

# Conclusion



## NOTES:

QNX, Momentics, Neutrino, Photon microGUI, and "Build a more reliable world" are registered trademarks in certain jurisdictions, and PhAB, Phindows, and Qnet are trademarks of QNX Software Systems.

All other trademarks and trade names belong to their respective owners.

### In this course you learned:

- how to create systems that:
  - are robust and resilient
    - use multiple processes that run in independent address spaces and communicate with each other such that they remain isolated
  - meet tight timing requirements
    - arrange priorities properly, make use of multiple threads if necessary
    - keep scheduling latency down by trying to get back to a blocking state fast
    - keep interrupt and scheduling latency down by not spending a lot of time in interrupt handlers or, whenever your timing requirements allow, simply have the kernel wake up a thread instead

*continued...*

NOTES:

### In this course you learned (continued):

- how to create systems that:
  - handle large, complex tasks in a modular and maintainable manner
    - once again, use multiple processes that communicate using various forms of IPC
  - fit into a relatively small memory footprint
    - configure your system to run only the processes you need
  - talk to a large variety of hardware
    - write processes that handle interrupts, do I/O, and take advantage of the resource manager framework
- a bit about the development tools
- a bit about configuring your target

NOTES:

### Where to go from here:

- other training courses
  - Developing, Debugging and Analyzing Performance with the QNX Momentics IDE
  - Writing Drivers for the QNX Neutrino RTOS
- support plans
  - contact your sales person for details
- custom engineering
  - we can write code for you (BSPs, drivers, ...)
- websites
  - [www.qnx.com](http://www.qnx.com)
  - [www.foundry27.com](http://www.foundry27.com) – source, forums, etc...

NOTES: