

Exercise RTOS-07P. FreeRTOS real time operating system - basics

REAL TIME OPERATING SYSTEMS IN CONTROL APPLICATIONS LABORATORIUM SYSTEMÓW STEROWNIA PRZEMYSŁOWEGO I AUTOMATYKI BUDYNKÓW

KATEDRA ENERGOELEKTRONIKI I AUTOMATYKI SYSTEMÓW PRZETWARZANIA ENERGII
WWW.KANIUP.AGH.EDU.PL

AKADEMIA GÓRNICZO-HUTNICZA
WWW.AGH.EDU.PL

Subject:

FreeRTOS real time operating system - basics

Tools:

Visual Studio Express, FreeRTOS sources

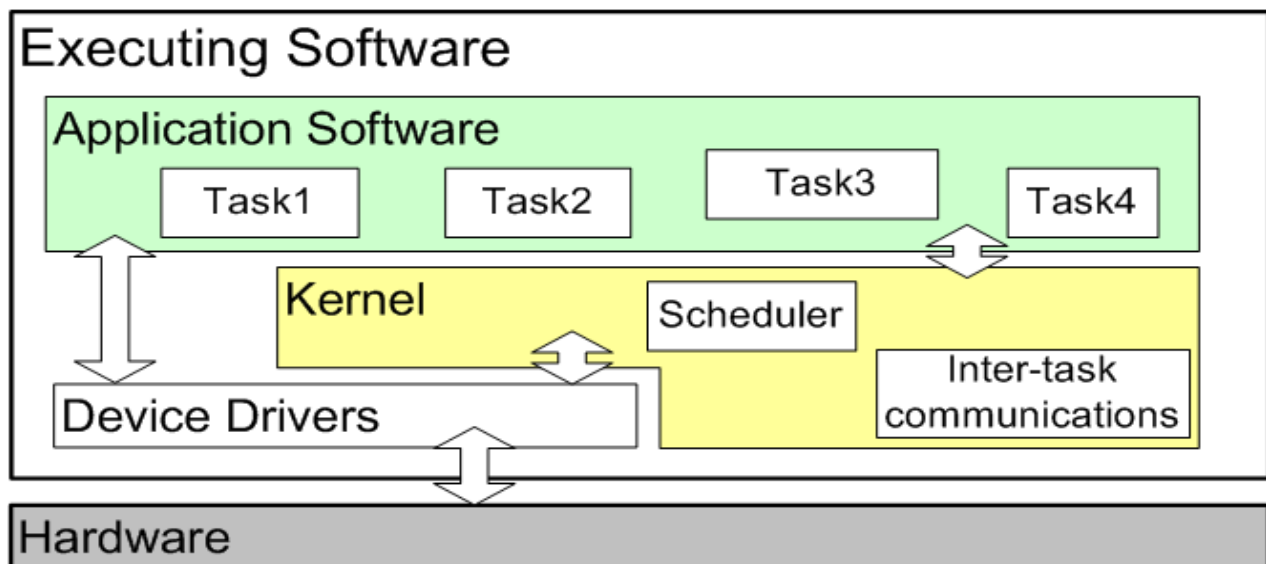
Required skills:

Basic knowledge of issues related to programming in C language and real-time systems

Introduction.

FreeRTOS is a real time operating system for embedded systems created and maintained by Real Time Engineers Ltd. It has been designed with the simplest and smallest source code in mind, thanks to which it can be run on small microcontrollers. In addition, it supports many architectures, making porting on most microcontrollers quite easy. Until recently, it was distributed under the GPLv2 license. In November 2017, it was taken over by Amazon and was included in the AWS (Amazon Web Services) open source project - since then the system has evolved towards IoT technology and collaboration with the Amazon cloud. It has since been released under the MIT license.

Example image architecture of the FreeRTOS system:



Purpose of the exercise.

The aim of the exercise is to learn the basics of FreeRTOS real-time operating system architecture and the principles of creating and running applications in its environment

Exercise program.

1. Download and install Visual Studio Express version 17 (note, installation is over 2.5 GB). The installation manager file can be downloaded from:
2. https://aka.ms/vs/15/release/vs_WDExpress.exe
3. Download the FreeRTOS_DEMO_lab_1 software sources posted on UPEL (Laboratory 1).
4. Run Visual Studio Express and open the project from the downloaded directory structure \FreeRTOS_DEMO_lab_1 \ FreeRTOS-Plus \ Demo \ FreeRTOS_lab_1. The project description file is FreeRTOS_lab_1.sln.
5. Compile, run and test the operation of the sample program with one task running in an infinite cyclic loop
6. Add a second identical task with a different loop activation time
7. Compile, run and test the operation of the application