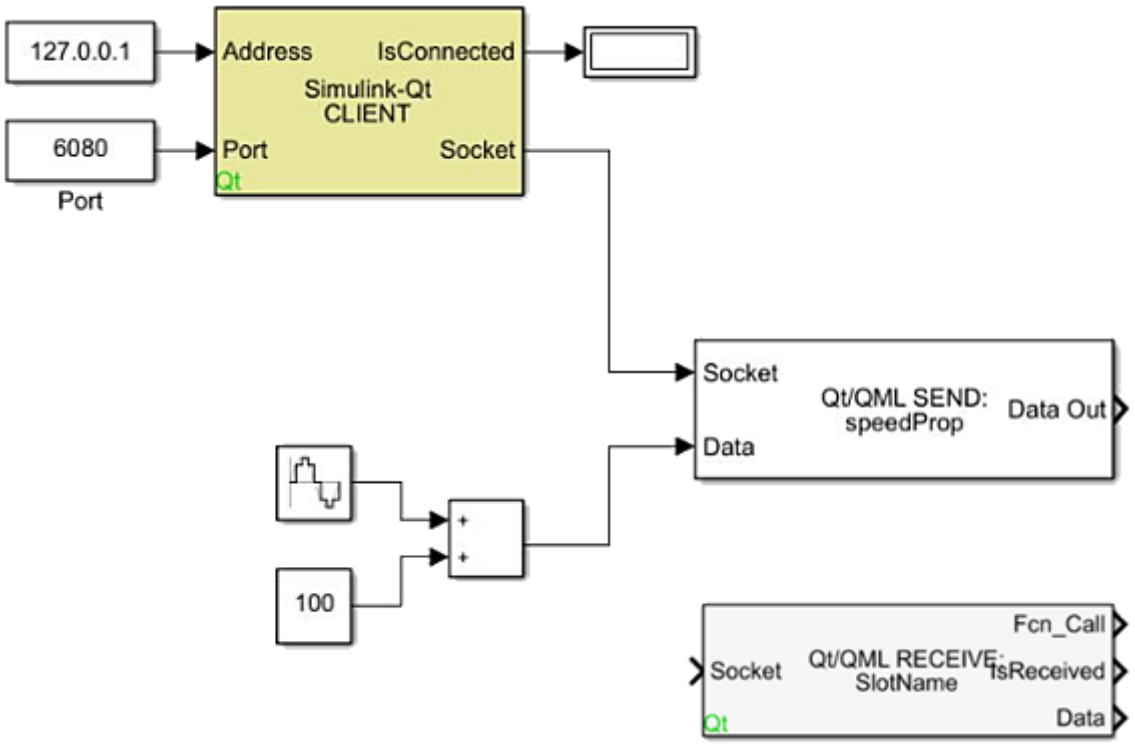
模拟动态系统

使用模拟链接连接器将模拟连接到 UI。Simulink 是一个基于 MATLAB 的图形编程环境，用于建模、仿真和分析多域动态系统。在 Windows 上，Qt 设计工作室提供与 Simulink 模型连接的内置支持，允许它们使用 Qt 设计工作室开发的应用程序发送和接收数据。在计算机上安装 Simulink，并与 Qt 设计工作室同时运行它，以实现应用程序之间的通信。

这里提供的信息主要集中在Qt设计工作室中Simulink连接器的集成上。有关如何使用 Simulink 环境的信息，请参阅数学工作提供的文档。

用于模拟链接的 Qt 模块集

将 Simulink Qt 模块集安装到您的计算机上，以便将 Simulink 模型连接到您的应用程序。Qt 模块集安装程序添加了与应用程序建立连接所需的 Simulink 模块。安装后，**SLQT 库**块集将添加到 Simulink 块库中。这些块允许随应用程序发送和接收**属性**、**信号**和**插槽**更新。Qt 块集包括**模拟链接-Qt 客户端**、**地址**、**Qt/QML 发送**和**Qt/QML 接收**块。



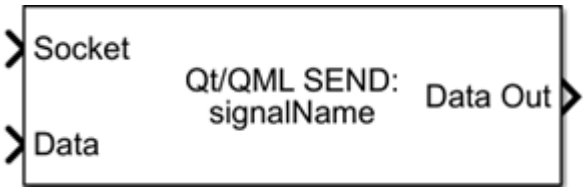
思慕林克-Qt 客户端

- **端口**输入指定 IP 地址的端口值，该值可以通过使用**端口**块或有效的 Simulink 整数值来确定。
- "**已连接**"输出为布尔信号。如果为 true，则指定已建立与服务器的连接。
- **套接字**输出发送一个信号，该信号显示连接的套接字 ID。该信号需要传递到相应的 **Qt/QML 接收**块和 **Qt/QML 发送**块。

地址和端口

地址块将服务器的 IP 地址作为典型的 IP 地址字符串传递给 **Simulink-Qt 客户端**块。**端口**块确定 IP 地址的端口值。对于在同一台计算机上运行 Simulink 模型和应用程序时的模拟，请使用 IP 地址 127.0.0.1 和任何可用的端口。

Qt/QML 发送



Qt/QML 发送块从 Simulink 发送**信号**或**属性**值更改。它用于 Simulink 需要发送到您的应用程序的每个属性。块的属性名称需要与应用程序中的属性或槽的名称相对应。

该块有两个输入和一个输出：

- **套接字**输入接收来自 **Simulink-Qt 客户端**模块的套接字信号。
- **数据**输入接收要作为**信号**或**属性**更新发送的数据。
- **数据输出**输出输出输出传递的数据，以便在需要时将其连接到其他 Simulink 模块。

Qt/量子点接收

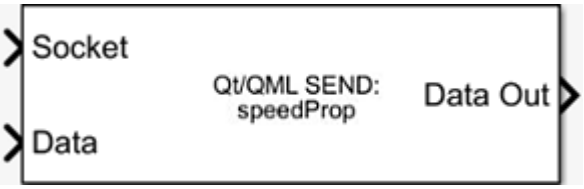
Qt/QML 接收块从应用程序接收**信号**或**属性**值更改。它用于 Simulink 需要从您的应用程序接收的每个属性。块的属性名称需要与应用程序中的属性或槽的名称相对应。

The block has one input and two outputs:

- The **Socket** input receives the socket signal from the **Simulink-Qt Client** block.
- The **Fcn_Call** output sends the function-call, which can either be terminated if idle, or connected to a valid function call subsystem.
- The **isReceived** output emits a scalar Boolean signal suggesting that a valid **Signal** or **Property** update was acquired from the connection.
- The **Data** output signals data payload from a **Signal** or **Property** value.

Specifying Property Names in Simulink

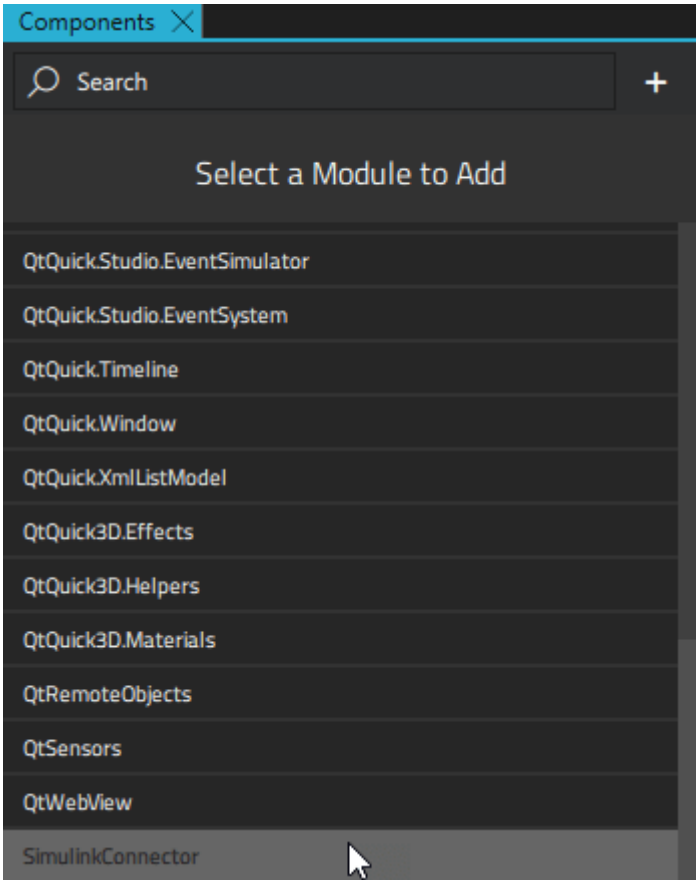
Double-click the **Qt/QML Send** or **Qt/QML Receive** block in Simulink to specify a property name. A pop-up for **Block**



Integrating the Simulink Model to Qt Design Studio

Importing the Simulink Connector

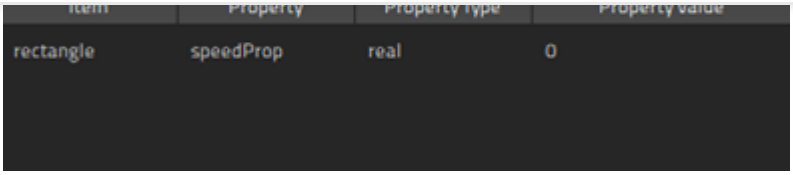
To integrate the Simulink model into Qt Design Studio, you first need to add the Simulink connector module to your project. In the **Components** view, select + > **SimulinkConnector**. Qt Design Studio is now ready to communicate with the Simulink model.



If you need to change the IP address and/or port, you need to select the **SimulinkConnector** item in Navigator and set the IP address and/or port in the Properties view. If you cannot see **SimulinkConnector** in **Navigator**, you need to click (**Filter Tree**) and unselect **Show only visible items**.

To communicate with a specific model in Simulink, you need to create properties matching the send and receive properties in the root of the application you are building. Select the root item in **Navigator** to add the properties on the **Properties** tab in the Connections view.

See Specifying Custom Properties for a detailed description of how to add a custom property. The name of the property and the data type need to match those of the send or receive property of the Simulink model.

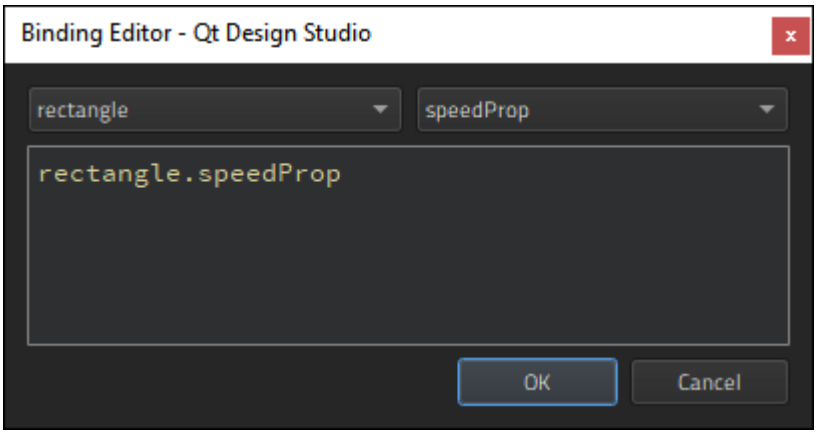


Creating Bindings

Next, you need to bind the value of the property you just created to the desired properties of UI components.

By binding the root item property to a component property you can use it, for example, to rotate an component. Binding a root item property of speed to a component property of rotation would result in the item rotating in the screen when the simulation is run.

To bind the root item property to a component property, select the component either by clicking on it on the canvas or in **Navigator**. In the **Properties** view, find the component property to which you want to bind the root item property. Select the (**Actions**) menu next to a property, and then select **Set Binding**. In the **Binding Editor**, select the text field and type in , for example . For more information, see Setting Bindings.`<id>.<property name>` `rectangle.speedProp`



Run the simulation by first clicking the **Run** icon in Qt Design Studio and then the **Run** icon in Simulink.

‹ Simulating Application Logic

Using QML Modules with Plugins ›

Newsroom
Careers
Office Locations

FAQ

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login
Contact Us
Customer Success

Community

Contribute to Qt
Forum
Wiki
Downloads
Marketplace

Feedback Sign In