Exercise RTOS -11P. FreeRTOS – synchronization

REAL TIME OPERATING SYSTEMS
LABORATORIUM SYSTEMÓW STEROWNIA PRZEMYSŁOWEGO I AUTOMATYKI BUDYNKÓW

KATEDRA ENERGOELEKTRONIKI I AUTOMATYKI SYSTEMÓW PRZETWARZANIA ENERGII AKADEMIA GÓRNICZO-HUTNICZA
WWW.KANIUP.AGH.EDU.PL WWW.AGH.EDU.PL

Subject:	**FreeRTOS real time operating system – synchronization methods**
Tools:	**Visual Studio Express, FreeRTOS sources**
Required skills:	**Basic knowledge of issues related to programming in C language and real-time systems**

Introduction.

Synchronization or protection of resources against simultaneous access of many processes can be implemented using various tools: inter-task communication, semaphores, mutexes.

A semaphore is a protected variable used to control access to a shared resource across many processes. To handle semaphores at the basic level, three functions can be used, which are used to create the semaphore, release the resource and block the resource.

- SemaphoreHandle_t = **xSemaphorCreateBinary(void)**;

or

- SemaphoreHandle_t = xSemaphoreCreateMutex();

- **xSemaphoreGive(SemaphoreHandle_t xSemaphore);**
  xSemaphore – handle to the previously created semaphore

- **xSemaphoreTake(SemaphoreHandle_t xSemaphore,TickType_t xTicksToWait);**
  xSemaphore – handle to the previously created semaphore
  xTicksToWait – waiting time for semaphore release

Exercise RTOS -11P. FreeRTOS – synchronization

Purpose of the exercise.

The purpose of the exercise is to become familiar with the synchronization mechanisms.

Exercise program.

1. Use project 020-Semaphores and test the operation in each of the following cases:
   1.1. Run two continuous tasks in which the text will be written to the output (terminal) without any synchronization
   1.2. Run two continuous tasks in which text output (terminal) with write protection (synchronization) will be performed using a binary semaphore (or mutex). In this case, output to the terminal (stdout) will be a protected resource to which we want to provide exclusive access (one task at a time only).