Application Bulletin

Date: Aug. 30, 2007 Re: Hot-Backup OCS System

Hot-Backup Operator Control Station (OCS) for High Availability Systems

Learn how to use OCS in a "hot backup configuration" to maintain operations should one system fail, or need to be shut down for updates (especially important for industries like food and pharmaceutical).

Application Overview

Any two Operator Control Stations (OCS) in combination with SmartStix™ I/O can be set up to be a High Availability System. In this case, High Availability is defined as the ability of the Slave controller to take over control of the I/O in the event of the 'Master' controller going off line for some reason and then the Slave giving back control when the Master is up and running again.

In the example discussed below, an XLe OCS is programmed to be the Master using Network ID 1 and a QX451 using Network ID 253 is programmed to be the Slave. (The QX451 could also be programmed to perform other tasks in addition to being the Slave controller). They are controlling two SmartStix I/O blocks, a 32 point output block at ID 10 and a 32 point input block at ID 11. (The actual Network ID numbers are not significant except to this program.)

The system looks something like this:

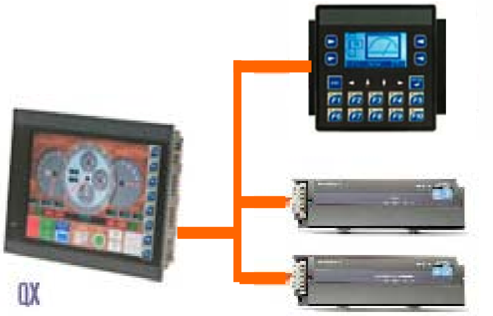


Figure 1 – High Availability System
XLe as Master, QX as Slave, and SmartStix I/O

How does the program work?

Programming Note: *Normally, when using SmartStix I/O in a system, the I/O is just configured and there is no ladder programming required. (I/O Configuration/Network I/O) When the I/O is configured, all input and output communication is automatic and the data just appears in the I/O tables. However, in this application, communication with the outputs must be under program control and therefore must be programmed with Net-Put Remote I/O functions. Outputs can only be controlled from one source. Also, the SmartStix Input block must receive at least one Net-Put*

Remote I/O command in order to function. This command is included in both the Master and Slave programs. Therefore ***do NOT configure either of these SmartStix I/O blocks in the I/O Configuration****. Other I/O, including other SmartStix I/O may be configured in the normal way.*

Duplicate control logic is placed in each controller. (Even though the control logic is the same in each controller, some changes may need to be made to the programs to adapt them to different screen sizes and types if the OCS models are different). Then appropriate changes are made in each program to accomplish the following: The Master sends a Heartbeat over the Network to the Slave. Both the Slave and the Master always monitor the SmartStix inputs. Only the Master controls the SmartStix outputs unless the Slave sees the Master Heartbeat disappear. Then the Slave starts controlling the outputs. When the Slave sees the Master Heartbeat return, the Slave maintains control of the outputs for a short time while the Master logic is updated and then the Master takes control of the outputs again.

Addressing Note: *Pay strict attention to the addressing requirements of the OCS models being used. Some models have internal I/O and therefore any Remote I/O must not overlap I/O addresses. For instance, the XL series (XLe, XLt, etc.) reserve the first 32 %I and the first 32 %Q so the first remote %I or %Q would be 33 or above. (This program example uses %Q0001 for the remote outputs and therefore would not directly work with the XLe OCS.)*

The concept is really quite simple.

1. The Master is programmed to control the I/O as required by the application.
2. A copy of the Master program is moved to the Slave and any necessary changes are made to adapt it to the different type controller including changing the Hardware Configuration.
3. Additions to the Master program:
    a. When the Master first starts, or the network is restored, the PULL_TMR timer is started.
    b. Status is pulled from the Slave.
    c. A copy of the current outputs is pulled from the Slave.
    d. Additions are made to the control logic to bring it in sync with the Slave status.
    e. After the PULL_TMR timer times out, outputs are sent to the SmartStix output block, and status and a copy of the outputs are sent to the Slave.
    f. A Heartbeat signal is made available to the Slave controller.
4. Additions to the Slave program:
    a. Monitor the Heartbeat from the Master.
    b. As long as there is a Heartbeat pull the status and a copy of the outputs from the Master.
    c. Additions are made to the control logic to bring it in sync with the Master status.
    d. If the Master Heartbeat stops, begin controlling the outputs.
    e. If the Heartbeat stops and then restarts, start the NET_PULL timer.
    f. Send Slave status and a copy of the current outputs to the Master.
    g. Stop controlling the outputs when the Master Heartbeat returns.

Setting up the Hardware

There are no special hardware setup requirements for the High Availability System. Just make sure each unit has a unique Network ID, and the CAN wiring is complete and terminated a both ends.

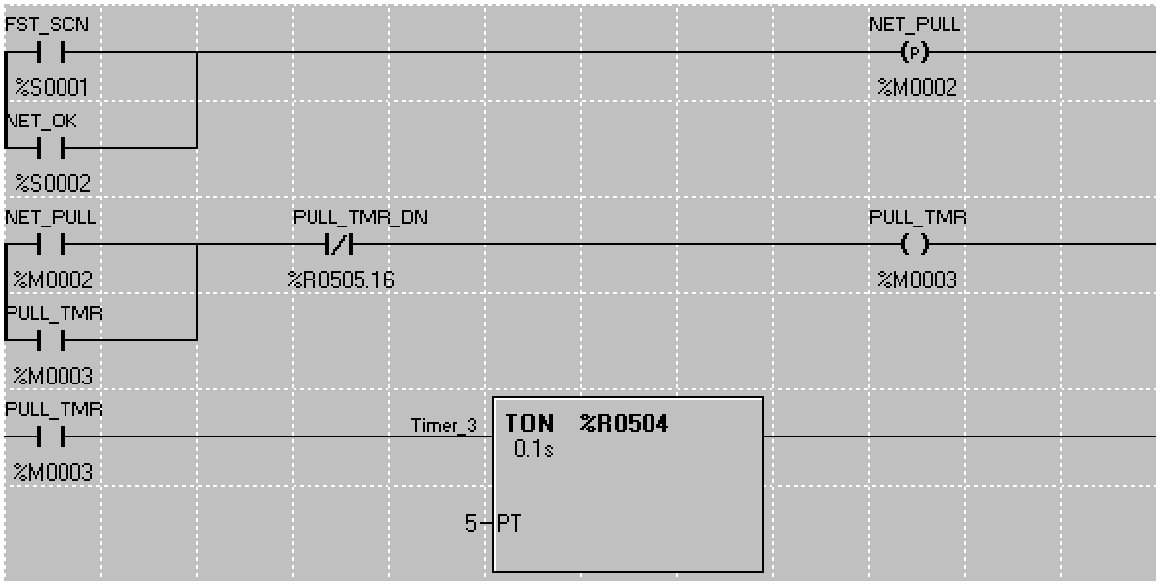
It would be prudent to supply power to the CAN network from a source other than the Master or Slave power supplies so the network can still operate when either controller is powered down.

Cscape setup

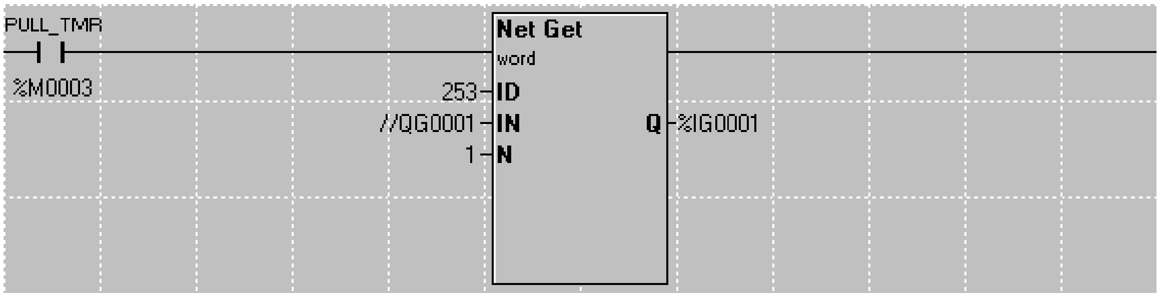
for additional Cscape information, please click here

Following are examples of the *Additions to the Master program* listed above:

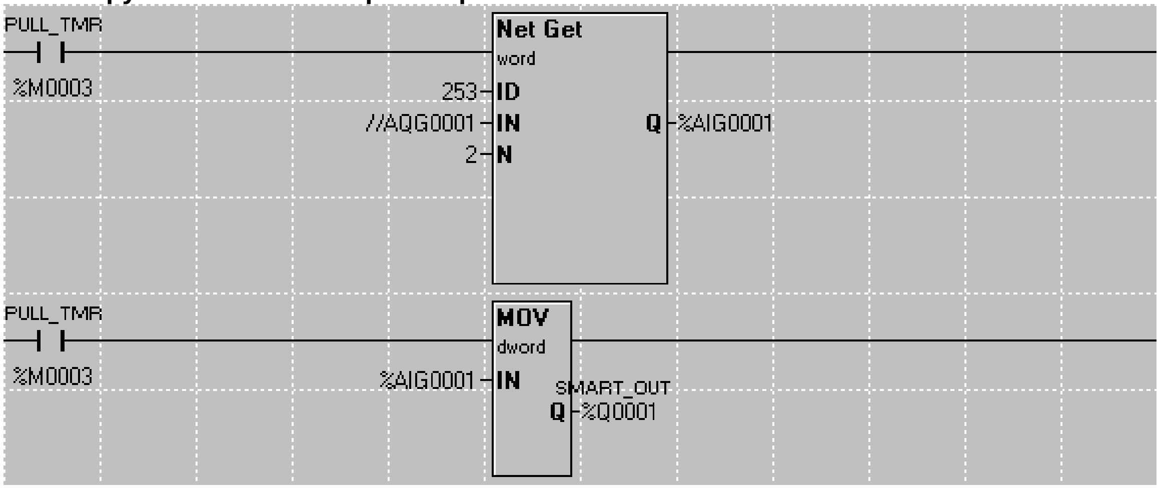
3a. When the Master first starts, or the network is restored, the PULL_TMR timer is started.



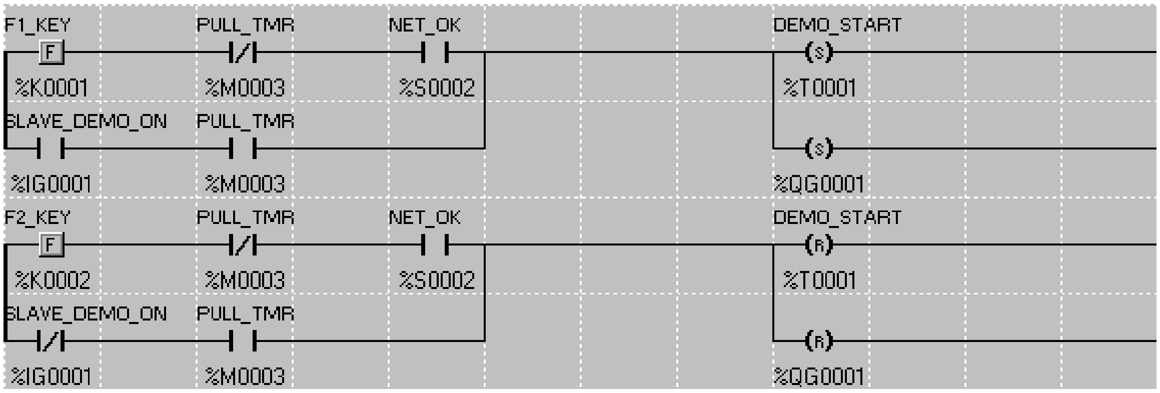
3b. Status is pulled from the Slave.



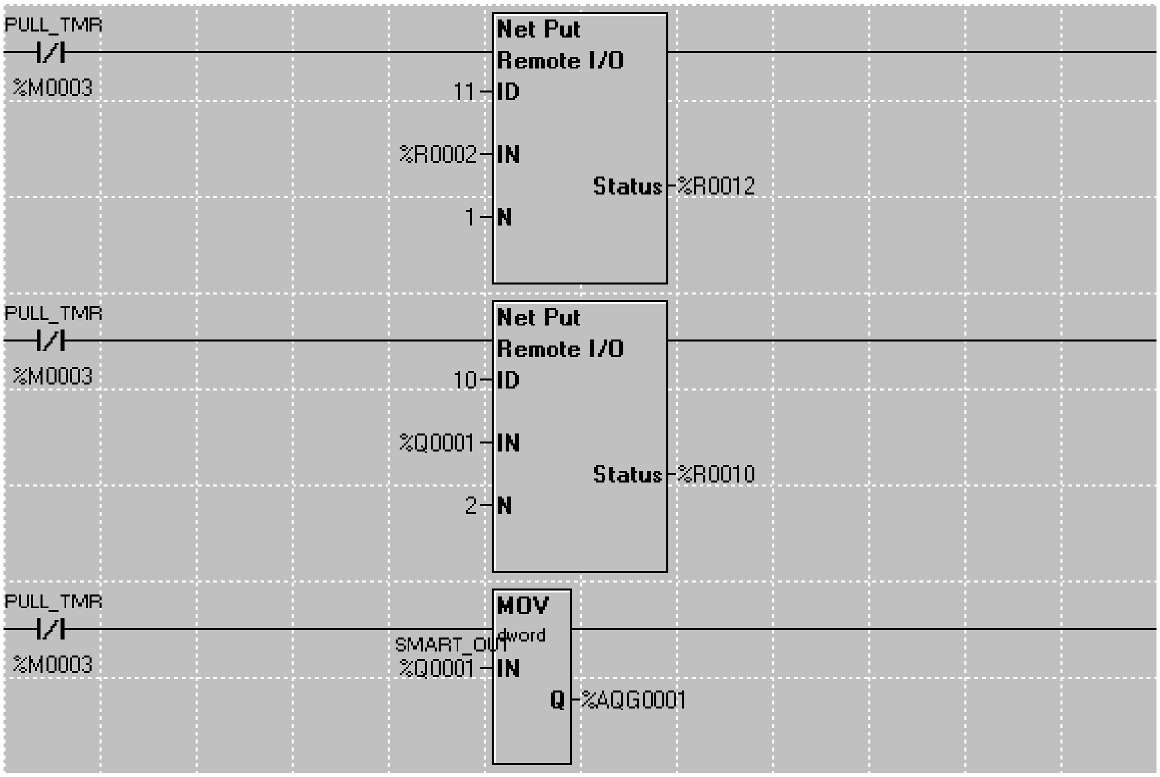
3c. A copy of the current outputs is pulled from the Slave.

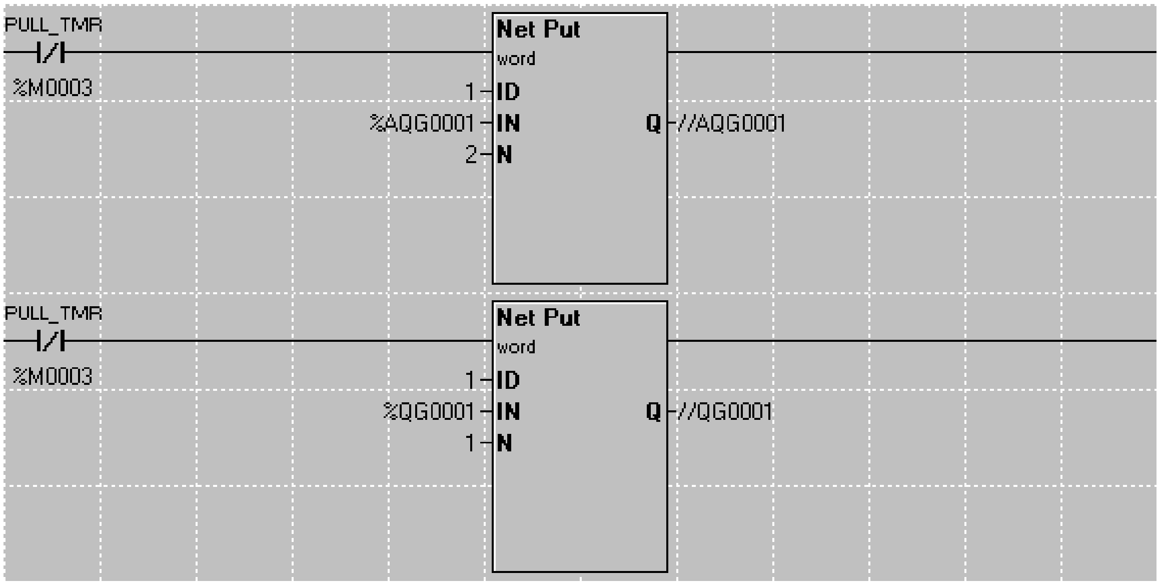


3d. Additions are made to the control logic to bring it in sync with the Slave status.

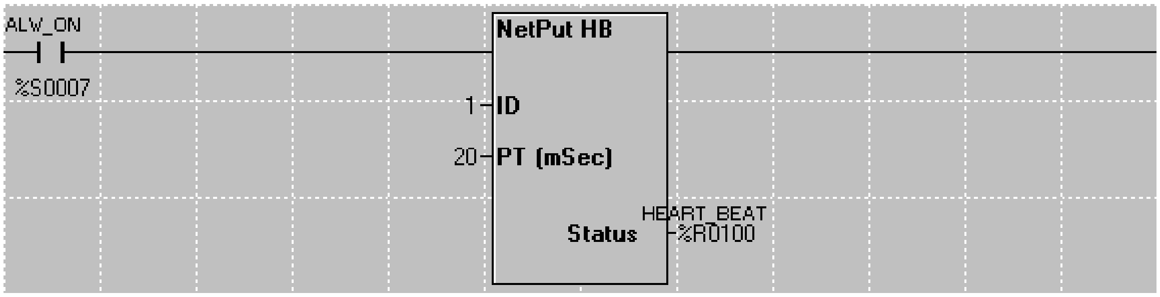


3e. After the PULL_TMR timer times out, outputs are sent to the SmartStix output block, and status and a copy of the outputs are sent to the Slave.



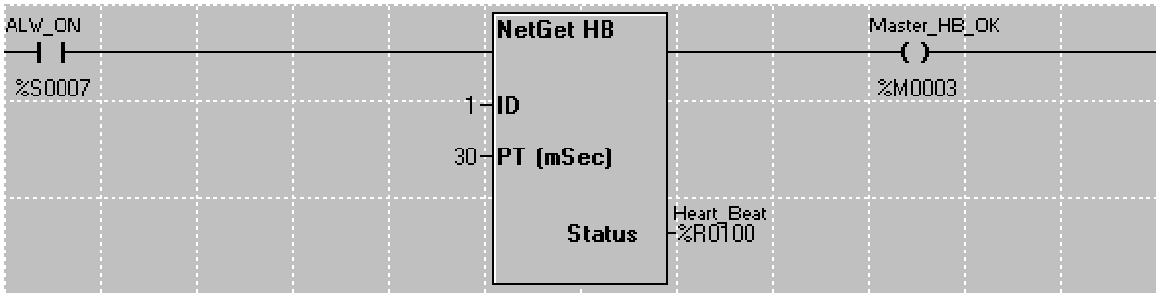


3f. A Heartbeat signal is made available to the Slave controller.

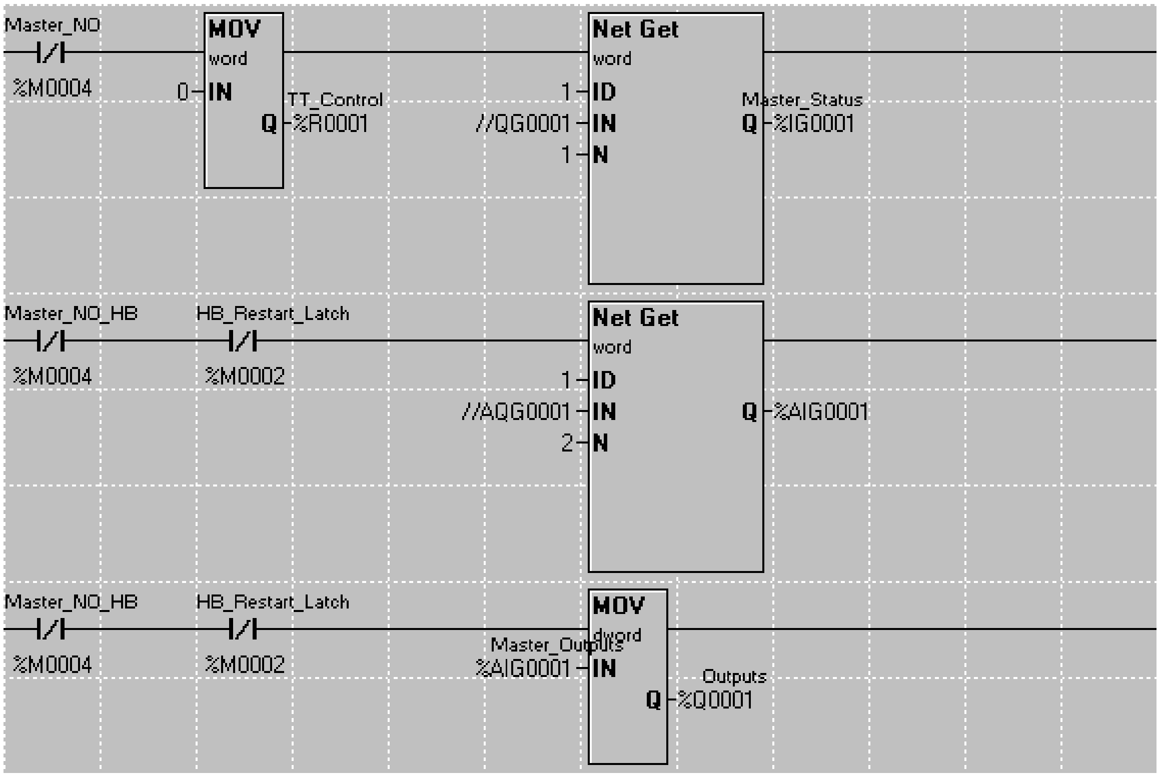


Following are examples of the *Additions to the Slave program* listed above:

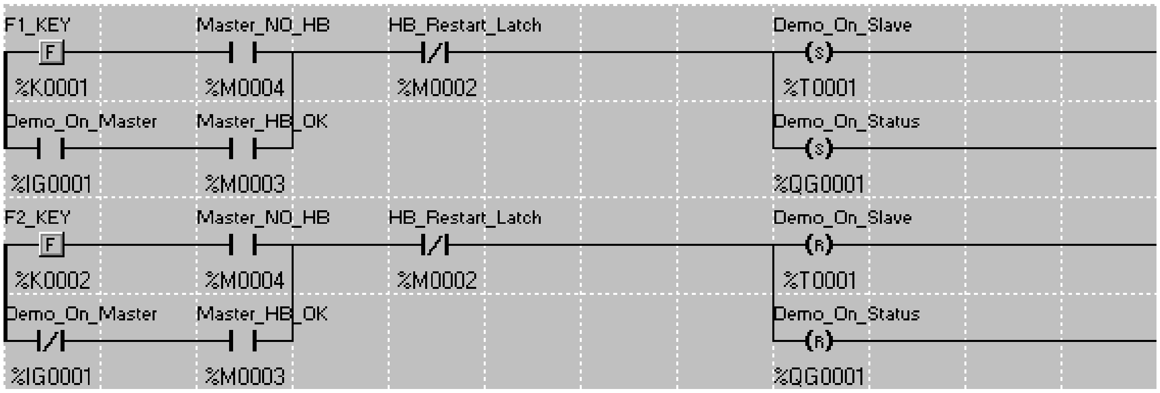
4a. Monitor the Heartbeat from the Master.



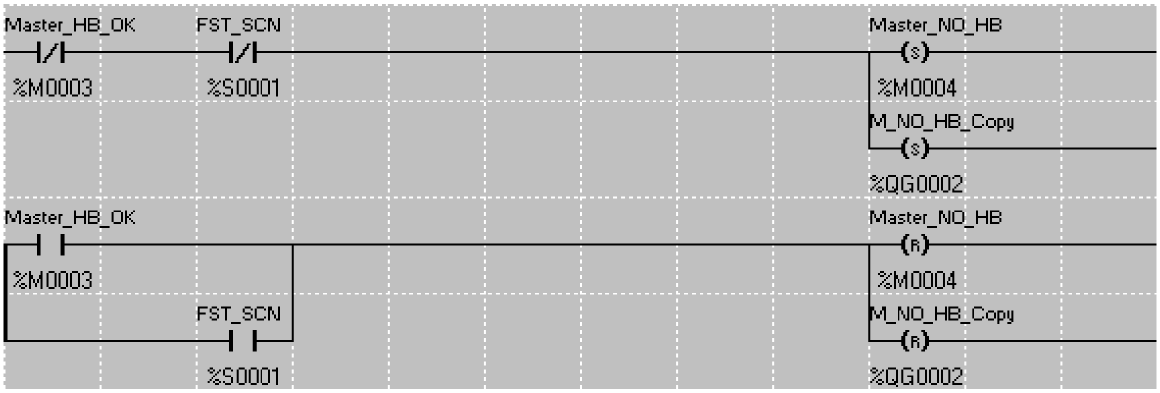
4b. As long as there is a Heartbeat pull the status and a copy of the outputs from the Master.

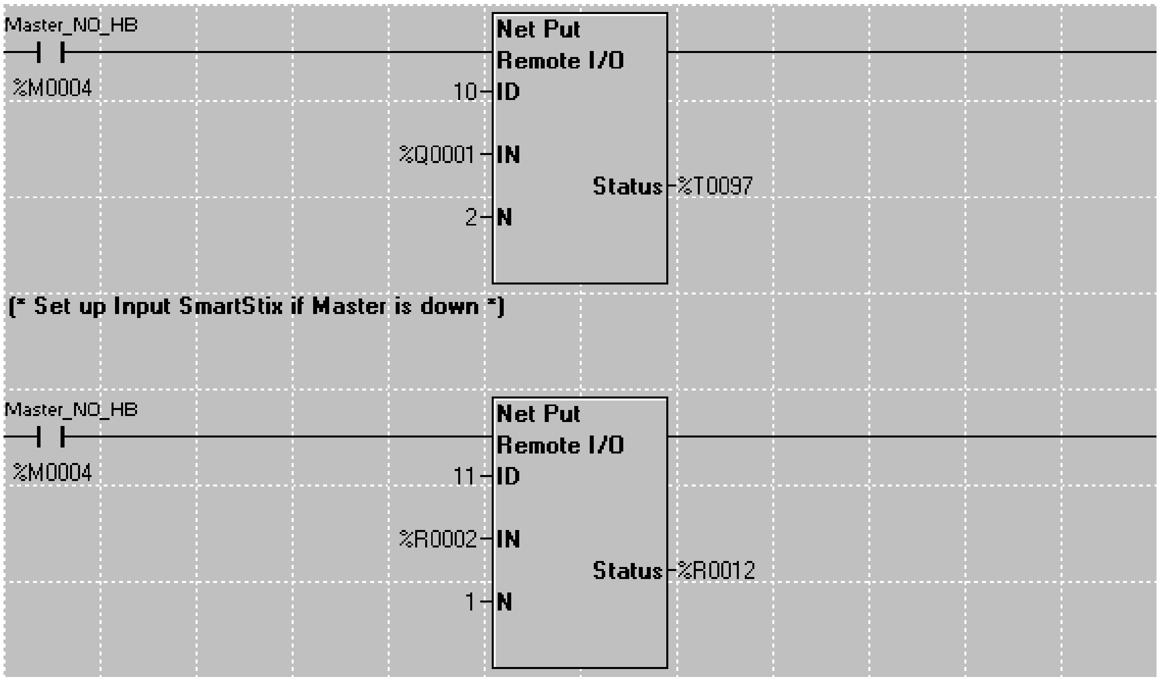


4c. Additions are made to the control logic to bring it in sync with the Master status.



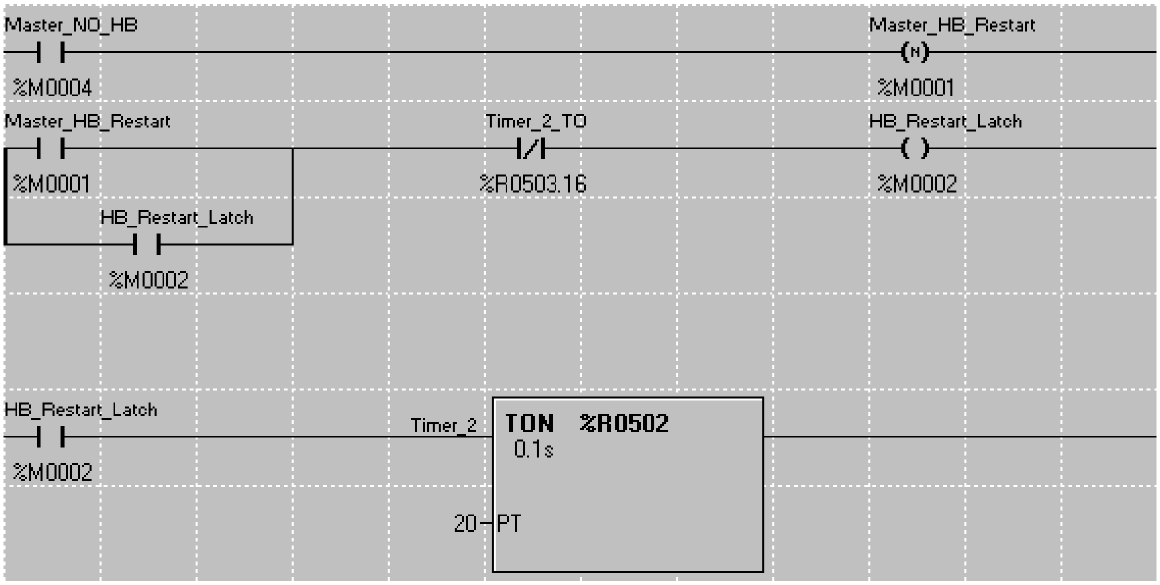
4d. If the Master Heartbeat stops, begin controlling the outputs.



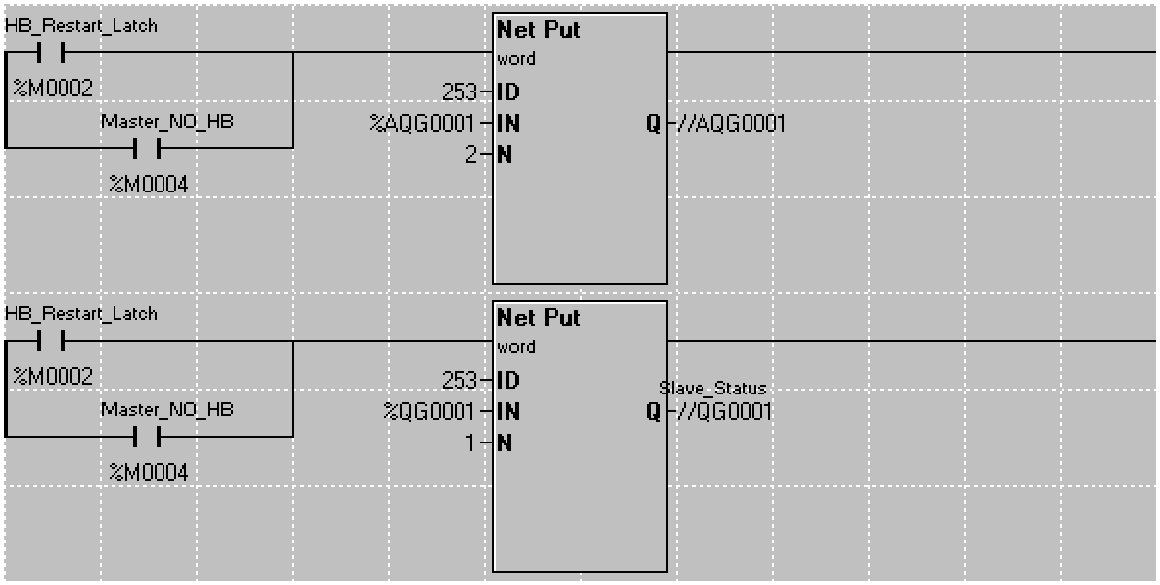


(* Set up Input SmartStix if Master is down *)

4e. If the Heartbeat stops and then restarts, start the NET_PULL timer.



4f. Send Slave status and a copy of the current outputs to the Master.



4g. Stop controlling the outputs when the Master Heartbeat returns.

Part of 4d.

Horner's Tech Support Dept. wrote this Application Note. If you have questions, please call our Tech Support Dept. **toll-free, 1-877-665-5666, press 3 (in the USA only) or 1-317-916-4274**. You can also e-mail techsppt@heapg.com.

Horner APG, LLC
59 South State Avenue – Indianapolis, IN 46201 – Ph: 317-916-4274 – Fax: 317-916-4280
For additional information, please visit: www.HornerOCS.com