

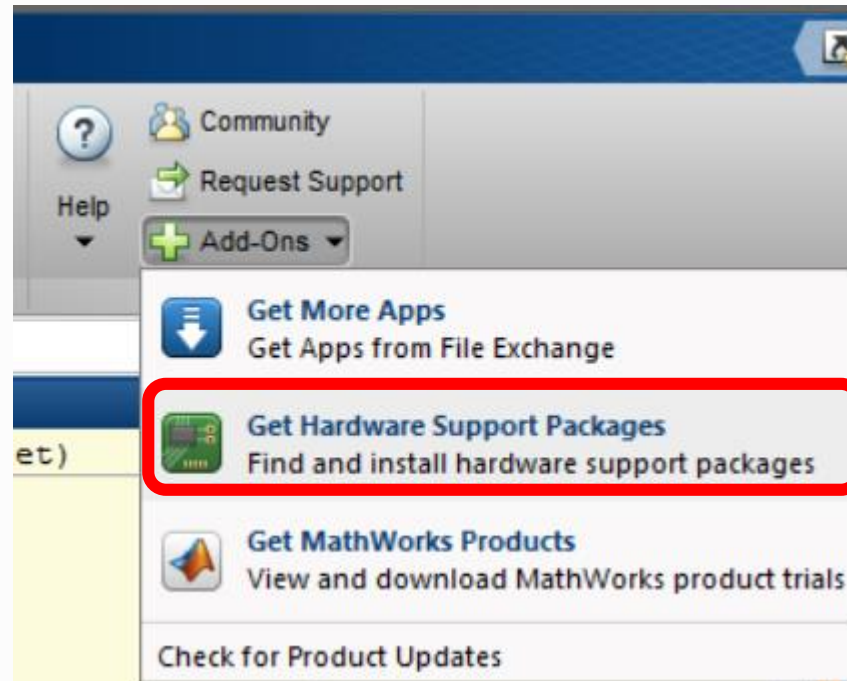


Guía paso a paso para usar Arduino Support from Simulink

Edilberto Carlos Vivas González

Instalación del paquete

En la pestaña “Home” click sobre “Add-ons” y luego “Get Hardware Support Packages”



Instalación del paquete

Buscar el soporte para Arduino y continuar con todos los pasos de la instalación

The screenshot shows the 'Support Package Installer' window. The title bar reads 'Support Package Installer'. The main area is titled 'Select support package to install'. A dropdown menu shows 'Show: All (61)'. Under 'Support for:', a list of hardware categories is shown, with 'Arduino Uno/Nano/Mega2560' selected and highlighted with a red box. To the right, a table lists 'Support packages:'. The first row is highlighted with a red box and contains the following data:

Action	Installed Version	Latest Version	Required Base Product	Supported Host Platforms
1 <input checked="" type="checkbox"/> Update	14.1.2	14.1.3	Simulink	Win32,Win64,Mac64

At the bottom, the 'Installation folder:' is set to 'C:\MATLAB\SupportPackages\R2014a'. The 'Next >' button is highlighted with a red box. A tooltip over the table area reads 'Choose a support package to install, reinstall, or update'.

Instalación del paquete

Desde la página de Arduino descargar e instalar los drivers



The screenshot shows the Arduino IDE download page. The browser address bar shows 'arduino.cc/en/Main/Software'. The navigation menu includes 'Buy', 'Download', 'Products', 'Learning', 'Forum', 'Support', and 'Blog'. The main heading is 'Arduino IDE' with the version '1.0.6'. Under the 'Download' section, there is a list of options: 'Windows Installer, Windows ZIP file (for non-administrator install)', 'Mac OS X', 'Linux: 32 bit, 64 bit', and 'source'. The 'Windows Installer, Windows ZIP file' option is highlighted with a red box. To the right, under 'Next steps', there are links for 'Getting Started', 'Reference', 'Environment', 'Examples', 'Foundations', and 'FAQ'. Below this, there is a section for 'Arduino 1.5.8 BETA (with support for Arduino Yún and Arduino Due boards)'.

Arduino - Software

arduino.cc/en/Main/Software

Buy Download Products Learning Forum Support Blog LOG IN SIGN UP

Arduino IDE

Arduino 1.0.6

Download

Arduino 1.0.6 (release notes):

- Windows Installer, Windows ZIP file (for non-administrator install)
- Mac OS X
- Linux: 32 bit, 64 bit
- source

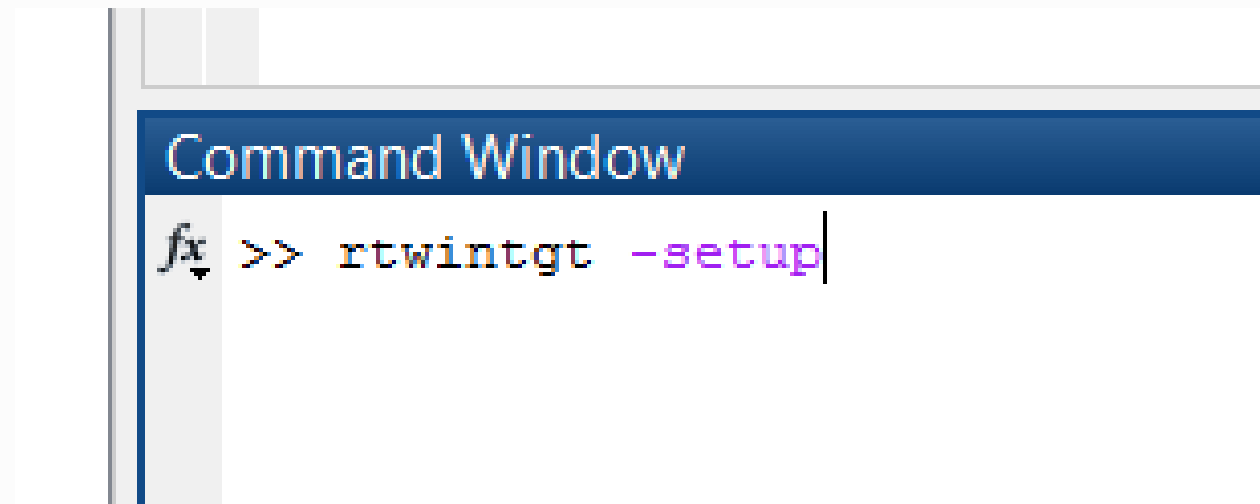
Next steps

- Getting Started
- Reference
- Environment
- Examples
- Foundations
- FAQ

Arduino 1.5.8 BETA (with support for Arduino Yún and Arduino Due boards)

Configuración de MatLab

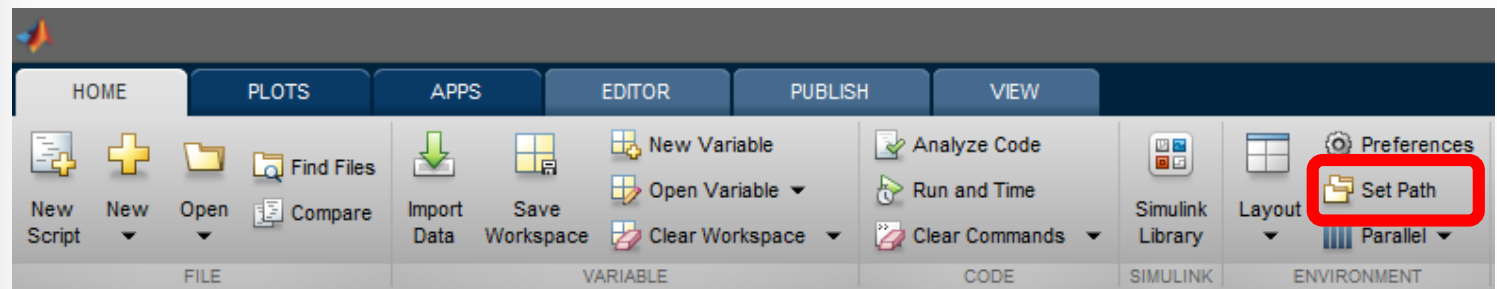
Instalar “Real Time Windows Target” escribiendo en la ventana de comandos
`rtwintgt -setup`



```
Command Window  
fx >> rtwintgt -setup|
```

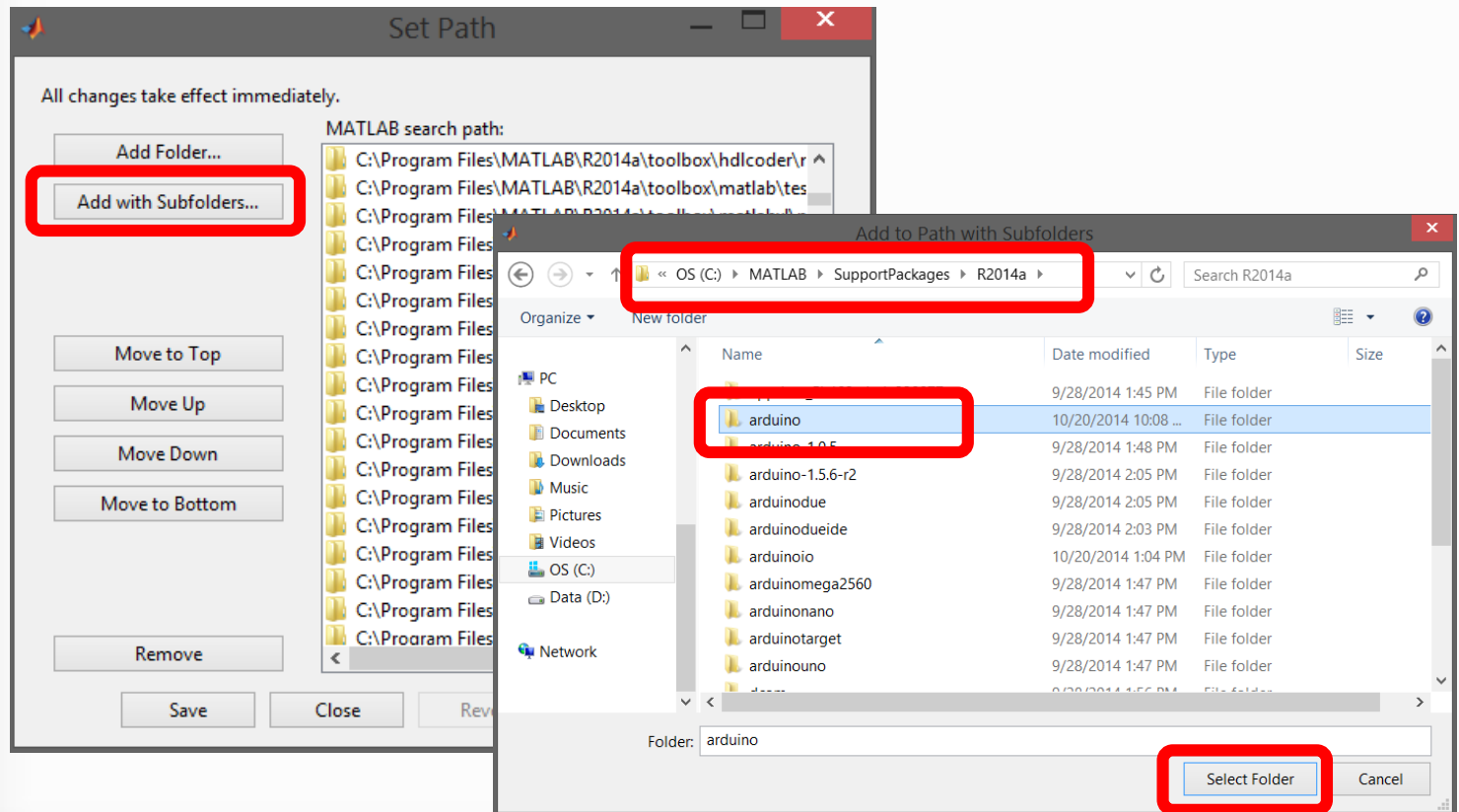
Configuración de MatLab

Agregar la carpeta donde se encuentran los archivos de Arduino al path de MatLab



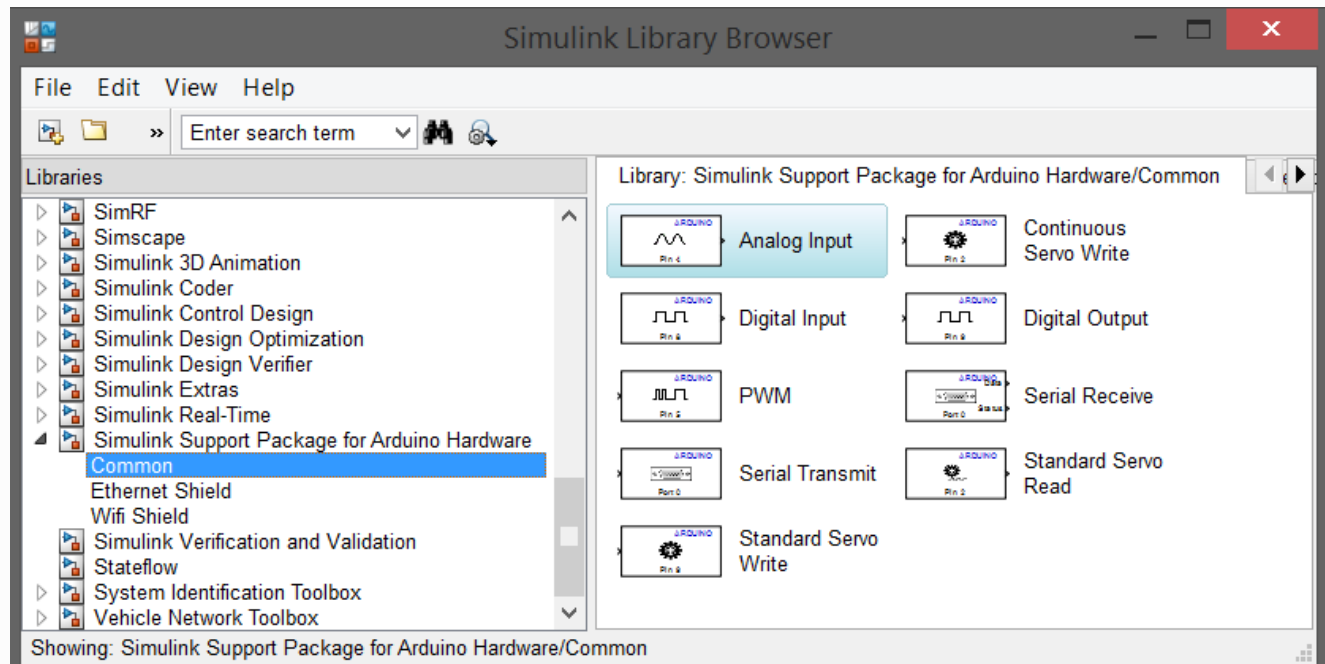
Configuración de MatLab

Agregar la ruta de la carpeta de Arduino. La carpeta de trabajo de Matlab debe estar en la misma ruta.



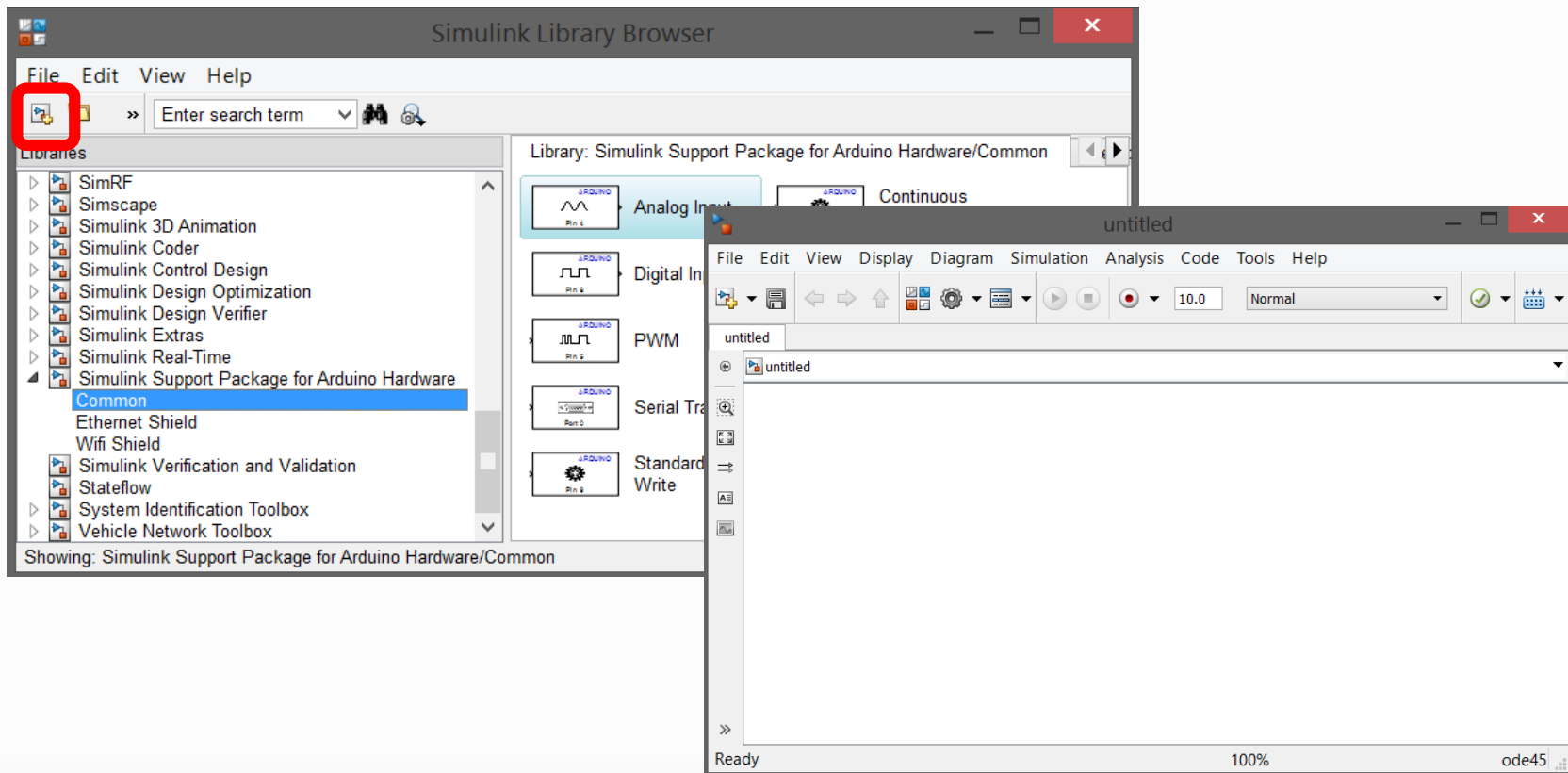
Configuración de MatLab

El soporte está listo para ser utilizado



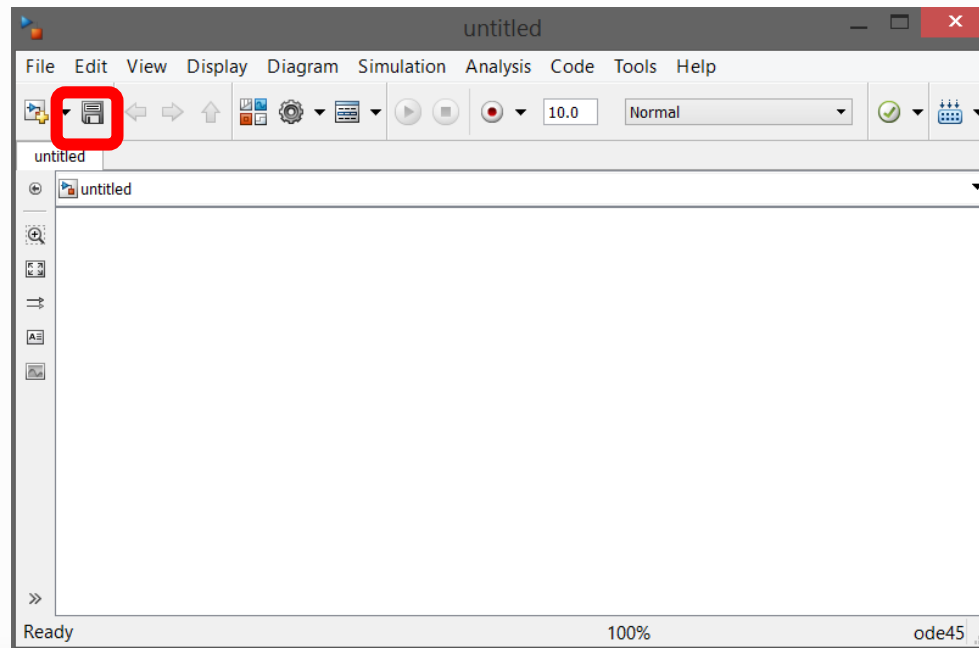
Prueba de funcionamiento

Crear un nuevo modelo



Prueba de funcionamiento

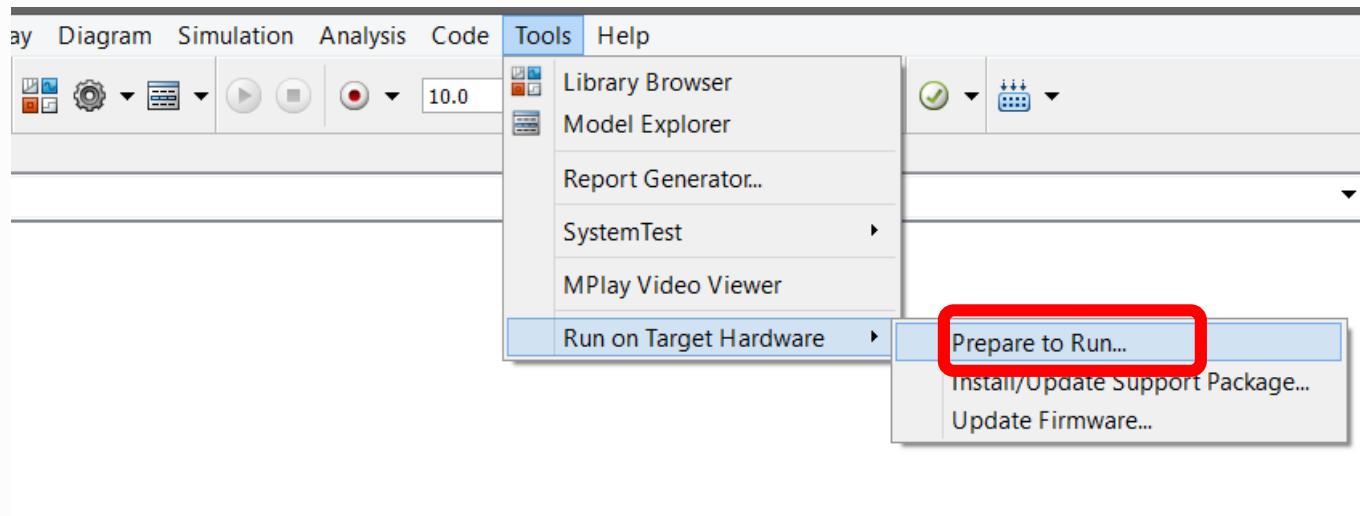
Guardar el archivo. El nombre no debe tener espacios o caracteres especiales, tampoco puede iniciar con un número



Prueba de funcionamiento

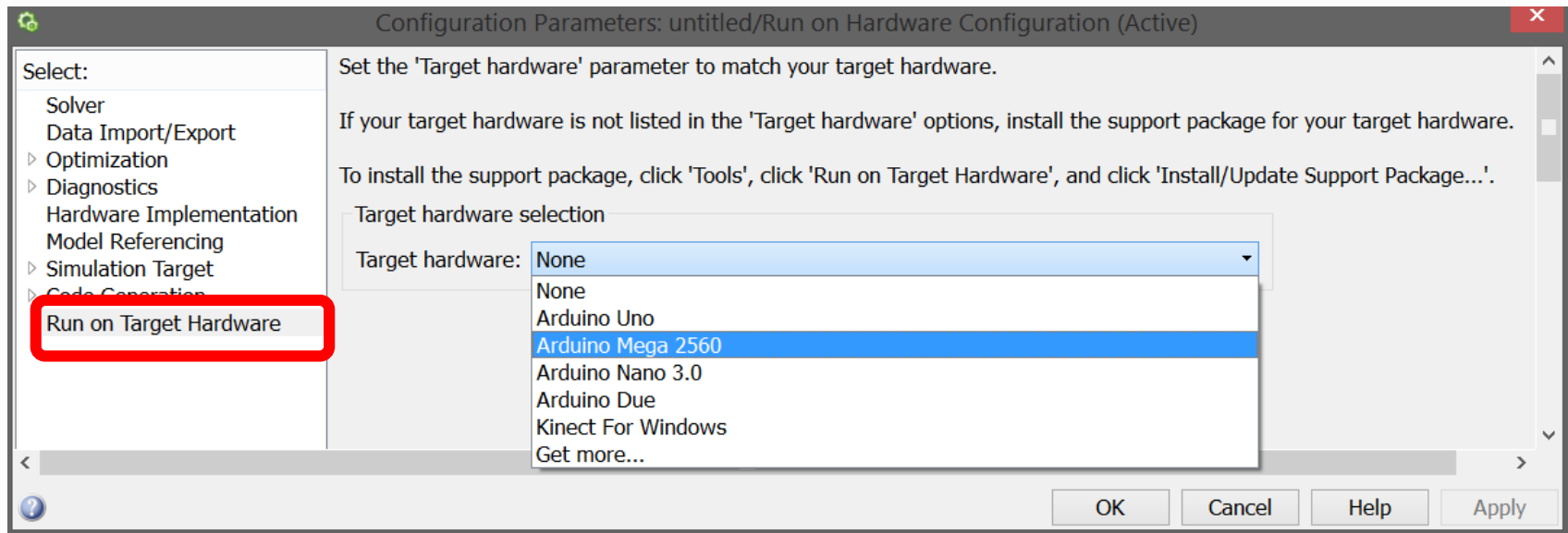
En el menú ir a:

*Tools => Run on Target Hardware =>
Prepare to Run*



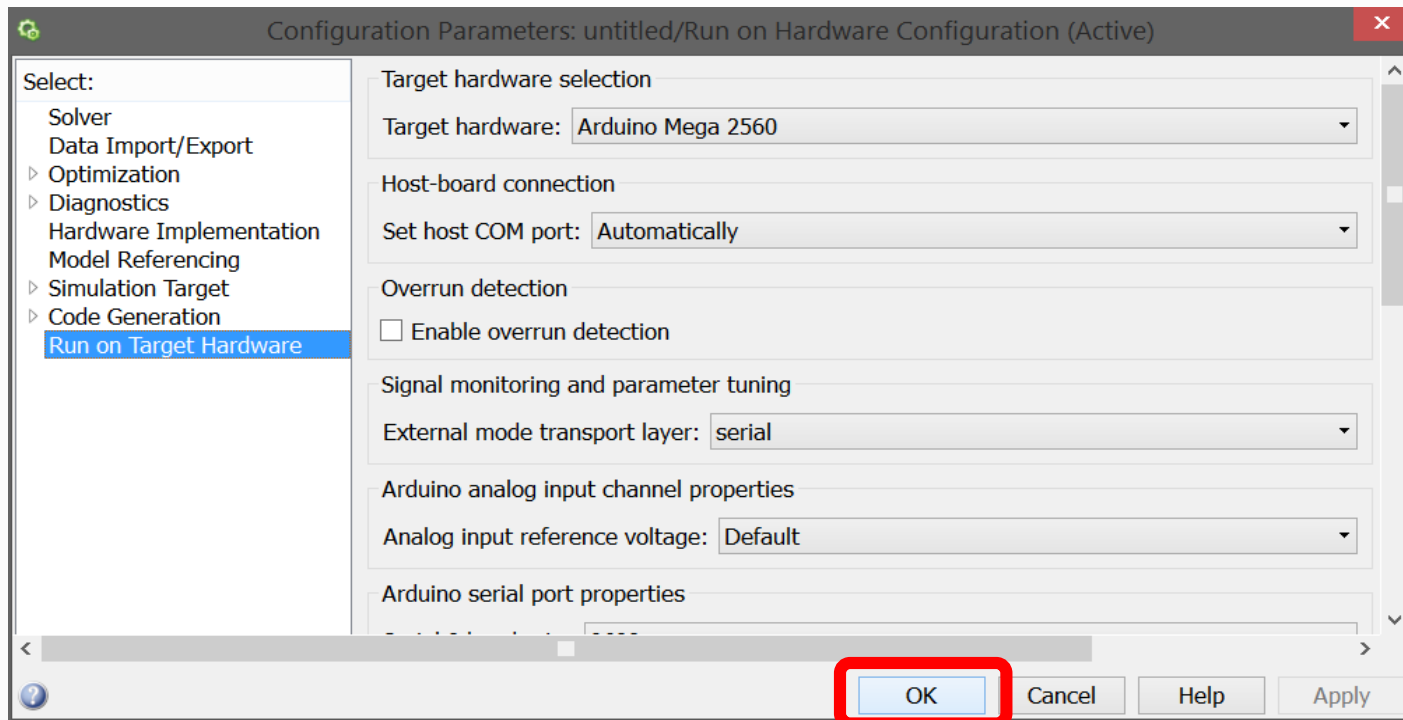
Prueba de funcionamiento

En “Run on Target Hardware” seleccionar el dispositivo en la lista desplegable



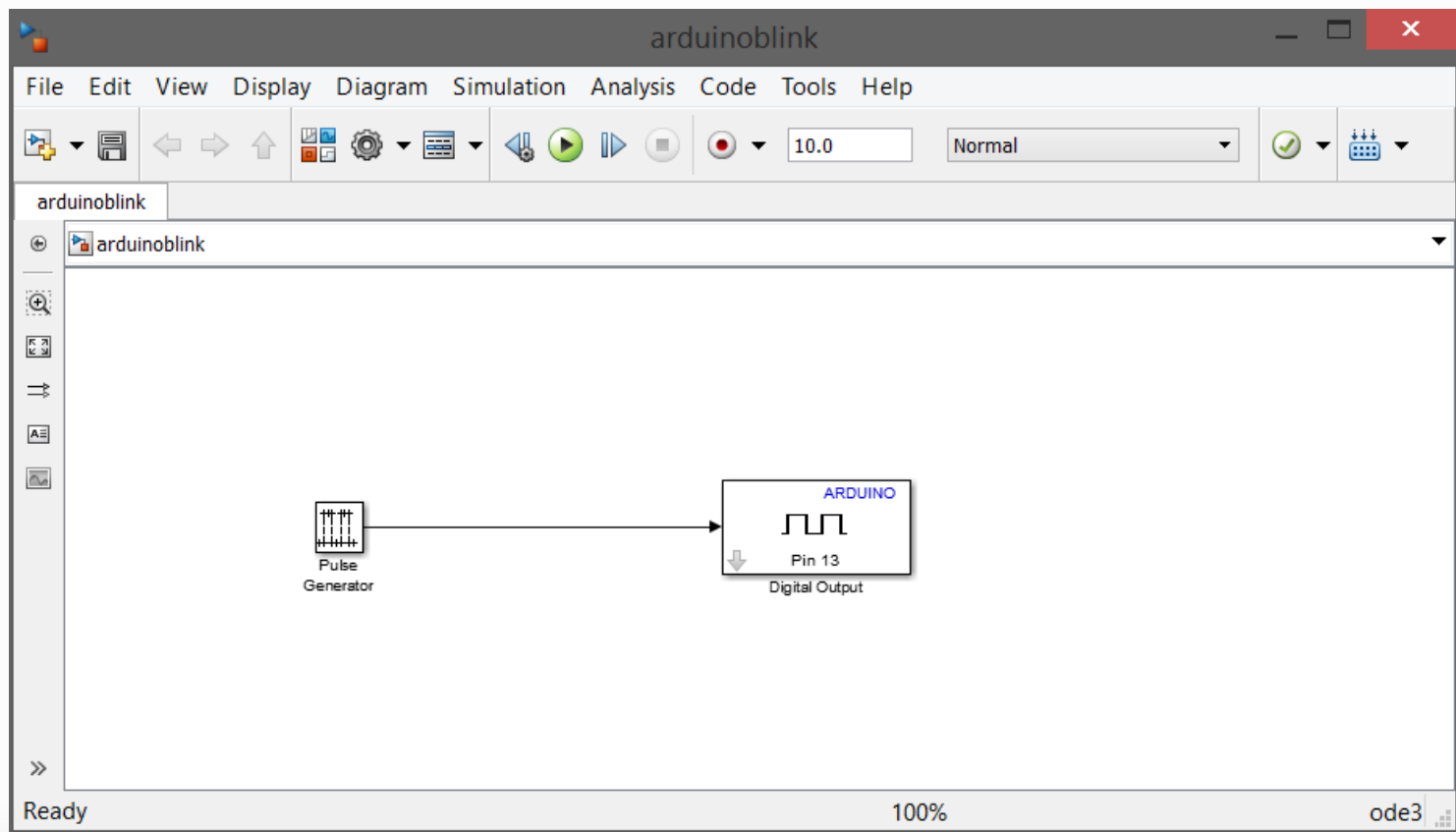
Prueba de funcionamiento

Aceptar dejando las opciones que quedan por defecto. En Matlab R2013 habilitar la opción “external mode” si se van a adquirir datos analógicos.



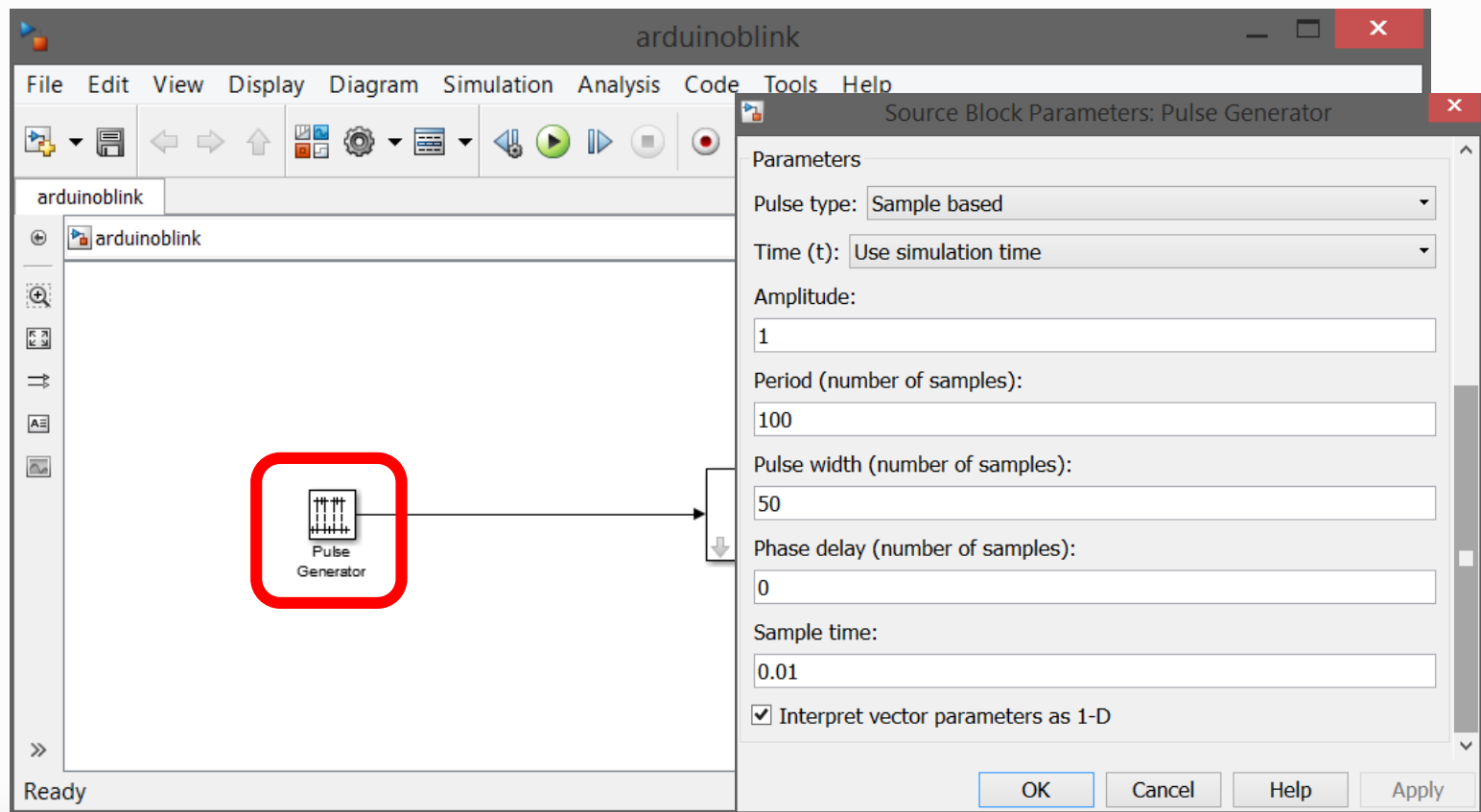
Prueba de funcionamiento

Agregar los bloques mostrados en la figura



Prueba de funcionamiento

Configurar el generador de pulsos



The image shows a screenshot of the 'arduinoblink' software interface. The main window displays a workspace with a 'Pulse Generator' block highlighted by a red square. A line connects the block to a terminal or output area on the right. Overlaid on the right side is the 'Source Block Parameters: Pulse Generator' dialog box, which contains the following settings:

- Parameters
- Pulse type: Sample based
- Time (t): Use simulation time
- Amplitude: 1
- Period (number of samples): 100
- Pulse width (number of samples): 50
- Phase delay (number of samples): 0
- Sample time: 0.01
- Interpret vector parameters as 1-D

Buttons at the bottom of the dialog include OK, Cancel, Help, and Apply.

Prueba de funcionamiento

Seleccionar el pin 13 de la salida digital

The image shows two overlapping windows from a simulation software. The foreground window is titled "Sink Block Parameters: Digital Output" and contains the following text:

Arduino Digital Output (mask) (link)
Set the logical value of a specified digital output pin.

Enter the number of the digital output pin (0-13 for Arduino Uno and Nano, 0-53 for Arduino Mega 2560 and Due). Do not assign the same pin number to multiple blocks within a model.

Parameters

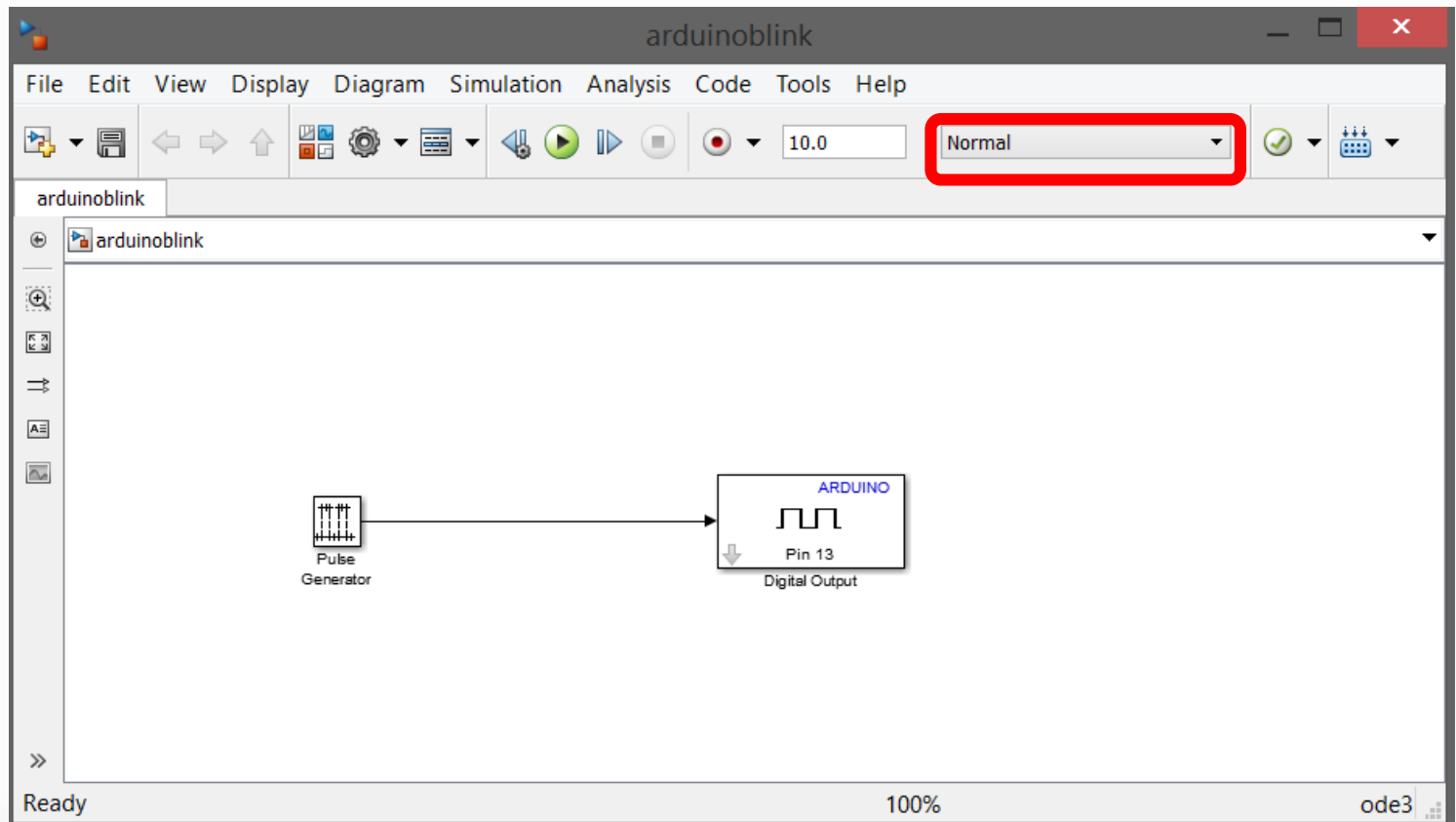
Pin number:

Buttons: OK, Cancel, Help, Apply

The background window is titled "arduinoblink" and shows a simulation interface. It has a menu bar (Analysis, Code, Tools, Help) and a toolbar with a play button, a stop button, a dropdown menu showing "10.0", and a "Normal" dropdown. The main workspace contains a block labeled "ARDUINO" with a square wave icon, "Pin 13", and "Digital Output". A red rounded rectangle highlights this block. A line connects the "Pin 13" block to the "Pin number" field in the dialog box. The status bar at the bottom shows "100%" and "ode3".

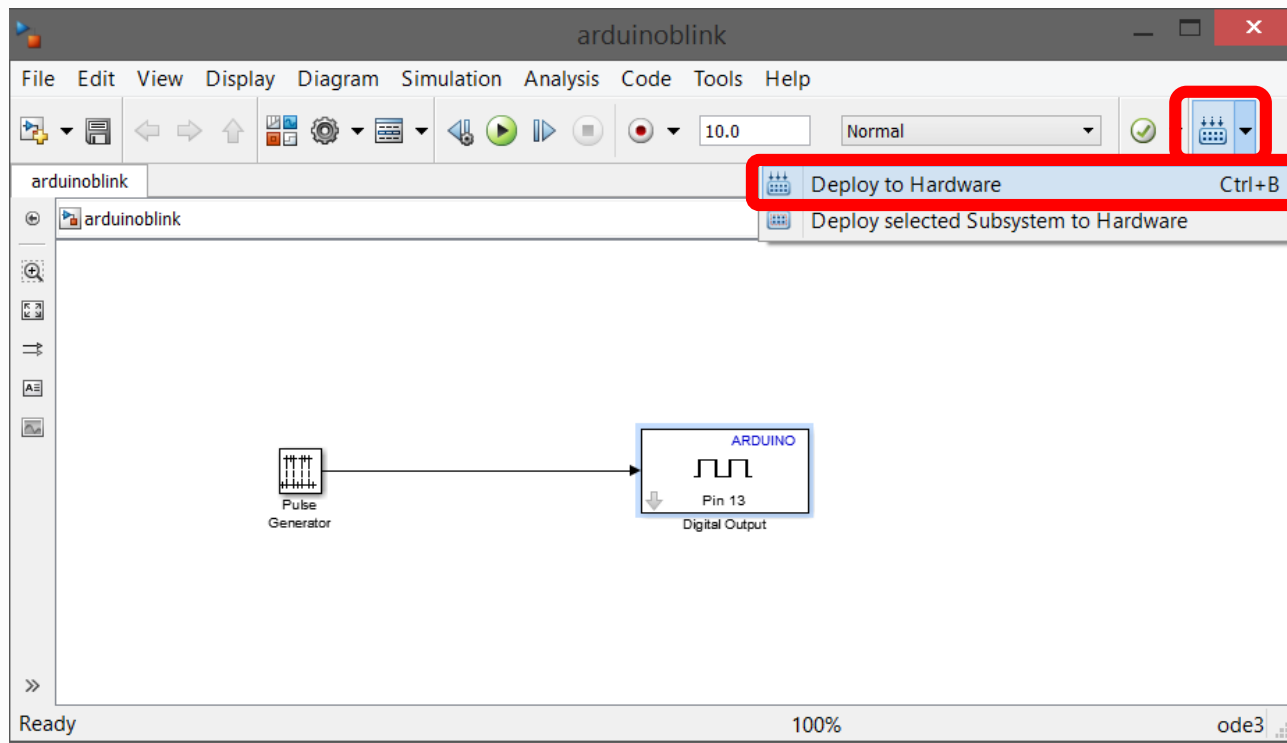
Prueba de funcionamiento

Verificar que esté en modo “Normal”.



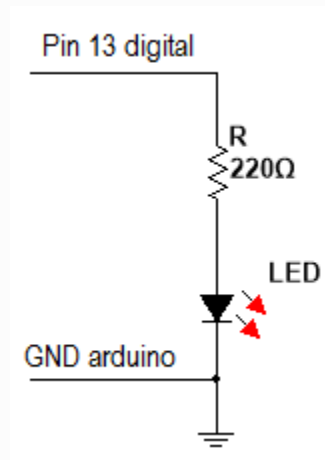
Prueba de funcionamiento

Programar el dispositivo.



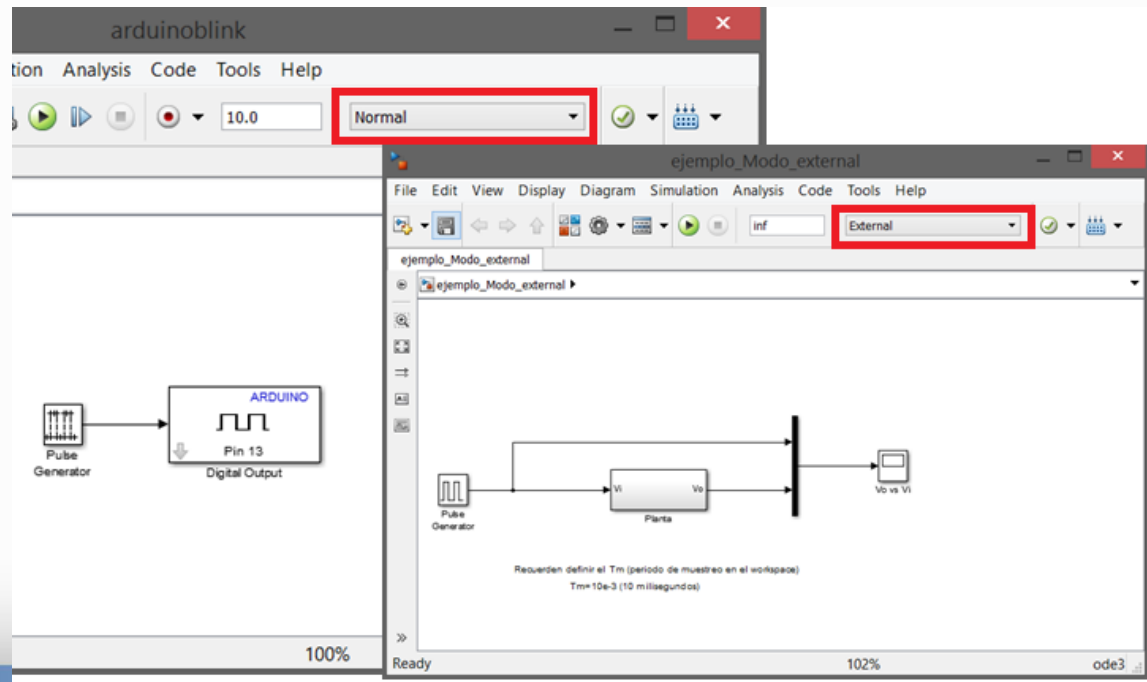
Prueba de funcionamiento

- El circuito que se debe implementar es el siguiente. Se adjunta el archivo llamado Blink.mdl ya listo para realizar la prueba.



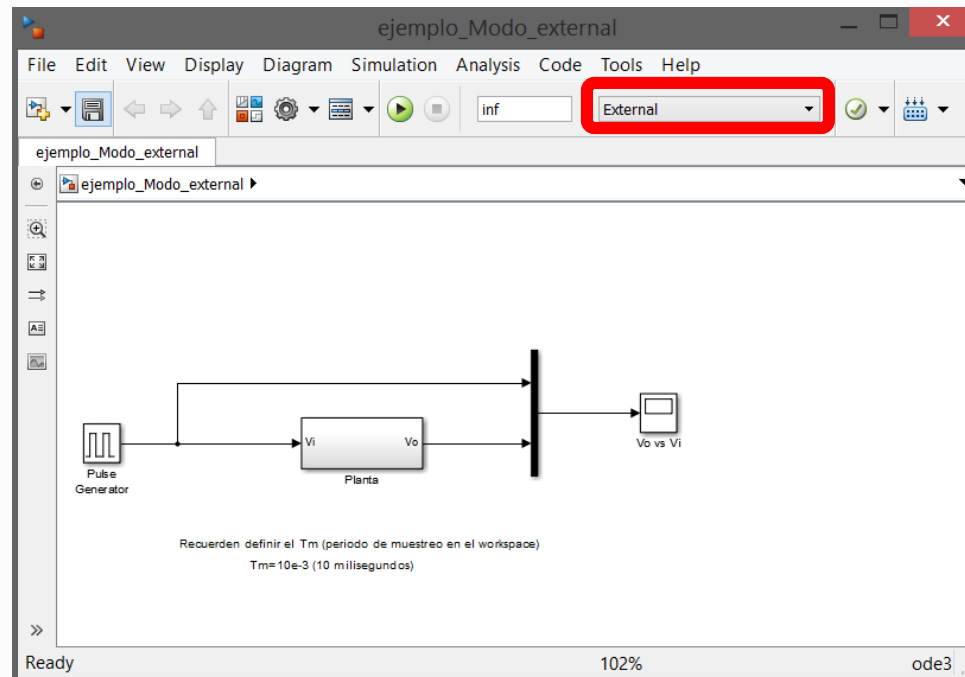
Modos de Ejecución

- **Modo Normal:** este modo se utiliza cuando el programa se deja embebido en el microcontrolador y no necesita la intervención del PC para funcionar, por ejemplo la oscilación de un LED.
- **Modo External:** este modo se utiliza cuando se va a utilizar la tarjeta Arduino como una tarjeta de adquisición de datos, por ejemplo leer el voltaje en un circuito RC y visualizar en tiempo real la carga y descarga del condensador.



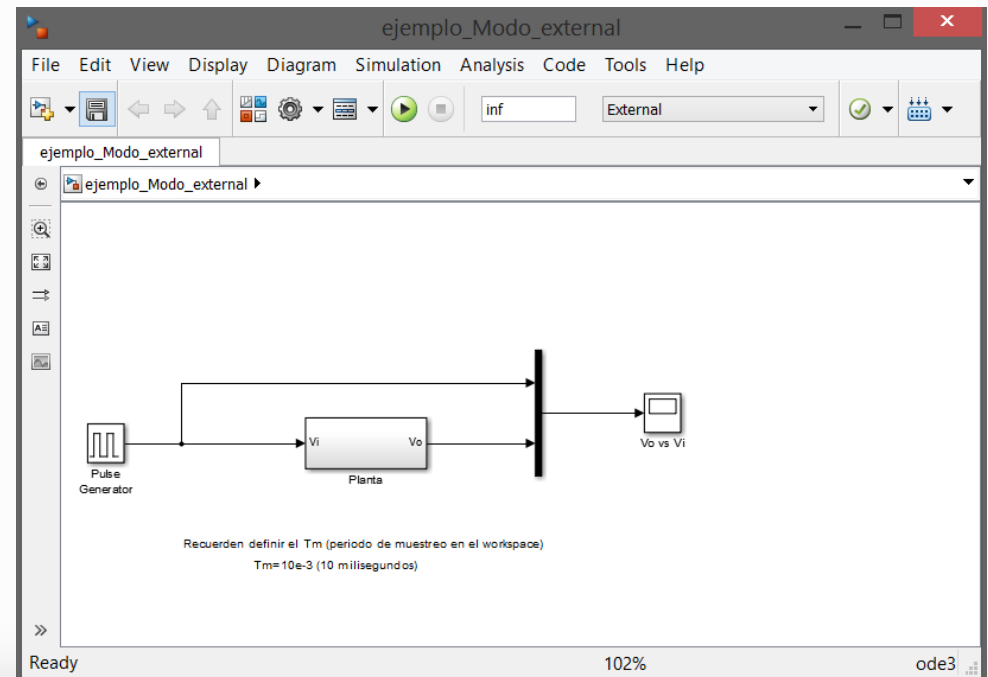
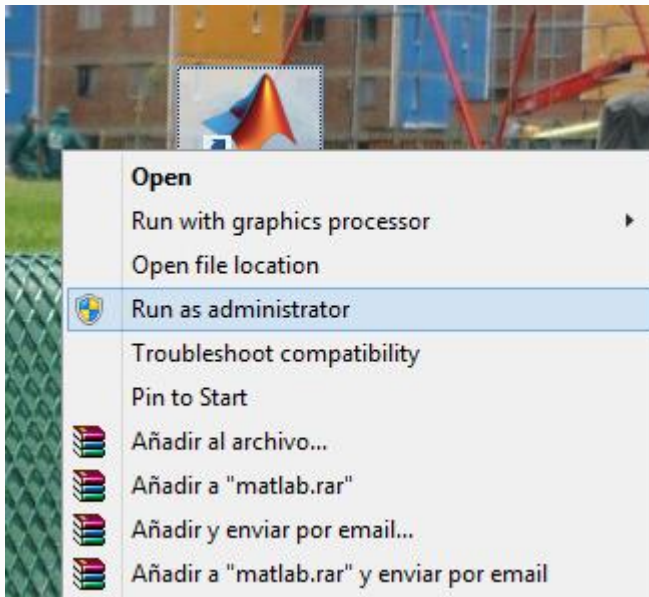
ejemplo_Modo_External

Ejemplo de una conexión en modo external usando como planta un circuito RC



ejemplo_Modo_External

1. Ejecutar MatLab en modo administrador
2. Abrir ejemplo_Modo_external.mdl



ejemplo_Modo_External

3. Definir en el Workspace el periodo de muestreo, T_m debe ser \geq a 0.01 segundos

```
Command Window
>> Tm = 0.01

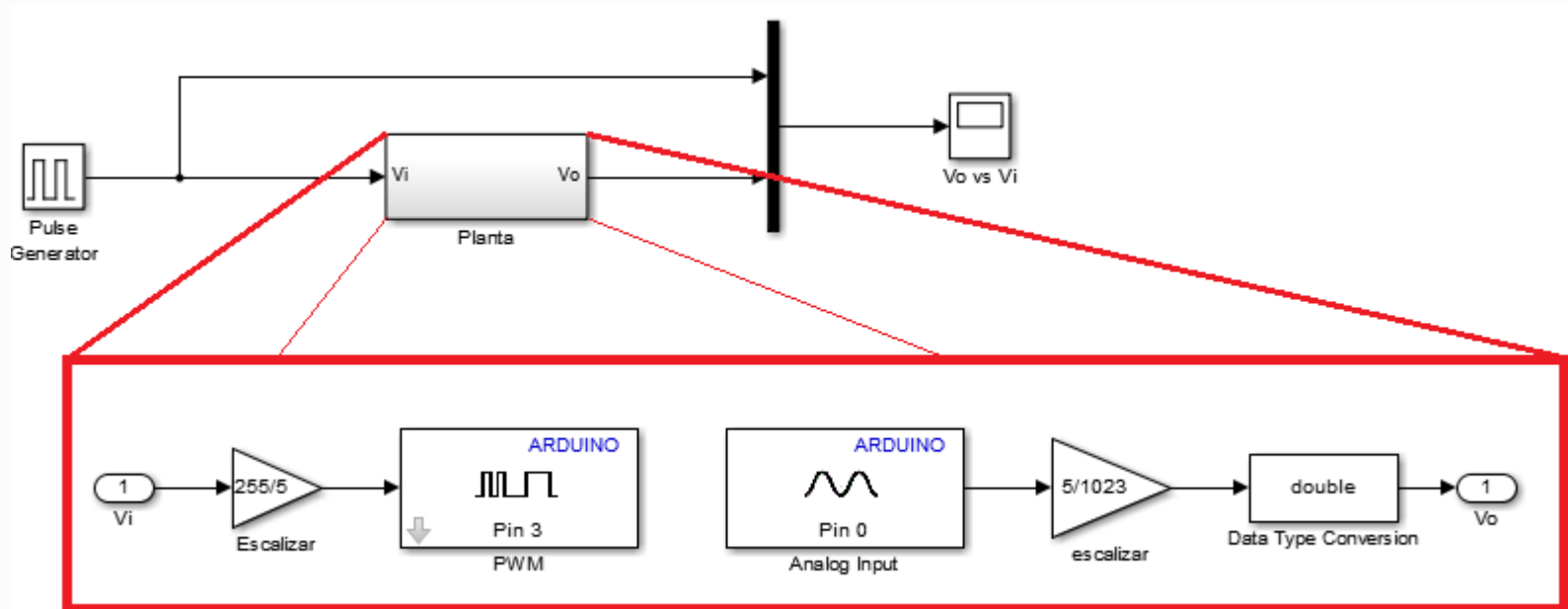
Tm =

    0.0100

fx >> |
```

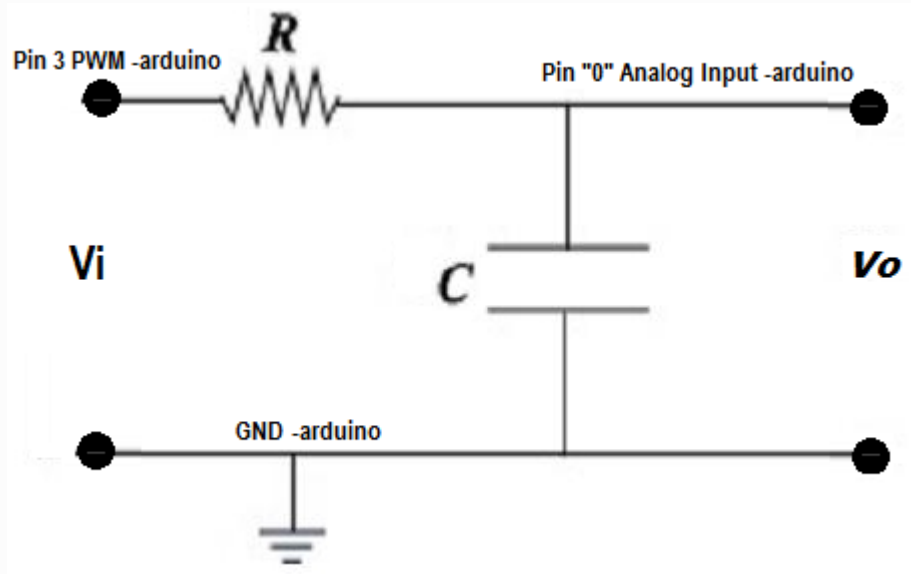
ejemplo_Modo_External

4. Definir los pines de entrada y de salida



ejemplo_Modo_External

5. Conectar la planta

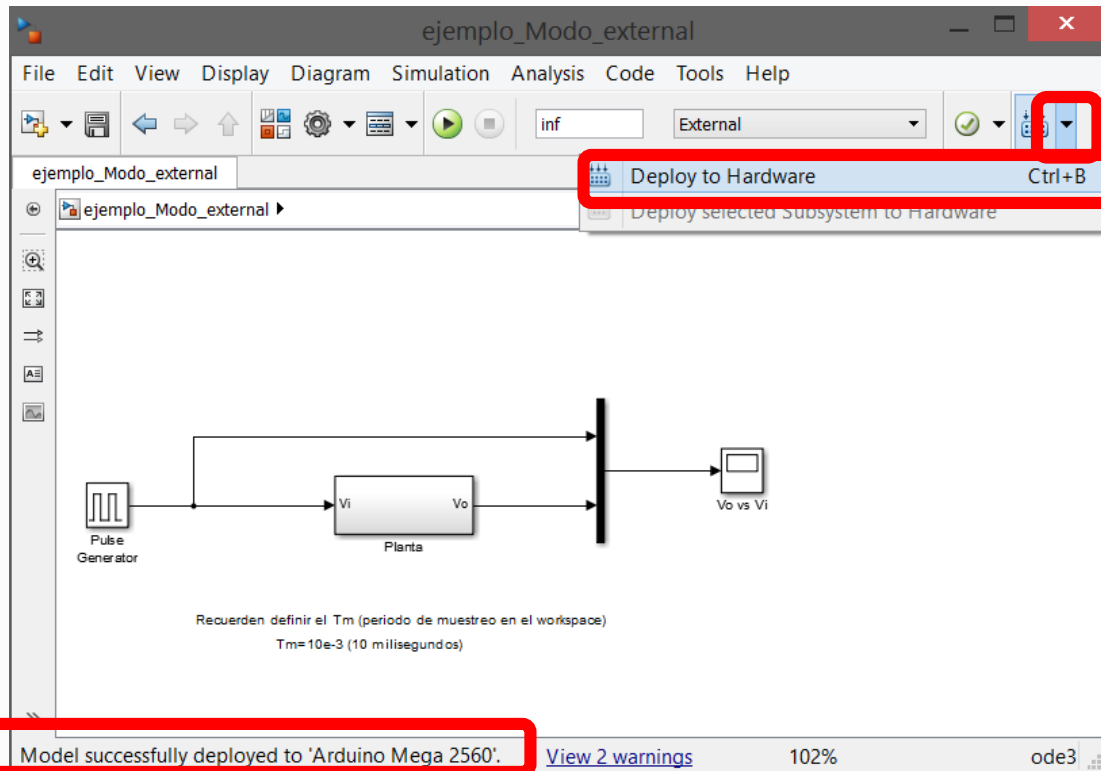


$$R = 1K \text{ ohm}$$

$$C = 220 \text{ uF}$$

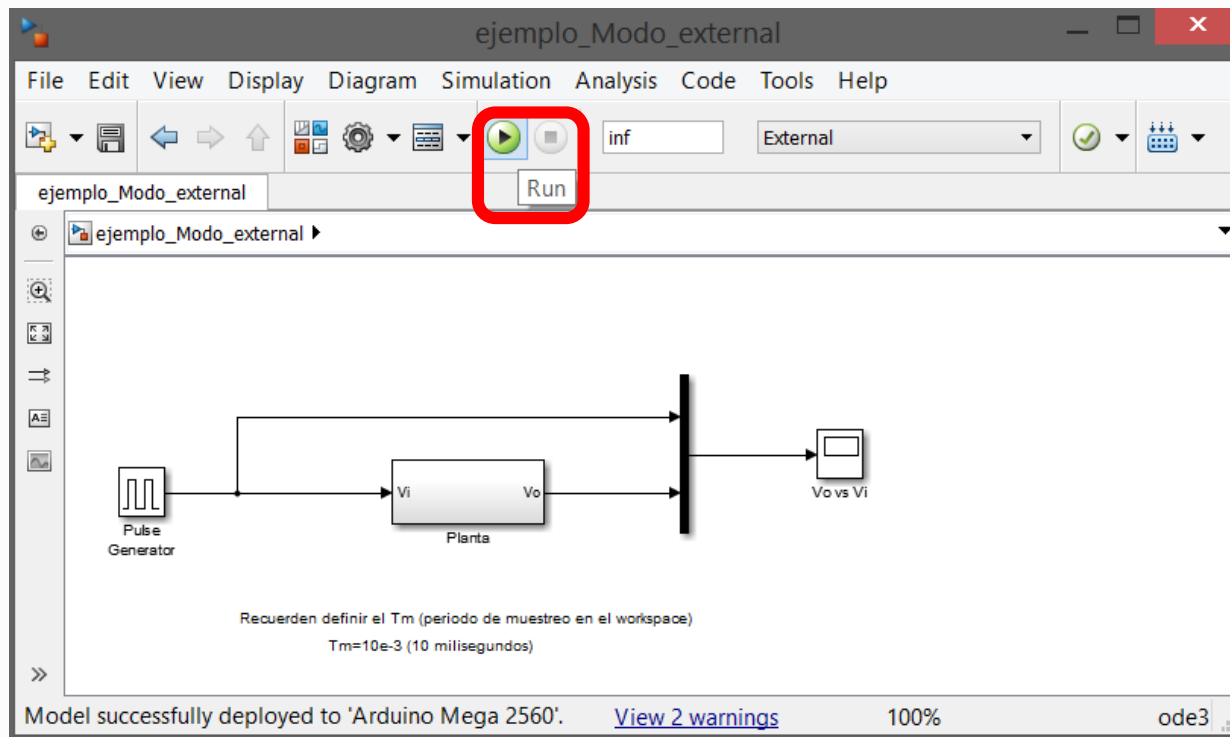
ejemplo_Modo_External

6. Programar el dispositivo



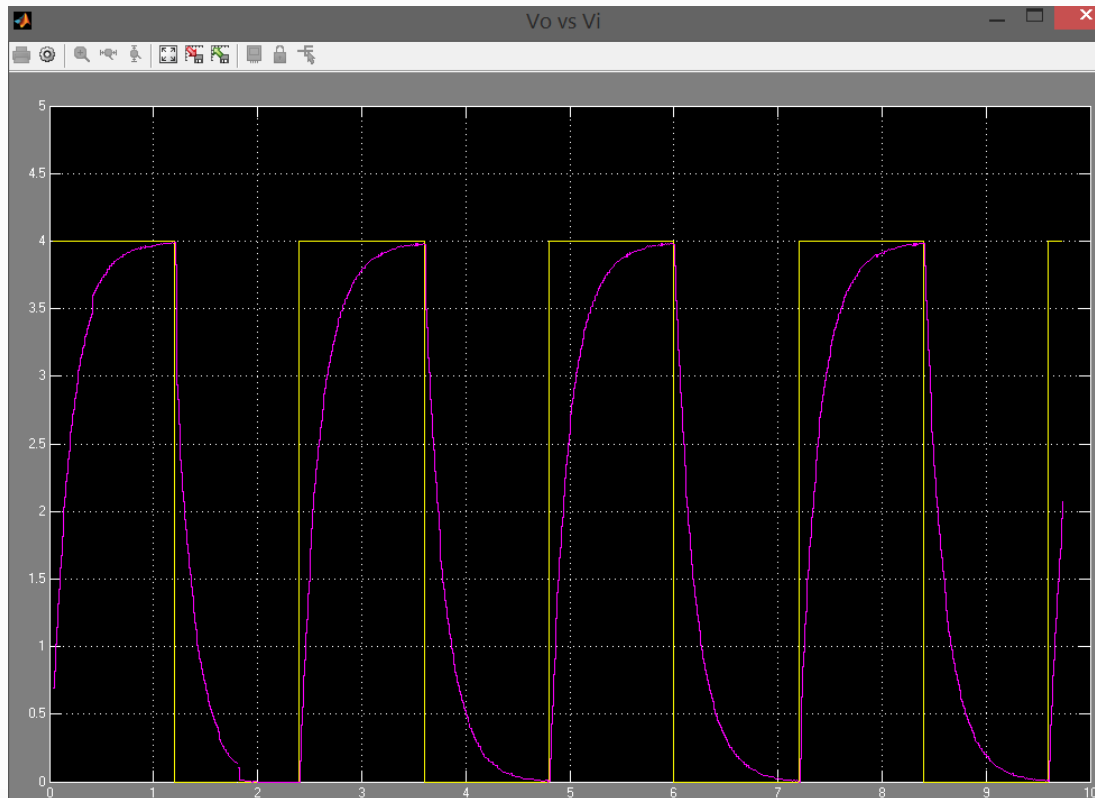
ejemplo_Modo_External

7. Ejecutar el programa



ejemplo_Modo_External

Resultado



ejemplo_Modo_External

8. Salvar datos

