Itasdi

Innovative Teaching Approaches in development of Software Designed Instrumentation and its application in real-time systems

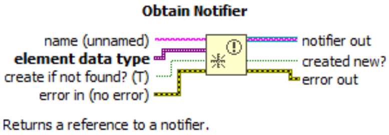
The Advanced Applications of LabVIEW

Lecture 2: Master/Slave Design Pattern

Co-funded by the Erasmus+ Programme of the European Union

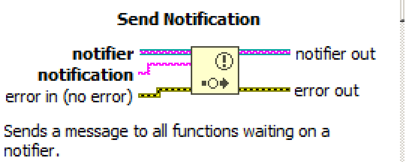
Application of Master/Slave Design Pattern

- The master/slave design pattern is used for parallel tasks.
- The pattern consist of one master loop and at least one slave loop. The master loop communicates with slave loops by notifier.
- The notifier is used to send the message to different part of code by using the functions:



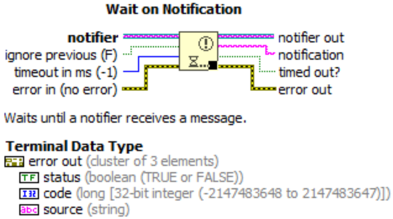
Obtain Notifier

Returns a reference to a notifier.



Send Notification

Sends a message to all functions waiting on a notifier.

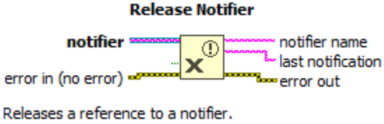


Wait on Notification

Waits until a notifier receives a message.

Terminal Data Type

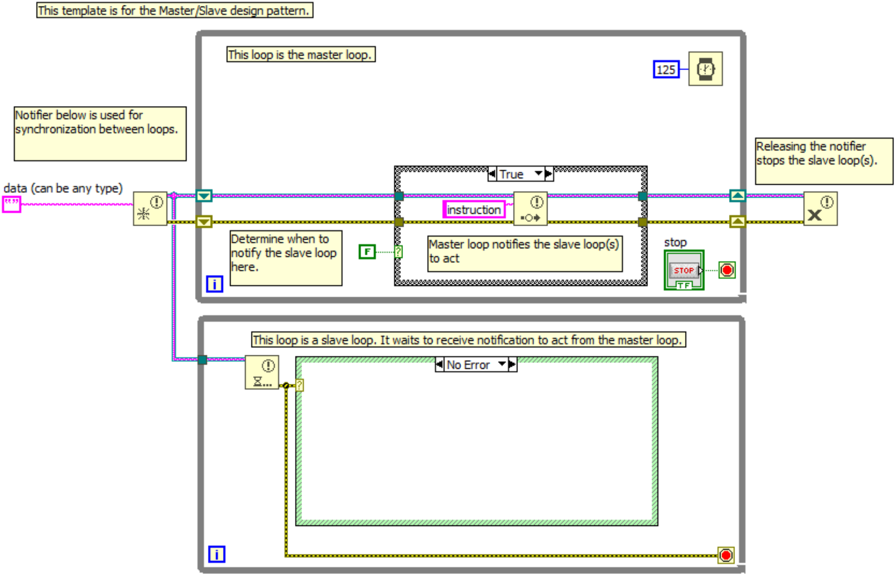
error out (cluster of 3 elements)
status (boolean (TRUE or FALSE))
code (long [32-bit integer (-2147483648 to 2147483647)])
source (string)



Release Notifier

Releases a reference to a notifier.

Structure of master/slave design pattern



This template is for the Master/Slave design pattern.

This loop is the master loop.

Notifier below is used for synchronization between loops.

Determine when to notify the slave loop here.

Master loop notifies the slave loop(s) to act

Releasing the notifier stops the slave loop(s).

This loop is a slave loop. It waits to receive notification to act from the master loop.

Comments

- The master/slave design pattern has one disadvantage. The notifier does not have the buffer to store the messages from master loop. When the slave loops work faster than the messages are added to notifier, it is not a problem. But you have to remember this and make sure it is designed like this.
- The master/slave design pattern is the simplest multiple loop design. In comparison to the producer/consumer design pattern you can create more than one slave loop.

Thank you for attention!

Lecture was prepared based on materials from: ”LabVIEW Core 3 Course Manual”.

This project has been funded with support from the European Commission. This communication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.