



BME
Budapesti Műszaki és Gazdaságtudományi Egyetem

KJIT
Közlekedés- és Járműirányítási Tanszék

Járműfedélzeti rendszerek 3

4. Előadás

Tartalom

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

- Ciklusok időzítése
- Alapvető programfelépítések
 - Szekvenciális (sequential programming)
 - Állapot (State programming)
 - Állapotgép (State machine)
 - Párhuzamosság (Paralellism)
 - Párhuzamos ciklusok
 - Master/slave
 - Producer/consumer

Ciklusok időzítése

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

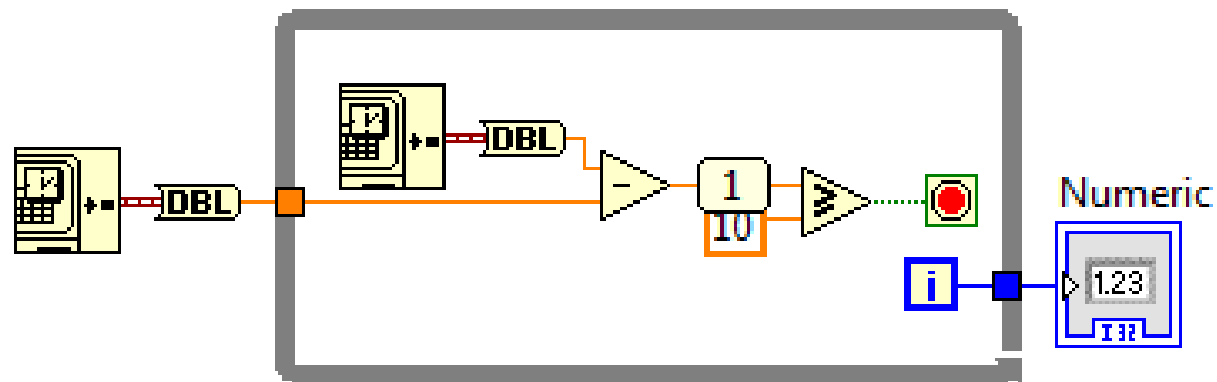
- Timerekkel
 - Elapsed time / Wait until ... (lásd 3. Előadás)
 - Dátumok összehasonlításával
- Szemaforokkal (Semaphores)
- Jelzőkkel (Notifiers)
- Sorokkal (Queues)
- Esetleg erre szolgáló lokális változókkal

Időzítés dátum segítségével 1

- A standard időzítésre használt subVI-ok kifuthatnak az ábrázolási helyből (integer!) és átfordulhatnak
- Ezért nagy futásidejű programoknál szükséges a kiváltásuk lebegőpontos számokkal
- Ekkor a dátumot (Get Date/Time is Seconds) Double-é alakítjuk és ezzel időzítünk

Időzítés dátum segítségével 2

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék



Probe Watch Window

Probe Display

| Probe(s) | Value | Last Update |
|------------|-----------|----------------------|
| Untitled 1 | | |
| [1] Probe | 10.001E+0 | 2013.03.05. 14:31:57 |

10,0006

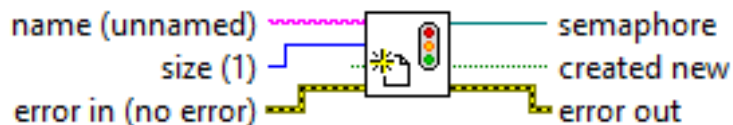
Szemaforok (Semaphores) 1

- Ugyanazon erőforrásokat használó ciklusok/subVI-ok időzítésére szolgál, kritikus kódrészletek véshetőek vele.
 - Pld. globális változók közös használata esetén, vagy
 - Külső eszköz használata esetén.
- Adatot nem továbbít, csak futást engedélyez (egyszerre csak egyet)
- Versenyhelyzetek megelőzhetőek vele
- Működés példaprogramban

Szemaforok (Semaphores) 2

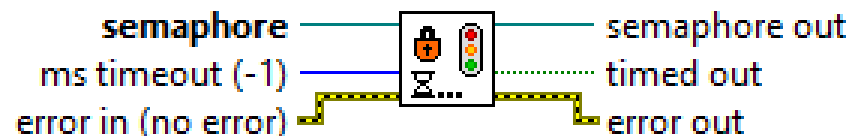
Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

Obtain Semaphore Reference.vi



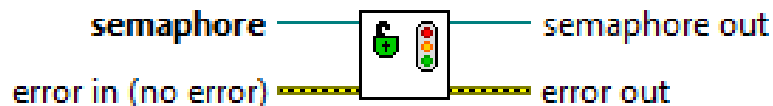
Obtains a reference to an existing semaphore or creates a new semaphore and returns a reference to that semaphore.

Acquire Semaphore.vi



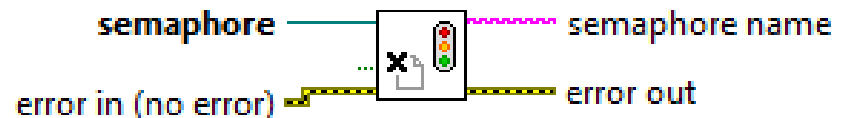
Acquires access to a semaphore.

Release Semaphore.vi



Releases access to a semaphore. If the Acquire Semaphore VI is waiting for the semaphore this VI releases, it stops waiting and continues execution.

Release Semaphore Reference.vi

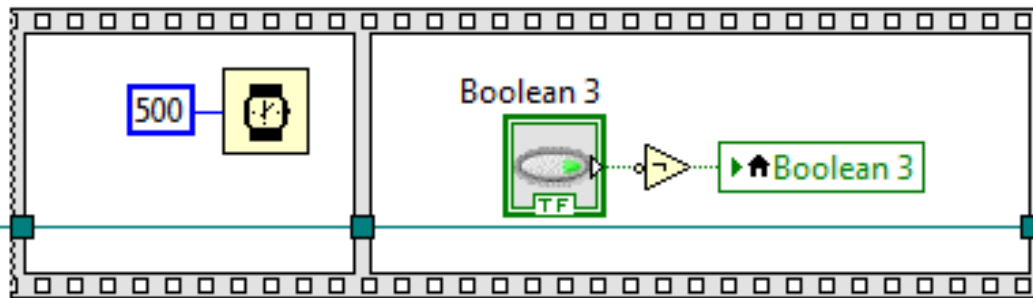


Releases a reference to a semaphore.

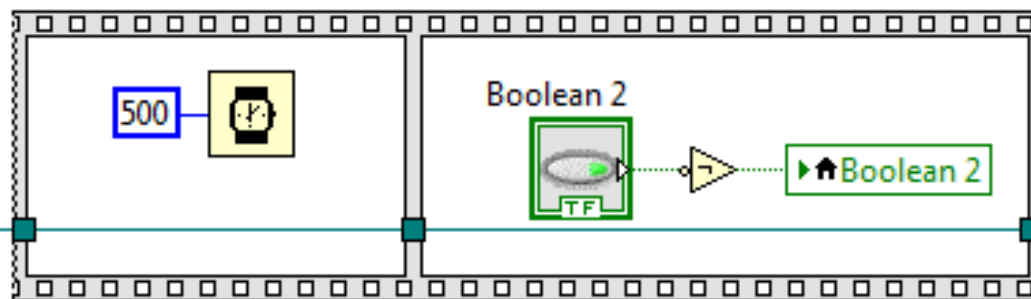
Szemaforok (Semaphores) 3

Obtain Semaphore Reference.vi

szemafor



Acquire Semaphore.vi



Release Semaphore.vi



stop 2

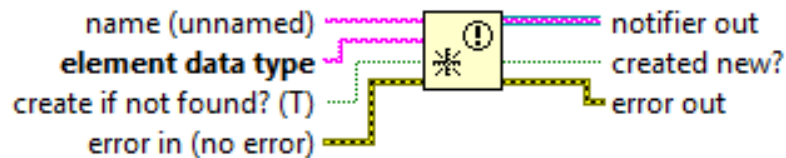


Jelzők (Notifiers) 1

- Legjobb kapacitáskihasználásra képes időzítés
- Felfüggeszthető vele egy block diagram futása, amíg az valamilyen külső információra vár.
- Küldhető vele bármilyen típusú üzenet (boolean, string, double, cluster, array, stb)
- Csak a legutóbbi üzenet kerül feldolgozásra (nincs memóriája)
- Működés példaprogramban

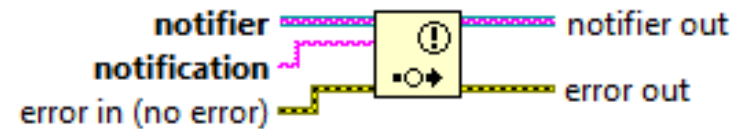
Jelzők (Notifiers) 2

Obtain Notifier



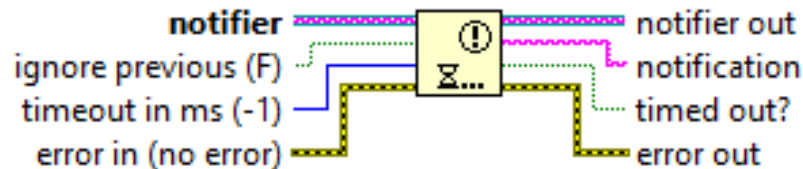
Returns a reference to a notifier.

Send Notification



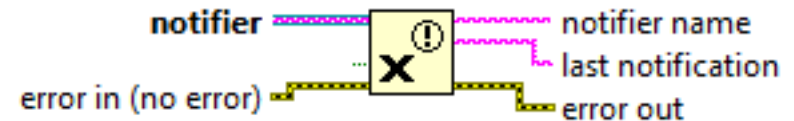
Sends a message to all functions waiting on a notifier.

Wait on Notification

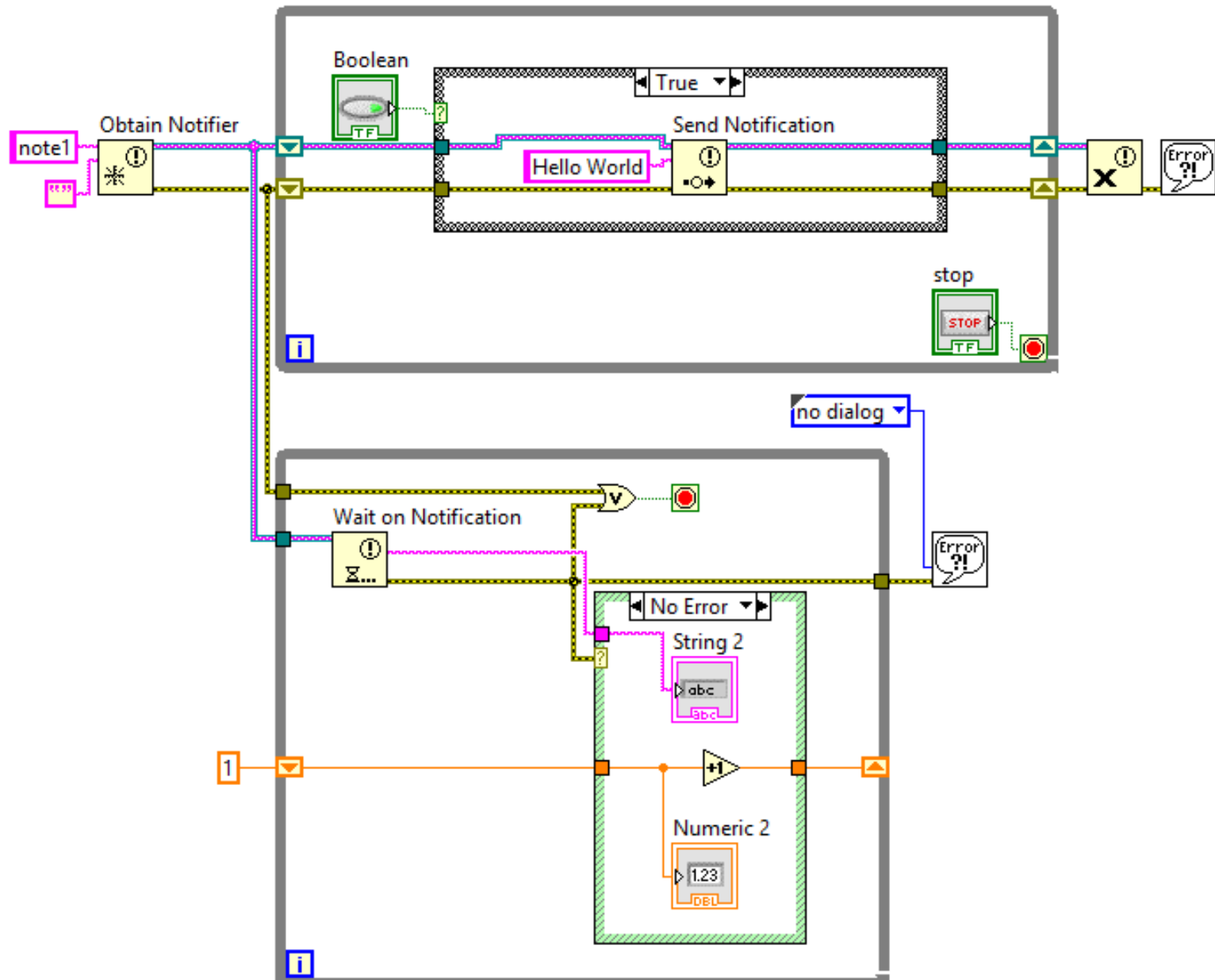


Waits until a notifier receives a message.

Release Notifier



Releases a reference to a notifier.

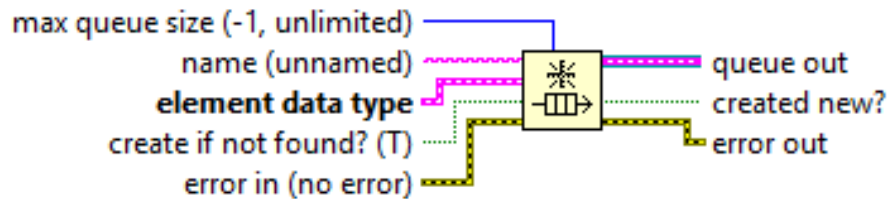


Sorok (Queues) 1

- Kommunikációt valósít meg, sorban tárolva az adatokat, bufferként működve.
- Rendelkezik memóriával (FIFO, vagy LIFO elv is megvalósítható vele)
- Bármilyen adattípus továbbítható vele
- Lekérhető a sor elemeinek száma, így hatékony kód készíthető
- Állapotgép enum bemenetét is tudja egymás után tárolni, így ha két állapotot hívunk meg egyszerre, nem veszik el adat
- Működés példaprogramban

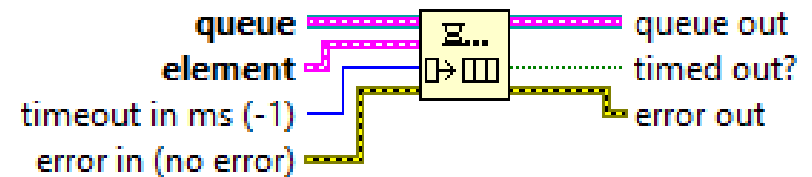
Sorok (Queues) 2

Obtain Queue



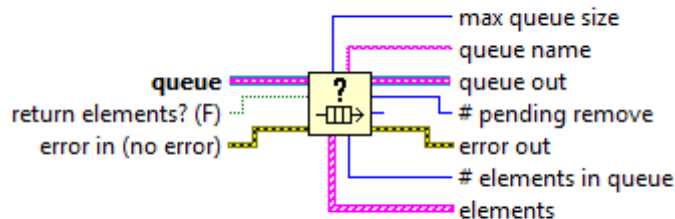
Returns a reference to a queue.

Enqueue Element



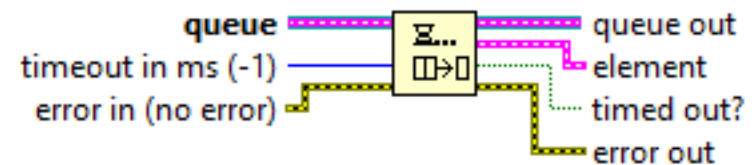
Adds an element to the back of a queue.

Get Queue Status



Returns information about the current state of a queue, such as the number of elements currently in the queue.

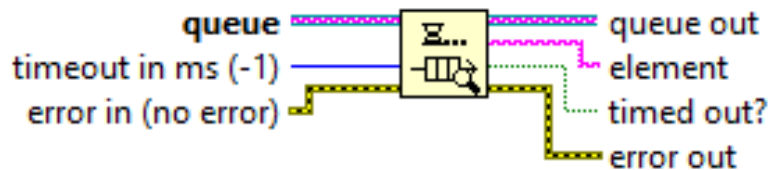
Dequeue Element



Removes an element from the front of a queue and returns the element.

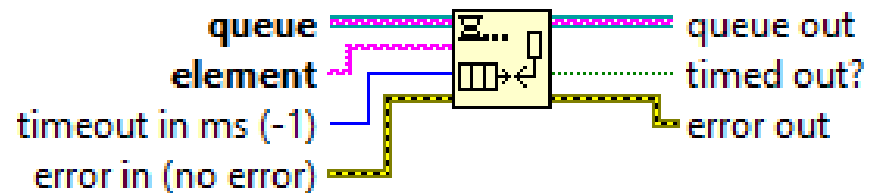
Sorok (Queues) 3

Preview Queue Element



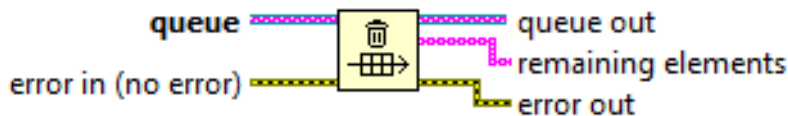
Returns the element at the front of the queue without removing the element from the queue.

Enqueue Element At Opposite End



Adds an element to the front of a queue.

Flush Queue



Removes all elements from a queue and returns the elements as an array.

Release Queue



Releases a reference to a queue.

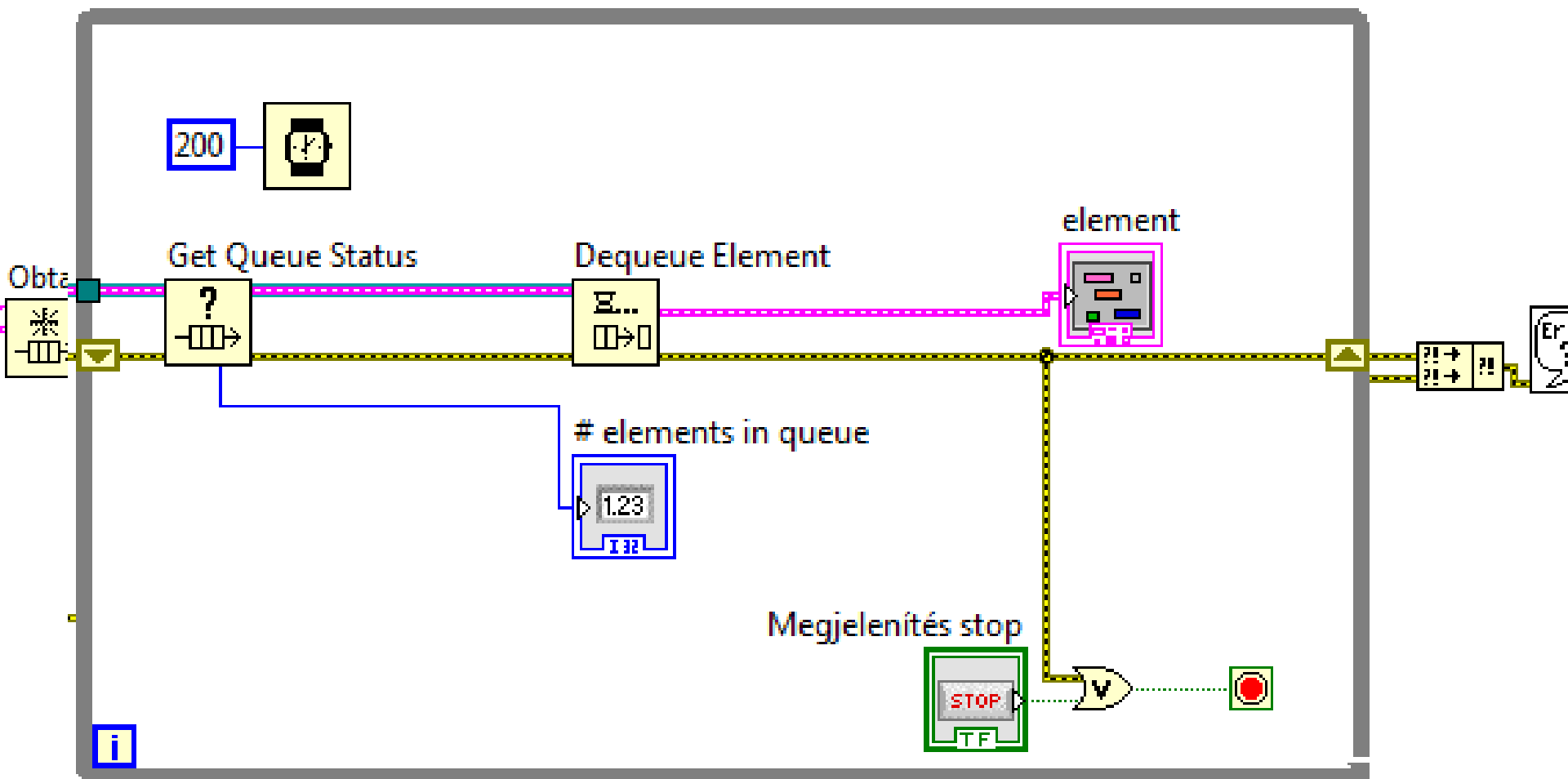
Sorok (Queues) 4

The screenshot shows a software window titled "Untitled 2" with a menu bar (File, Edit, View, Project, Operate, Tools, Window, Help) and a toolbar with icons for play, refresh, stop, and pause. The main area contains two "STOP" buttons, one for "Készítés stop" and one for "Megjelenítés stop". A label "# elements in queue" is positioned above a text box containing the value "183". Below this, there are two columns of data: "output cluster" and "element". Each column has two rows of data: "Numeric" and "String".

| output cluster | element |
|--------------------|--------------------|
| Numeric 168 | Numeric 915 |
| String 16:31:42 | String 16:31:33 |

Sorok (Queues) 5

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék



Alapvető programstruktúrák

Budapesti Műszaki és Gazdaságtudományi Egyetem *Közlekedés- és Járműirányítási Tanszék*

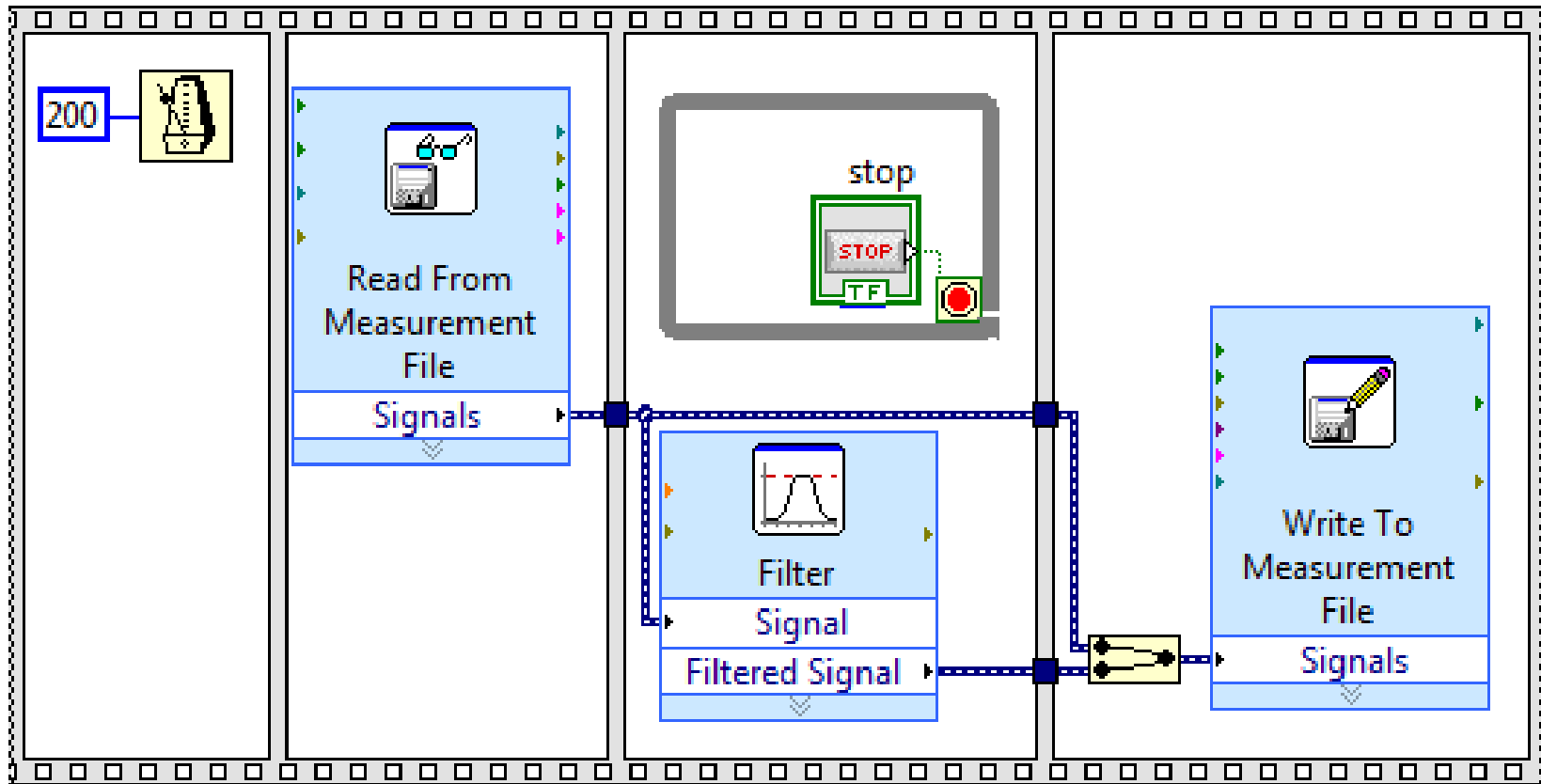
- Egyszerű ciklus
 - Szekvenciális (lehet ciklus nélküli is)
 - Állapotgép
- Több ciklus
 - Párhuzamos (független)
 - Master-Slave
 - Producer-Consumer
- Ezek egymásba is ágyazhatók, kombinálhatók

Szekvenciális struktúra

- A lépések egymás után hajtódnak végre
- Hiba esetén az azonos szekvenciában lévő egyéb subVI-ok még végrehajtódnak (ha bemenetként nem kapják meg az error clustert)
- Példák:
 - Egyszerű Dataflow VI
 - Stacked Sequence
 - Formula Node
- Ritkán használt, állapotgéppé szokás átalakítani

Szekvenciális Block Diagram ismétlés

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék



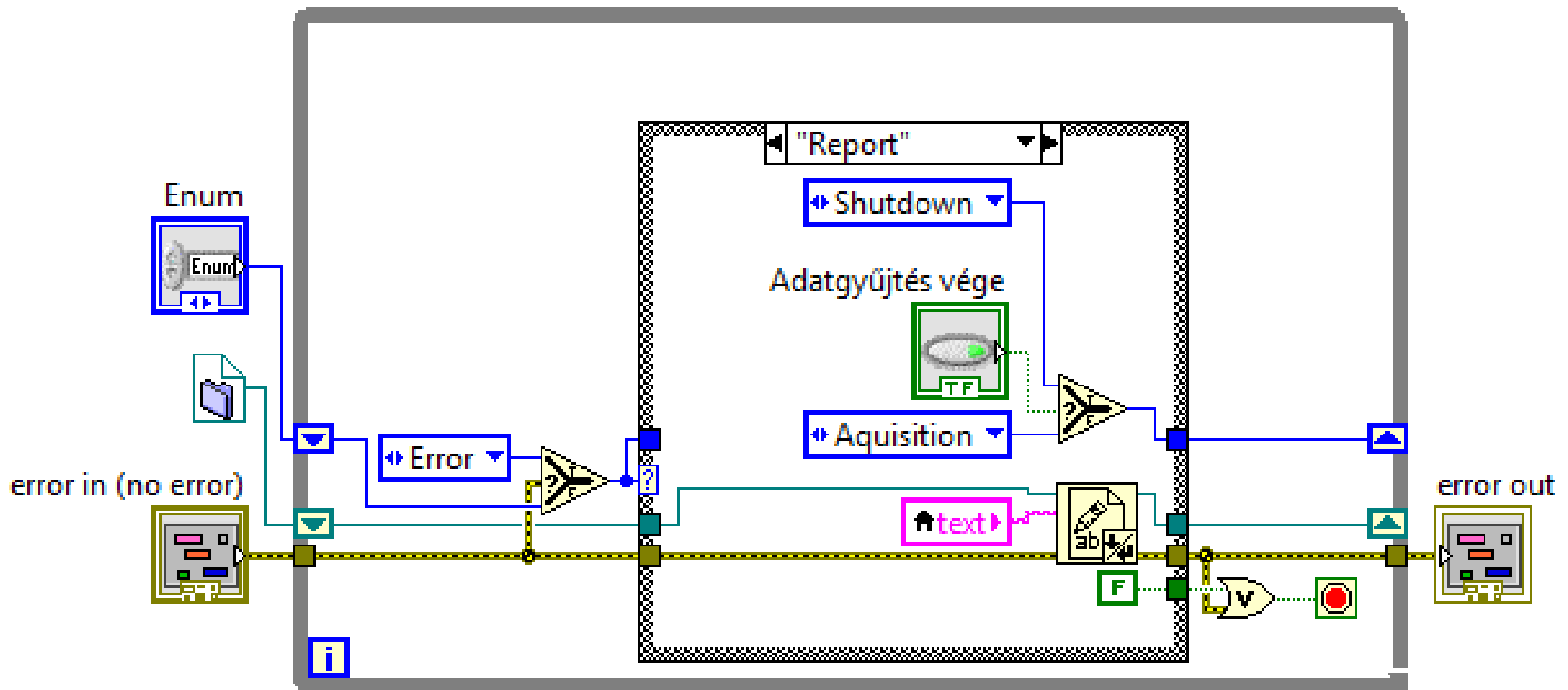
Állapotgép

Budapesti Műszaki és Gazdaságtudományi Egyetemtem Közlekedés- és Járműirányítási Tanszék

- Általánosan használt
- While ciklusba ágyazott Case
- A Case választó bemenete egy enum
- A Case választó bemenete a while ciklusban shift regiszterként kerül eltárolásra, így az előző állapottól, és annak bemeneteitől függ a következő
- Általánosságban van inicializáló, befejező és hibakezelő/hibatovábbító ága

Állapotgép – Block diagram ismétlés

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék



Független, párhuzamos ciklusok

- Több párhuzamos ciklus (lehetnek más időzítésűek)
- Olyan esetekben alkalmazott, amikor a ki-, illetve bemenetek (egy csoportja)
 - Egyidejűek
 - Függetlenek
 - Külön kezelhetőek
- Önmagában felhasználóval való kommunikációra használt

Master/Slave

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

- Több párhuzamos ciklusból áll
- 1 Master, 1 vagy több Slave
- Célszerű egy részfeladathoz 1 Slave-et rendelni (pl. robotkar mozgásánál minden aktuátorhoz) – beavatkozási idő csökken
- 2 Slave nem írhat ugyanabba a változóba (versenyhelyzet!)
- Bizonyos esetekben a slave gyorsabban kell hogy fusson mint a master loop (egyébként: adatvesztés)

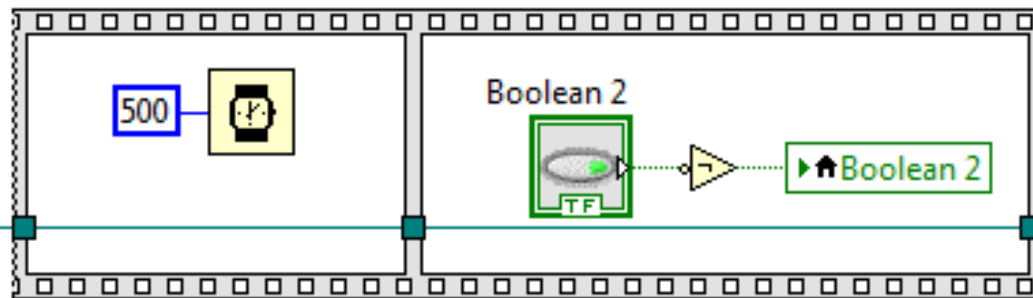
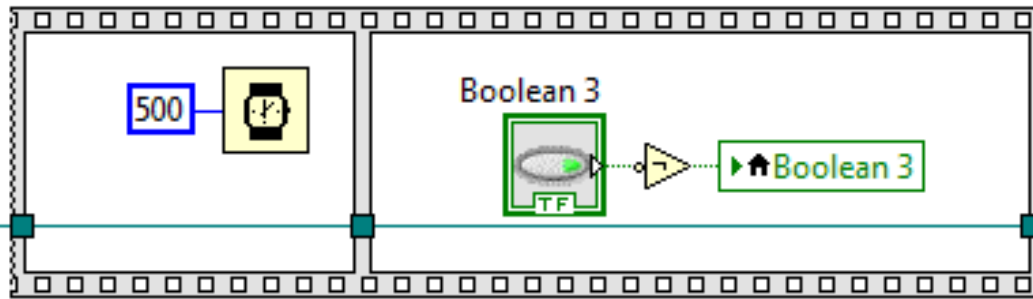
Váltott működés

- Amennyiben ugyanazon erőforrást (változó, fájl, hardvereszköz) két subVI akarja használni, meg kell akadályozni a versenyhelyzetet
- Ekkor Szemaforok használatával biztosítjuk mindkét programrészlet futását

Szemaforok (Semaphores) 2

Obtain Semaphore Reference.vi

szemafor



Acquire Semaphore.vi



Release Semaphore.vi

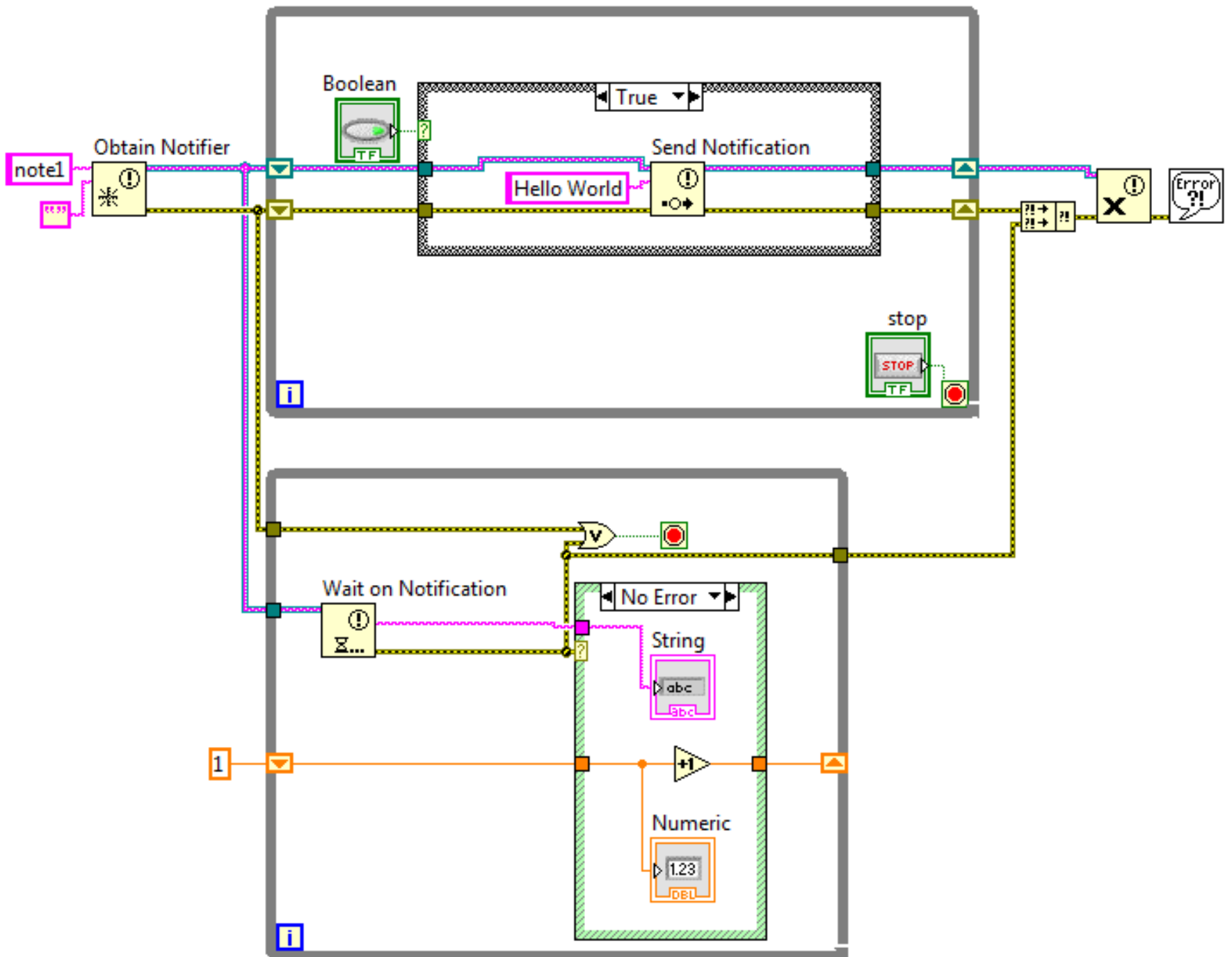


stop 2



Meghívott működés

- A Slave csak akkor futtatja le a programmagját, amennyiben arra a Mastertől utasítást kap (kisebb erőforrásigény)
- Megvalósítása általában Jelzőkkel (Notifier) történik



Producer-Consumer

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

- A Master-Slave-en alapul
- Különböző időzítésű ciklusok termelik és dolgozzák fel az adatokat
- Akkor használjuk, amikor több adatot kell kinyernünk, mit aktuálisan fel tudnánk dolgozni
- Az adatvesztés elkerülésére sorokat (Queue) használunk

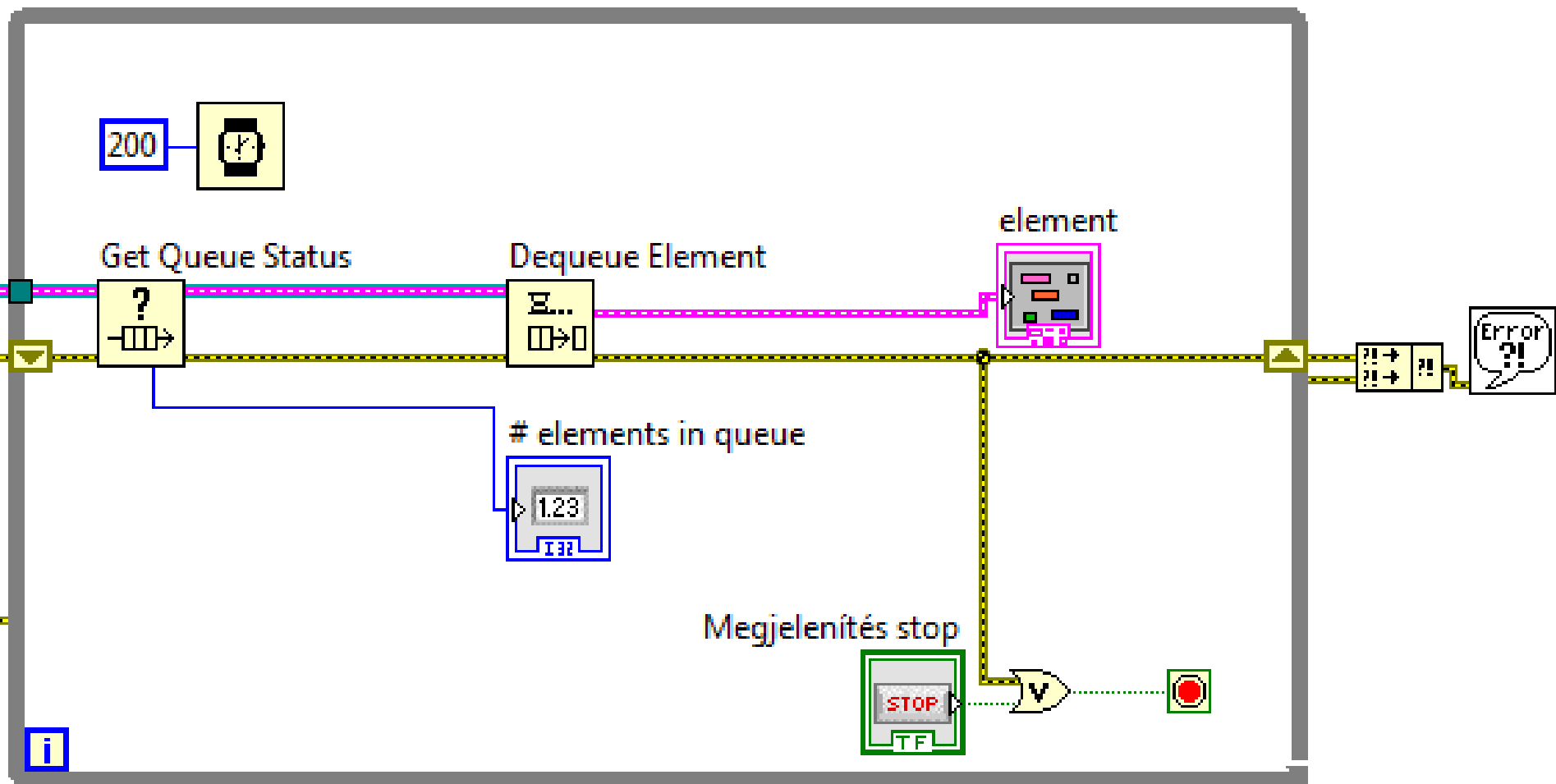
Sorok (Queues) 2

The screenshot shows a software window titled "Untitled 2" with a menu bar (File, Edit, View, Project, Operate, Tools, Window, Help) and a toolbar with icons for play, refresh, stop, and pause. The main area contains two "STOP" buttons, one for "Készítés stop" and one for "Megjelenítés stop". A label "# elements in queue" is positioned above a text box containing the number "183". Below this, there are two columns of data: "output cluster" and "element". Each column has two rows of data, one for "Numeric" and one for "String".

| output cluster | element |
|--------------------|--------------------|
| Numeric 168 | Numeric 915 |
| String 16:31:42 | String 16:31:33 |

Sorok (Queues) 3

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék





BME

Budapesti Műszaki és Gazdaságtudományi Egyetem

KJIT

Közlekedés- és Járműirányítási Tanszék

Köszönöm a figyelmet