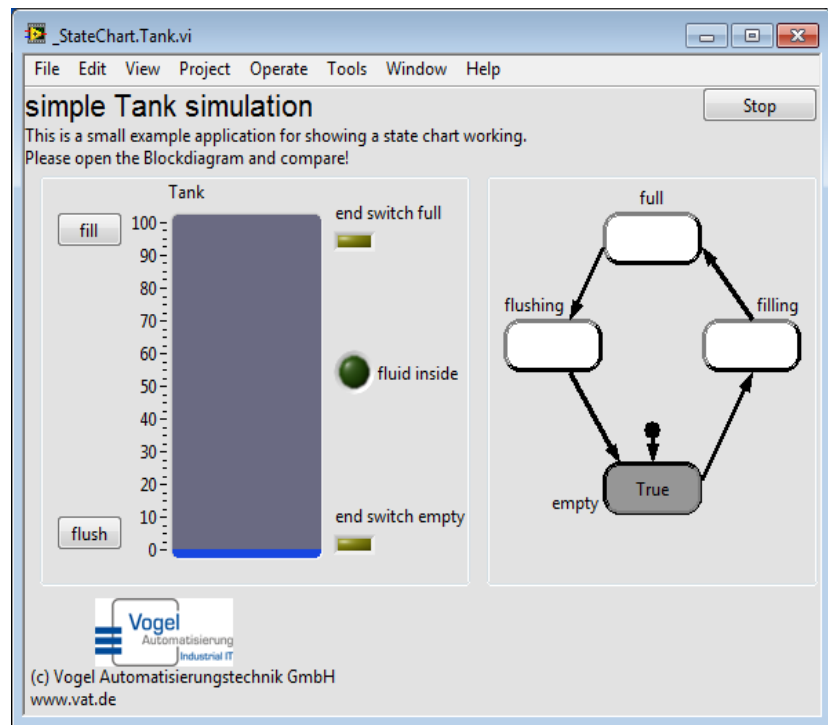
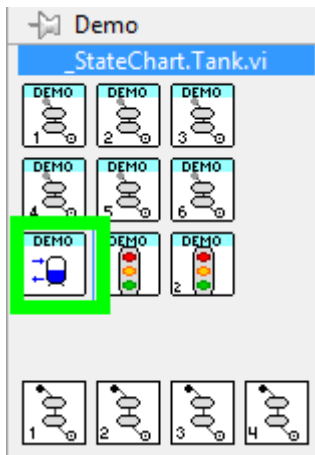


Tutorial open StateChart

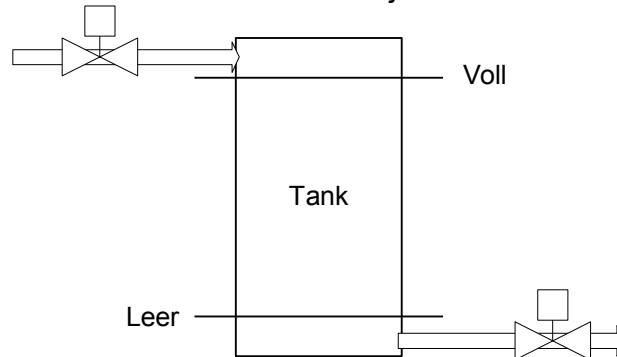
In diesem Dokument soll gezeigt werden, wie die Funktionen der open StateChart Funktionsbibliothek verwendet werden können.

Als Grundlage der Demonstration soll das Beispiel Tank dienen. Sie finden das Beispiel in der Palette „user libraries/StateChart.“



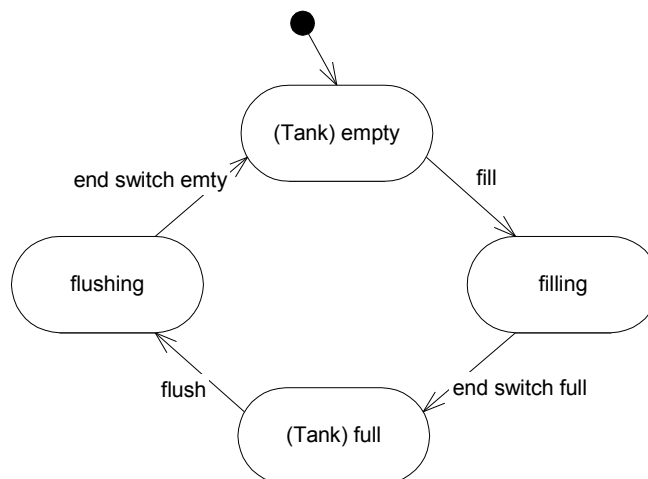
Vorbetrachtung

Es soll zu Veranschaulichung ein sehr einfacher Fall dienen.
 Ein Tank besitzt einen Zu- und einen Ablauf sowie je einen Sensor für Leer und Voll.



Der Füllstand des Tanks soll zwischen Voll- und Leermeldung pendeln. Der Tank soll immer vollständig entleert bzw. gefüllt werden.

Hierzu kann man sich ein einfaches Ablaufdiagramm entwerfen:

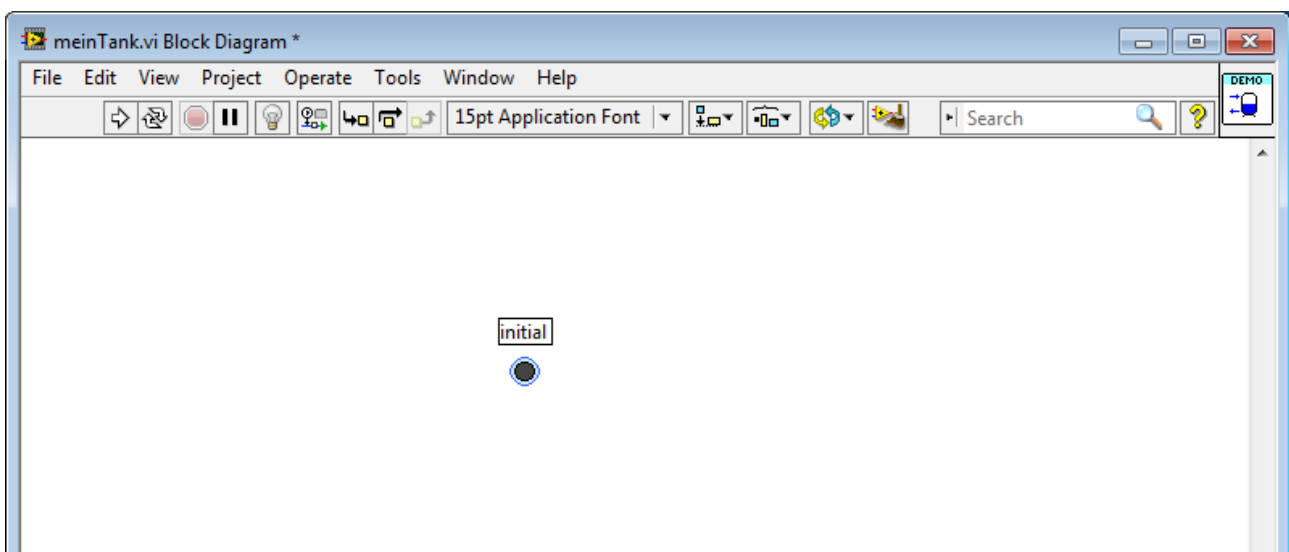
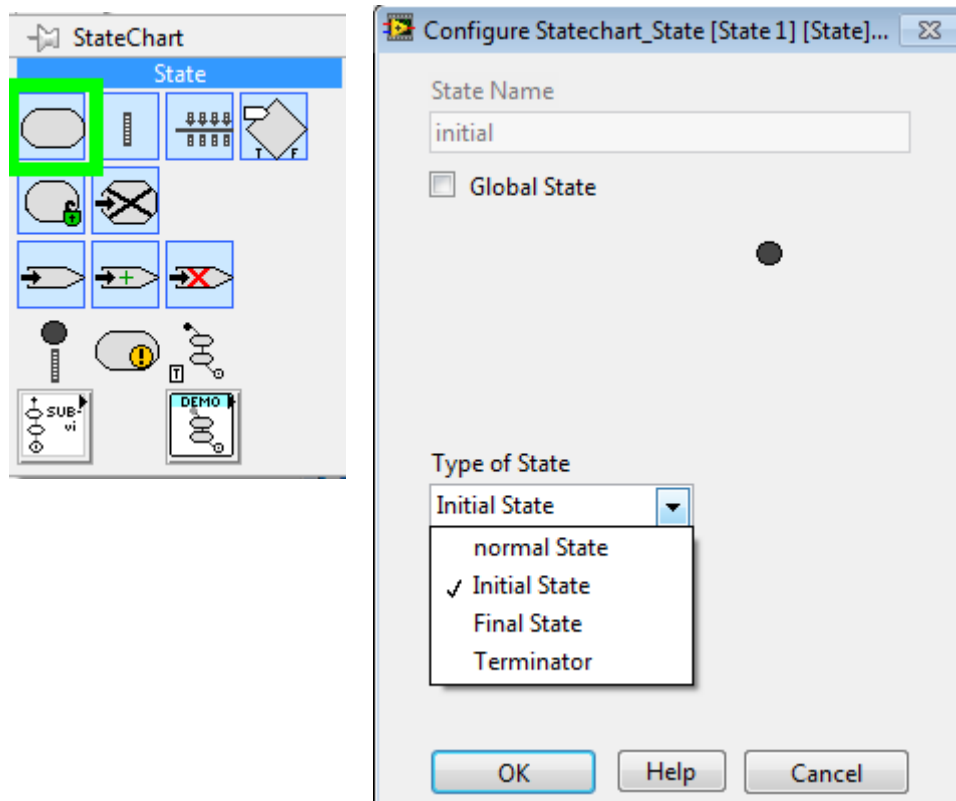


Umsetzung mit open StateChart

Als erstes werden die Anzeigeelemente und Funktionen für die Zustände der Statemaschine auf dem VI platziert.

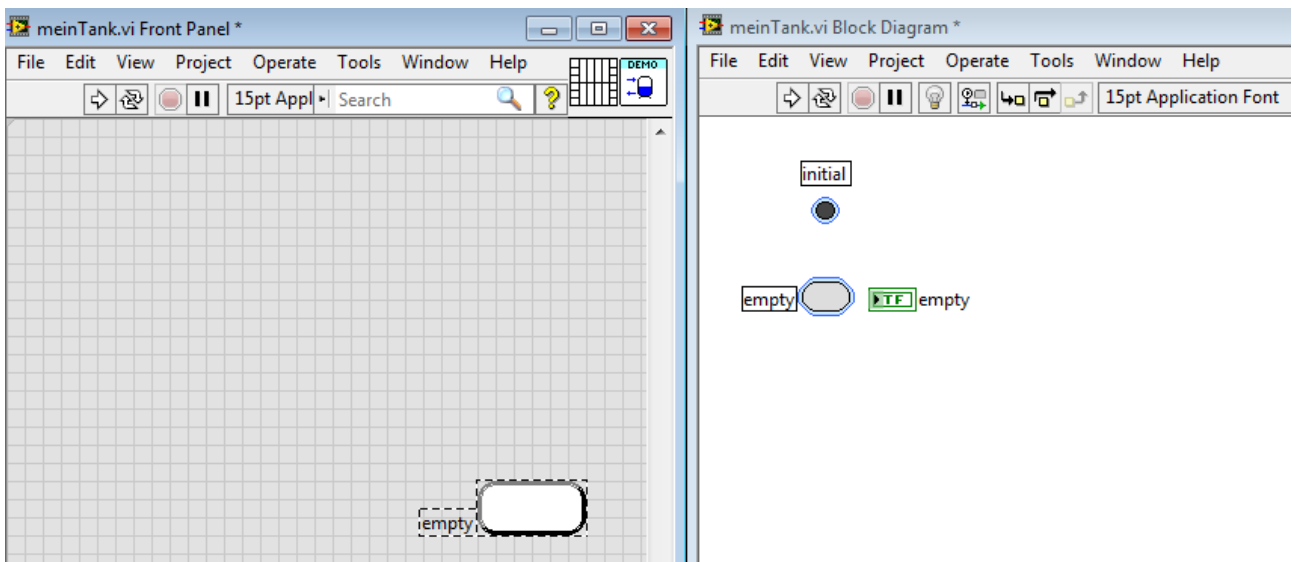
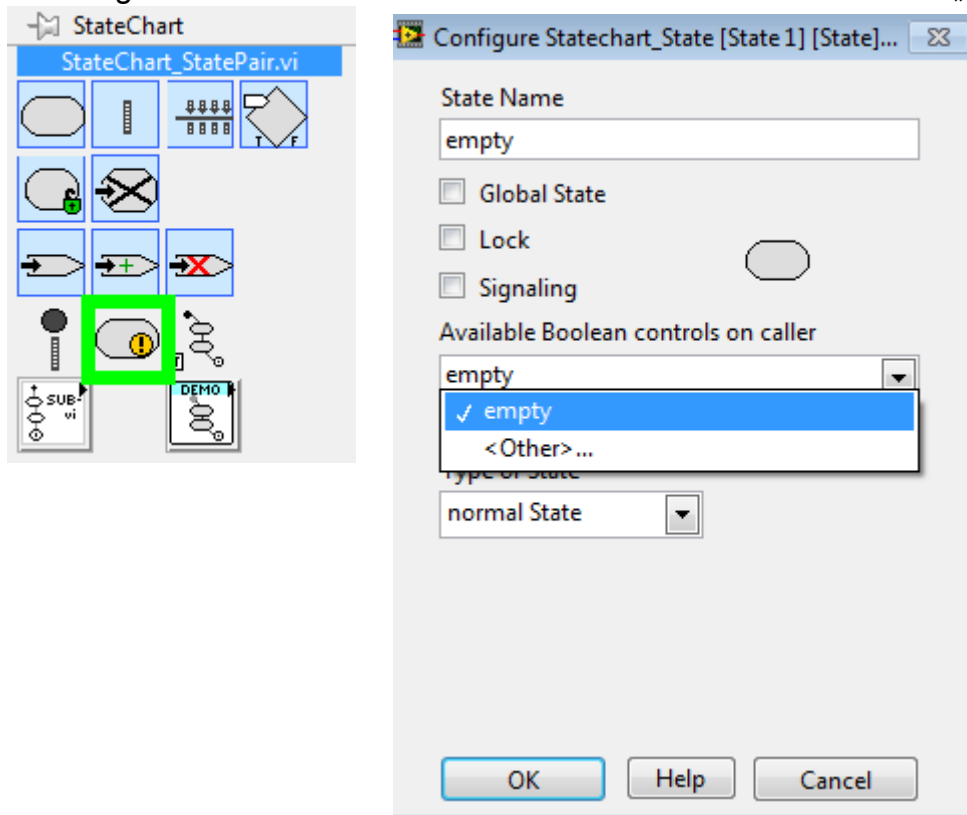
1. Schritt Zustände

Fangen wir mit dem Startzustand an. Platzieren Sie ein State VI auf dem Blockdiagramm.

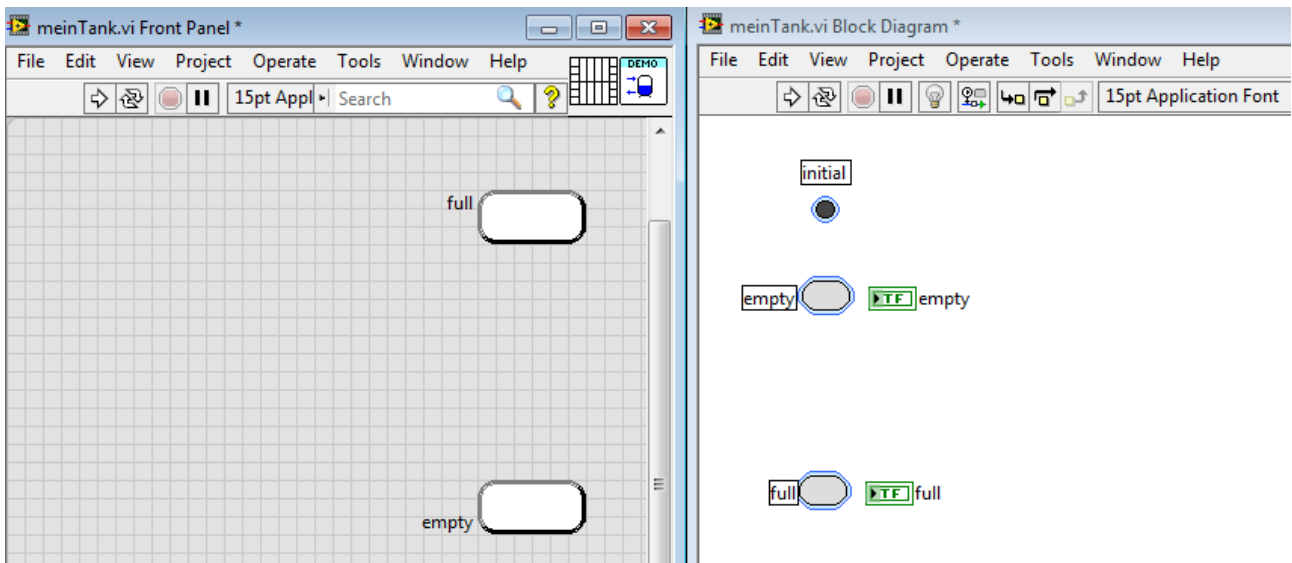


Konfigurieren Sie den Zustand als Initial State.

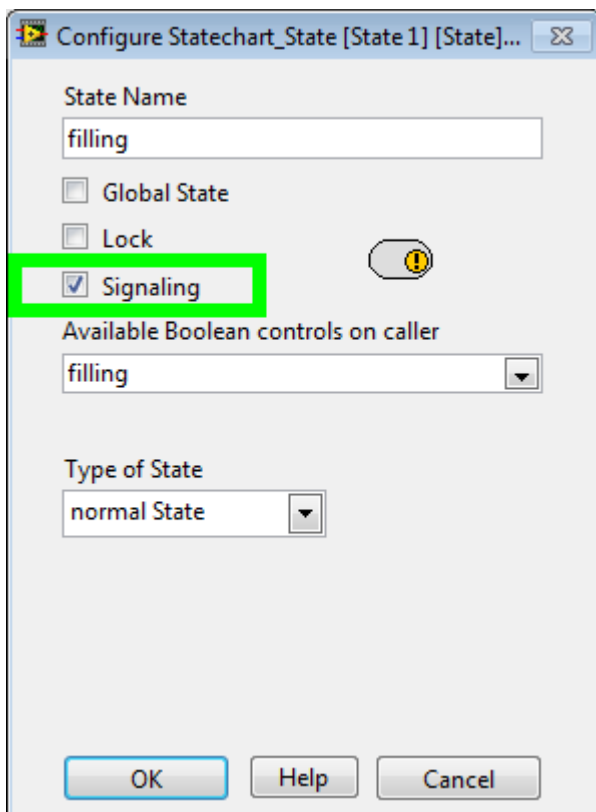
Fügen Sie nun den Zustand „empty“ hinzu. Da wir ein Anzeigeelement für den Zustand auf dem Frontpanel haben wollen, können wir ein State Pair einfügen. Ein State Pair ist die Kombination State VI und Anzeigeelement. Benennen Sie den Indikator nach „empty“ um und konfigurieren Sie das State VI als normal State mit dem Namen „empty“.

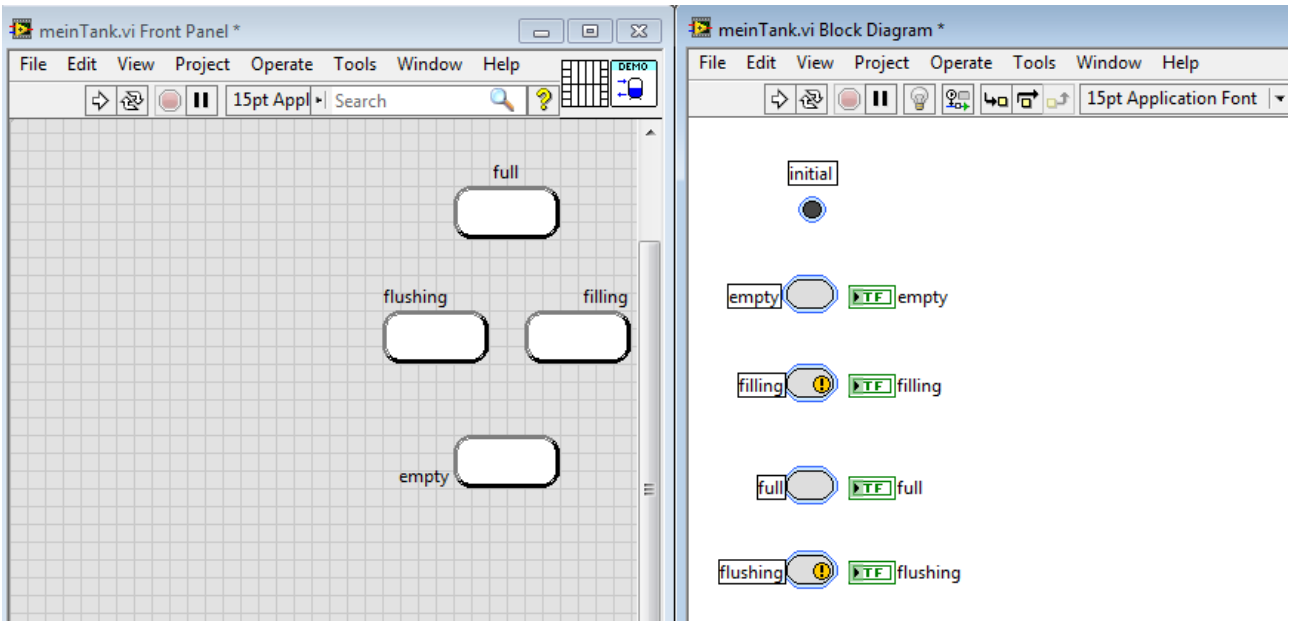


Wiederholen Sie den Vorgang für den Zustand „full“.

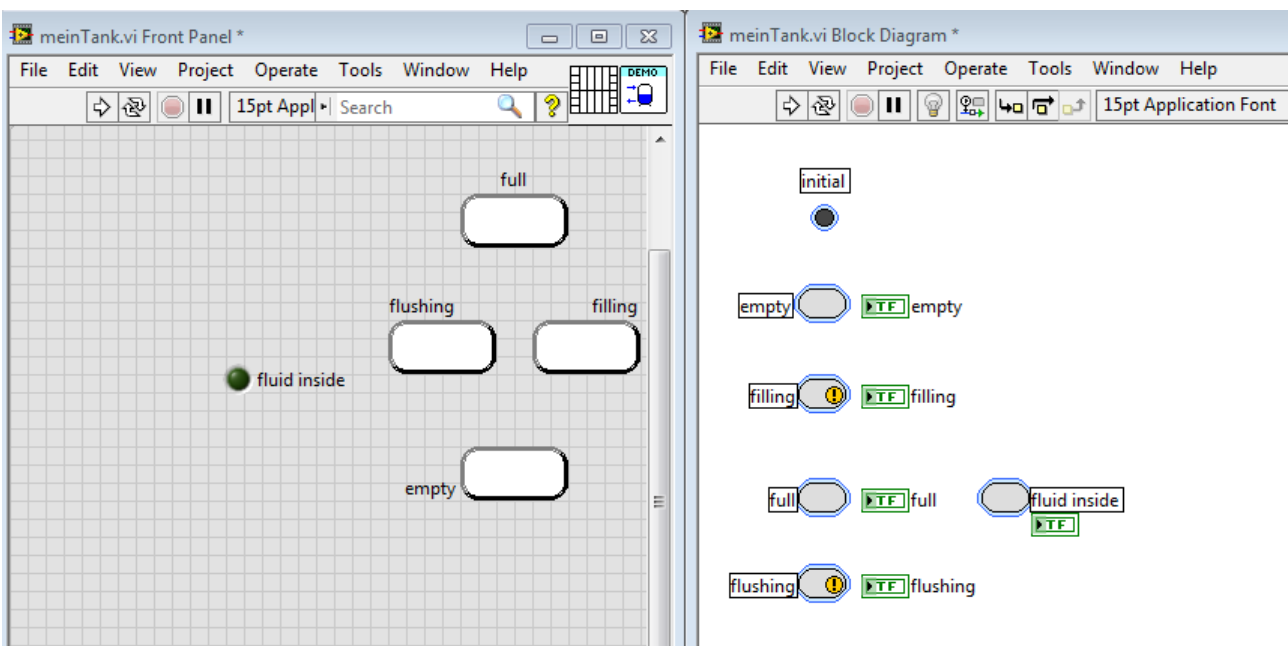


Als nächstes kommen die Zustände „filling“ und „flushing“. Hierbei ist zu beachten, dass für beide Zustände Aktionen bei Zustandsänderung zu programmieren sind. Wir wollen also ein Ereignis bei Zustandswechsel bekommen. Selektieren Sie dafür das Kästchen „Signaling“.





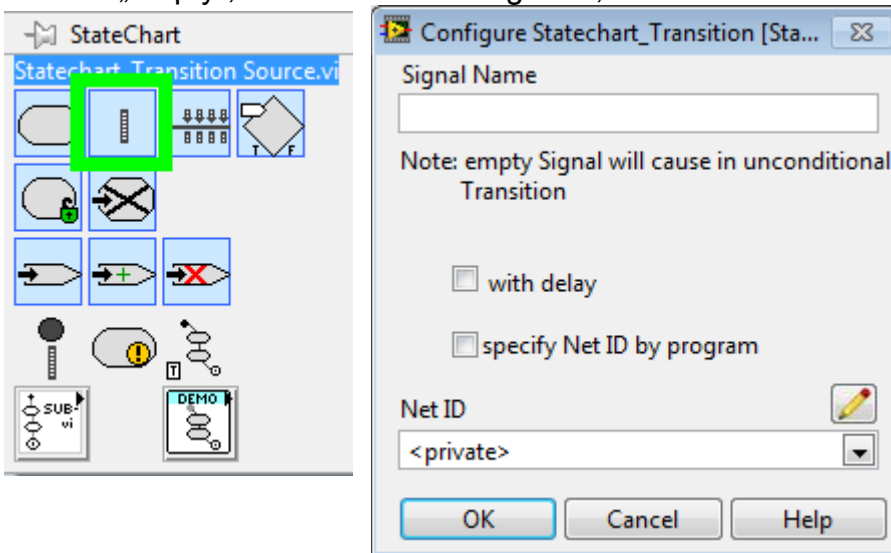
Wir wollen zusätzlich wissen, ob Flüssigkeit im Tank ist. Dies wollen wir mit einem zusätzlichen Zustand „fluid inside“ anzeigen.



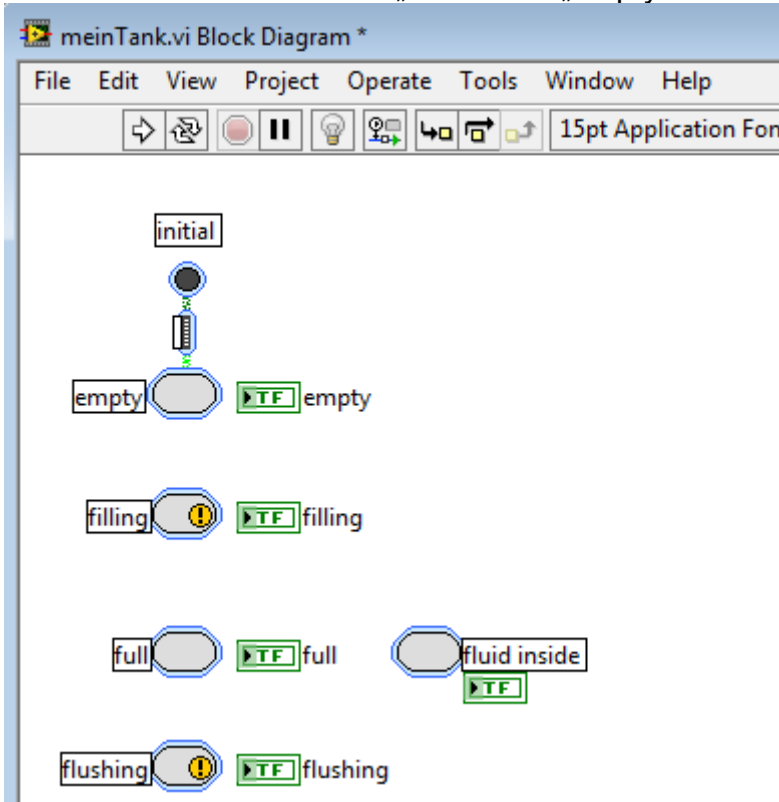
Die Zustände für unsere kleine Anwendung sind fertig.

2. Schritt Transitionen

Als nächstes programmieren wir die Schaltbedingungen für unser Netz. Zuerst verbinden wir unseren Startzustand mittels einer unbedingten, sofort schaltenden Transition mit dem Zustand „empty“, weil wir davon ausgehen, dass der Tank bei Systemstart leer ist.

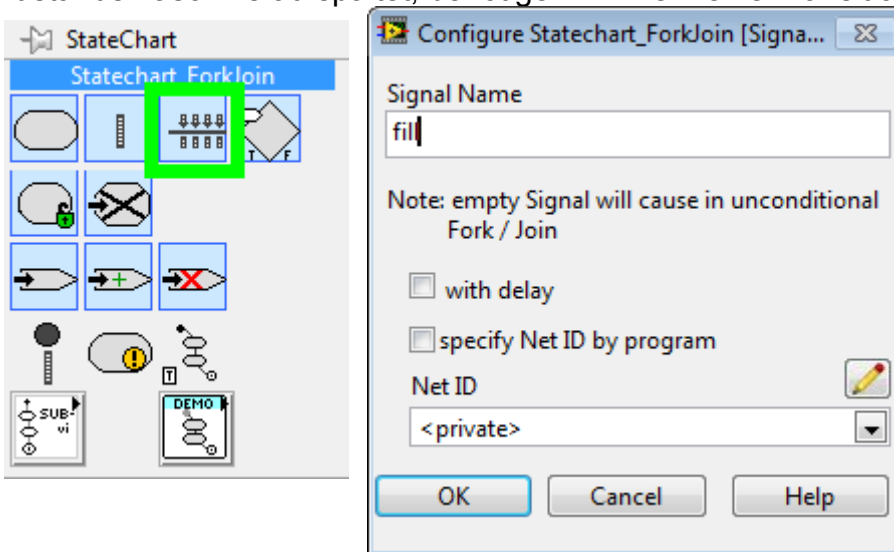


Lassen Sie hierfür den Namen für die Schaltbedingung leer. Wir wollen ein privates Netz erstellen (Der Netzname zur globalen Identifikation wird gleich den Namen der VI Instanz des aufrufenden Vi's gesetzt. - In unserem Fall „mein Tank.vi“). Verbinden Sie die Zustände „initial“ und „empty“ mit der Transition.

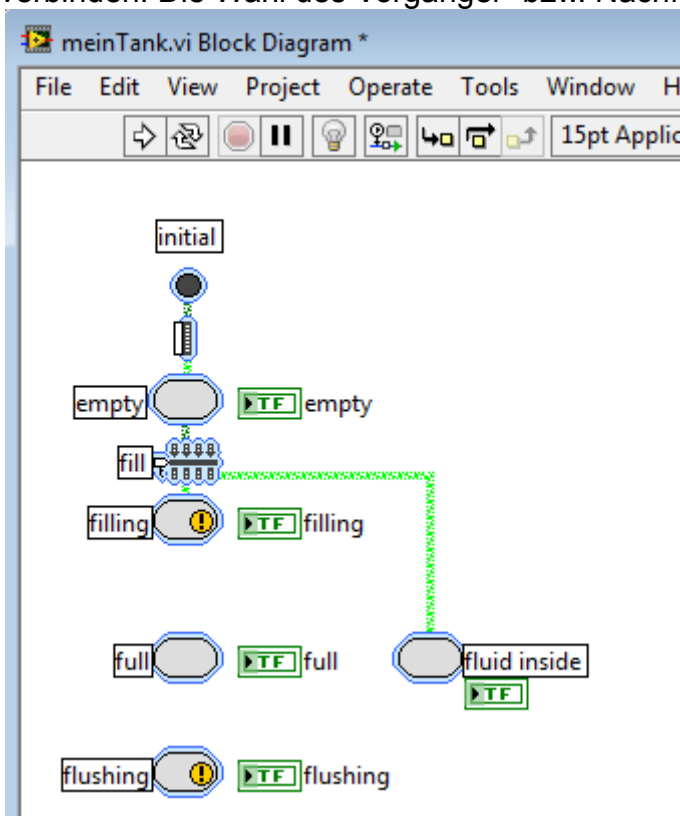


Der Übergang von „empty“ nach „filling“ soll erst erfolgen, wenn der Befehl „fill“ den Wert

„True“ besitzt. Zusätzlich soll der Zustand „fluid inside“ gesetzt werden. Da sich hier die Zustandsmaschine aufspaltet, benötigen wir hier keine Transition, sondern eine Gabelung.

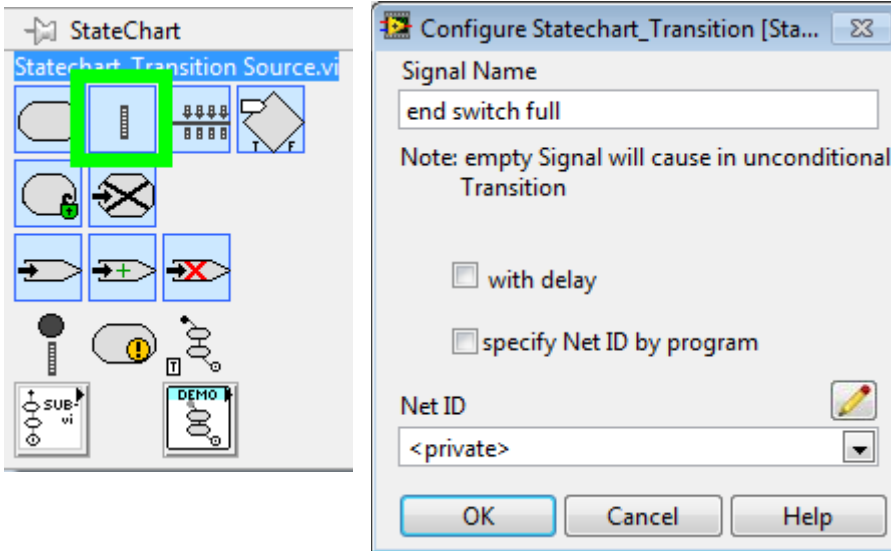


Tragen Sie im Konfigurationsdialog den Signalnamen „fill“ als Weiterschaltbedingung ein. Nun können Sie die Gabelung mit den Zuständen „empty“ und „filling“ sowie „fluid inside“ und „flushing“ verbinden. Die Wahl des Vorgänger- bzw. Nachfolger-Anschlusses ist frei.

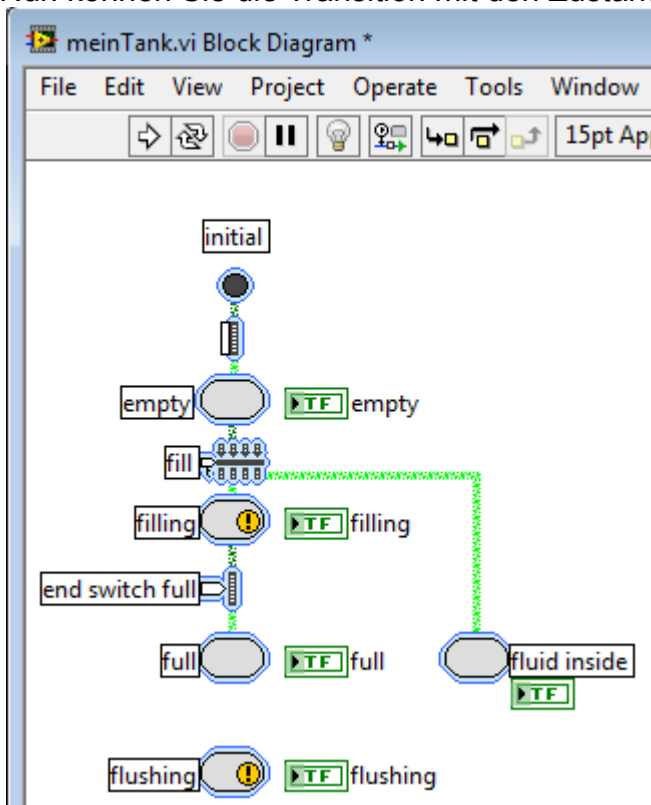


Wir wollen mit dem Befüllen des Tanks aufhören, wenn der Endschalter „end switch full“ den Wert „True“ erreicht. Hierfür benutzen wir eine bedingte Transition.

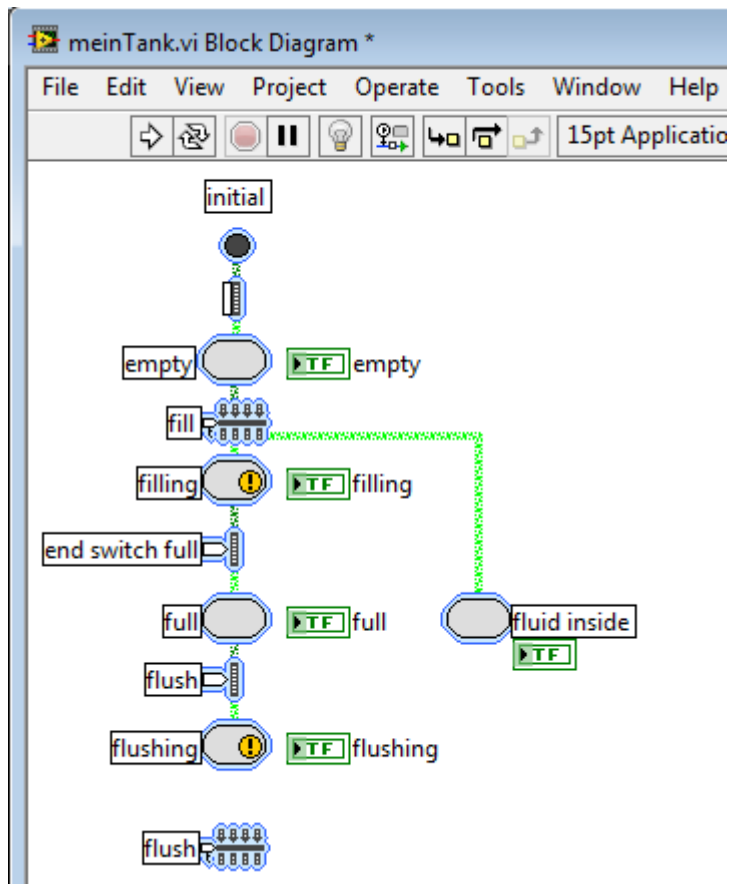
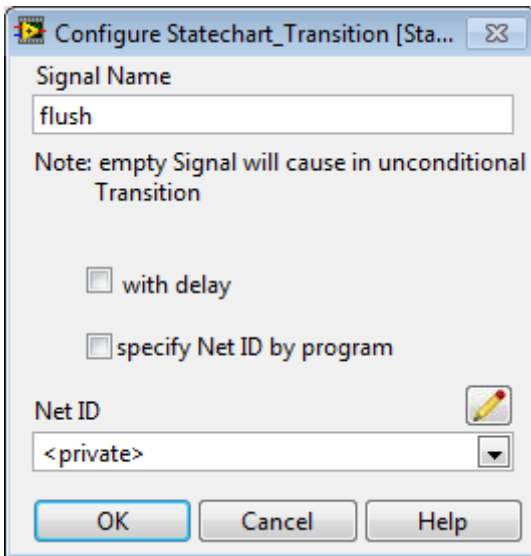
Tragen Sie im Konfigurationsdialog der Transition den Signalnamen „end switch full“ als Weiterschaltbedingung ein.



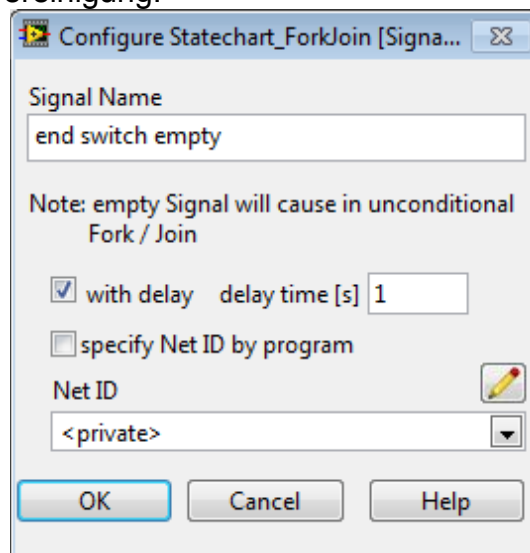
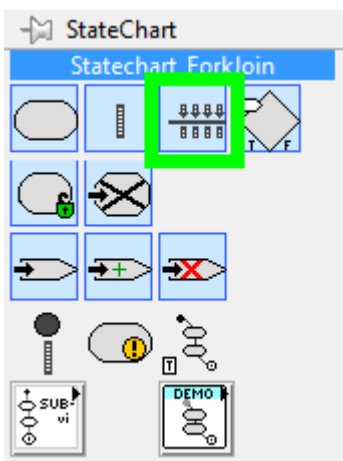
Nun können Sie die Transition mit den Zuständen „filling“ und „full“ verbinden.



Der Übergang von „full“ nach „flushing“ soll erst erfolgen, wenn der Befehl „flush“ den Wert „True“ besitzt.

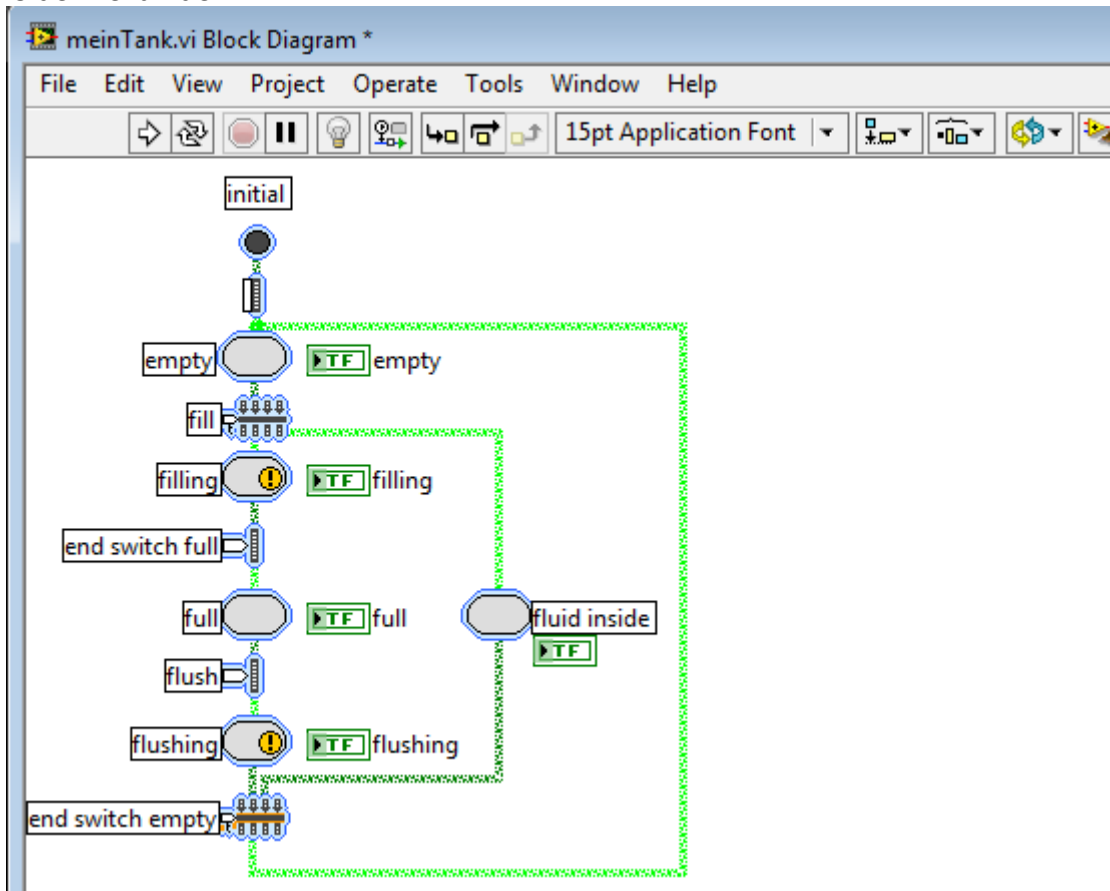


Wir wollen mit dem Entleeren des Tanks aufhören, wenn der Endschalter „end switch empty“ den Wert „True“ erreicht. Zusätzlich soll der Zustand „fluid inside“ rückgesetzt werden. Da sich hier die Zustandsmaschine vereint, benötigen wir auch hier keine Transition, sondern eine Vereinigung.



Tragen Sie im Konfigurationsdialog den Signalnamen „end switch empty“ als Weiterschaltbedingung ein. Zusätzlich können Sie eine Verzögerungszeit für die Weiterschaltbedingung konfigurieren, um den Tank restlos zu entleeren. (Hier als Beispiel 1s)

Nun können Sie die Vereinigung mit den Zuständen „flushing“ und „empty“ sowie „fluid inside“ verbinden.



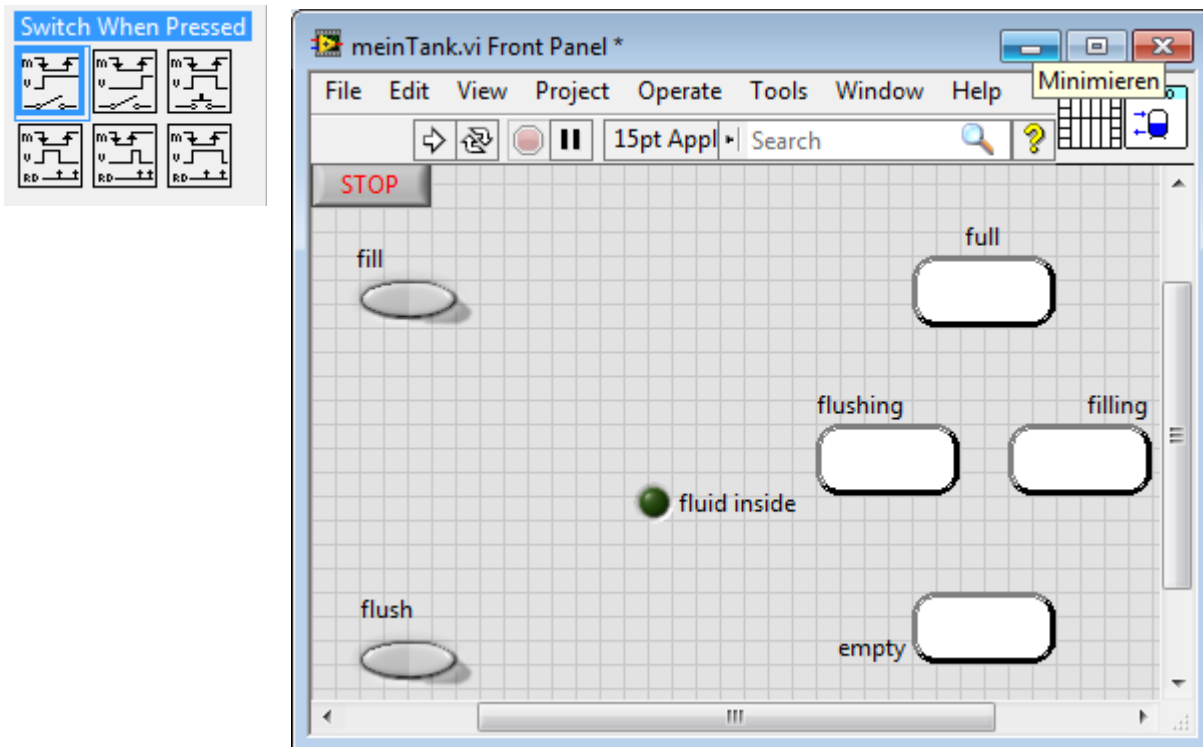
Unsere Zustandsmaschine ist hiermit fertig.

Setzen der Schaltbedingungen

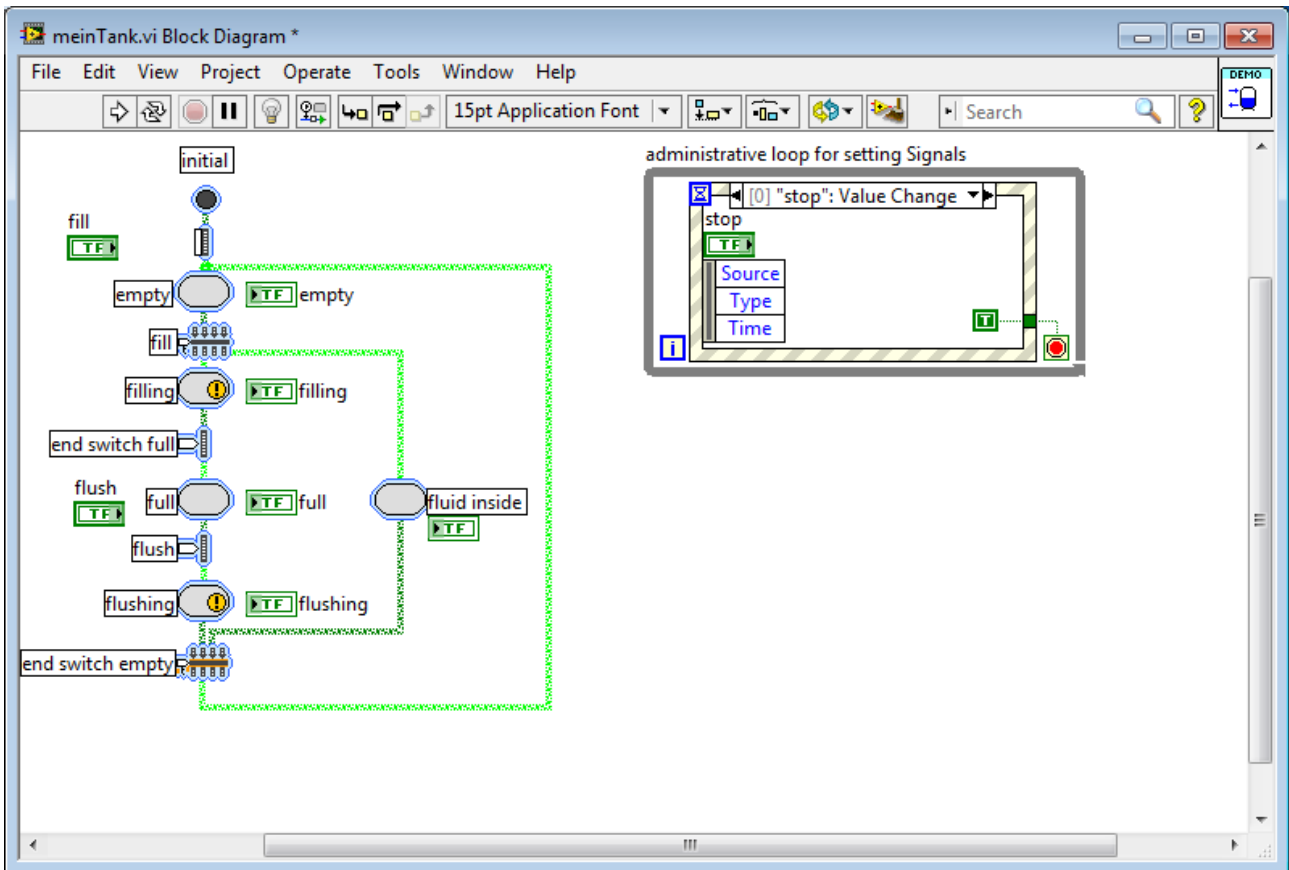
Das Zustandsdiagramm ist bereits grundsätzlich arbeitsfähig, es fehlen jedoch noch das Setzen der Schaltbedingungen und aus Ermangelung eines wirklichen Tanks, eine Simulation.

Erstellen Sie nun Schaltelemente für die Schaltbefehle „fill“, „flush“ und zum Beenden des Beispielprogramms.

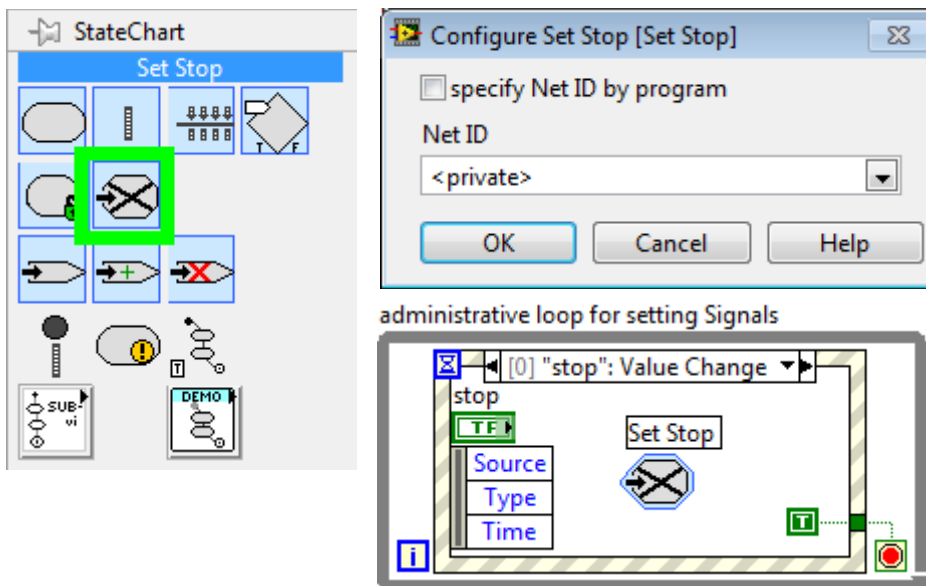
Ändern Sie die Mechanical Action aller Schaltelemente auf „Switch When Pressed“



Für die Auswertung der Tasten benötigen wir eine „Event Struktur“

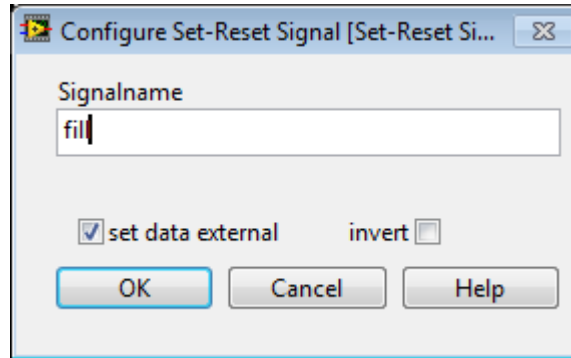
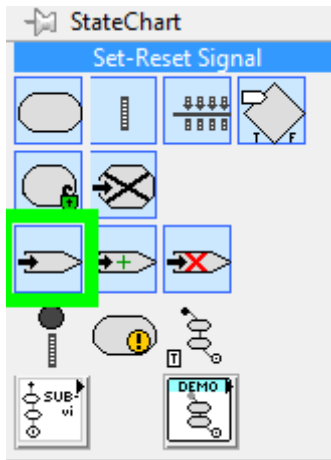


Erzeugen Sie je ein "Event Case" für die Wertänderung der Tasten „stop“, „fill“ und „flush“.
 Im Fall „stop“ soll das Programm und somit auch das Netz beendet werden. Sie können das Netz mit einem Terminator oder der Funktion SetStop beenden. Wir benutzen hier die Funktion SetStop.

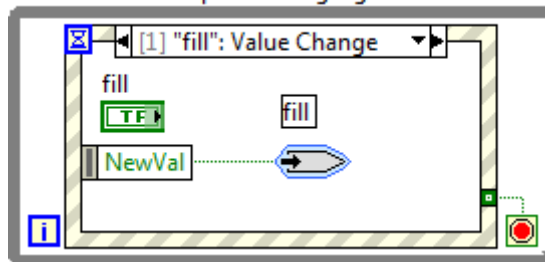


Die Freigaben des Füllens und Entleerens werden mit der Funktion „Set-Reset Signal“

realisiert. Tragen Sie im Konfigurationsdialog den Signalnamen „fill“ ein. Setzen Sie das Häkchen für „set data external“.

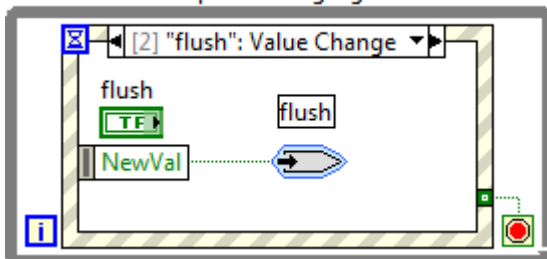


administrative loop for setting Signals



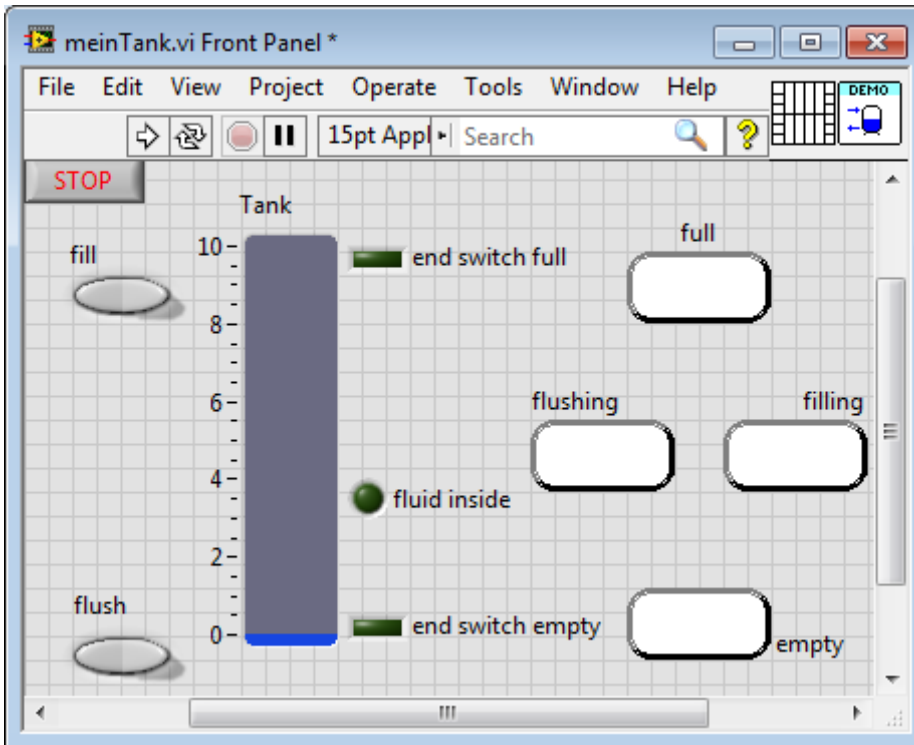
Wiederholen Sie das Gleiche für den Signalnamen „flush“.

administrative loop for setting Signals

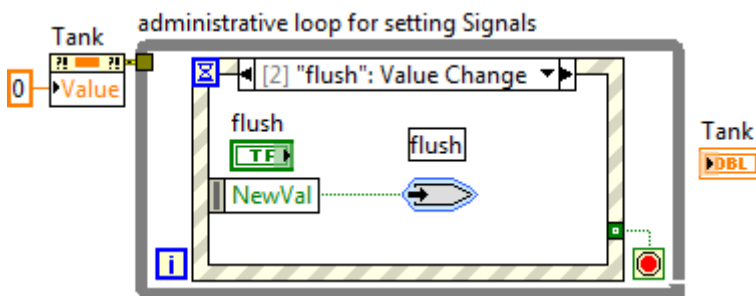


Simulation des Tanks

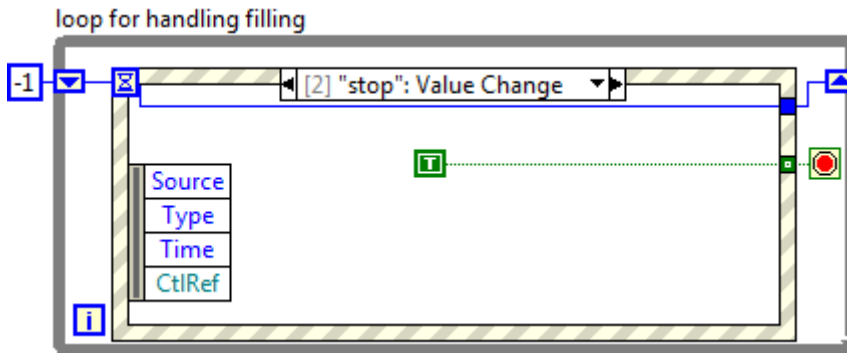
Erstellen Sie ein numerisches Anzeigeelement für den Tank sowie für die Endschalter voll und leer.



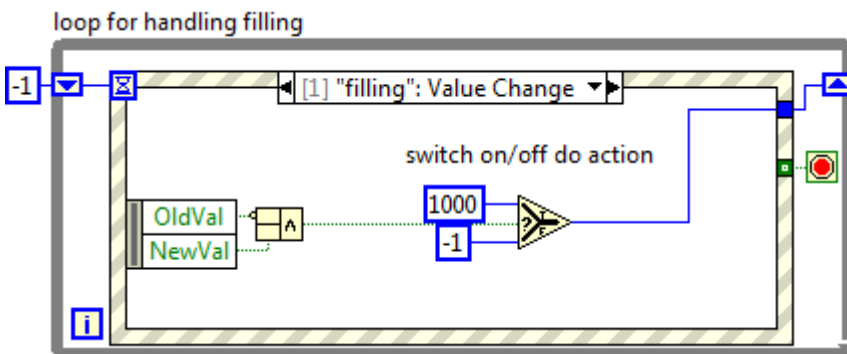
Der Wert des Tanks wird bei Programmstart auf 0 gesetzt.



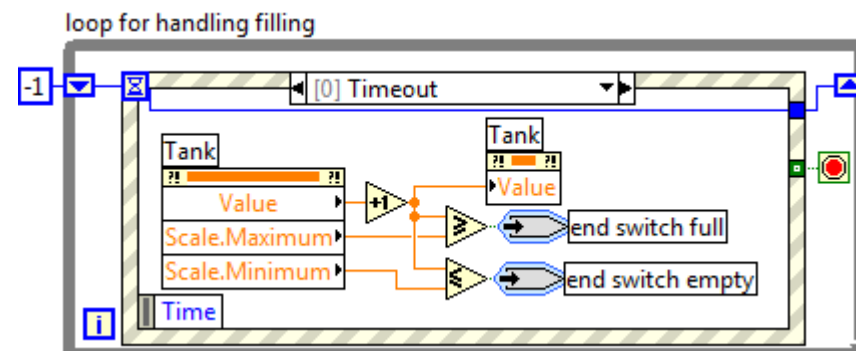
Wir wollen nun das Befüllen simulieren. Erzeugen Sie einen "Event Case" für die Simulation des Befüllens. Der "Event Case" soll mit „stop“ beendet werden.



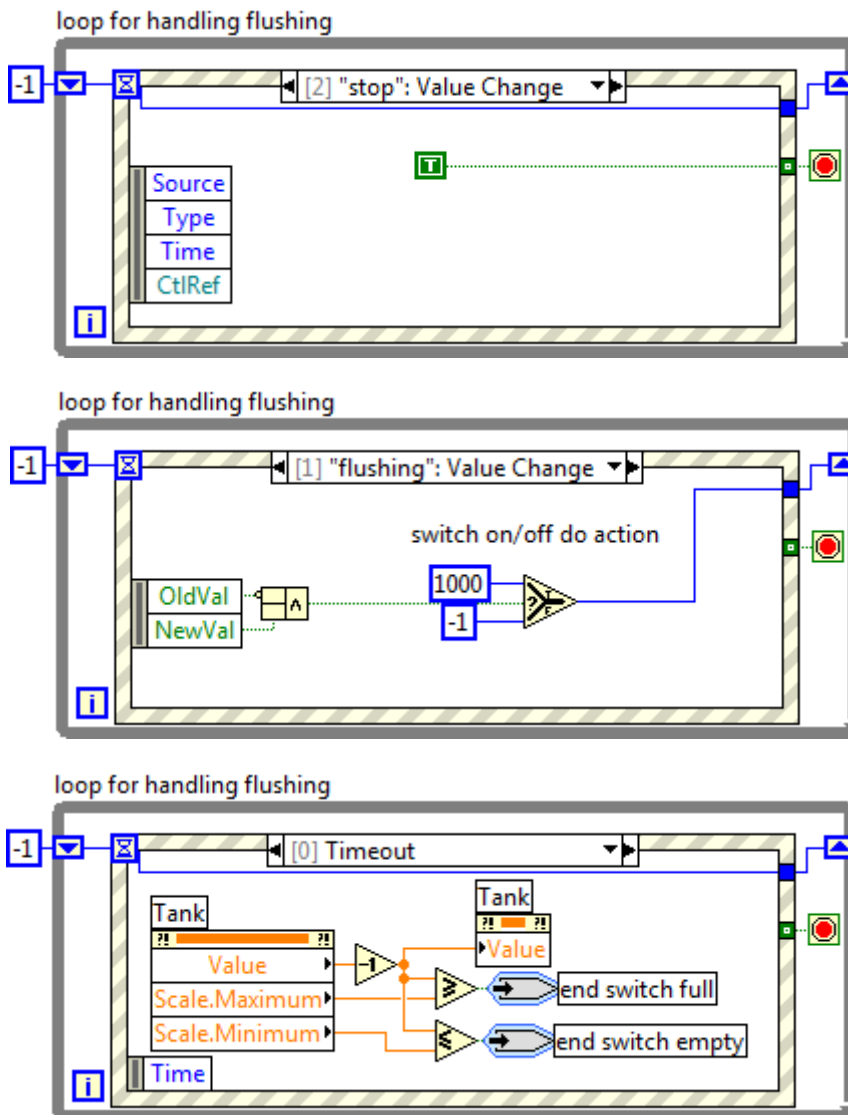
Wenn der Zustand „filling“ auf „True“ steht, dann wollen wir den Tankinhalt zyklisch inkrementieren. Dazu nutzen wir den Timeout Fall der Ereignisstruktur. Wir schalten bei Zustandsänderung des Zustandes „filling“ den Timeout Fall ein oder aus. Da wir bei der Erstellung des Zustandes „filling“ die Option „Signaling“ aktiviert haben, bekommen wir ein Event bei der Änderung des Zustandes „filling“ auf dem Anzeigeelement „filling“.



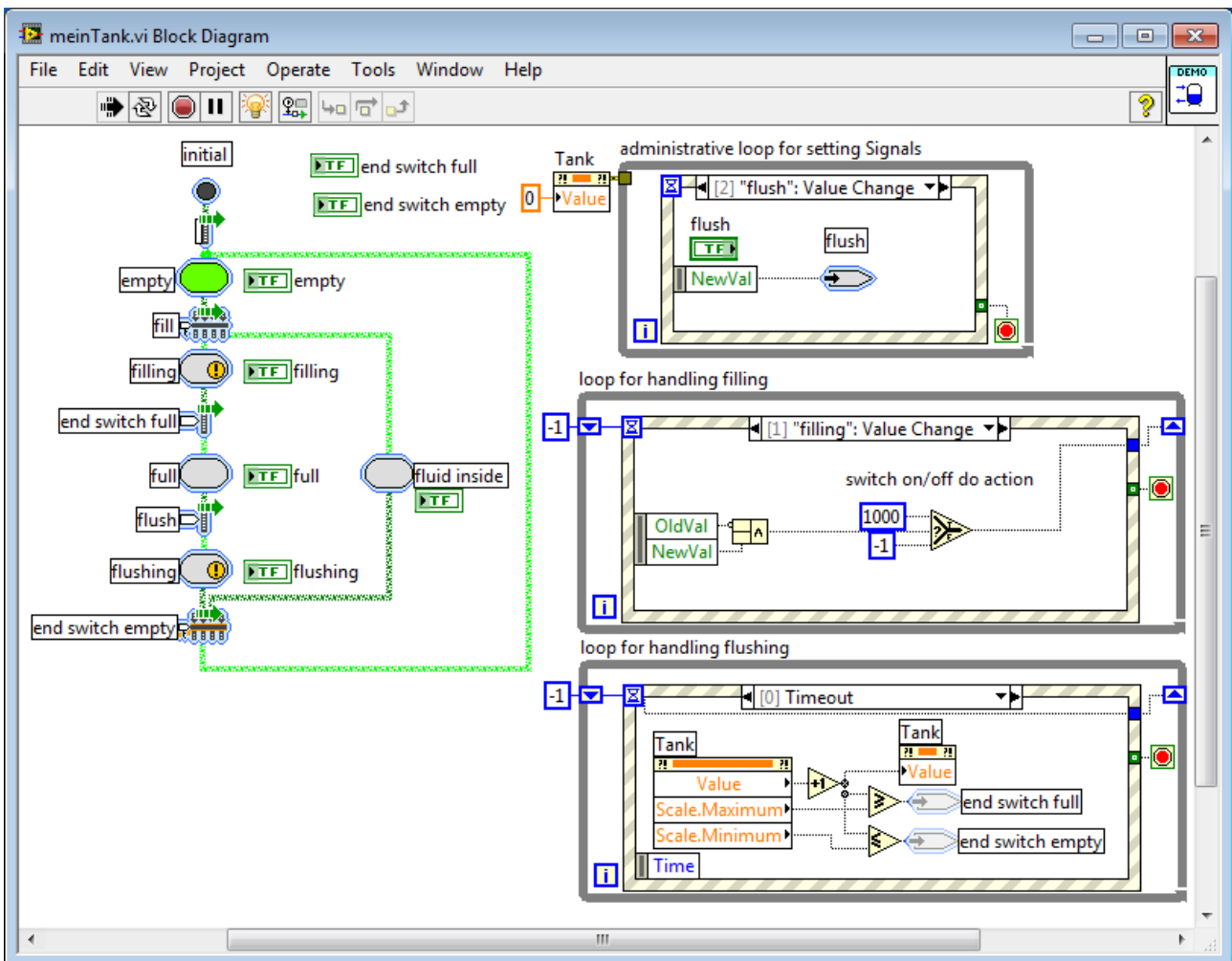
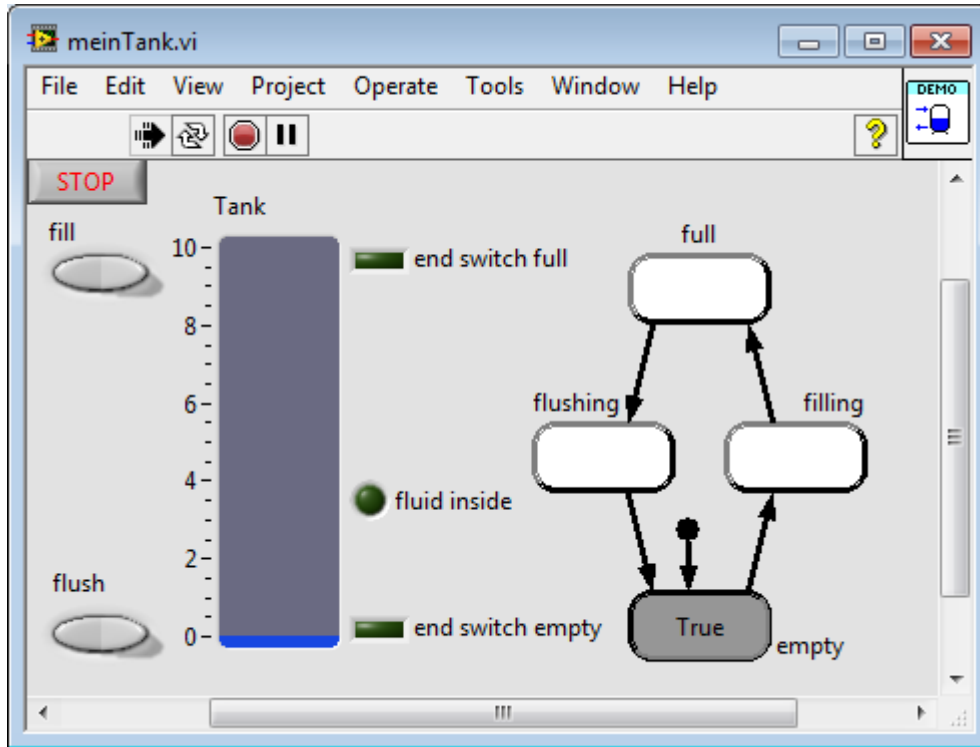
Die Timeout Aktion erhöht den Füllstand und generiert die Signale der Endschalter.



Für die Simulation des Entleerens erzeugen Sie eine weitere Ereignisstruktur analog der des Befüllens, jedoch mit dem Unterschied, dass dieses mal die Änderung „flushing“ die Timeout Aktion ein- bzw. ausschaltet und die Timeout Aktion den Füllstand verringert.

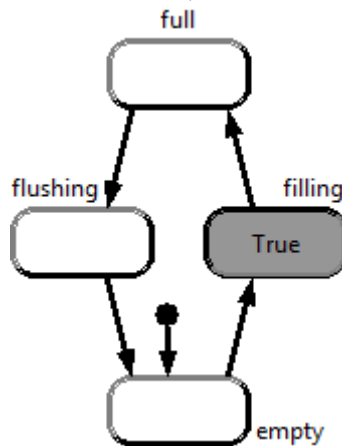


Sie können nun noch das Frontpanel etwas verschönern und Ihre Anwendung zum ersten mal starten.

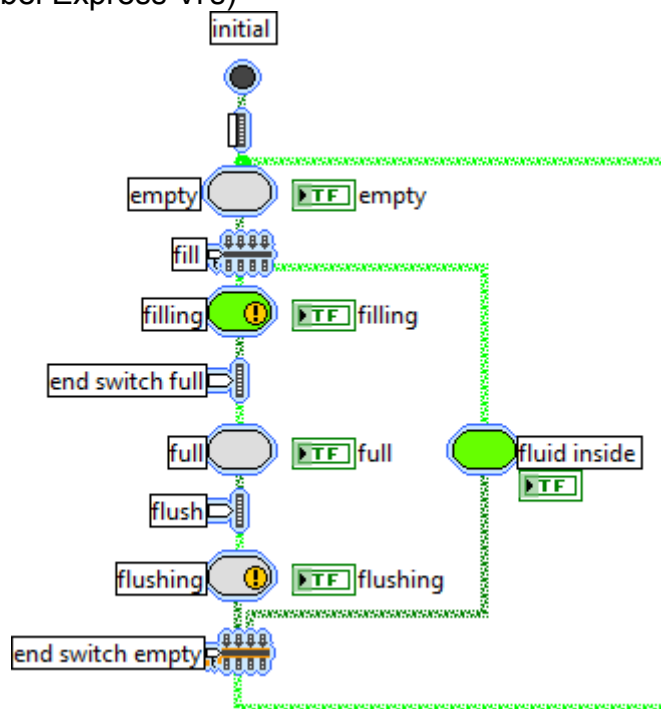


Beobachten des Netzes

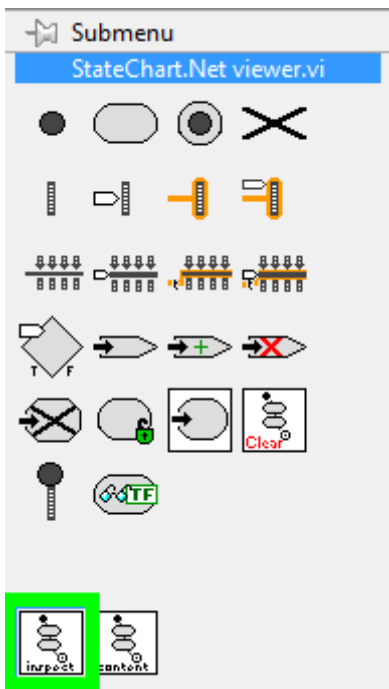
Alle Zustände, zu denen es boolesche Anzeigeelemente gibt, werden durch diese dargestellt. (In unserem Beispiel sind dies alle, außer der initial-Zustand.)



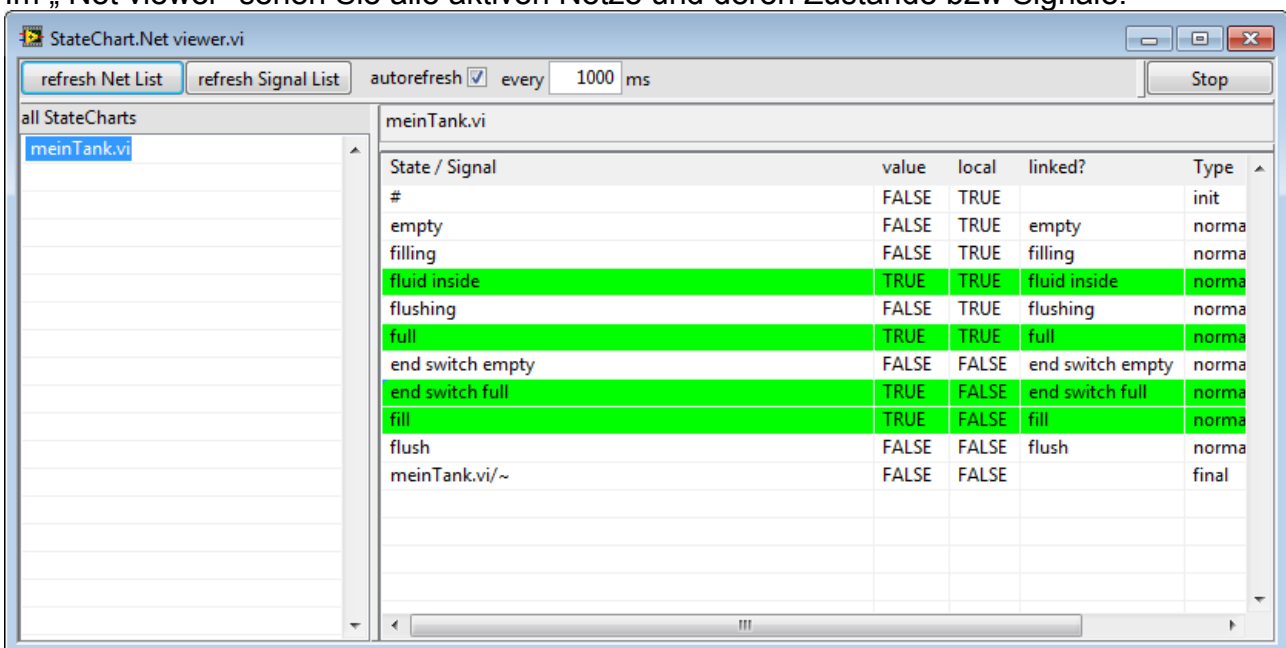
Zusätzlich werden im Blockdiagramm die Icons der aktiven Zustands-Vi's farblich gekennzeichnet (nur bei Express Vi's)



Sie können das Netz aber auch mit dem „StateChart.Net viewer“ beobachten. Dieser ist in der Palette StateChart – Submenu oder im Menü unter „Tools → open StateChart Netinspector“ zu finden.



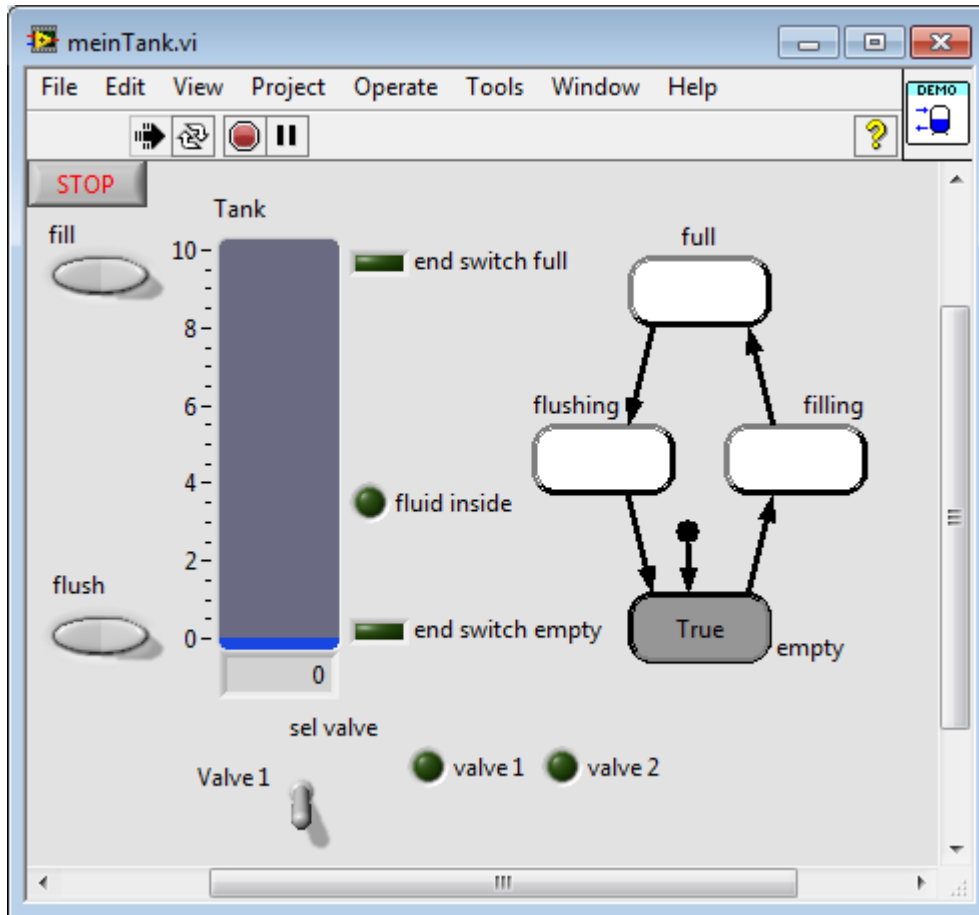
Im „Net viewer“ sehen Sie alle aktiven Netze und deren Zustände bzw Signale.



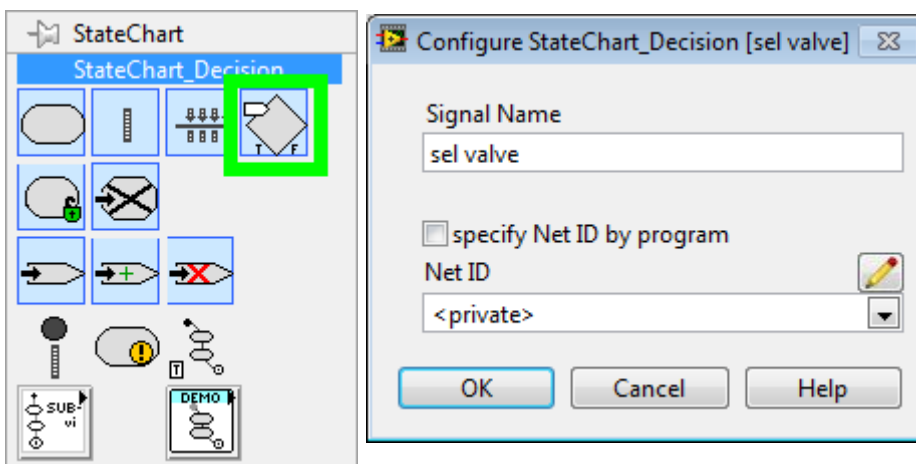
Für unser Netz sind dies die programmierten Zustände „full“, „empty“, „filling“, „flushing“, „fluid inside“ sowie der initial state („#“). Weiterhin wird immer ein finaler Zustand generiert, hier „meinTank.vi/~“. Zusätzlich gehören auch die Schaltbedingungen „fill“, „flush“, „end switch full“ und „end switch empty“ zum Netz.

Erweiterungen

Man könnte sich vorstellen, dass man zwei Auslassventile an den Tank anschließen möchte. Von den Ventilen soll immer nur eines aktiv werden.



Zur Implementierung dieses Verhaltens benötigen wir eine Entscheidung. Erstellen Sie ein Boolesches Bedienelement mit dem Namen „sel valve“ und platzieren die Funktion „Decision“ im Blockdiagramm. Konfigurieren Sie die Entscheidung wie nachfolgend abgebildet.



Erstellen Sie einen Fall in der Ereignisstruktur zum Setzen der Signale.

Erweitern Sie Ihr Netz um die Zustände „valve1“ und „valve 2“ sowie „?valve“ und verbinden Sie die Elemente.

