SINGAPORE POLYTECHNIC
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

ET0104 Embedded Computer Systems Laboratory

Laboratory 9 - Multitasking

Objectives

- To create separate threads in addition to the main application thread.
- To use event objects for threads synchronization.

Introduction

Multitasking is a method by which multiple tasks, also known as process and threads share common processing resources such as a CPU. In the case of a computer with a single CPU, only one task is said to be running at any point in time, meaning that the CPU is actively executing instructions for that task. Multitasking solves the problem by scheduling which task may be the one running at any given time, and when another waiting task gets a turn. The act of reassigning a CPU from one task to another one is called a context switch. When context switches occur frequently enough the illusion of parallelism is achieved.

Tasks can interact with each others. Sometimes they interact directly with each other; other times they interact through shared resources. These interactions must be coordinated, or synchronized. Event objects are typically used for synchronizing tasks processing rather than for controlling access to shared resources. It is a mechanism that lets a task go to sleep until, for example, some data is ready for it to process or a request is ready for it to service.

When a program runs, the main application thread will be created. From the main application thread more threads can be created as necessary. To create additional threads, the following function can be used.

```
Unsigned long _beginthreadex(void *security,
unsigned stack_size,
unsigned ( __stdcall *start_address )( void * ),
void *arglist,
unsigned initflag,
unsigned *thrdaddr);
An event object can be created by the following function.
HANDLE CreateEvent(LPSECURITY_ATTRIBUTES lpEventAttributes,
BOOL bManualReset,
BOOL bInitialState,
LPCTSTR lpName,);
```

Events can be set to the signal state by using the following function

```
BOOL SetEvent(HANDLE hEvent);
```

Experiment

From the D:\ECSLAB\LAB9 folder, open the lab9.dsw workspace. This workspace has the multitasking features of the ETS enabled. This workspace consists of two source files (lab9.c and lcd.c) and one header file (lcd.h). Fill in the blanks in the program lab9.c and compile the program. When the program has compiled successfully, download the program and run it. Take note of the multitasking aspect of the program.

Modify the program to add one more thread to display the number of steps remaining for each stepper motor command, on the first row of the LCD display.