[Comments] 30-Mar-22

593-Using Combines in Netlinx to Take Control of Another System

Symptoms

A master-to-master system is setup where one master (e.g. the Control Room) sometimes needs to take control of another room (e.g. the Boardroom) or rooms. The control room system issues a **COMBINE_DEVICES** command, but never gets control of the other system.

Cause

A master cannot take control of the functionality in another master. Instead, the other master must grant control to the remote system. The **COMBINE_DEVICES** statement only has an effect in the master issuing the command. It is possible to combine local devices with devices that are on another system, but this has no direct effect on the other system.

Example 1:

In the following example, there are two masters. Master 1 is the Control Room master, and master 2 is the Boardroom master. All of the control is in the Boardroom master, so this system can run standalone. At times, the Control Room needs to combine itself with the Boardroom. The code below shows how this is done.

The Control Room code: Notice that this code consists of nothing more than a **DEFINE_DEVICE** section; the Boardroom code has all the functionality and will be responsible for granting control to the Control Room system. To make this easier, there is a separate device on the Control Room touch panel (129:1:0) that is just for the buttons that will combine and un-combine the Control Room with the Boardroom.

```
PROGRAM_NAME='Control Room'

(***********************************************************************
***********)

(* DEVICE NUMBER DEFINITIONS GO BELOW *)

(***********************************************************************
***********)
```

```
DEFINE_DEVICE

// The first device is for room control when combined.

dvControlRoom = 128:1:0

// The second device is for pushes to the Boardroom for requesting
access or relinquishing access to the room.

dvRoomSelect = 129:1:0

(***********************************************************************
***********)

(* END OF PROGRAM *)

(* DO NOT PUT ANY CODE BELOW THIS COMMENT *)

(***********************************************************************
***********)
```

The Boardroom code: See comments in the code below.

```
PROGRAM_NAME='Boardroom'

(***********************************************************************
************)

(* DEVICE NUMBER DEFINITIONS GO BELOW *)

(***********************************************************************
************)

DEFINE_DEVICE

dvTP = 128:1:0 // The local touch panel.

vTP = 33001:1:0 // The local virtual device for combines.

// The next devices are on the remote master - Control Room panel.
```

```
// The first device has general functionality. This will only work
when this device has been combined with the local virtual.

dvControlRoom = 128:1:1

// The second device only has the buttons for requesting that the
remote device be granted access or relinquish access.

// This device will never be combined with the local virtual.

dvRoomSelect = 129:1:1

(***********************************************************************
*************)

(* STARTUP CODE GOES BELOW *)

(***********************************************************************
*************)

DEFINE_START

// Local control only for this room at startup.

// Control Room does not have control yet.

COMBINE_DEVICES(vTP,dvTP)

(***********************************************************************
*************)

(* THE EVENTS GOES BELOW *)

(***********************************************************************
*************)

DEFINE_EVENT

// Control Room system is requesting that access to this room be
granted.

BUTTON_EVENT[dvRoomSelect,1]

{
```

```
PUSH:

{

// Release any previous combine.

UNCOMBINE_DEVICES(vTP)

// Combine remote device with the local virtual.

// Note that local device also still has control.

COMBINE_DEVICES(vTP,dvTP,dvControlRoom)

}

}

// Control Room system is requesting that access to this room be
terminated.

BUTTON_EVENT[dvRoomSelect,2]

{

PUSH:

{

// Release any previous combine.

UNCOMBINE_DEVICES(vTP)

// Combine local device only with the local virtual.

// Control Room will no longer have control.

COMBINE_DEVICES(vTP,dvTP)

}

}

// Boardroom control code goes below.
```

```
// These are pushes from the local virtual. If Control Room system
has been combined with virtual then it will have control also.

BUTTON_EVENT[vTP,1] // Do something in boardroom.

{

PUSH:

{

// Do something.

}

}

(***********************************************************************
*************)

(* END OF PROGRAM *)

(* DO NOT PUT ANY CODE BELOW THIS COMMENT *)

(***********************************************************************
*************)
```

Example 2:

In Example 2, you do not need to define the touch panel in both systems. Instead, you define a virtual device in both masters as described in TN435.

You may want to do something like this if you have many systems that will connect to one with a shared device (or devices) - you won't have to predefine all of the possible panels that may connect to the shared system.

• System 7121 (The Control Room) has the touch panel and a virtual device with system number 7122.

• System 7122 (The Boardroom) contains the controlled devices and the virtual device.

• System 7121 sends a message to system 7122, via the virtual device, telling the virtual device to combine with the touch panel in system 7122.

• The subroutines DPS_TO_STRING and 'STRING TO DEV', from TN461, are used to convert the DEV data to and from an ASCII string that can be sent / received by the virtual device.

Note: DO NOT combine the panel and virtual in both systems. This will cause problems.

```
Control Room Code:

PROGRAM_NAME='PanelSystem7121'

#include 'debug.axi' // see TN461

DEFINE_DEVICE

dvTP = 10001:1:0

vdvSys2 = 33000:1:7122

DEFINE_VARIABLE

integer btnCombinePanelWithSystem7122[] =

{

1, // combine

2 // un-combine

}

DEFINE_EVENT

BUTTON_EVENT[dvTP,btnCombinePanelWithSystem7122]

{

RELEASE: // always combine / un-combine on releases, not pushes

{

IF (GET_LAST(btnCombinePanelWithSystem7122) = 1)

SEND_COMMAND vdvSys2,"'combine ',DPS_TO_STRING(dvTP)"

ELSE
```

```
SEND_COMMAND vdvSys2,'uncombine'

}

}

// end
```

The Boardroom code:

```
PROGRAM_NAME='DeviceSystem7122'

#include 'debug.axi' // see TN461

DEFINE_DEVICE

dvRS232 = 5001:1:0

dvRelay = 5001:7:0

vdvTP = 33000:1:0

DEFINE_VARIABLE

INTEGER nBtns[] = {1,2,3,4,5,6,7,8,9,10,11,12}

DEFINE_EVENT

BUTTON_EVENT[vdvTP,nBtns] // functional test

{

PUSH:

{

SEND_STRING dvRS232,"ITOA(nBtns[GET_LAST(nBtns)])"

TO[dvRelay,GET_LAST(nBtns)]

TO[BUTTON.INPUT] // feedback statement for debugging
```

```
}
}
DATA_EVENT[vdvTP]
{
COMMAND: // combine here
{
STACK_VAR DEV dvCombine
SELECT
{
ACTIVE (FIND_STRING(LOWER_STRING(DATA.TEXT),'uncombine',1)):
{
UNCOMBINE_DEVICES(vdvTP)
}
ACTIVE (FIND_STRING(LOWER_STRING(DATA.TEXT),'combine ',1)):
{
// the assumes message form is 'combine <dv>:<prt>:<sys>'
CALL 'STRING TO DEV' (DATA.TEXT,dvCombine)

UNCOMBINE_DEVICES(vdvTP)
COMBINE_DEVICES(vdvTP,dvCombine)
}
}
}
```

```
}

// end
```

Example 2 code and touch panel attached.