

Mectron/Sfwr Eng 4aa4 - Lab 5

Introduction:

In this lab you will learn how to interface with some of the devices that are not part of myRIO box. DIO pins will be used to blink an LED.

Goals:

- Learn how to create periodic real time tasks for blinking an LED connected to myRIO Expansion Port (MXP) Breakout.

Note: Before you go for your lab sessions, please read the following documents at your own convenient time, in addition to the class notes:

- [C Support for myRIO User Guide6.0.pdf](#). Most of the software components mentioned in this document have already been installed and configured on lab computers.
- [Getting Started with C Development Tool\(Eclipse\).pdf](#)
- Document at this link: <http://www.drdoobbs.com/soft-real-time-programming-with-linux/184402031>. Or, a PDF version can be found in folder of "ref" for Lab 3 .
- Document at this link: <https://hpc-tutorials.llnl.gov/posix/threads api/>

Activities in this lab:

NOTE: Please use a 220 resistor to protect external LED in the lab.

Part 1: [50]

Blinking an external LED using a periodic real time task:

- The NI myRIO-1900 has 3.3 V general-purpose DIO lines on the MXP and MSP connectors. MXP connectors A and B have 16 DIO lines per connector and MSP connector C has eight DIO lines. Any of these pins can be used to input or output a digital signal from/to an external device. The following description first gives the background information for which it is assumed that pin 7 of MXP B is used for this part of the lab. Please understand the background material by reading different documents referenced here before you start writing code for this part of the lab.
- Refer to the section describing DIO channels in myRIO Shipping Personality V3.0 reference manual (page 18). The DIO channels are divided into 8 channel banks.
- There are three registers to access each DIO channel bank namely: Data Direction Registers, Pin Input Registers and the Pin Output Registers.
- In order to set and use a particular channel in C programming language, a struct is defined in file 'DIO.h' as follows:

```
/**
 * Registers for a particular digital IO.
 */
typedef struct
{
```

```

uint32_t dir;    /**< DIO direction register */
uint32_t out;   /**< DIO output value register */
uint32_t in;    /**< DIO input value register */
uint8_t bit;    /**< Bit in the register to modify */
} MyRio_Dio;

```

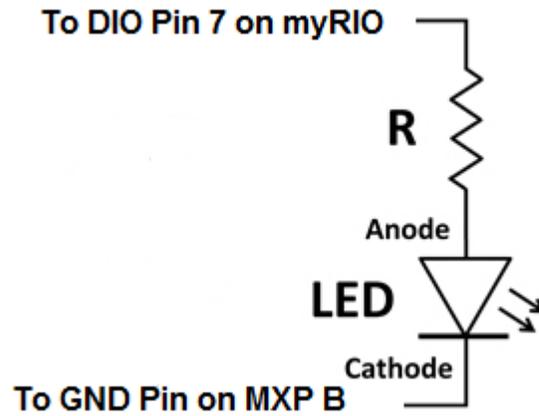
- Please refer to the files 'DIO.h' and 'DIO.c' in your workspace (used by you in lab 4) to familiarize yourselves with several functions to read from and write to a particular DIO channel.
- For the purpose of this part of the lab, you need a variable of type 'MyRio_Dio' to represent a DIO channel e.g.

```
MyRio_Dio B7;
```

- Next you need to assign B7.dir, B7.out and B7.in to names of registers as specified in myRIO Shipping Personality V3.0. For example for the direction register:

```
B7.dir = DIOB_70DIR;
```

- Specify B7.bit = 7; as you want to use pin 7 in this part of the lab.
- Once a channel is set and is ready for use, you can use a function 'Dio_WriteBit()' defined in 'DIO.c' to write a boolean value to a DIO channel.
- Create a template project in your work space and rename it to a suitable name.
- Using the file 'main.c' of your project, create a real time periodic task that will toggle the value written to the DIO channel B7 in each period.
- Adjust the time period to 1ms or more so that blinking of LED can be observed.
- Connect the breakout board to MXP B and create the LED circuit as shown in the figure below. **NOTE: Please use a 220 resistor to protect external LED in the lab.**
- Compile and run the program and show the result to your TA.



Part 2 [50]

Blinking an external LED using a threaded periodic real time task:

- Create a template project in your workspace and rename it to a suitable name.
- Repeat what was done in part 1 of this lab except that this time use a threaded task and DIO pin 5.
- Show the result to your TA.

Marking scheme:

Lab 5 grade will be based on the student's performance on the codes and results.

- Part 1 (50%)
- Part 2 (50%)