ASI Controls Technical Note Series

ASIC/1-8655 I/O Slave Module

Affects: ASIC/1-8040, -7040, -8540, -7540

Date: 25 September 2007

Note No. TN-037

Problem: You have used up all the available points on your ASIC/2, but you still need more.

Solution: Use an ASIC/1-8655 as a slave module.

When designing a project, you may come across a situation where you need more points that you can land on an ASIC/2, but you don't want to add another ASIC/2 to the project. What do you do? You can add an ASIC/1-8655 to the Local Bus of that ASIC/2, and make it an Input/Output slave module.

Options

But why do this? Isn't it better to just add another ASIC/2? Well, as with everything, it all depends on what you are trying to accomplish. Here are your main options.

- Option 1: If using an ASIC/2-8540, upgrade to an ASIC/7540.
- Option 2: If already using an ASIC/2-7540, add another ASIC/2 to the project, and transfer data using Remote Points on the System Bus.
- Option 3: Use open and available points on existing controllers, and transfer data on the appropriate communication bus.
- Option 4: Add an ASIC/1-8655, and use Polling and Broadcasting to transfer data on the ASIC/2 local bus.

Option 1-Upgrade to ASIC/2-7540

Upside

- Adding more physical points means adding more programming to utilize them. This option keeps all of that additional programming in the same controller.

Downside

- Only available if you were planning on using an ASIC/2-8540.
- Upgrading to an ASIC/2-7540 will add to your hardware costs.
- The ASIC/2-7540 is larger, causing you to redesign your enclosure and power supply needs.

Option 2-Add another ASIC/2

Upside

- The addition of a second ASIC/2 adds another processor to the project, giving you more computing power to meet your Sequence of Operation.
- Adds another Local Bus to the project.

Downside

- Adding another controller will raise your hardware costs.
- The additional controller will require more enclosure space, and you will need to provide additional power supplies, terminal strips, etc.
- These two controllers now become dependent on information from each other. Depending on the Sequence of Operation, you will need to carefully plan for how this data transfer will occur.

Option 3-Utilize open points on existing controllers

Upside

- Doesn't add to your hardware costs.

Downside

- The existing controllers may not be close to the physical points that you need to add, which raises your labor and material costs to run wires over a distance.
- You will need to carefully plan for how you will transfer data between the controller that is wired to the physical point and the controller that will utilize it.

Option 4-Add an ASIC/1-8655

Upside

- Hardware costs are lower than adding another ASIC/2.
- Footprint and power requirements are smaller than an ASIC/2.

Downside

- This option can not be used if you were planning to use the local bus for Modbus or an ePAD device.
- This option may not be feasible if you were planning to already have a lot of polling and broadcasting occurring on the Local Bus to other ASIC/1's.
- Some project specifications forbid the use of expanders to get more points.

If you have weighed all of your options, and you have decided on Option 4, then read on. We will now deal with how to implement this option.

Implementation

The first step is to determine what the inputs and outputs will be used for. Don't forget to modify the pull-up resistors accordingly to match the type of input that you are using. While we are on the subject, it is not advisable to plan to use the analog outputs. For most control sequences, the response time is not quick enough to properly control the device, so we will not cover that in this document.

Secondly, the ASIC/1-8655 is an application-specific controller. It runs pre-programmed Sequences of Operation that are determined by setting its Personality. You must set the Personality to 0 (or None) to use the ASIC/1-8655 as a slave board. This will cause the physical inputs and outputs to be disconnected from any control sequences, making it a "dumb" device. Now you can be assured that all automatic operation inside the ASIC/1-8655 will cease to affect the outputs.

The third step would be to add the necessary programming to the ASIC/2. You will have to make sure that you have enough instances of the Poll List, Poll Manager, Monitor and Broadcast objects to meet your sequences. We will use the Poll Manager to get data from the ASIC/1-8655, and put it into the Poll List. The Monitor Object will be used to assemble and collect all of the input data. The Broadcast Object will be used to control the 8 binary outputs on the ASIC/1-8655.

How to use this example project

This example project includes two .asi files, and two .fgp files. The best way to use the project is to load the A1-8655-18655.asi file into an ASIC/1-8655. This configuration is already set up to use the ASIC/1-8655 as a slave module. Once loaded, set the Address (**Table 1,1-2: *A1_DeviceAddress***) to 18655, and the Group Address (**Table 1,9: *A1_GroupAddress***) to 0. The controller will be set up to use inputs 1 through 6 for temperature inputs, and inputs 7 and 8 as binary inputs. You will have to modify the pull-up resistors accordingly, but for the purposes of testing, the default ones will work fine.

It is very important to set the Group Address (**Table 1,9: *A1_GroupAddress***) to a unique value, like 0. If you have any other ASIC/1's on the local bus, you will want to make sure that the slave module does not respond to Broadcast messages intended for these controllers. To be sure, it must have a unique Group Address.

You will also need an ASIC/2 to complete testing the configuration. You can do one of two things. Load the A2-7540-7540.asi file into your controller (ASIC/2-7540 or ASIC/2-8540), or import the two Function Groups that are in the project files into your existing ASIC/2 controller configuration.

Reference Manuals you may need to complete this test project

All of the information used in this example can be found in the manuals that are listed below.

- Visual Expert 3.0 User Manual, DOC-1545. This manual will give more detail on how to use Function Groups in controller configurations.
- ASI Controls Object Definitions, DOC-1400. This manual gives descriptions of all the Objects in an ASIC/2. You can use this manual for a more in-depth explanation of the Objects we are using in this example, which are the Poll List, Poll Manager, Broadcast and Monitor Objects.
- ASIC/1-8655 Engineering Guide, DOC-1568. This is the source to get detailed information on the Tables in an ASIC/1-8655, and the Commands used by the Broadcast and Poll Manager Objects. Check out the Appendix for information on addressing the ASIC/1-8655.
- ASIC/1-8655 Installation Manual, DOC-1592. Check out this manual for information on configuring input pull-up resistors.

Overview

Here is the overview of what happens when this all comes together.

For the Inputs:

- The Poll Managers look at their Poll Group Authority (**MGR-XX-07-LB_ONLY, *MGR_AuthorityCode1***), and looks for Poll List instances with the same value.
- The Poll Manager then asks the addresses that have that Poll Group Authority for the information that they are configured to get, and puts it into those Poll List instances, in the proper place.
- Then the Monitor object is used to assemble the separate Lo and Hi Bytes of the Word value together into a single Word value.

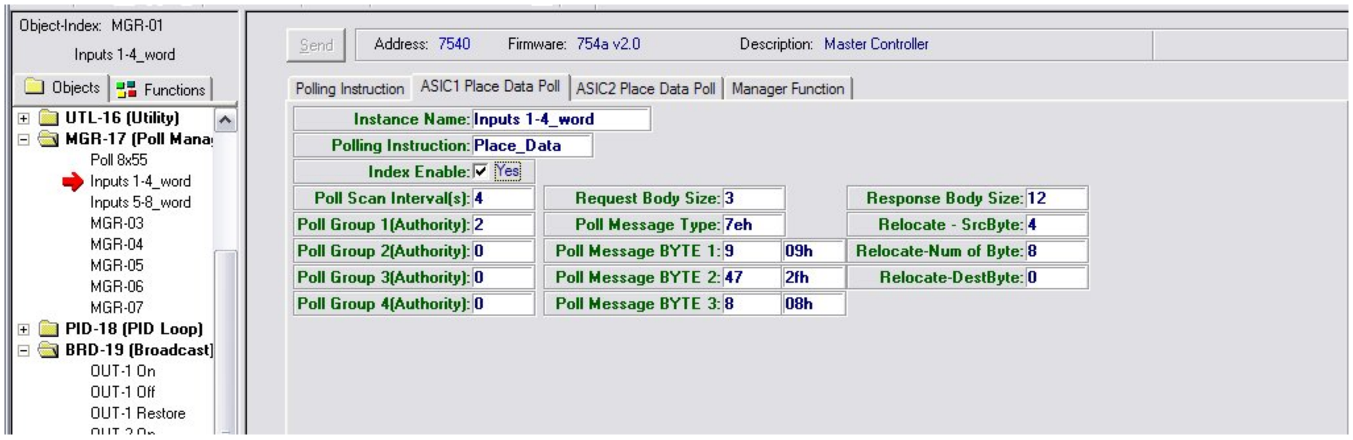
For the Outputs:

- The Broadcast instances have been set up to use Message Type 20h (**BRD-XX-17-LB_ONLY, *BRD_MessageType***), which will override the binary outputs on the ASIC/1.
- The Broadcast instance with the Static Message Byte 1 (**BRD-XX-18-LB_ONLY, *BRD_StaticMsgByte1***) value of 1h will override the Output On.
- The Broadcast instance with the Static Message Byte 1 (**BRD-XX-18-LB_ONLY, *BRD_StaticMsgByte1***) value of 2h will override the Output Off.

- The two Broadcast instances (one for On, one for Off) have been tied together, so that only a single Trigger instance is needed, connected to the Trigger Handle of the Broadcast instance used to send the override On message.

Working with the Inputs

If you would like to know the full version of what is happening, read on. I will breakdown what is happening using the Poll Manager instance named "Inputs 1-4_word" as shown in *Figure 1*.



Address: 7540 Firmware: 754a v2.0 Description: Master Controller

Instance Name: Inputs 1-4_word
Polling Instruction: Place_Data
Index Enable: ✓ Yes

Poll Scan Interval(s):	4	Request Body Size:	3		Response Body Size: 12
Poll Group 1(Authority):	2	Poll Message Type:	7eh		Relocate - SrcByte: 4
Poll Group 2(Authority):	0	Poll Message BYTE 1:	9	09h	Relocate-Num of Byte: 8
Poll Group 3(Authority):	0	Poll Message BYTE 2:	47	2fh	Relocate-DestByte: 0
Poll Group 4(Authority):	0	Poll Message BYTE 3:	8	08h	

Figure 1.

- This Poll Manager is configured to look for Poll List instances that are only using a Poll Group Authority (**MGR-XX-07-LB_ONLY, *MGR_AuthorityCode1***) value of 2. This means that other Poll List instances will not be used by this Poll Manager, they will be left alone.

- It has a unique Scan Interval (**MGR-XX-09-LB_ONLY, *MGR_PollrScanInterval***) value of 4. The Scan Interval determines how often the Poll Manager will execute its Polling Instruction. A lower number means more often, a higher number means less frequently. Remember, it is very important that each Poll Manager that is executing a Polling Instruction has to have a unique Scan Interval.

- It has a Polling Instruction (**MGR-XX-06-LB_LSN, *MGR_PollingInstruction***) of *Place_Data*. This tells the Poll Manager that a custom Polling Instruction will be used. Two tabs will appear, one marked *ASIC/1 Place Date Poll*, and one marked *ASIC/2 Place Data Poll*. We will be entering data on the *ASIC/1* tab.

- The Request Body Size (**MGR-XX-14-LB_ONLY, *MGR_RequestBodySize***) refers to how many Bytes in the body of the Poll Message. This value will always be 3 for an ASIC/1 *Place_Data* Instruction.

- The Message Type (**MGR-XX-09-HB_ONLY, *MGR_PollMessageType***) is 7eh, which is a request to get specific table data from the ASIC/1 controller.

- Poll Message Byte 1 (**MGR-XX-10-LB_ONLY, *MGR_PollMessageBYTE1***) is used to specify which Table in the ASIC/1 we need to access. In this case we want Table 9, which is where RAM Input Values are held.

- Poll Message Byte 2 (**MGR-XX-10-HB_ONLY, *MGR_PollMessageBYTE2***) is used to specify which Entry we want to start with. We want to start with Entry 47, which is the first Byte used to hold the Word value for Input 1.

- Poll Message Byte 3 (**MGR-XX-11-LB_ONLY, *MGR_PollMessageBYTE3***) tells the Poll Manager how many Bytes to bring back; starting with the Byte defined in Poll Message Byte 2, in this case 8 Bytes. This means the Poll Manager will get and bring back 8 Bytes of information: Table 9, Entry 47, 48, 49, 50, 51, 52, 53 and 54. If you look at the ASIC/1 Engineering Guide, you can see that this will give us the Word values for Input 1 through Input 4.

- The Response Body Size (**MGR-XX-14-HB_ONLY, *MGR_ResponseBodySize***) tells us how many total bytes of data we are bringing back. This total is derived from the 8 actual Bytes of data we are bringing back, and 4 more Bytes that are the rest of the message. If we were bringing back just 4 Bytes of data, the value would be 8, and so on.

- The Relocate Source Byte (**MGR-XX-06-HB_LSN, *MGR_Relocate-SrcByte***) indicates which Byte holds the first piece of actual data we need. This is always Byte 4 for a 7eh message type.

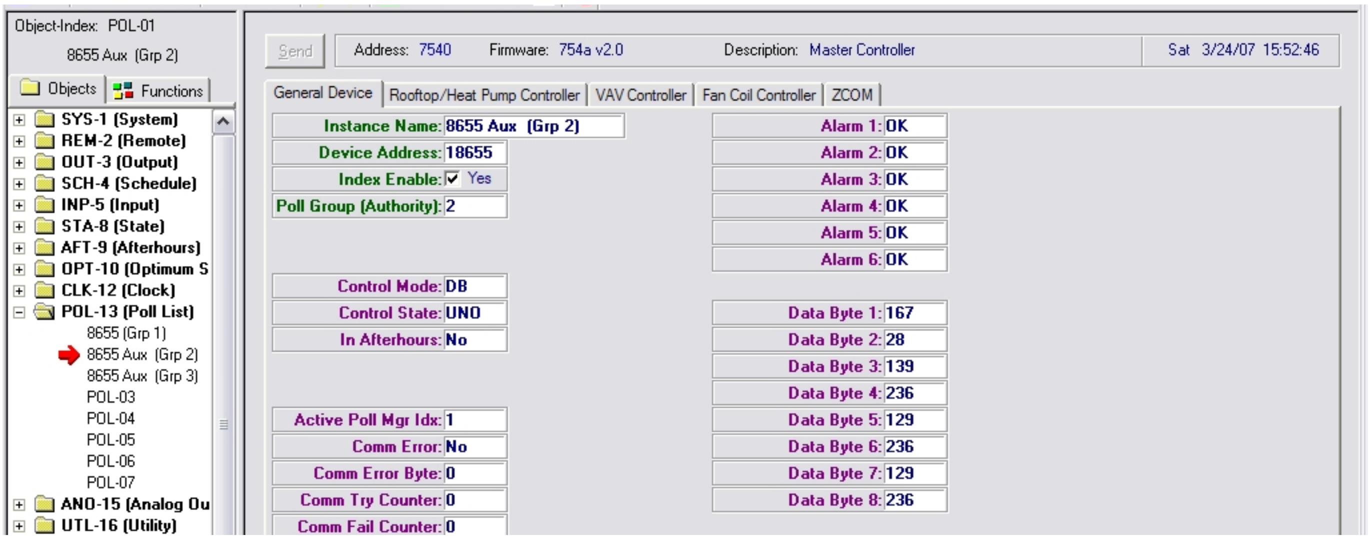
- The Relocate Number of Bytes (**MGR-XX-06-LB_MSN, *MGR_Relocate-NumofBytes***) value tells us how many Bytes we are going to store in the Poll List. We are storing 8 Bytes, which is the maximum number of Bytes we can move. This number should always match the number entered in Poll Message Byte 3.

- And finally, we have the Relocate Destination Byte (**MGR-XX-06-HB_MSN, *MGR_Relocate-DestByte***). This is the landing place for the 8 bytes of data in the Poll List. If you look at the General Tab of a Poll List instance, you will see that there are 8 values marked Data Byte 1 through Data Byte 8 (**POL-XX-06-LB_ONLY**, ***POL_DataByte1*** through **POL-XX-13-LB_ONLY**, ***POL_DataByte8***). These are the 8 Bytes that we can use to store data in the Poll List. In our example, we have a value of 0, which translates to Data Byte 1. For the record, Data Byte 2 would be represented by a value of 1, Data Byte 3 would be represented by a value of 2, etc.

When this particular Poll Manager does its thing, this is the path the data takes.

- Table 9, Entry 47 is retrieved, and put into Destination Byte 0, which is Poll List Data Byte 1.
- Table 9, Entry 48 is retrieved, and put into Destination Byte 1, which is Poll List Data Byte 2.
- Table 9, Entry 49 is retrieved, and put into Destination Byte 2, which is Poll List Data Byte 3.
- Table 9, Entry 50 is retrieved, and put into Destination Byte 3, which is Poll List Data Byte 4.
- Table 9, Entry 51 is retrieved, and put into Destination Byte 4, which is Poll List Data Byte 5.
- Table 9, Entry 52 is retrieved, and put into Destination Byte 5, which is Poll List Data Byte 6.
- Table 9, Entry 53 is retrieved, and put into Destination Byte 6, which is Poll List Data Byte 7.
- Table 9, Entry 54 is retrieved, and put into Destination Byte 7, which is Poll List Data Byte 8.

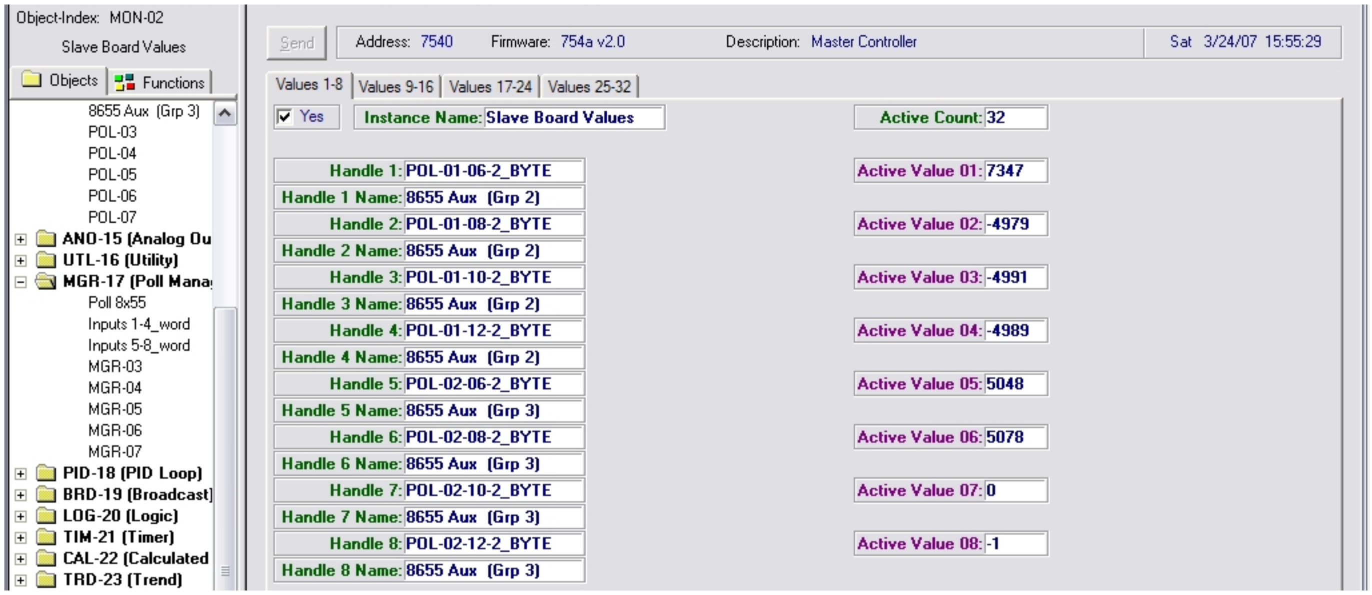
The last step in the whole process is to assemble the 8 Byte values into 4 Word values, so we can use them in the controller and/or WebLink. If you notice in *Figure 2*, Data Byte 1 contains a value of 167 and Data Byte 2 contains a value of 28. This won't mean anything to the end user, or the controller. They are each a part of the Word value that is the temperature value read on Input #1. We can't simply add them together. We have to put them together in such a way, that each Byte represents its share of a full Word value. We could reassemble them using the Calculate Object, but that is overkill here. The Poll List has an attribute that allows us to grab Data Byte 1 and Data Byte 2 at the same time (**POL-XX-06-2_BYTE, *POL_DataWord1-2***). We can use that attribute in a Monitor instance, and it will reassemble the two Bytes for us, as shown in *Figure 3*. You can also see in *Figure 3* how to reassemble the rest of the ASIC/1-8655 Input values, including Inputs #7 and #8, which are configured as N.O. Binary inputs. Input #8 (*Figure 3*, Active Value 8) is closed, or made, and is displaying a value of -1, just as you are used to seeing in an ASIC/2.


Object-Index: POL-01
8655 Aux (Grp 2)

Address: 7540 Firmware: 754a v2.0 Description: Master Controller Sat 3/24/07 15:52:46

Instance Name:	8655 Aux (Grp 2)	Alarm 1:	OK
Device Address:	18655	Alarm 2:	OK
Index Enable:	✓ Yes	Alarm 3:	OK
Poll Group (Authority):	2	Alarm 4:	OK
		Alarm 5:	OK
		Alarm 6:	OK
Control Mode:	DB		
Control State:	UNO	Data Byte 1:	167
In Afterhours:	No	Data Byte 2:	28
		Data Byte 3:	139
		Data Byte 4:	236
Active Poll Mgr Idx:	1	Data Byte 5:	129
Comm Error:	No	Data Byte 6:	236
Comm Error Byte:	0	Data Byte 7:	129
Comm Try Counter:	0	Data Byte 8:	236
Comm Fail Counter:	0		

Figure 2. Example of a Poll List instance used with a Poll Manager using the Place Data polling instruction


Object-Index: MON-02

Slave Board Values

Objects | Functions

- 8655 Aux (Grp 3)
- POL-03
- POL-04
- POL-05
- POL-06
- POL-07
- ANO-15 (Analog Ou
- UTL-16 (Utility)
- MGR-17 (Poll Mana
  - Poll 8x55
  - Inputs 1-4_word
  - Inputs 5-8_word
  - MGR-03
  - MGR-04
  - MGR-05
  - MGR-06
  - MGR-07
- PID-18 (PID Loop)
- BRD-19 (Broadcast)
- LOG-20 (Logic)
- TIM-21 (Timer)
- CAL-22 (Calculated
- TRD-23 (Trend)

Send | Address: 7540 | Firmware: 754a v2.0 | Description: Master Controller | Sat 3/24/07 15:55:29

Values 1-8 | Values 9-16 | Values 17-24 | Values 25-32

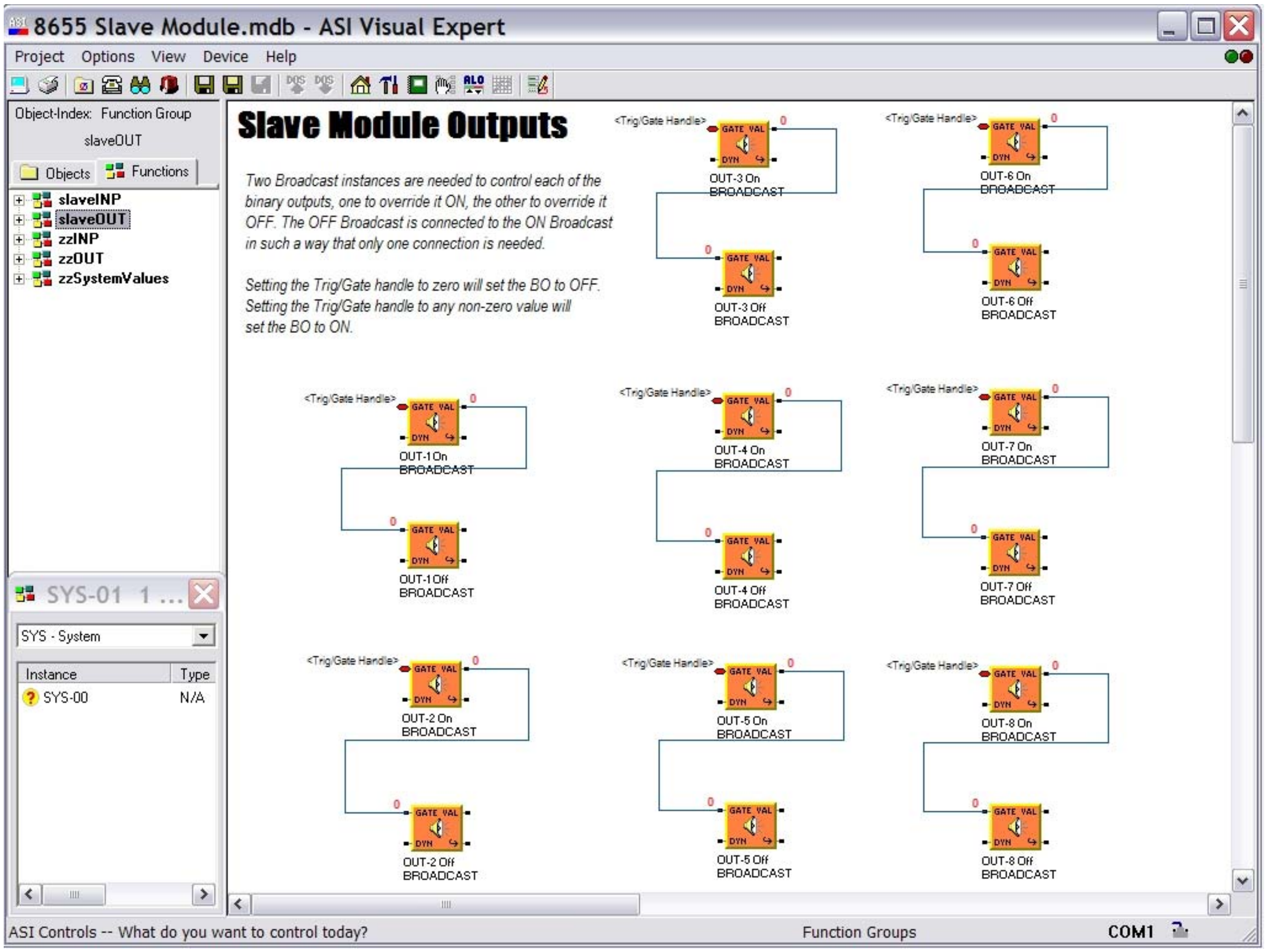
Yes [✓] | Instance Name: Slave Board Values | Active Count: 32

Handle 1:	POL-01-06-2_BYTE	Active Value 01:	7347
Handle 1 Name:	8655 Aux (Grp 2)		
Handle 2:	POL-01-08-2_BYTE	Active Value 02:	-4979
Handle 2 Name:	8655 Aux (Grp 2)		
Handle 3:	POL-01-10-2_BYTE	Active Value 03:	-4991
Handle 3 Name:	8655 Aux (Grp 2)		
Handle 4:	POL-01-12-2_BYTE	Active Value 04:	-4989
Handle 4 Name:	8655 Aux (Grp 2)		
Handle 5:	POL-02-06-2_BYTE	Active Value 05:	5048
Handle 5 Name:	8655 Aux (Grp 3)		
Handle 6:	POL-02-08-2_BYTE	Active Value 06:	5078
Handle 6 Name:	8655 Aux (Grp 3)		
Handle 7:	POL-02-10-2_BYTE	Active Value 07:	0
Handle 7 Name:	8655 Aux (Grp 3)		
Handle 8:	POL-02-12-2_BYTE	Active Value 08:	-1
Handle 8 Name:	8655 Aux (Grp 3)		

Figure 3. Example of using a Monitor instance to assemble Poll List Data Byte Values

Working with the Binary Outputs

The work needed to override the Binary Outputs doesn't involve as many steps as reading the Inputs. Each physical Output on the ASIC/1-8655 will need two separate Broadcast instances allocated so we can override it on and off. In the sample configuration files the "Off" instance (*Figure 6*) is controlled by the state of the "On" instance (*Figure 5*), so we can use a single Trigger Handle. This allows us to mimic the normal operation of Binary Outputs, for easier programming. This can be seen in a screen capture, shown if *Figure 4*.



Two Broadcast instances are needed to control each of the binary outputs, one to override it ON, the other to override it OFF. The OFF Broadcast is connected to the ON Broadcast in such a way that only one connection is needed.

Setting the Trig/Gate handle to zero will set the BO to OFF. Setting the Trig/Gate handle to any non-zero value will set the BO to ON.

Figure 4. Function Group that contains all Broadcast instances needed to override an ASIC/1-8655's Binary Outputs

You can see in this Function Group that the Trigger Handle of each "On" Broadcast is configured to be a Public Point. The matching "Off" instance is configured to look at the status of the "On" instance's Trigger Value (**BRD-XX-01-WD_VAL, *BRD_TrigValue***), giving us the operation we desire. You can use Public Points in other FGP's to link to the "On" Broadcast instance.

To use this setup in your project, there are just a few things that may need editing. One would be the address of the ASIC/1, which is put into the Destination Address attribute (**BRD-XX-15-LB_ONLY and BRD-XX-16-LB_ONLY, *BRD_DestnAddress***). This example uses 18655 for a value, your project may differ. Adjust as necessary.

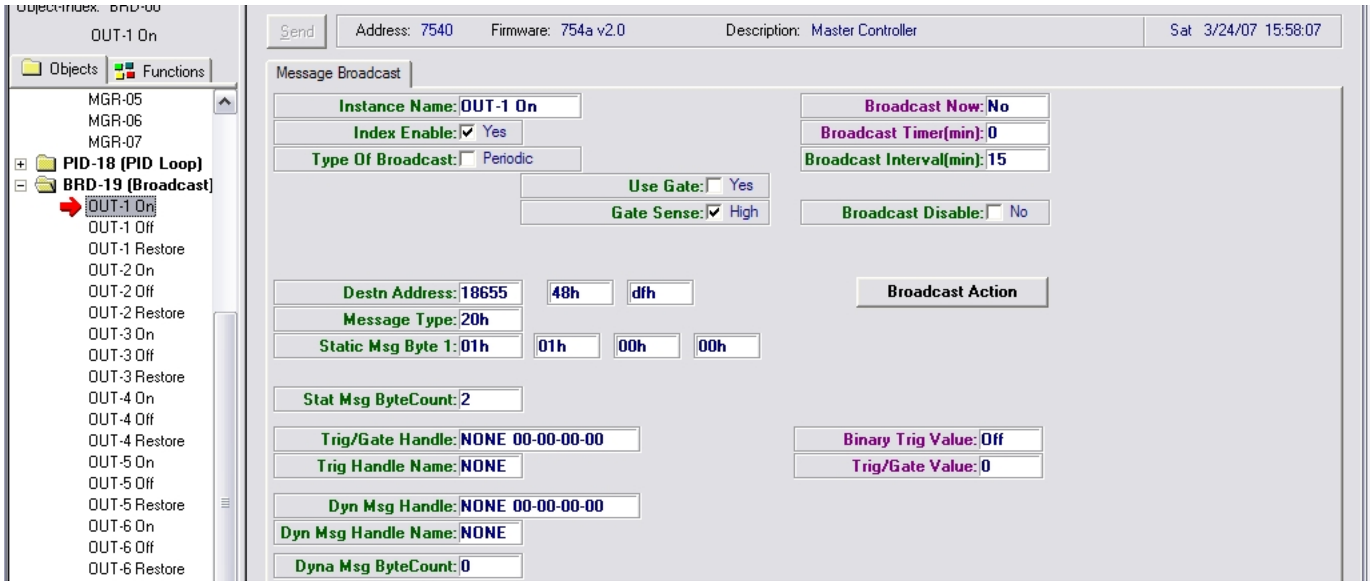
The Destination Address can also be a Group Address value, not just an address for a single controller. That would be for a different application, such as lighting control, but it is worth mentioning

You may also wish to change the Type of Broadcast value (**BRD-XX-05-IFLoB1S, *BRD_TypeOfBroadcast***) that you are using. You can choose to use a Periodic or Triggered version. The Periodic version will use the Broadcast Interval attribute (**BRD-XX-14-LB_ONLY, *BRD_BroadcastInterval***) to determine how often the Broadcast Message is sent.

If the Use Gate (**BRD-XX-05-IFLoB3S, *BRD_GateOnAnyChange***) value is “Yes” and the Gate Sense is "High" (**BRD-XX-05-IFLoB2S, *BRD_GateSenseZero***), the Periodic Type will only send a message when the Trigger Value is non-zero. If the Use Gate value is “Yes” and the Gate Sense is "Low", the Periodic Type will only send a message when the Trigger Value is equal to zero.

The Triggered version sends out the message only once, based on the status of the Trigger Value. For this type of application, I recommend the Periodic version be used, but there may be cases where you want to use the Triggered version instead.

The rest of the attributes do not need any adjustment, and should be left as they are.



Object-Index: BRD-00
OUT-1 On
Objects | Functions

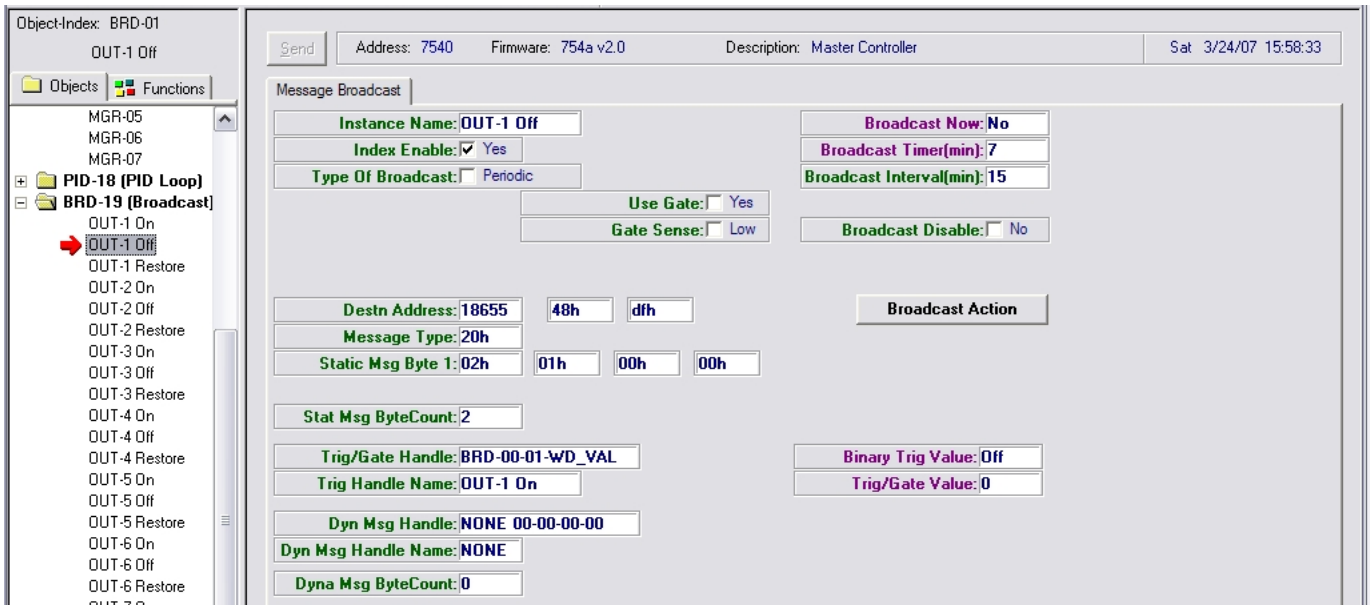
MGR-05
MGR-06
MGR-07
PID-18 (PID Loop)
BRD-19 (Broadcast)
OUT-1 On
OUT-1 Off
OUT-1 Restore
OUT-2 On
OUT-2 Off
OUT-2 Restore
OUT-3 On
OUT-3 Off
OUT-3 Restore
OUT-4 On
OUT-4 Off
OUT-4 Restore
OUT-5 On
OUT-5 Off
OUT-5 Restore
OUT-6 On
OUT-6 Off
OUT-6 Restore

Send | Address: 7540 | Firmware: 754a v2.0 | Description: Master Controller | Sat 3/24/07 15:58:07

Message Broadcast

Instance Name: OUT-1 On
Index Enable: ✓ Yes
Type Of Broadcast: Periodic
Use Gate: Yes
Gate Sense: ✓ High
Broadcast Now: No
Broadcast Timer(min): 0
Broadcast Interval(min): 15
Broadcast Disable: No
Destn Address: 18655 | 48h | dfh
Broadcast Action
Message Type: 20h
Static Msg Byte 1: 01h | 01h | 00h | 00h
Stat Msg ByteCount: 2
Trig/Gate Handle: NONE 00-00-00-00
Trig Handle Name: NONE
Binary Trig Value: Off
Trig/Gate Value: 0
Dyn Msg Handle: NONE 00-00-00-00
Dyn Msg Handle Name: NONE
Dyna Msg ByteCount: 0

Figure 5. Broadcast instance to override an ASIC/1-8655 Binary Output “On”



Object-Index: BRD-01
OUT-1 Off
Objects | Functions

MGR-05
MGR-06
MGR-07
PID-18 (PID Loop)
BRD-19 (Broadcast)
OUT-1 On
OUT-1 Off
OUT-1 Restore
OUT-2 On
OUT-2 Off
OUT-2 Restore
OUT-3 On
OUT-3 Off
OUT-3 Restore
OUT-4 On
OUT-4 Off
OUT-4 Restore
OUT-5 On
OUT-5 Off
OUT-5 Restore
OUT-6 On
OUT-6 Off
OUT-6 Restore

Send | Address: 7540 | Firmware: 754a v2.0 | Description: Master Controller | Sat 3/24/07 15:58:33

Message Broadcast

Instance Name: OUT-1 Off
Index Enable: ✓ Yes
Type Of Broadcast: Periodic
Use Gate: Yes
Gate Sense: Low
Broadcast Now: No
Broadcast Timer(min): 7
Broadcast Interval(min): 15
Broadcast Disable: No
Destn Address: 18655 | 48h | dfh
Broadcast Action
Message Type: 20h
Static Msg Byte 1: 02h | 01h | 00h | 00h
Stat Msg ByteCount: 2
Trig/Gate Handle: BRD-00-01-WD_VAL
Trig Handle Name: OUT-1 On
Binary Trig Value: Off
Trig/Gate Value: 0
Dyn Msg Handle: NONE 00-00-00-00
Dyn Msg Handle Name: NONE
Dyna Msg ByteCount: 0

Figure 6. Broadcast instance to override an ASIC/1-8655 Binary Output “Off”

This example also includes Broadcast instances that are configured to restore the Binary Outputs. These instances will remove the override commands and return the Output to its normal operation. They were included for reference, and should not really be used in a slave module application. Keep them in mind for other applications.

Conclusion

Considerations

This technology is not limited to just the ASIC/1-8655. You are probably wondering if you can also use the ASIC/1-6000-PD-MB, which will lower your hardware costs even further. Yes, you can. However, this controller is not as flexible as the ASIC/1-8655. Here are the main differences.

- The ASIC/1-6000 platform does not have as many physical Inputs and Outputs.
- You don't get Form C dry contacts for the outputs, you get Triacs.
- The ASIC/1-6000 platform does not give you a pull-up resistor for each physical input. This means you can not configure the Inputs any way you want to, you have to use them as they come. Input 5 is the only one that has a removable pull-up resistor.

So when you are tempted to use an ASIC/1-6000 instead of an ASIC/1-8655, keep these factors in mind.

Finally, carefully consider any safety issues caused by a lack of communication between the ASIC/2 and the ASIC/1. Things do fail, and problems will occur. If your application needs 20 physical inputs, use the ASIC/2 for the 16 most important inputs, and use the ASIC/1 for the 4 inputs that are the least important or least likely to cause a major sequence error if they can not be read. Do the same for the physical outputs. If you have any outputs that can be classified as "just for show", such as controlling panel lights or buzzers, consider putting them on the ASIC/1, and leaving the equipment outputs on the ASIC/2. It would also be a good idea to set up a Notify instance to monitor the communication status (**POL-XX-02-IFLoB5S,** ***POL_CommError***) of the ASIC/1. You may also want to consider using this attribute as an interlock in your controller sequence. Remember that the Poll List will retain the last information it got, until the controller is reset, or until the polled controller comes back online. This interlock would prevent your ASIC/2 controller from continuing with a sequence of operation that is using old or bad data values. Lastly, protect the Local Bus cable a little more diligently, especially if the ASIC/1 is not located in the same enclosure as the ASIC/2. Do what you can to make sure that it can not be damaged easily.

Summary

Using an ASIC/1-8655 as a slave module is a good way to extend the physical points available to an ASIC/2, and do it without loading up the System Bus with Remote Points. Carefully consider your application and see if it is a workable solution.

If you have any further questions please contact: ASI Controls Technical Support support@asicontrols.com, or call 925-866-8808