

***CONTROL EN TEMPS  
REAL MITJANÇANT EL RTW  
DE MATLAB-SIMULINK***

***Bianca M. Innocenti***

# **CONTROL EN TEMPS REAL MITJANÇANT EL RTW DE MATLAB-SIMULINK**

## **Introducció**

Per poder realitzar aquesta pràctica cal tenir clars uns quants conceptes relacionats amb el Simulink. A continuació es defineixen els més importants:

### *Real-time workshop*

El *Real-Time Workshop* (RTW) proveeix les utilitats per convertir els models de *Simulink* en codi C i, posteriorment, amb un compilador de C, compilar el codi i convertir-lo en un executable de temps real.

Per a compilar el codi C es necessita instal·lar un compilador de C, el *Real-Time Workshop* permet els següents compiladors: *Microsoft Visual C/C++*, *Watcom C/C++*. En el nostre exemple ja s'hi troba instal·lat el **Microsoft Visual C/C++** per tant utilitzarem aquest programa per compilar el codi C.

El compilador de C crea el codi executable a partir del codi C generat a partir del *Real-Time Workshop* i del codi C de les *s-functions* que hi ha creades. En realitat no utilitzarem el *Real-Time Workshop*.

### *Real-Time Windows Target*

El *Real-Time Windows Target* (RTWT) està dissenyat per a tenir una major flexibilitat i rapidesa que el *Real-time Workshop*. Aquesta rapidesa permet la sintonia de paràmetres i senyals durant el funcionament en temps real, però augmenta la mida del codi generat.

La integració entre el *Simulink* en mode extern i el *Real-Time Windows Target* permet utilitzar els models de *Simulink* com una interfície gràfica per:

- **Visualitzar senyals:** visualització de senyals mentre funciona l'aplicació en temps real, s'utilitzen els mateixos blocs *Scope* del Simulink emprats per a visualitzar senyals en processos simulats.
- **Sintonia de paràmetres:** es fa mitjançant la utilització de les caixes de diàleg dels blocs de paràmetres a l'aplicació mentre aquesta es troba funcionant en temps real.

D'altres aplicacions per al *Real-Time Windows Target* inclouen:

- **Control Real-time:** Prototipus automotrius, perifèrics computeritzats, sistemes de control instrumentals.
- **Real-Time hardware a la simulació en llaç tancat:** Prototipus de controladors connectats a plantes físiques. Plantes prototipus connectades a controladors actuals
- **Tutorial:** Ensenya conceptes i procediments per a modelar, simular i testejar sistemes en temps real.

### *Real-time kernel*

El *Real-Time Windows Target* utilitza el *Real-Time Kernel* per a assegurar-se que la aplicació de temps real funciona efectivament en temps real. El *Real-Time Kernel* funciona quan la CPU es posa a zero (*kernel mode*) i utilitza el rellotge intern del PC per a mesurar el temps. El real-time kernel també comunica amb el hardware I/O (Entrada/Sortida) utilitzant drivers, blocs de I/O i verifica la instal·lació de la placa I/O.

**Mode Extern de Simulink:** La comunicació entre el *Simulink* i la aplicació *real-time* es fa a través del mòdul d'interfície de mode extern. Aquest mòdul connecta directament amb el *real-time kernel* i és utilitzat per a començar la aplicació *real-time*, canviar paràmetres i recuperar dades. Es a dir, és la interfície de comunicació per a poder passar dades externes al *Simulink*.

### *Acció real-time*

L'aplicació *real-time* permet funcionar el temps real en el PC i té les següents característiques:

- **Codi compilat:** Creat a partir del codi C generat a partir del compilador Microsoft Visual C/C++ o el Watcom C/C++.
- **Relaciona el model Simulink:** L'executable conté una forma binària de tots els components del model de Simulink connexions entre blocs, dependències de temps i variables en els blocs de Simulink.
- **Relació amb el kernel:** L'executable ha de ser carregat i executat directament pel *real-time windows target kernel*. No pot ser executat sense el kernel. El kernel executa un driver de mode kernel, intercepta interrupcions del rellotge intern del PC, manté les senyals del rellotge per al sistema operatiu de Windows, i assegura l'execució en temps real de la aplicació real-time.

- **Checksum:** S'assegura que si canviem els paràmetres del nostre model durant l'execució aquests es portin a terme. Si fem canvis a la part estructural del nostre model el checksum no els té en compte i caldria tornar a compilar.

### *Adquisició de senyals i anàlisi*

Els senyals poden ser adquirits, representats i gravats utilitzant blocs Scope. Això permet observar el comportament del model durant la simulació o l'aplicació mentre funciona en temps real. Es poden adquirir senyals mentre la aplicació es troba funcionant en temps real utilitzant:

- **Signal Tracing:** És el procés d'adquisició i visualització de senyals durant el funcionament en temps real. Això permet la adquisició de senyals i la visualització d'aquestes a l'ordinador mentre funciona l'executable.
- **Signal Logging:** És el procés d'adquisició de senyals durant el funcionament en temps real. Després d'arribar al final de l'execució es poden dibuixar i analitzar les dades. Aquestes dades es poden emmagatzemar en variables del Matlab o en disquets.

### *Procediment per a la instal·lació de la tarja d'adquisició de dades en el matlab (simulink):*

A continuació es donen els passos a seguir per a poder instal·lar la placa amb la que enviarem i rebrem dades del procés.

1. Si s'introdueix la següent comanda:

***rtwho***

El Matlab ens informa de l'estat de la connexió amb temps real amb la tarja:

***Real Time Windows Target version 1.00 (C) The MathWorks, Inc.  
1994-1998***

***Matlab performance = 100.0%***

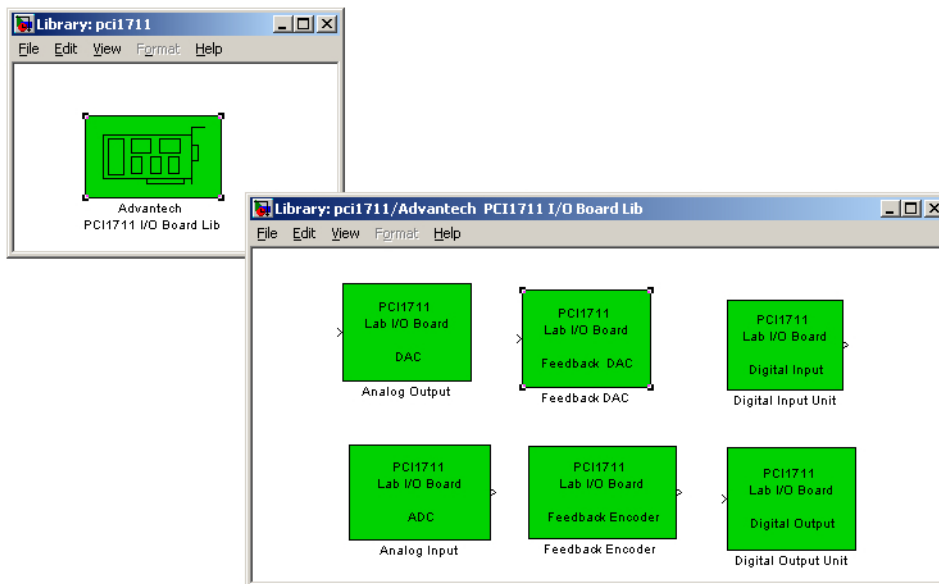
***Kernel timeslice period = 1 ms***

2. Configurar com a compilador predeterminat el Microsoft Visual C/C++ version 6.0 , s'utilitza la comanda:

***mex –setup***

Triant l'opció corresponent al Microsoft Visual C/C++ version 6.0.

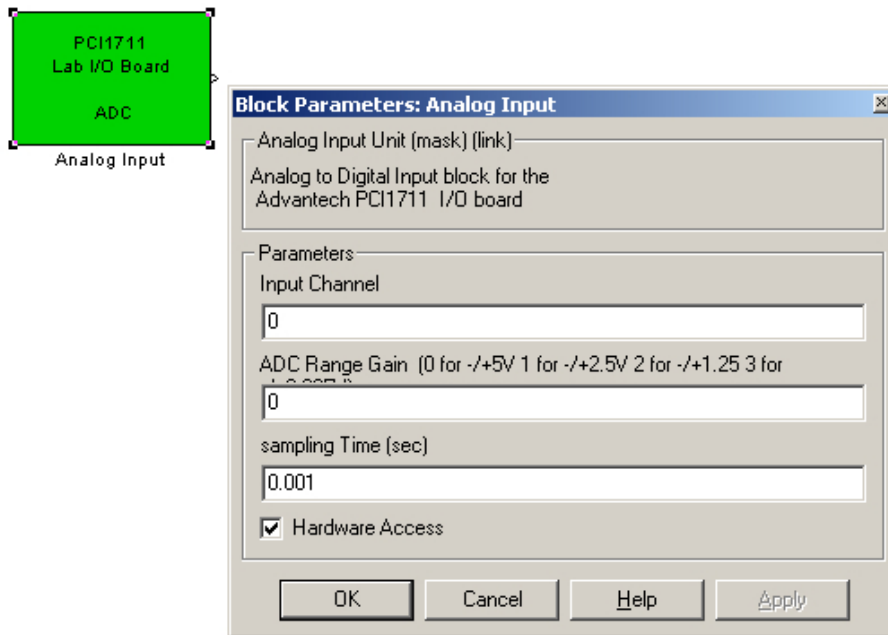
3. Iniciar el Simulink i al prompt de Matlab escriure pci1711. La pantalla següent apareixerà. Clickant dues vegades a sobre s'obre la finestra que conté les funcions de lectura i escriptura de les dades.



***Figura 1. Llibreria PCI1711 i els seus components.***

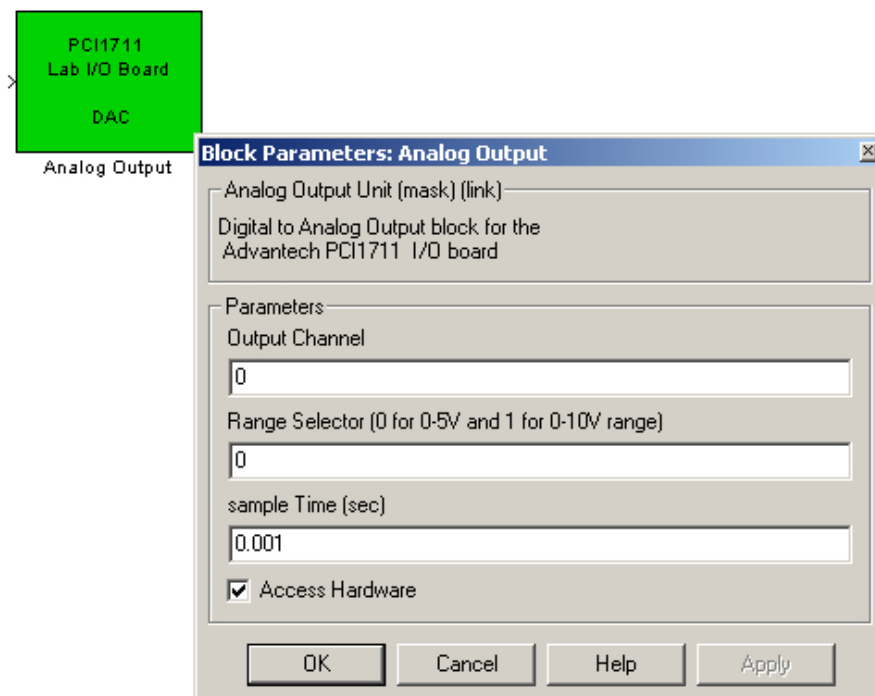
4. Crear un model Simulink, i copiar els blocs d'entrada i sortida analògiques.
5. Fer doble click sobre el bloc Analog Input.

En aquesta finestra es poden seleccionar el canal d'entrada (0-15), el rang de tensió d'entrada (que ha de ser obligatòriament 0 així la tensió podrà variar de -10 a 10 V, encara que a la finestra digui de -5 a 5 V) i el temps de mostreig.



**Figura 2.** Finestra de configuració de l'entrada analògica.

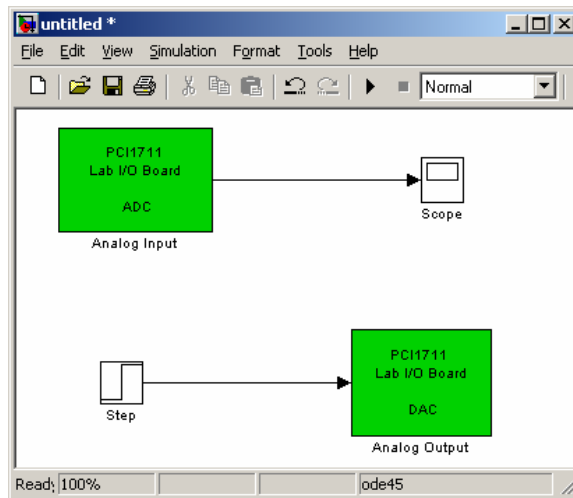
6. Configurar el bloc de sortida.



**Figura 3.** Finestra de configuració de la sortida analògica.

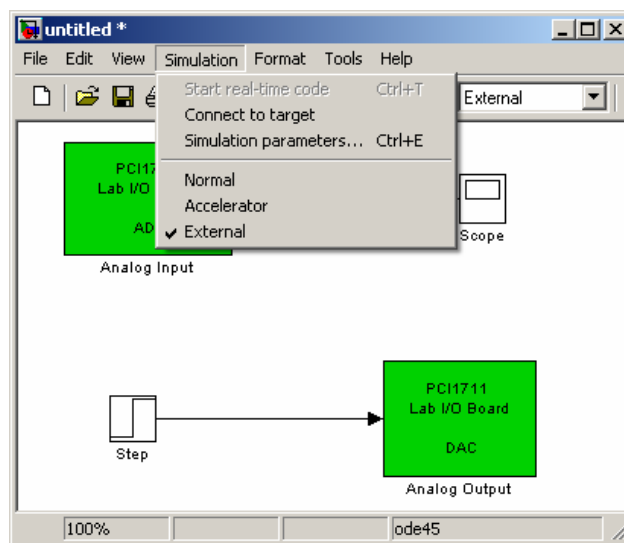
De la mateixa manera que per a l'entrada, s'han d'escollir els canals de sortida (0-1), la tensió de sortida (0-10V) i el temps de mostreig, que ha de ser el mateix que el seleccionat en l'entrada.

7. Crear el diagrama de blocs, per exemple com el mostrat a continuació:

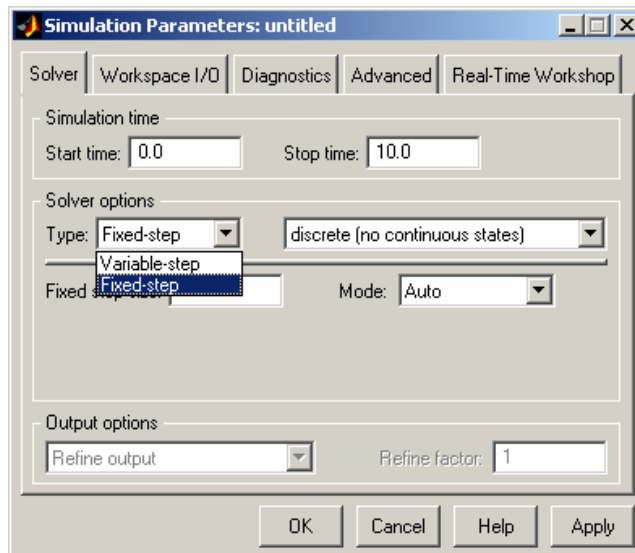


8. Un cop generat el diagrama de blocs cal guardar-lo.

9. Seleccionar l'opció *External* al menú *Simulation*.

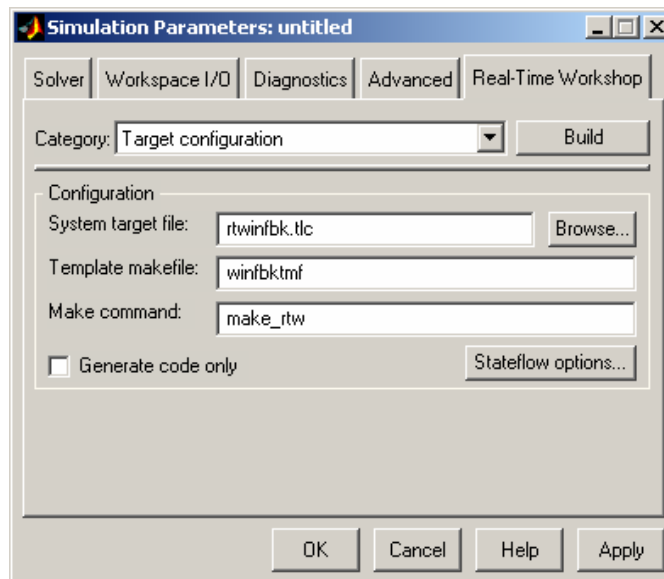


10. Seleccionar l'opció *Simulation parameters* al menú *Simulation*.



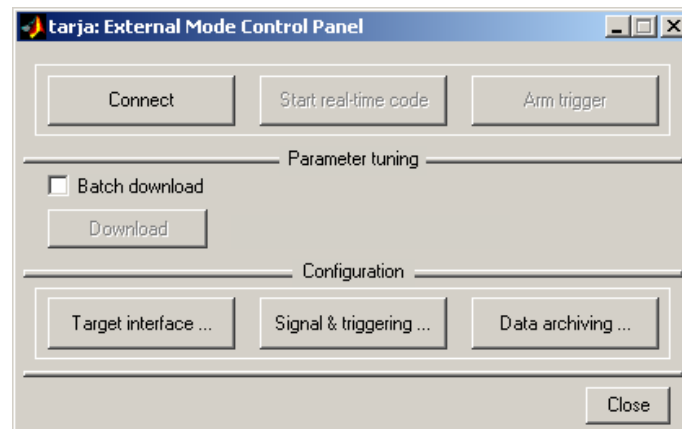
S'ha d'escollir *Fixed-step* com a valor de *Type* en *Solver Options*. També s'ha de triar el temps que es vol durir l'execució (*Stop time*) i el temps de mostreig (*Fixed step size*).

11. En aquesta mateixa finestra, anar a *Real-Time Workshop*.

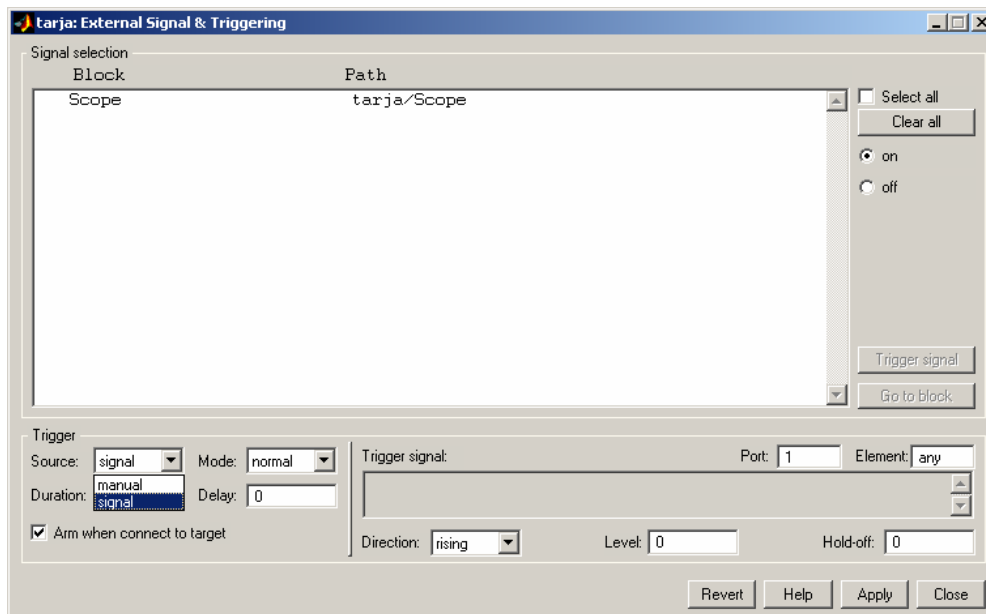


Clicar a sobre de *Browse* a *System target file* i triar l'opció ***rtwinfbk.tlc***. Una vegada fets els passos següents cal tornar a aquesta finestra i clicar a sobre el botó *Build*.

12. Anar a *External model control panel* de *Tools*.



13. Clicar a sobre de *Signal & triggering*



En aquesta finestra cal seleccionar cadascun dels blocs *Scope* i posar-los a **on** per a que mostrin les dades una vegada comenci a executar-se l'aplicació en temps real. Permet seleccionar quins d'ells estaran actius durant la simulació. A més es necessari, seleccionar *signal* a *Source* en *Trigger* i el checkbox *Arm when connected to target*.

El més important és definir correctament el paràmetre *Duration*. L'opció *Duration* es refereix al nombre de temps de mostreig que seran necessaris per fer la simulació d'acord amb la duració de la mateixa i el temps de mostreig. Per tant:

***Duration = Temps simulació/Temps de mostreig.***

Si *Duration* és inferior a la requerida per la simulació, aquesta es parará abans que el temps final establert.

14. Clickar en *Apply* i *Close* i tornar a l'opció *Simulation parameters* al menú *Simulation* en la subfinestra *Real-Time Workshop* i fer un *Build*.

15. En aquest moment es disposa del model compilat i es pot procedir a executar l'aplicació en temps real. Per això cal anar a l'opció *Connect to target* del menú *Simulation*.

16. Una vegada Simulink s'hagi connectat a la tarja, l'opció *Start real-time code* del menú *Simulation* estarà visible i es podrà començar l'execució de l'aplicació.

17. Consideracions generals:

- Entrades analògiques: 0 - 3.
- Sortides analògiques: 0-1.
- No funcionarà correctament per temps de mostreig inferiors a 0.001 segons.
- Durant l'execució del programa, amb la comanda ***rtwho*** es pot veure quin és l'estat de recursos del sistema. Es recomana que mai sigui inferior al 50%.

*Real Time Windows Target version 1.00 (C) The MathWorks, Inc. 1994-1998*

*Matlab performance = 93.6%*

*Kernel timeslice period = 1 ms*

*TIMERS: Number Type Period Running*

*1 In 0.001 Yes*

*2 In 0.001 Yes*

*32 Out 0.001 Yes*

*HISTORIES: Name Size Mode Link \_history\_ext 1 by 1 0*

*DRIVERS: Name Address Inputs Outputs Advantech PCL-812PG 0x220*

*32 18 Real Time Model 0 1 0*