86/
src iotdk-core-2-32 http://iotdk.intel.com/repos/3.0/iotdk/core2-32/

For mraa-upm.conf:

src mraa-upm http://iotdk.intel.com/repos/3.0/intelgalactic/opkg/i586/

Intel® Galileo and Embedded Systems– Useful books

- Embedded Linux Development with Yocto Project by Otavio Salvador; Daiane Angolini – Packt, 2014
- Intel® Galileo and Intel® Galileo Gen 2: API Features and Arduino Projects for Linux Programmers by Manoel Carlos Ramon – Apress, 2015
- Internet of Things with Intel Galileo by Miguel de Sousa – Packt, 2015
- Node.js for Embedded Systems by Patrick Mulder; Kelsey Breseman – O'Reilly, 2016 (early release)

Intel® Galileo – Useful links

- Galileo downloads: https://software.intel.com/en-us/iot/hardware/galileo/downloads
- Galileo IDE downloads: https://software.intel.com/en-us/iot/software/ide
- Yocto Project Development Manual: http://www.yoctoproject.org/docs/current/dev-manual/dev-manual.html
- Videos Competition - Intel Embedded Systems 2015: https://www.youtube.com/playlist?list=PLdv8QZ_rwBOdwKX-LVTrv62MQfYiJ2XTk
- Our website: http://sbesc.lisha.ufsc.br/sbesc2016/Intel+Embedded+Systems+Competition

Don’t forget...

- Keep your schedule up-to-date weekly
- Got a question? Ask us!

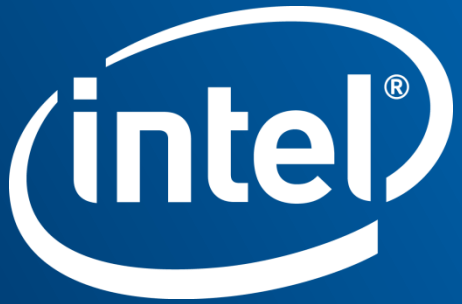
  submissaocompeticaointel@gmail.com

Next Webinar…

- August 23 – 15h00
- August 24 – 10h30

Next Deadline…

- September 20 – Partial Report Submission (through JEMS)


intel®