

# Integrating OS Changer with your RTOS

## Overview

“ . . . I was involved with enabling OS Changer on the MQX RTOS as well as branding OS Changer as an ARC International product named ARC-OS Changer. As such, my focus is primarily on how we were able to modify OS Changer to create ARC-OS Changer for MQX.”

## Benefits

“One of the things I appreciated while creating ARC-OS Changer for MQX is that the overall OS Changer design is very modular and provided a number of benefits while starting.”

### □ How Easy Was It?

“ . . . the version of OS Changer we used as a starting point provided support for different build environments. Because makefiles and MetaDeveloper project files were provided we were able to quickly get started with OS Changer \*with our choice\* of build environments and had examples to rely upon while modifying and creating new makefiles and new project files for inclusion into ARC-OS Changer for MQX. This enabled us to provide users with the ability to choose how they want to build ARC-OS Changer for MQX without having to create everything from scratch.”

### □ How Long Did The Port Take You to Complete?

OS Changer allowed him to sidestep many of the non-engineering roadblocks typically involved with integrating a third-party product. “I'd estimate that meeting the functional requirements of the product with MQX requires ~2 1/2 weeks to develop from scratch and ~1 week to formally create the test software and perform verification with a reasonably experienced engineer unfamiliar with OS Changer. . . Meeting additional performance requirements, creating the performance benchmark software and revalidating the changes I'd estimate at ~1 1/2 weeks.”

### □ How was the Technical Support?

“I was very impressed with how quickly I received technical responses to my questions and with the quality of the information I received. I felt the training was well structured and covered all of the areas end-users would need to know about.”

### □ How Was The Performance?

“ . . . Inside the ARC-OS Changer for MQX product, we've made a number of OS-specific performance enhancements such as replacing OS-independent function calls made within ARC-OS Changer with the equivalent MQX calls directly; sidestepping the middle-man. Adding these performance enhancements was quick and painless due to how the OS Changer's internal abstraction layer was designed.”

“The VxWorks and pSOS API layers are well separated to the point where I could quickly include or exclude the VxWorks and pSOS libraries them from my code. . . Considering performance was a crucial metric I was doing this quite a bit and was quite happy to find that it was easy to switch among different APIs. . . because of the internal structure of ARC-OS Changer we were able to integrate the MQX lightweight memory allocator to improve the overall performance of the product without making modifications to the architecture of ARC-OS Changer. To me, this indicates that future improvements to ARC-OS Changer can be easily and quickly added without requiring significant modifications to the existing code base.”