Lab 7 - APSC 160 DAQ Interface Handout

Associated with the DAQ module are a number of functions for setup, reading and writing values. These DAQ functions are similar, in concept, to the functions used for file I/O, as illustrated in the following table:

DAQ functions	*File I/O functions*

`setupDAQ`
`continueSuperLoop`
`digitalRead/analogRead`
`digitalWrite`
	`fopen`
	testing for EOF
	`fscanf`
	`fprintf`

DAQ Function Details

In the description below, the following defines and includes are assumed to exist in the program:

```
#include <DAQlib.h>

#define TRUE 1
#define FALSE 0
```

```
int setupDAQ(int setupNum);
```

- Initializes the DAQ module and returns TRUE if it was successful; otherwise it returns FALSE.
- No DAQ functions must be called, if FALSE is returned.
- If `setupNum` of 0 is given, then it sets up the hardware, assuming that the physical hardware is connected. A second console window is opened, and when the user is done, they need to quite this second window.
- If a `setupNum` of 1 to 7 is given, then the simulator is started. The particular configuration provided by the simulator is determined by the value of `setupNum`. When the user is done with the simulator, they need to quit the simulator window.

```
int continueSuperLoop(void);
```

- When the user is done and has quit the window opened by `setupDAQ`, this function returns **`FALSE`**; otherwise it returns **`TRUE`**.

```
int digitalRead(int channelNumber);
```

- Reads the given digital input channel (**`channelNumber`**), and returns its value (which is either 0 or 1).
- The `channelNumber` is used to determine which connection or I/O device we are referring to.
- The channel numbers for reading and writing are independent.

```
void digitalWrite(int channelNumber, int val);
```

- Writes the given value (**`val`**) to the indicated digital output channel (**`channelNumber`**).
- The value must be either 0 or 1.

```
void displayWrite(int data, int position);
```

- Writes the given value (**`data`**) to the LED display at the given position (**`position`**).
- **`data`** determines which segments are off, and which are on.
- **`position`** must be a value between 0 and 7.
- `position` 0 is the right-most LED display and `position` 7 is the left-most LED display.

General Framework

The general algorithm will be used by all our programs:

- Initialize the DAQ module;
  - if not successful, print an error message and exit.
- Otherwise, if initialization is successful, then while the user has not quit, perform the following steps:
  - Read from input (if any)
  - Process data, including any inputs that were read
  - Generate output (if needed)

Hardware Devices

LED (write)	*Switch (read)*
1 = ON 0 = OFF	1 = ON 0 = OFF

Using the Sleep function

To use the `Sleep` function you must include the `Windows.h` header file. The function prototype is:

```
void Sleep(unsigned long delayInMilliSeconds);
```

- Suspends the execution of the program for the given number of milliseconds.

Note: this handout does not include info on the `analogRead()` function.