



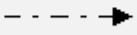

# الدرس 11

## التعامل مع الإشارات

يتم التعبير عن الإشارات في السميولينك بواسطة خطوط تصل بين البلوكات وبعضها البعض وهذه الخطوط لها علاقات رياضية تحدد بها ويتم عمل الإشارات باستخدام بلوكات **source** ويمكن أيضا استخدام **Signal & Scope Manager** لعمل الإشارات بدون استخدام البلوكات

### أشكال خطوط الإشارات

يستخدم السميولينك مجموعة من أشكال الخطوط التي تعبر عن الأنواع المختلفة من الإشارات كما موضح في الجدول الآتي

Signal Type	Line Style	Description
<a href="#">Scalar</a> and <a href="#">Nonscalar</a>		Simulink uses a thin, solid line to represent the diagram's scalar and nonscalar signals.
<a href="#">Nonscalar</a>		When the <b>Wide nonscalar lines</b> option is enabled, Simulink uses a thick, solid line to represent the diagram's nonscalar signals.
<a href="#">Control</a>		Simulink uses a thin, dash-dot line to represent the diagram's control signals.
<a href="#">Bus</a> <a href="http://almohandes.org">almohandes.org</a>		Simulink uses a thick, composite line to represent the diagram's signal buses.

ويمكنك من التحكم في أنواع الخطوط التي تعرضها البرنامج كما تشاء  
ويجب ملاحظة انه عند الرسم في البداية فان الخطوط تكون بالنمط العادي ولكن بعد عمل **update** فان الخطوط تظهر مثل الموجودة في الجدول السابق

## علامات الاشارات Signal Labels

وهي الكلام الذى يظهر على خطوط الاشارات وتظهر اسم الاشارة وتحديد نوعها سواء افتراضية او غير افتراضية  
ويمكنك ايضا التعديل فى هذا الكلام كما تشاء

### أنواع الاشارات

تكون فى الغالب جميع انواع الاشارات فى السميولينك من النوع **double** ولكن يمكنك عمل اشارات من انواع مختلفة كما تشاء

## أبعاد الاشارات Signal Dimensions

توجد فى السميولينك بلوكات يكون خرجها ذو بعد واحد او ذو بعدين  
والاشارة ذات البعد الواحد تتكون من متسلسلة ذات بعد واحد تخرج عند تردد معين لمتسلسلة متجهة فى بعد واحد عند كل زمن محاكاة

**A one-dimensional (1-D) signal consists of a stream of one-dimensional arrays output at a frequency of one array (vector) per simulation time step**

وبالمثل الاشارة ذات البعدين تتكون من متسلسلة فى بعدين تخرج عند تردد متسلسلة مصفوفة فى بعدين عند كل زمن تقطيع

**A two-dimensional (2-D) signal consists of a stream of two-dimensional arrays emitted at a frequency of one 2-D array (matrix) per block sample time.**

ويعبر فى السميولينك غالبا عن الاشارة ذات البعد الواحد بمتجه اما الاشارات ذات البعدين تكون على هيئة مصفوفة  
اما اذا كانت ذات قيمة واحدة مثل الثوابت فتكون فى صورة كمية قياسية  
ويجب ملاحظة ان السميولينك لا يدعم التغيير فى ابعاد الاشارات اثناء المحاكاة حيث يجب ان تكون الاشارة ثابتة اثناء المحاكاة.

## Complex Signals

الاشارات ذات القيم المركبة

ويمكننا عمل اشارات ذات قيم مركبة عن طريق الاتى :

1- تحميل بيانات من الماتلاب

2- استخدم بلوك **constant** وجعله يخرج قيمة **complex**

3- عمل اشارات تحتوى على قيمة حقيقية وقيمة تخيلية معا ثم تحويلها الى قيمة مركبة **complex** باستخدام بلوك

**Real-Imag to Complex**

ويجب مراعاة مراجعه انواع البلوك لمعرفة الانواع التى تدعم الاشارات المركبة قبل استخدامها

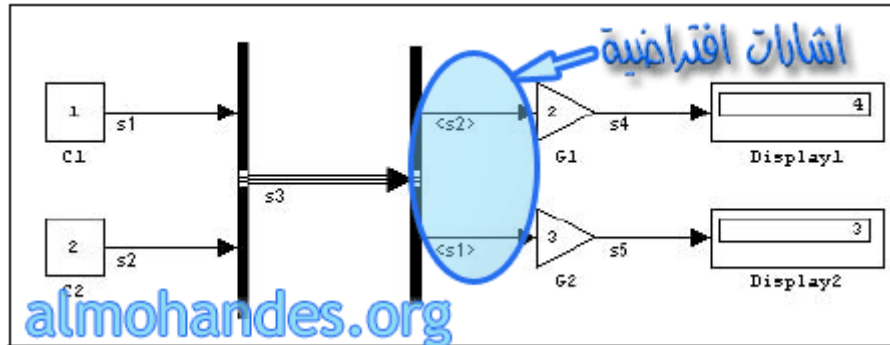
# Virtual Signals

## الاشارات الافتراضية

وهي اشارات تستخدم للتعبير عن اشارات اخرى وغالبا تستخدم مع البلوكات الاتية **Bus Creator**,

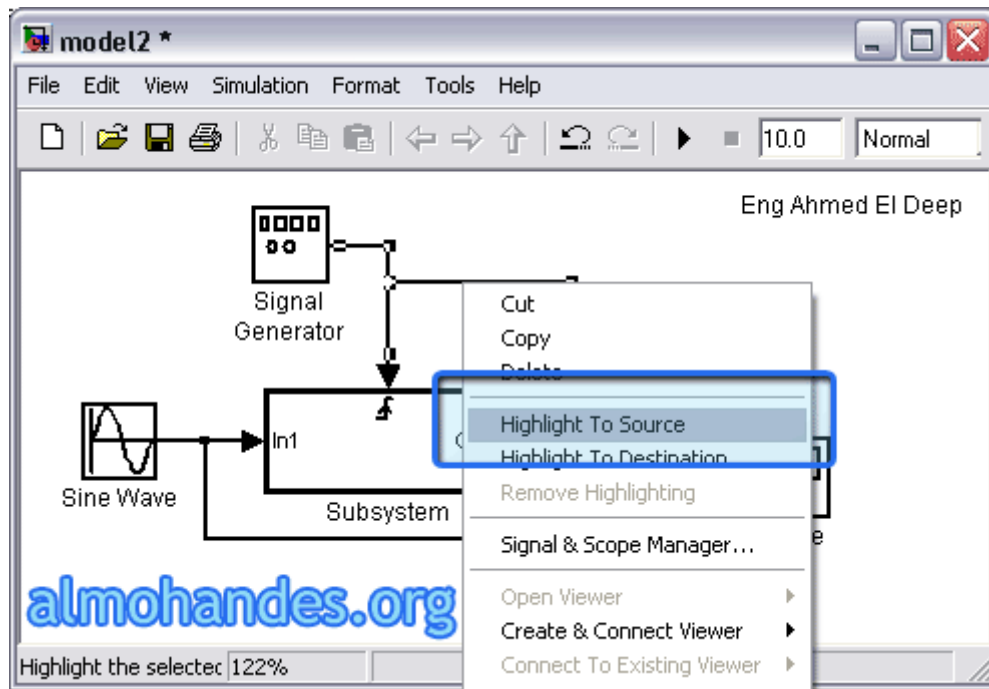
**Inport, and Outport blocks**

وتكون هذه الاشارات في صورة مرسومة وليس لها اي علاقات رياضية ويهملها السميوليك عند البدء في المحاكاة ويقوم بتحديد القيم الاصلية لهذه الاشارات عن طريق استخدام **signal propagation** كما عرفنا سابقا ويمكننا ملاحظة هذا في الشكل الاتي حيث يقوم السميوليك بتحديد الاشارات المناسبة لى **S1** و **S2**

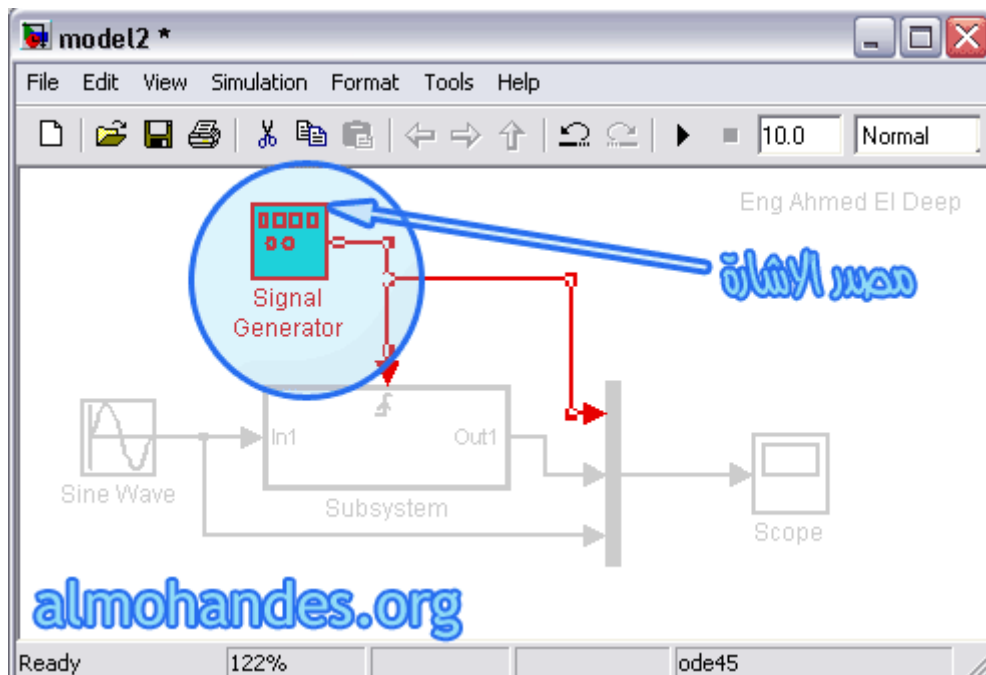


## Displaying Virtual Signal Sources and Destinations

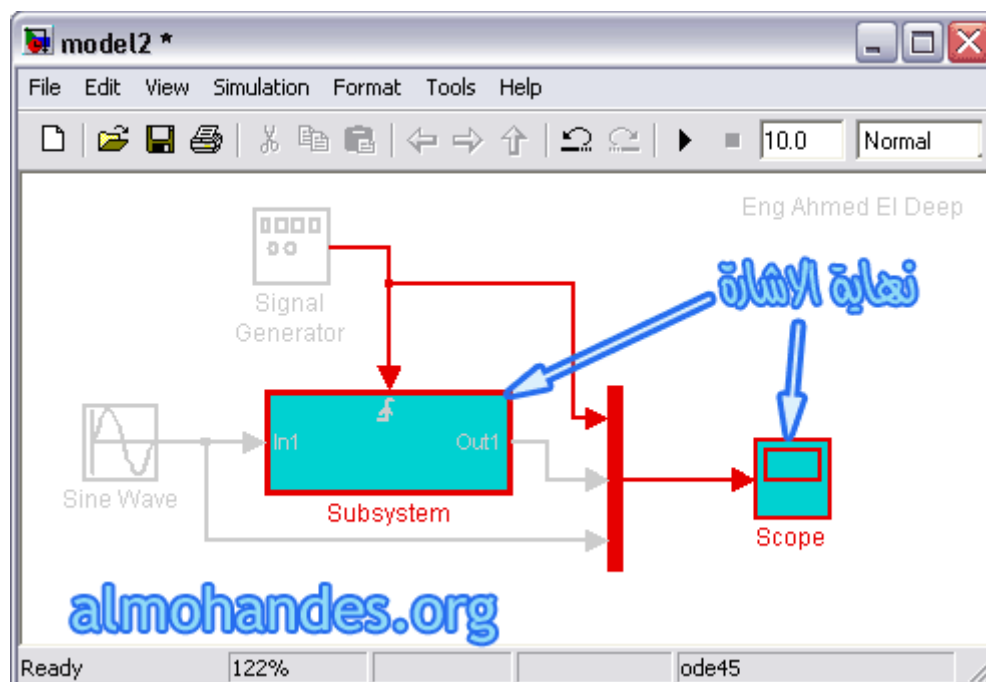
لعرض مصدر ونهاية الاشارة نقوم بتحديد الاشارة والضغط بالماوس يمين واختيار **Highlight to Source**



وسيكون شكل الإشارة كما يلي :



ولتحديد نهاية الإشارة او المكان التي تخزن بها  
بعد كليك يمين اختار **Highlight to Destination**  
ويكون شكل الموديل كما يلي :

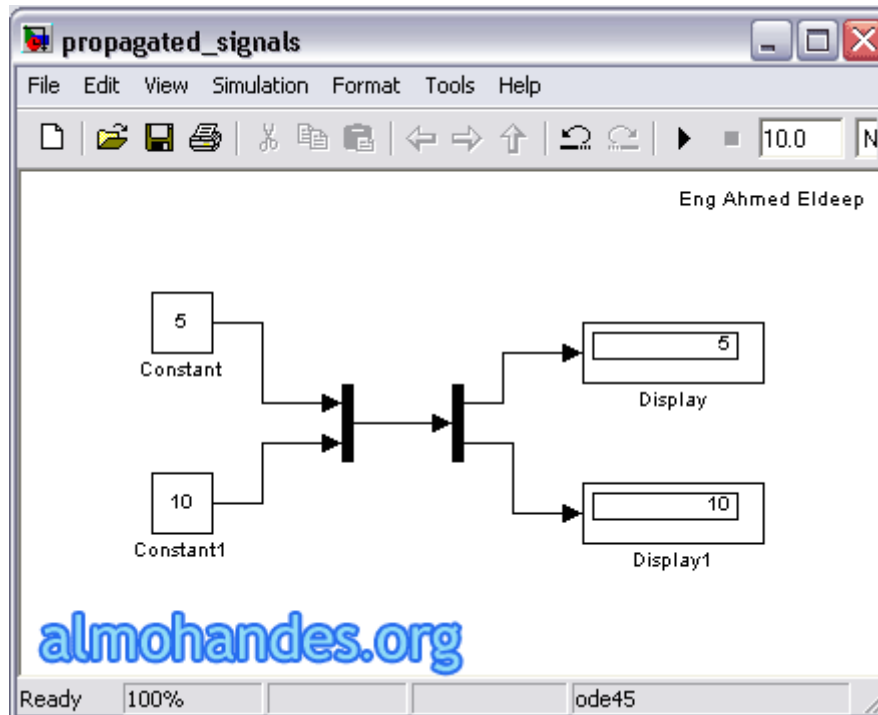


ولإزالة هذه التحديدات اختار بعد عمل كليك يمين  
**Remove Highlighting**

عرض الاشارات الغير افتراضيه المقابلة للاشارات الافتراضيه

## Displaying the Nonvirtual Components of Virtual Signals

سنقوم الان بعمل مثال بسيط وهو يحتوى على بلوكين **constant** و يحتوى على **mux** و **demux** و **display** كما ياتى



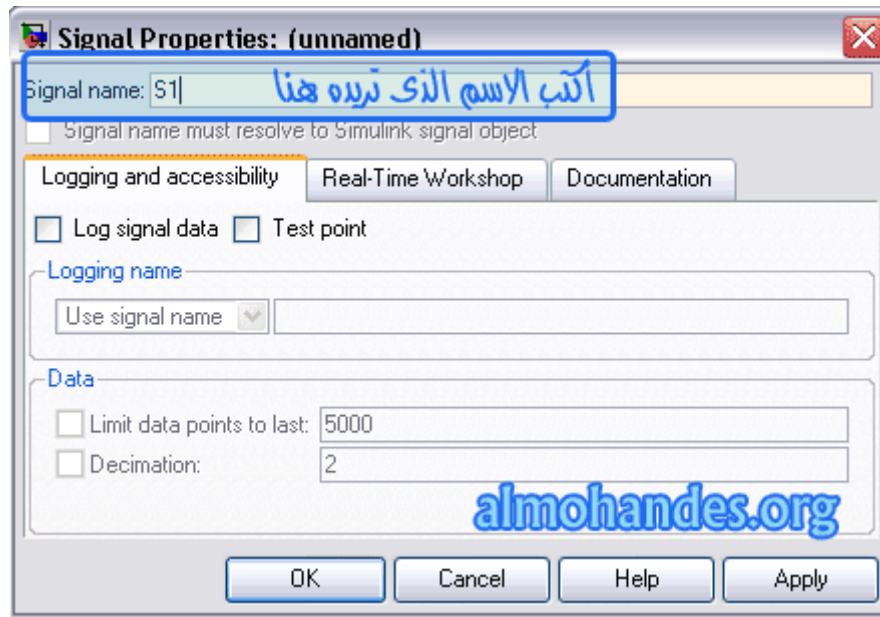
والان سنقوم بتحديد اسماء الاشارات الخارجة من الثوابت وهى اشارات غير افتراضيه عن طريق الدخول الى

### Signal Properties Dialog Box

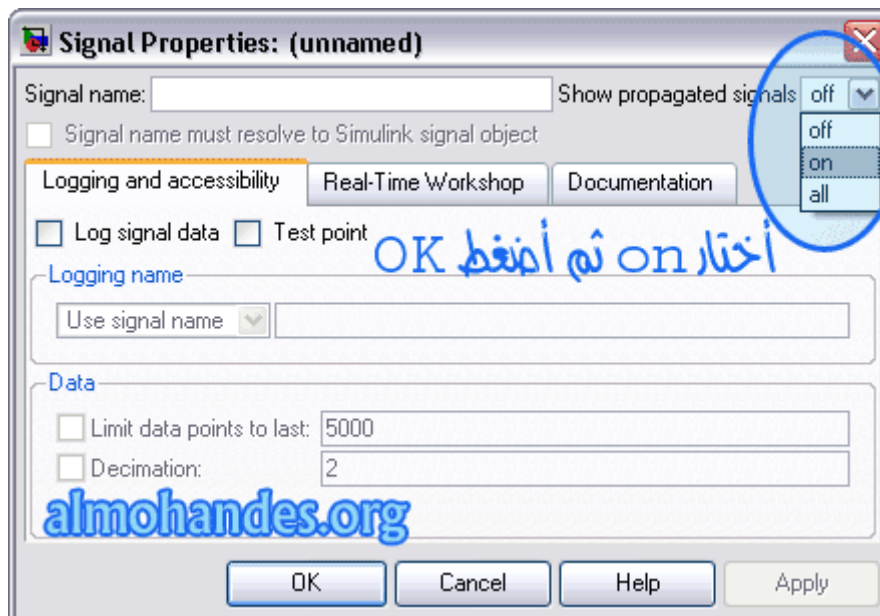
اختار الاشارة التى تريد تسميتها ثم كليك يمين واختار **Signal Properties** كما يلى :



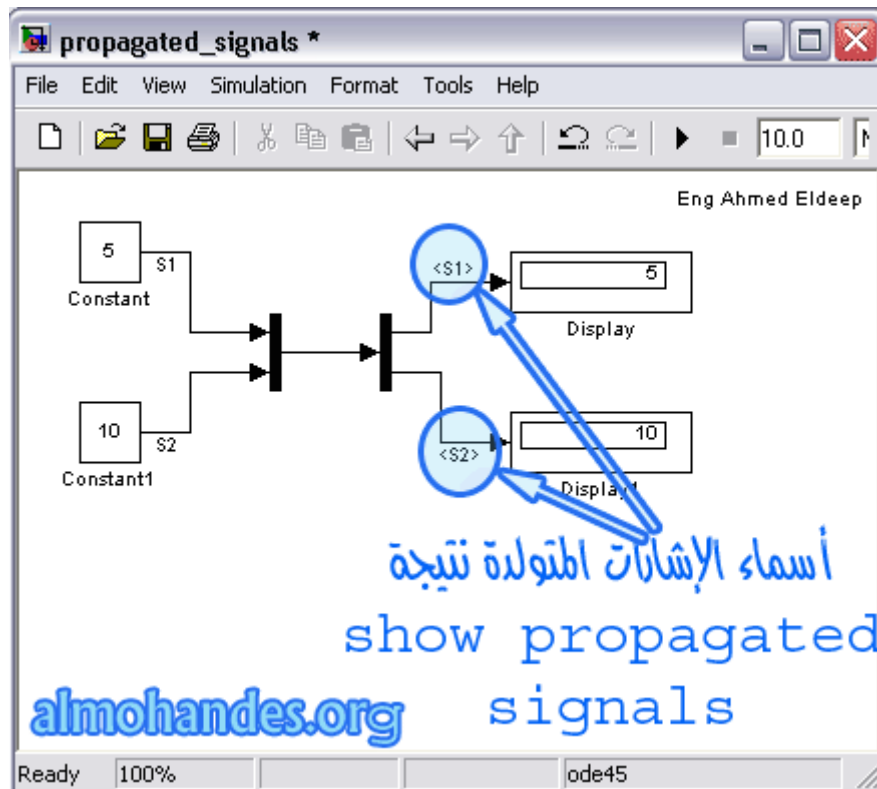
وبالدخول الى **Signal Properties** اكتب الاسم الذى تريده كما بالشكل الاتى



وبالمثل مع الإشارة الاخرى  
والان ننتقل الى الاشارات الافتراضيه  
وعند الدخول الى **Signal Properties** نلاحظ وجود خيار على اليمين يسمى  
**Show propagated signals**  
كما بالشكل الاتى :



وعند اختيار **on** فان هذا سيؤدي الى توليد اسماء الاشارات الغير الافتراضيه المقابلة لها كما فى الشكل الاتى :

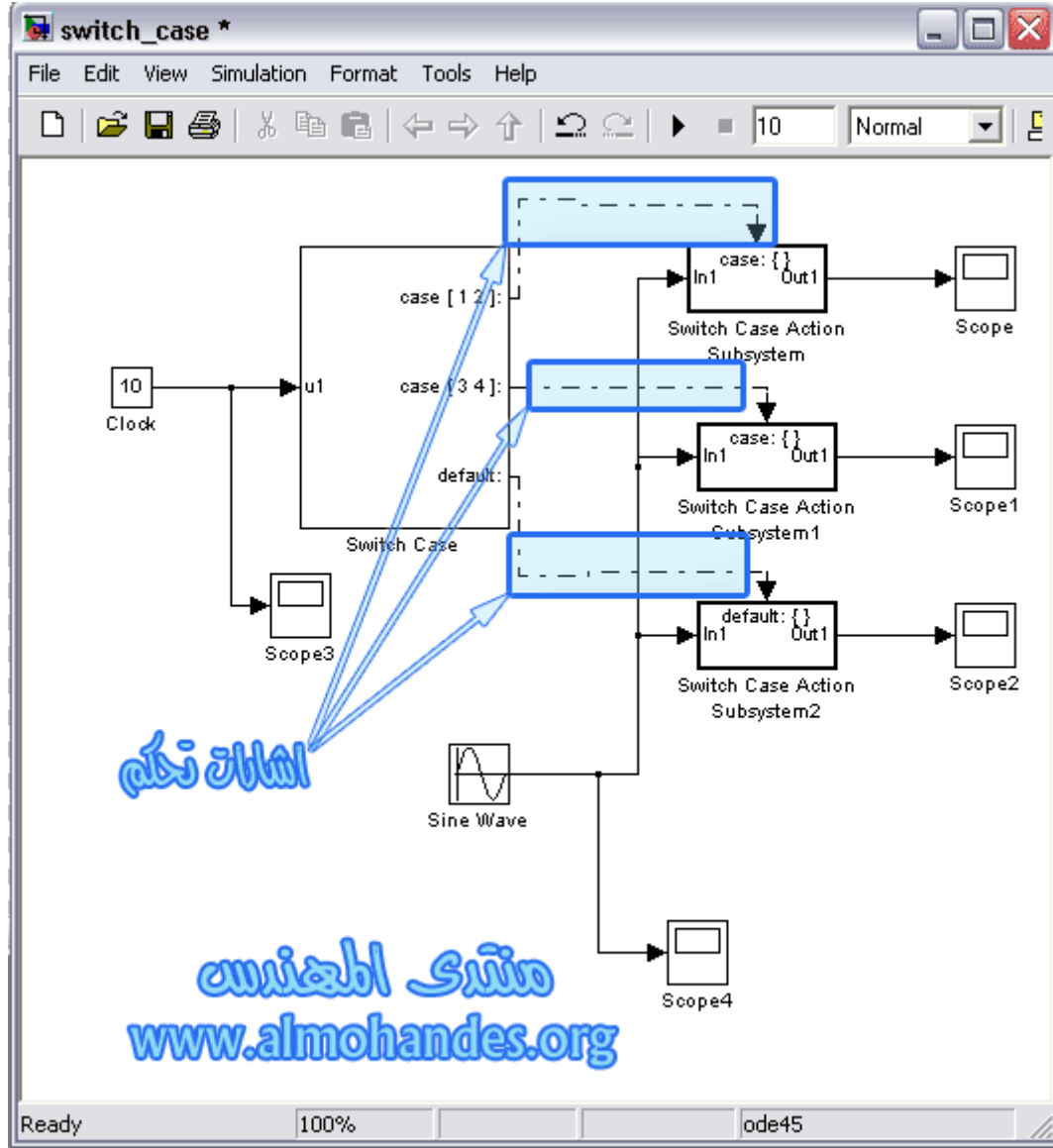


وعند تغيير اسم الاشارة الغير افتراضيه فان السميولينك يقوم بتغيير الاشارة الافتراضيه المقابلة لها مباشرة .

## Control Signals

### إشارات التحكم

هي إشارات تستخدم بواسطة بلوك واحد حتى تبدأ في عملية تنفيذ بلوك آخر ويستخدم السميولينك خط متقطع للتعبير عن إشارات التحكم كما في الشكل الآتي :  
وهو لمثال قمنا بعمله سابقا

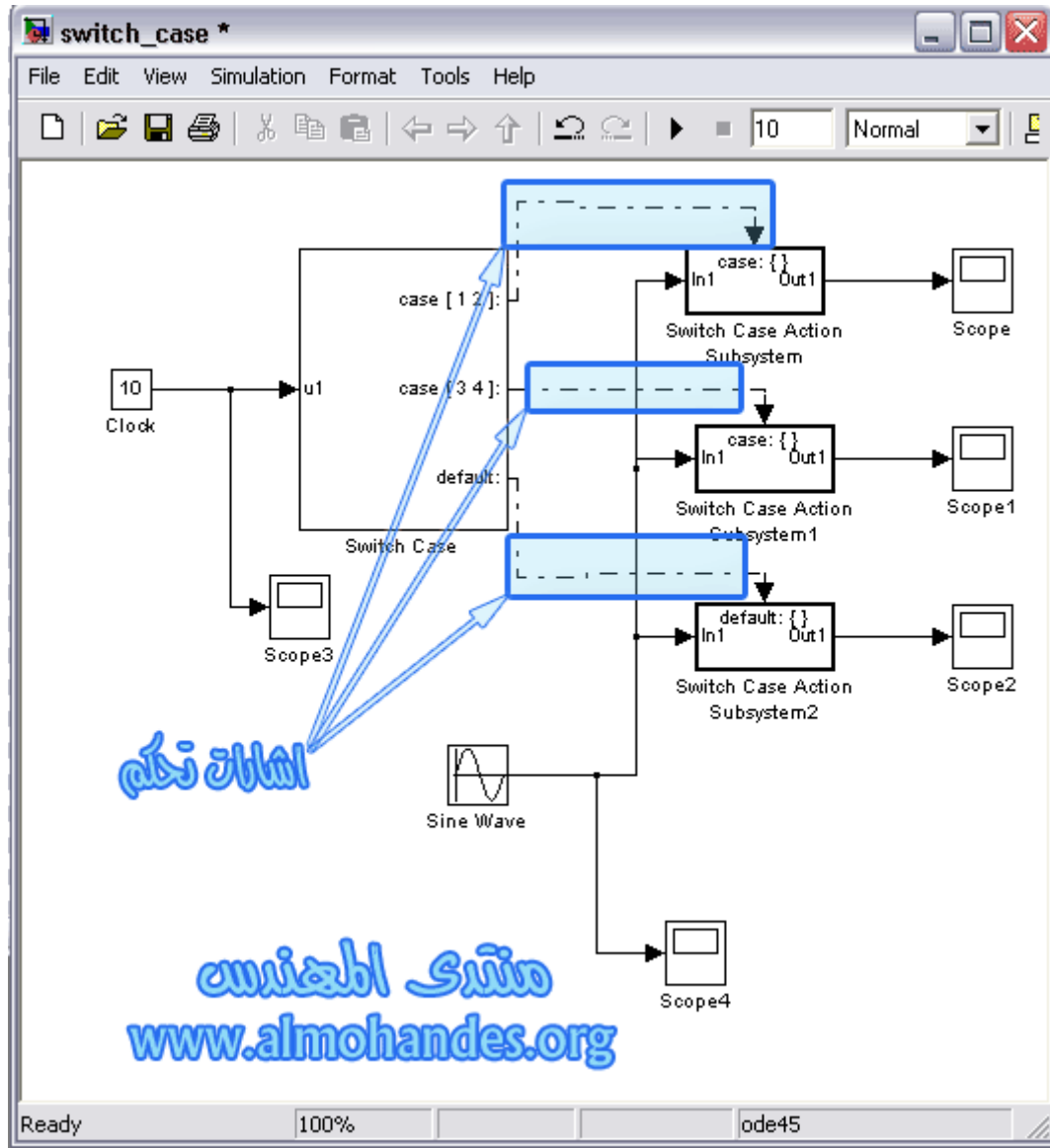




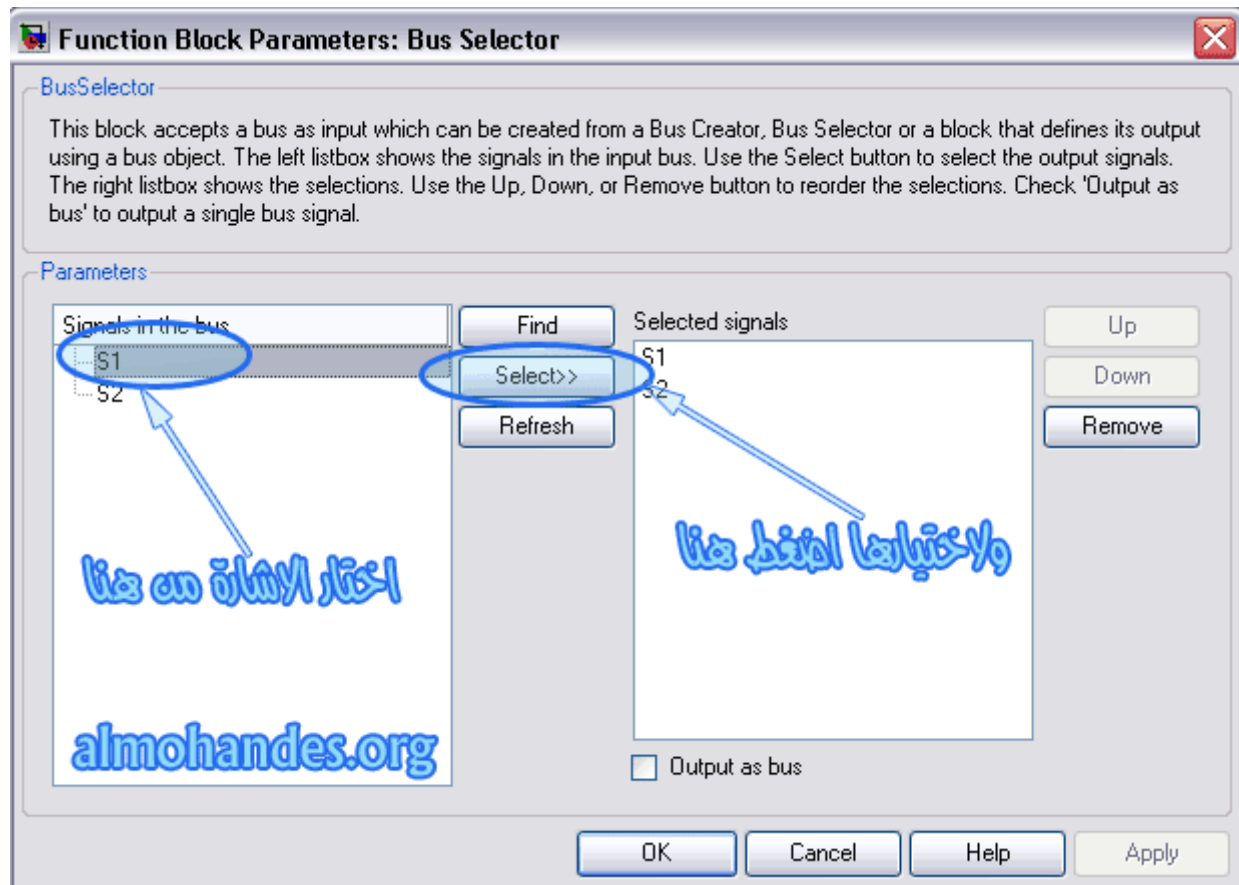
## Signal Buses ( ناقل الاشارات ) ( باص )

وهو عبارة عن اشارة مركبة من مجموعة من الاشارات ويرمز له في الطبيعة بالبلاستيك الذي يغلف مجموعة من الاسلاك معا

والان حاول تطبيق المثال السابق مع استخدام بلوك  
**Bus Creator** و **Bus Selector** كما يلي :



ولا تنسى عند استخدام **Bus Selector** من تحديد الاشارات التي سوف يستخدمها بالدخول الى خصائص البلوك بالضغط عليه مرتين واختيار الاشارات ثم الضغط على **select** كما في الشكل التالي:



## Initializing Signals and Discrete States

من خصائص برنامج السميوليك انه يتيح لنا ان نقوم بتحديد القيم الابتدائية للاشارات والحالات المتقطعة

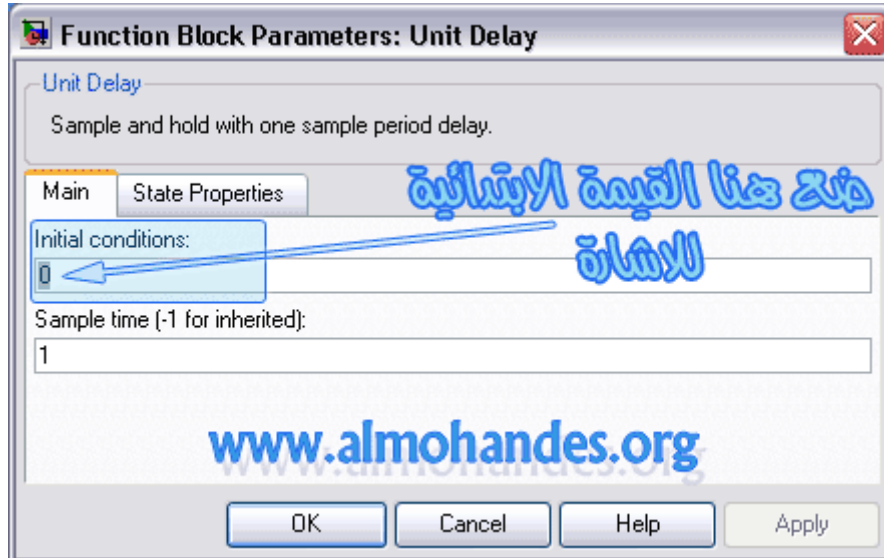
والقيم الابتدائية تعنى قيمة الاشارة عند الزمن صفر

ويمكننا عمل هذا عن طريق ال **signal objects**

ويمكننا عمل هذا ايضا بالدخول الى خصائص البلوك نفسه كما فى البلوكات الاتية

**Outport, Data Store Memory, or Memory**

فمثلا قم بادراج بلوك **Unit Delay** من قائمة **Discrete** فى نموذج فارغ وقم بالضغط عليه لتظهر نافذه خصائص البلوك وستجد بها **initial condition** حيث يمكنك تحديد القيمة الابتدائية للاشارة



اما لتحديد القيم الابتدائية عن طريق ال **Signal Object** فهذا سوف نتعرض له لاحقا عن التعامل مع البيانات

**التعامل مع نقط الاختبار**

## Working with Test Points

نقطة اختبار او **Test point** تعنى فى السميوليك ان هذه الاشارة من الموكد ان يتم ملاحظتها عند استعمال سكوب

**Scope** ويتيح لنا السميوليك ان نجعل اى اشارة نقطة اختبار .

وعند تعريف الاشارة على انها نقطة فان هذا يعنىها من عمليات ال **model optimizations** مثل

**signal storage reuse**

وهى طريقة لتقليل حجم الذاكرة المستخدمة فى المحاكاة عن طريق تحديد اماكن عامة فى الذاكرة لاستخدامها مع جميع البلوكات والاشارات بدلا من جعل مكان مخصص لكل بلوك او اشارة وسوف نتعرض لها لاحقا ان شاء الله.

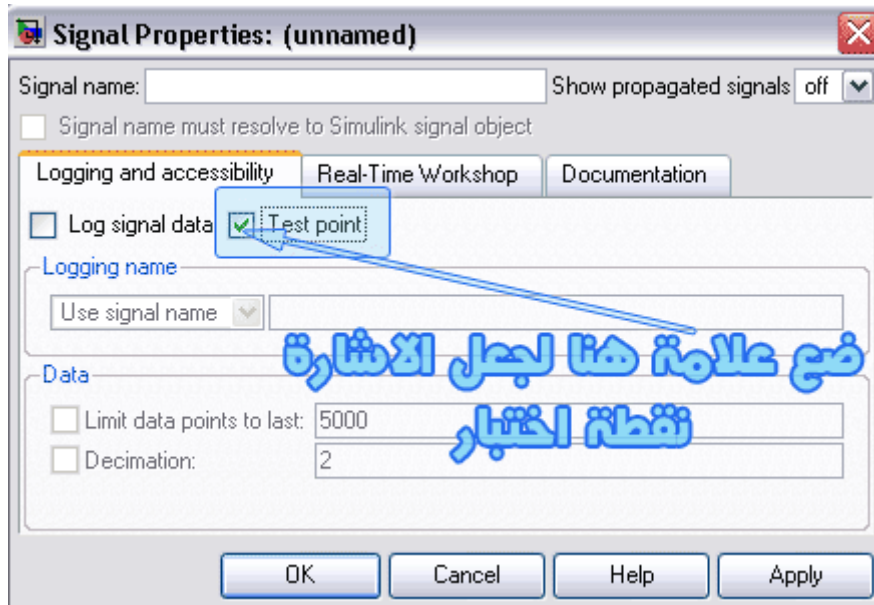
**block reduction**

وهى عملية يقوم بها السميوليك لتجاهل الاشارات الداخلة او الخارجة الى كل بلوك اثناء عملية المحاكاة.

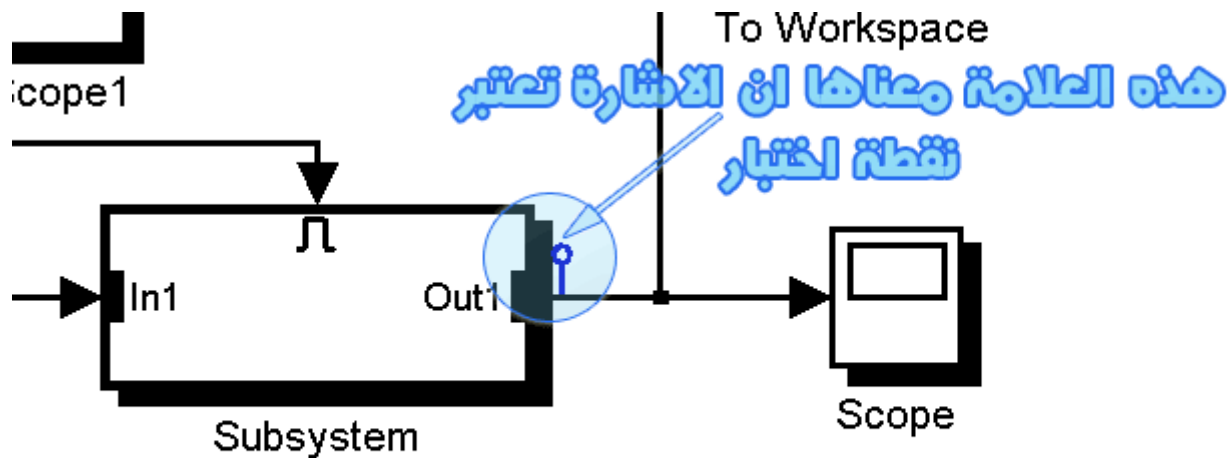
## Designating a Signal as a Test Point

تصميم إشارة كنقطة اختبار

قم بالدخول الى خصائص الإشارة المطلوبة **Signal Properties** وقم بوضع علامة على **Test point**



وايضا عند وضع علامة على **signal logging** فان السميولينك يقوم باعتبار الإشارة نقطة اختبار اتوماتيكيا والان ستلاحظ العلامة الموجودة على الإشارة بعد تحويلها الى نقطة اختبار كما يلي:



[www.Almohandes.org](http://www.Almohandes.org)

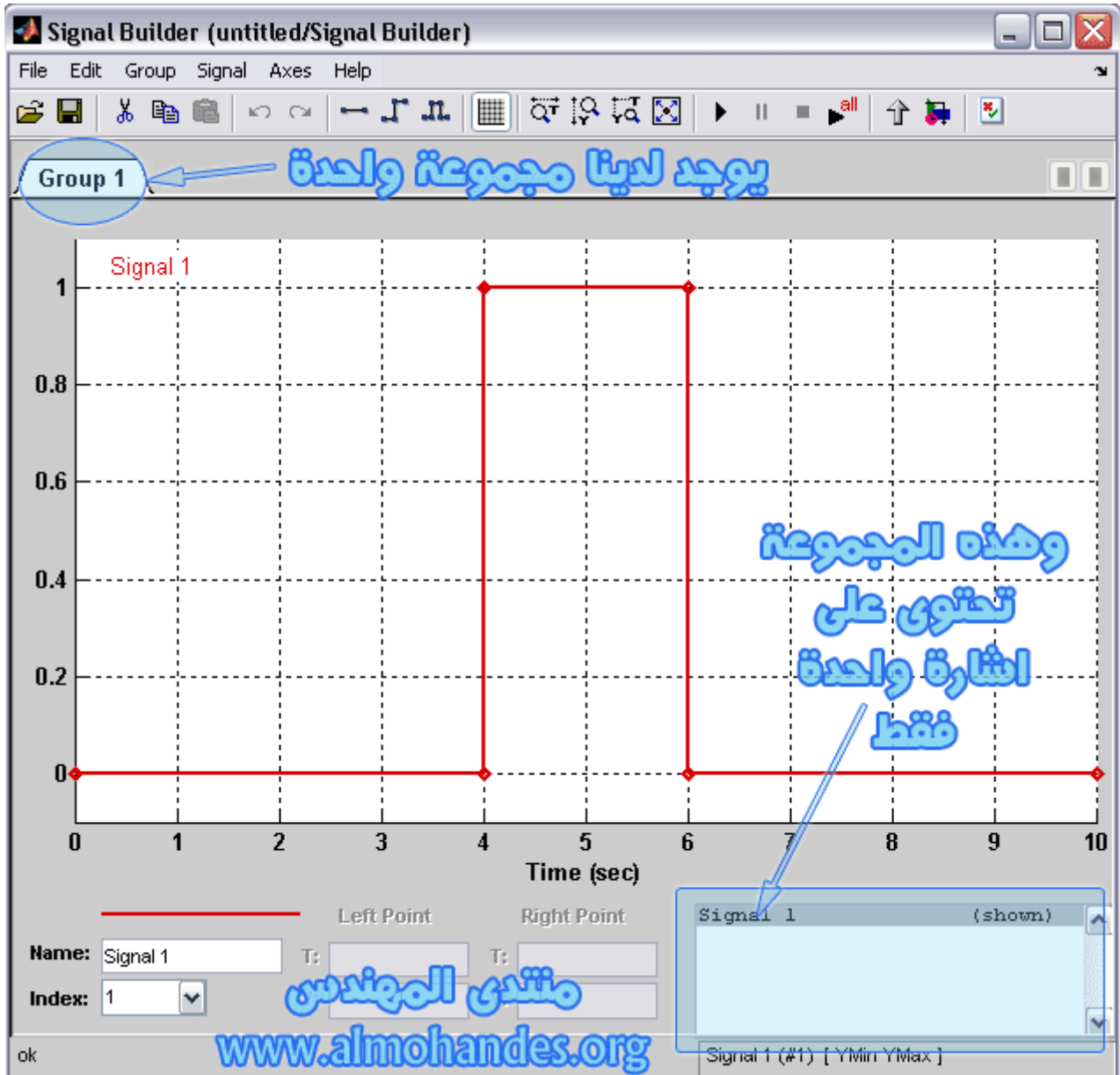
ويمكننا ايضا استخدام **Signal Objects** وسوف نتعرض له لاحقا.

## Working with Signal Groups

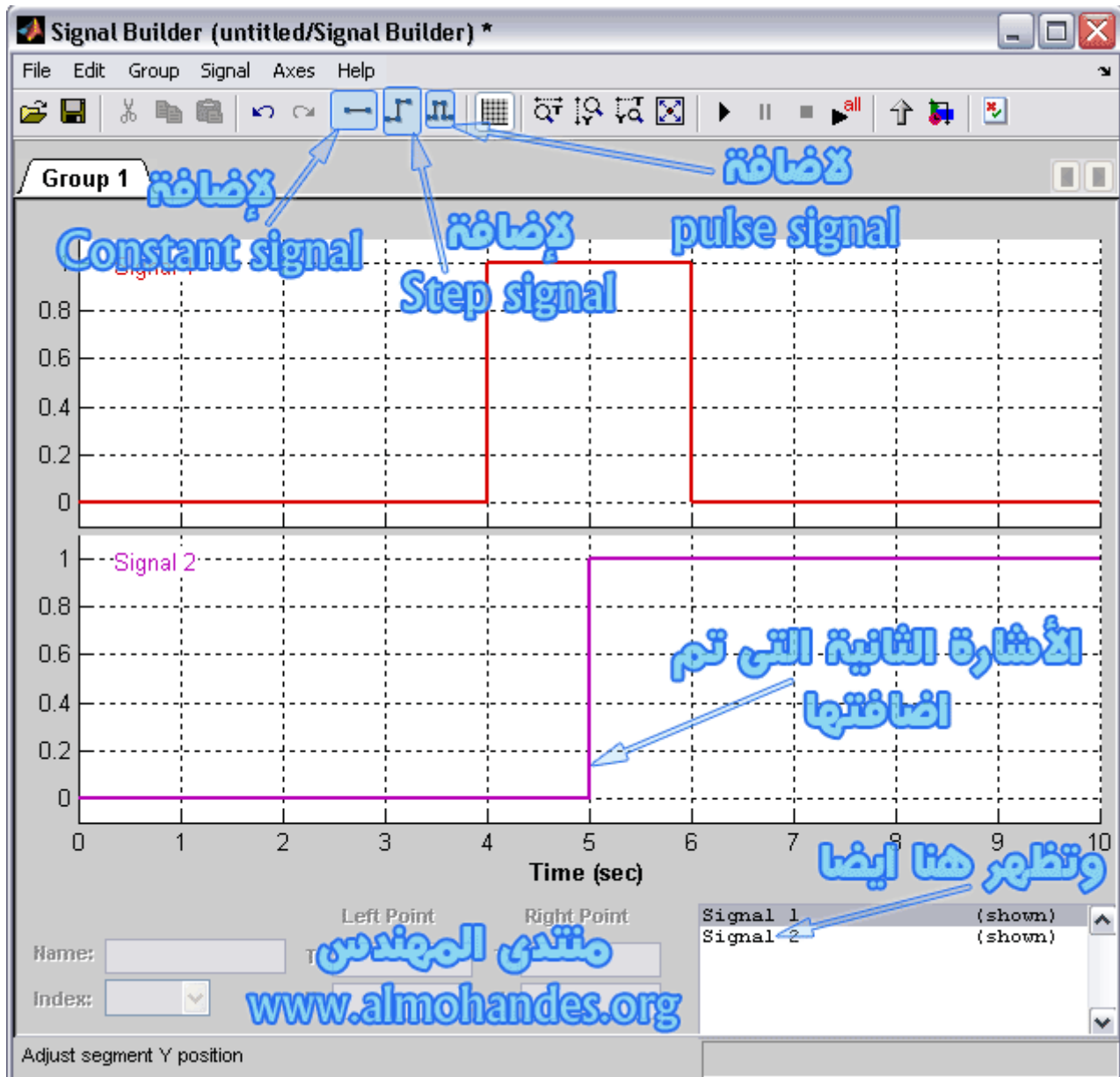
### التعامل مع مجموعات الاشارات

يتيح لنا بلوك **Signal Builder** الموجود فى قائمة **Sources** عمل مجموعات من الاشارات يمكنها ان تتبادل بين بعضها ومن فائدة هذا انه يسهل عملية اختبار الموديل وخصوصا عند استخدام بلوك **Assertion** الذى يستخدم فى اختبار للاشارات اذا كانت تساوى صفر ام لا.

والان قم بعمل موديل فارغ وضع به بلوك **Signal Builder** وعند الضغط عليه فانه يظهر لنا مجموعة واحدة وتحتوى على اشارة واحدة وهذه الاشارة عبارة عن موجة مربعة **square wave** كما بالشكل الاتى



والان قم باضافة اشارة اخرى الى المجموعة الموجودة عن طريق الدخول الى قائمة **Signal** واختيار **new** او عن طريق استخدام الايقونات الموجودة في ال **Toolbar** كما يلي

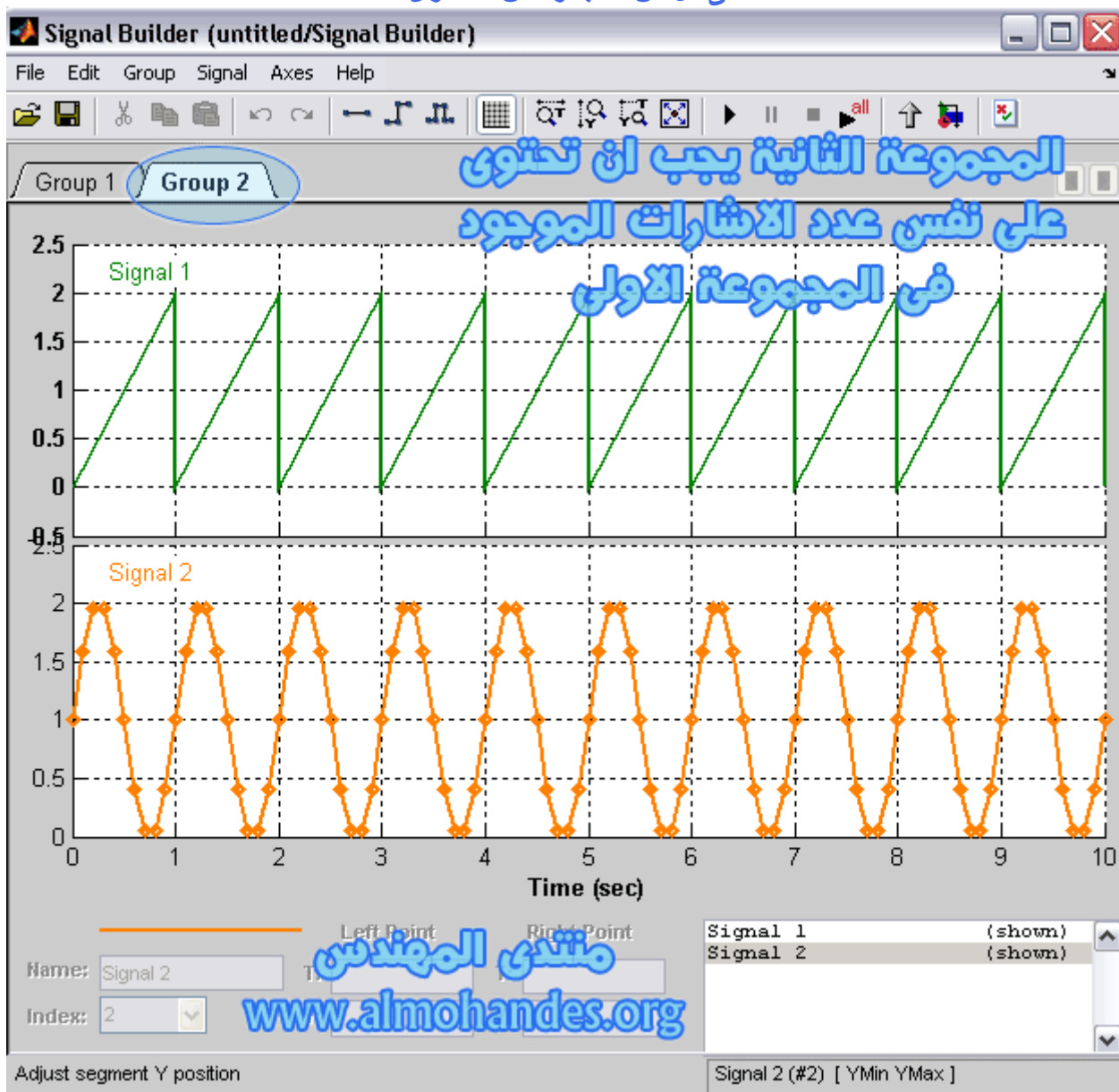


والان سنقوم بعمل مجموعة اخرى

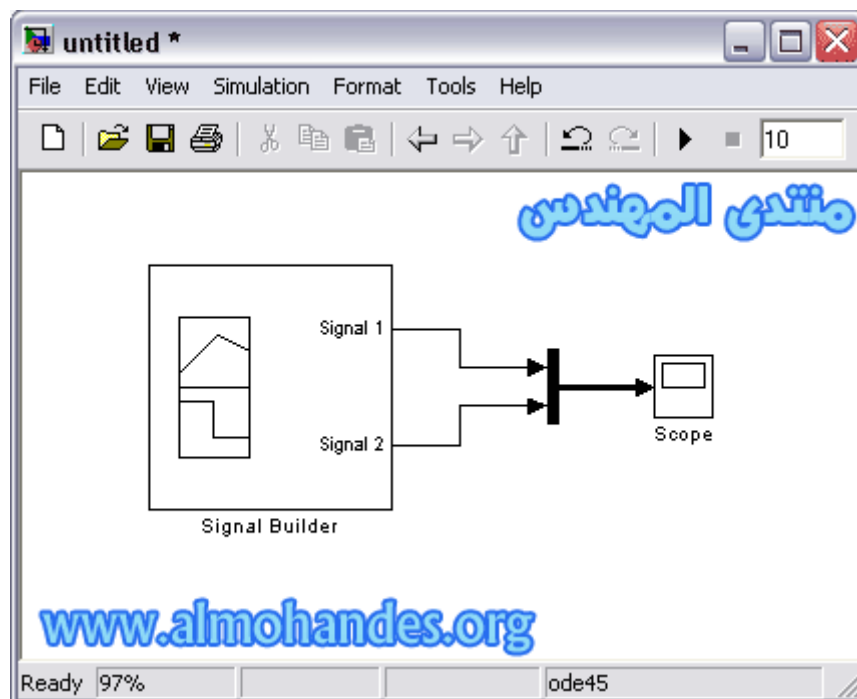
قم بالدخول الى قائمة **Group** واختار **Copy**

وسيقوم البرنامج بعمل مجموعة اخرى تحتوى على نفس اشارات المجموعة السابقة

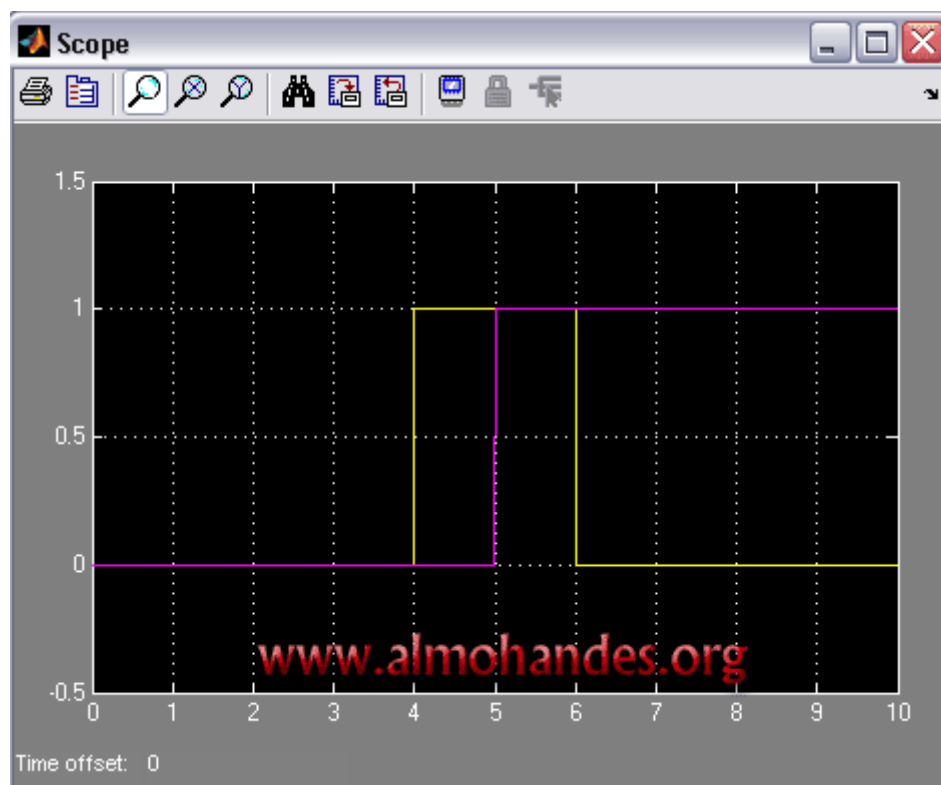
ادخل على المجموعة الثانية وقم بالتعديل فى الاشارات الموجودة بها كما تريد ويجب ملاحظة ان يكون عدد الاشارات فى كل من المجموعتان متساوى



والان قم بتحديد المجموعة التي تريد تخرجها من البلوك علما بان البلوك لا يخرج الا مجموعة واحدة  
ثم قم بتكملة الموديل كما يلي

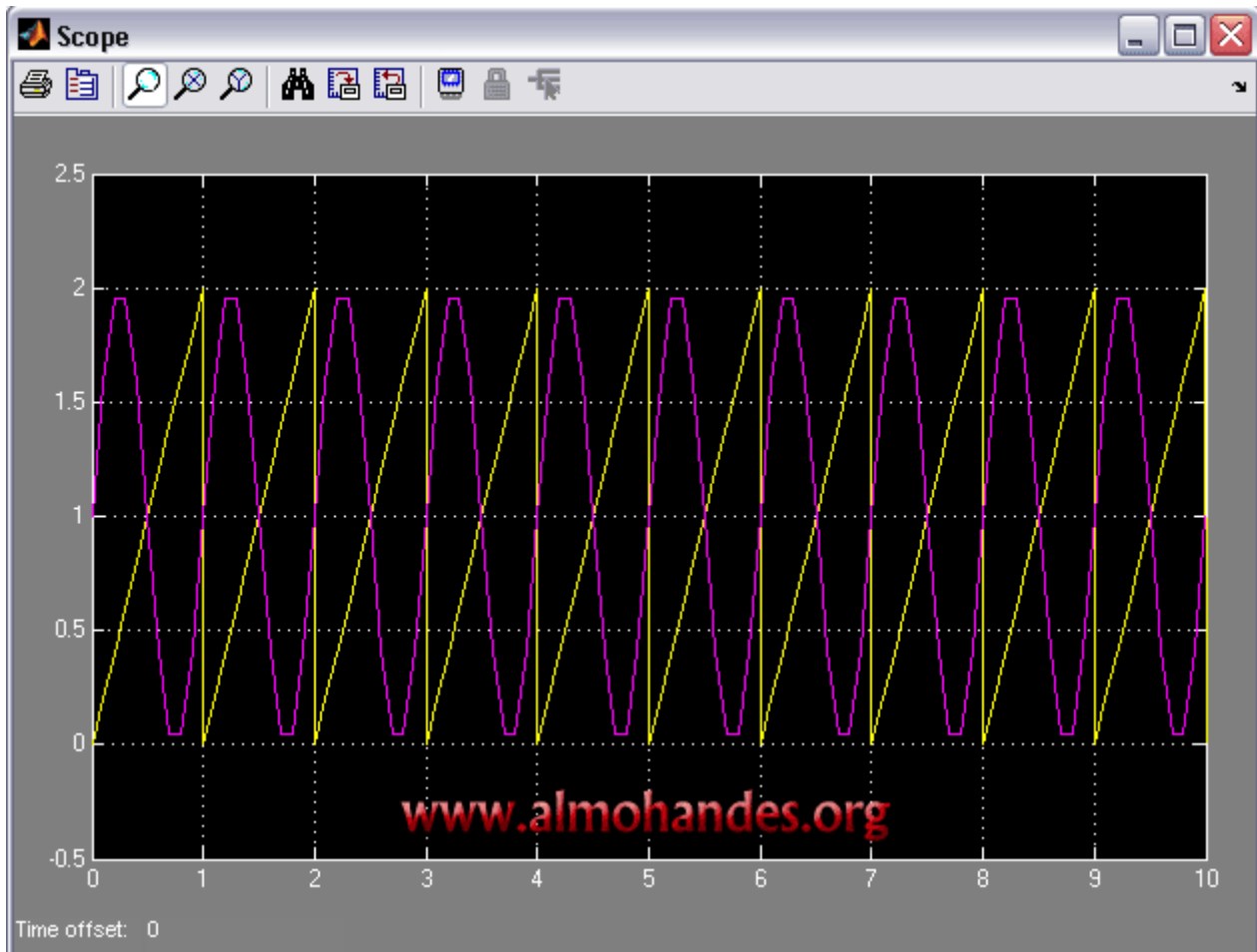


قم الان بتنفيذ عملية المحاكاة وستكون نتيجة التنفيذ فى حالة المجموعة الاولى كما يلي:





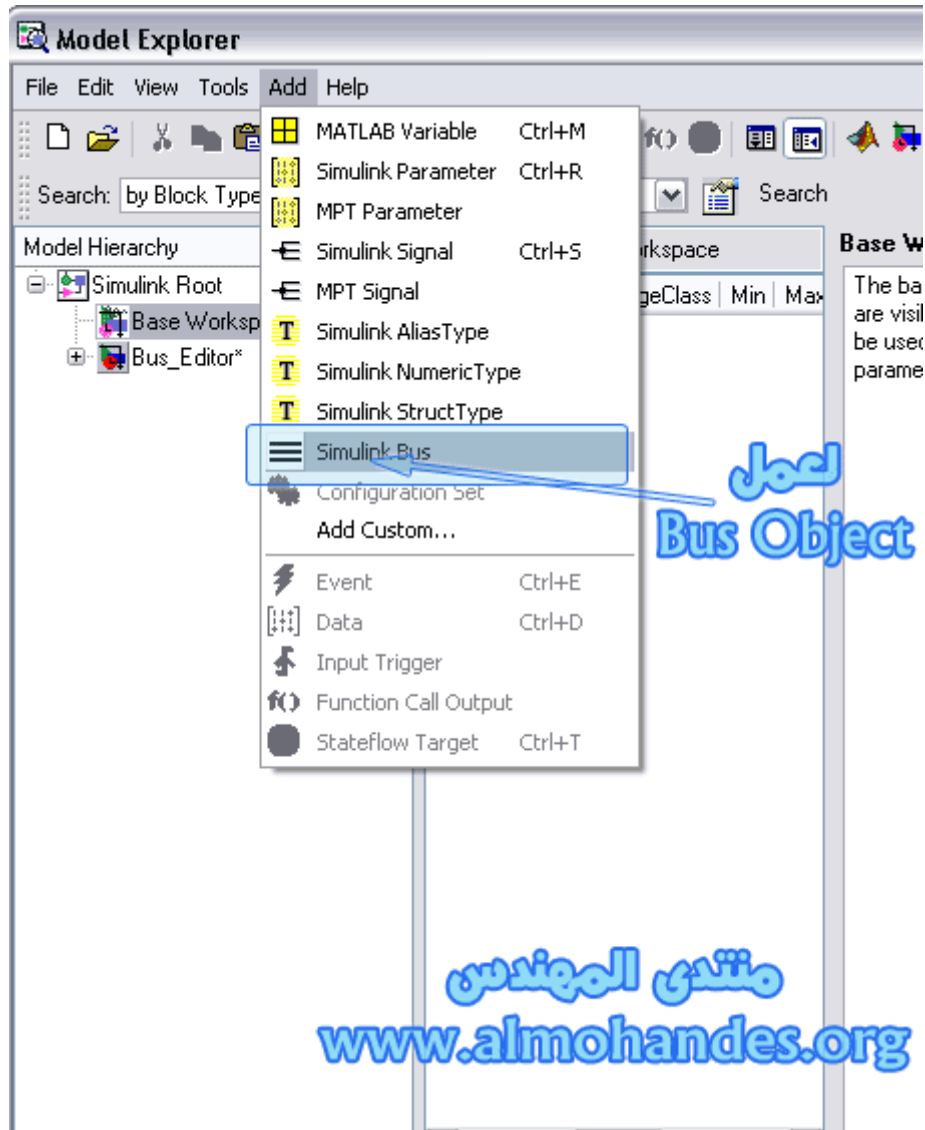
وفي حالة المجموعة الثانية تكون كما يلي:



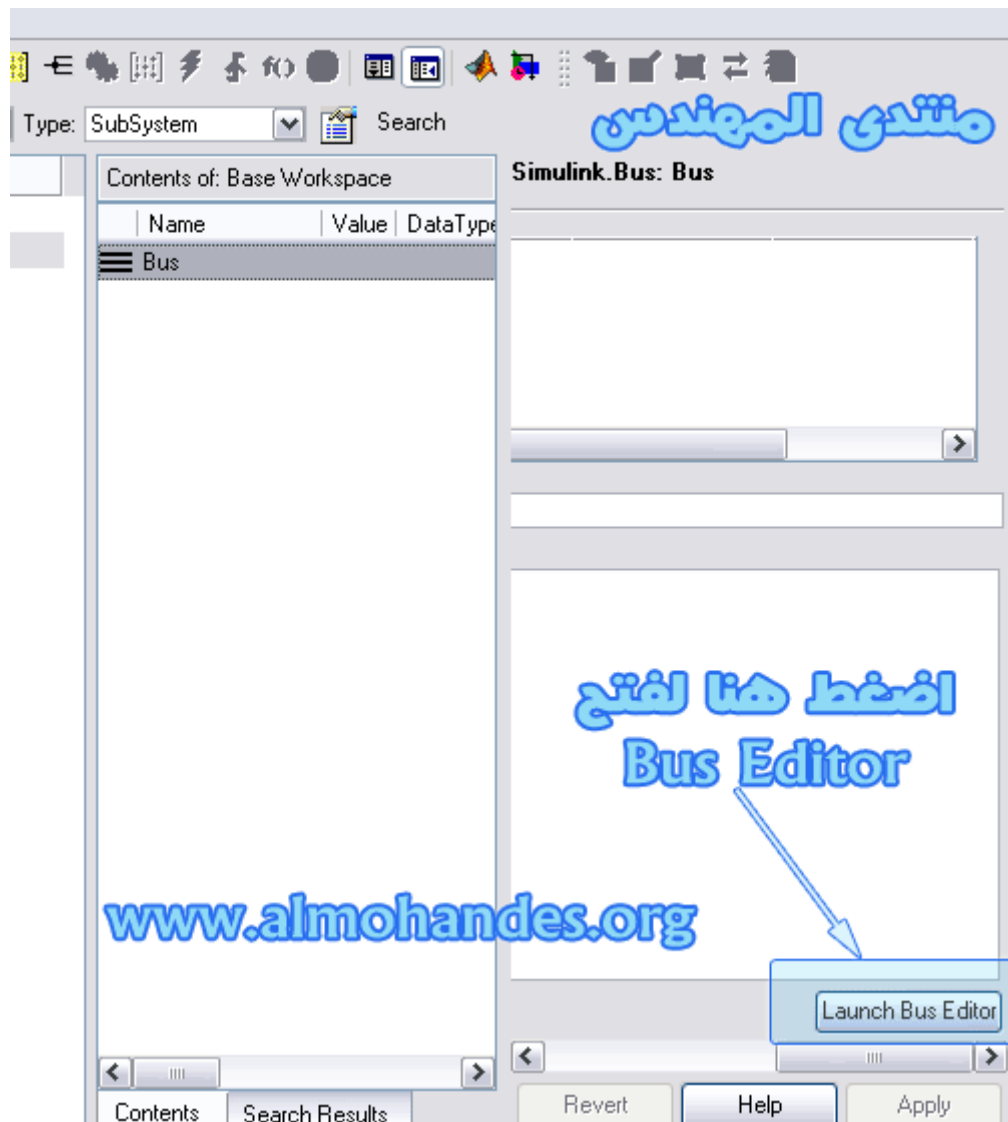
# Bus Editor

يتيح لنا ال **Bus Editor** تغيير خصائص ال **bus** المستخدم في نقل مجموعة من الاشارات  
اولا لفتح ال **Bus Editor** نستخدم احدى الطرق الاتية  
١- أختار **Bus Editor** من قائمة **tools**

٢- من ال **model explorer** اضغط على ايقونه **Bus Editor** الموجودة في ال **Bus object**  
فمثلا قم بعمل موديل فارغ وادخل على قائمة **view** واختر **Model explorer**  
وبعد ذلك في ال **model explorer** ادخل على قائمة **add** وقم باضافة **simulink bus** كما يلي:



والان سيظهر لنا ال **Bus object** وعند الضغط عليه تظهر لنا ايقونة  
**Launch Bus Editor**  
اضغط عليها لفتح **Bus Editor**

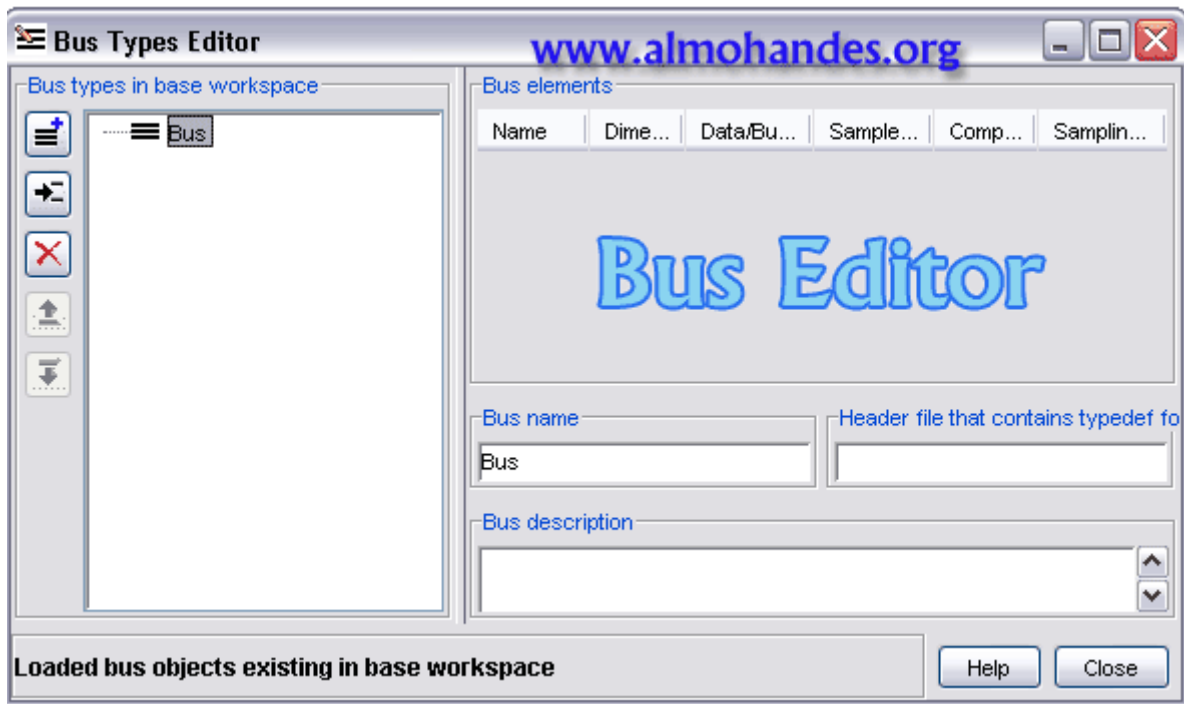


٣- واخر طريقة لفتح ال Bus Editor عن طريق كتابة الامر الاتي

رمز:

Buseditor

في سطر اوامر الماتلاب  
والان سيظهر لنا ال Bus editor كما بالشكل الاتي



وتوجد على اليسار قائمة يمكنك من خلالها اضافة **Bus**  
والايقونة الثانية لاضافة ال **Bus** الى النموذج وسيظهر على اليمين في خانة **Bus elements** مكونات ال **Bus**  
الذي تم اضافته .

سوف نتعرض ان شاء الله لمعرفة مميزات وخصائص ال **Bus editor** عن قرب عند الدخول الى فصل **working with data object**

النسبة المئوية لما تم انهاءه في الدورة حتى الان الى اجمالي الدورة يساوي ٥٠%

**Eng Ahmed Eldeep**  
[www.almohandes.org](http://www.almohandes.org)  
[ahmed@almohandes.org](mailto:ahmed@almohandes.org)

