FYS4240

Lab5:

Queued message handler (QMH)

- **File reader**
- **UDP sender**

Introduction

In this lab you will first learn about the Queued Message Handler (QMH) template, which is fundamental to create **high performance, scalable and large programs running in parallel** on multiple CPU cores. Then you will make two programs based on the QMH template. Lastly, you will make an application (exe file) from your LabVIEW program.

See lecture 4 for basic information about UDP and the use of UDP in LabVIEW. See also Lecture 5-6 for DAQ with LabVIEW examples.

In this lab much of the LabVIEW code is provided, an your task is to complete the code.

Exercise 1 – Queued message handler (QMH) template

1. Read the PDF document called QMH description.
2. Open the QMH template project:
    - File - Create Project – Templates – Queued Message Handler
3. Open the program **Main.vi**.
4. Run the program and look at the code in the block diagram.
5. Look at the project and see how the sub-VIs used in the template is found under **Support VIs**. Have a look at how they are made by opening the VIs (by double click).
6. Look at the QMH html documentation, found under **Project Documentation**.

After finishing this exercise, you should understand the basics for how the QMH template works and how it can be used (with modifications/additions) to make different application programs. You will get a deeper understand after having done exercise 2, 3 and 4.

Exercise 2 – File reader program based on the QMH template

In this Exercise you will make a program to read data from a provided **binary file** and plot them in graph.

1. Open the provided project called **FileReaderQMH.lvproj**
    a. Hint: File – Open project
2. Open the program **Main.vi.** This program is made based on the QMH template. Only the functions to read data from the binary file are missing.
3. You need to complete the following states (cases) in the Message Handling Loop
    a. OpenFile
    b. ReadData
    c. CloseFile
4. When the program is completed, run the program and select the provided file **Example Binary File.dat** from the GUI and verify that you are able to read and graph the data (a sinusoidal signal) with your program.

Exercise 3 – UDP send program based on the QMH template

In this Exercise you will make a program to send data over a network using UDP.

1. Open the project **UDP_send**.
2. Open **Main_UDPsend.vi**
3. In the Event handling loop (EHL), add the **Enqueue Message VI** (by drag-drop from project window, or copy-paste from another state), with the Message **Start**
4. In the UDP send loop:
    a. Case (state) Start: Open UDP connection, local port.
    b. Case (state) SendData: Send data to remote address and remote port.
    c. Case (state) Exit: Close UDP connection.
5. Make sure there are no programming errors (broken grey arrow).
6. Make a **UDP receiver** program.
    a. Port number should be the same number as remote port number in sender.
    b. **Hint:** if you search for UDP under *Help - Find Examples* you will find the project Simple UDP, with the receiver code you need.
7. Verify that the **UDP receiver** get the data distributed by the **Main_UDPsend** program. Hint: to test with both sender and receiver on the same computer you can use the localhost (=127.0.0.1) IP address.

Exercise 4 – Extending the UDP send program

In this exercise we will look at how to extend the program we made in exercise 3. You do not need to do any programing (but you can if you want to).

Question: How can we extend our UDP send program to also include writing the data to a binary file, in parallel with sending data over UDP? Describe what you need to add (e.g. states and queues) and where.

Hint: Remember what you learned about the QMH template in exercise 1. You can also look at the sample project *Continuous measurement and logging*.

Exercise 5 – Build an application

In this Exercise you will make an executable (exe) program of the LabVIEW VI program you made in exercise 2.

1. Open the project from exercise 2.
2. Right-click the ***Build Specification*** and select ***New – Application (EXE)***.
3. Type FileReader as the specification name and target (exe) name.
4. Select ***Source Files***, click **Main.vi** and press the *blue right arrow* to add the Main.vi program to the Startup VIs.
5. Click ***Build***
6. When finished, click ***Done***
    - You should now get a new folder named builds, an in there you will find the application program FileReader.exe.

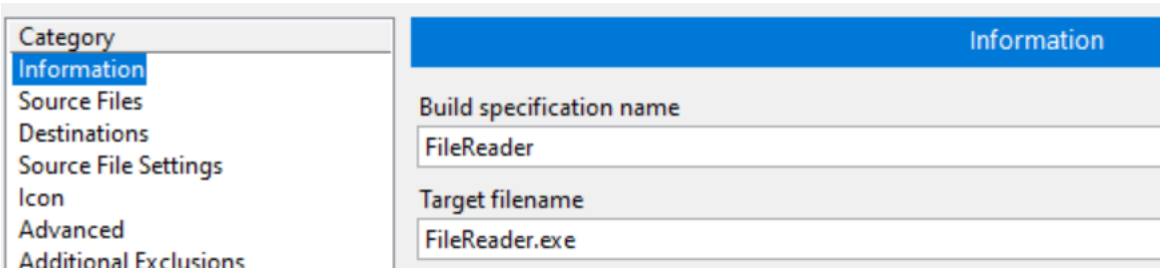
- You should now also see FileReader (target) under *Build Specification* in the project window. (Double clicking on this build and you can edit it and make a new based on the previous build settings, if you wanted to change something).

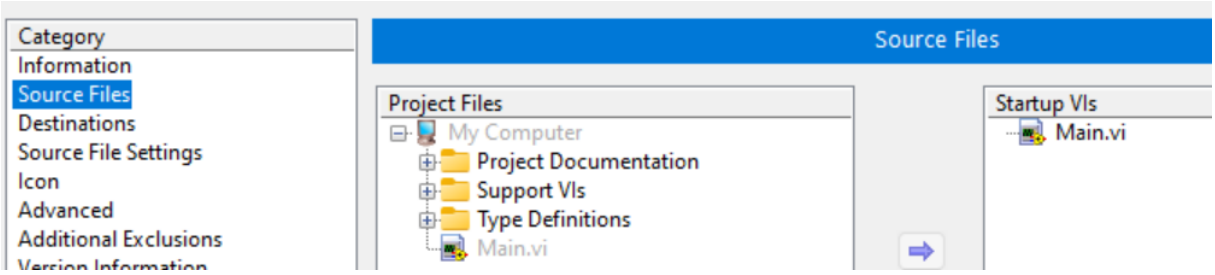
7. Run the FileReader.exe program (double click) and see that it is working like the program did when you did run it in the LabVIEW environment.

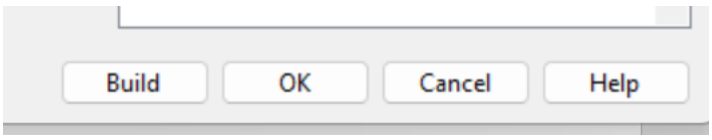
Notes:

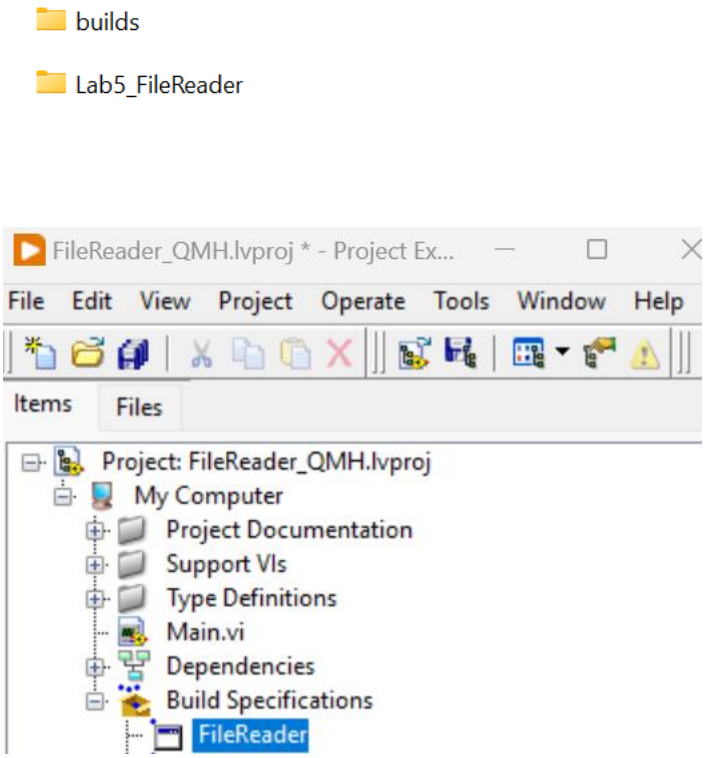
- This is only the minimum settings needed to make an exe, but it is usually sufficient. However, many options are available!
- If you wanted to run the FileReader.exe program on a computer where you do not have LabVIEW installed, you need to install the LabVIEW runtime environment (RE) on that machine. In order to get libraries that is required but not part of Windows. You can make an installer that includes the LabVIEW RE and the application program by:
  - Right-click the *Build Specification* and select ***New – Installer***
  - Under Source files you now add the FileReader application (exe) that you made previously.

Hints: See figures below.









What to hand in:

- The finished code from exercise 2 and exercise 3.
- The answer to the question in exercise 4.
- The exe file you made in exercise 5.