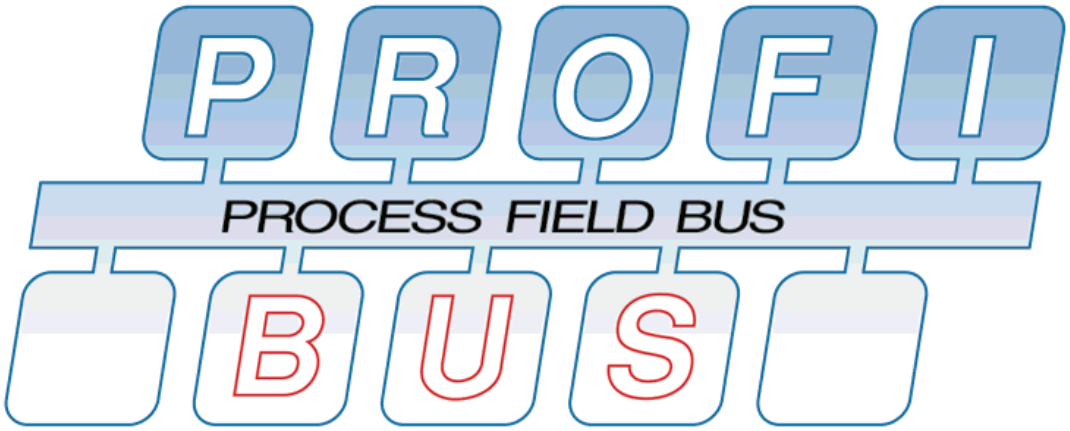
GSE Scale Systems

GSE 60-Series Instruments

Profibus Communications Option and Application Note

Ver 1.0.1
3/13/09
P/N 39-10-39398



Published by
GSE Scale Systems
1525 Fairlane Circle.
Allen Park, MI 48101
USA

Standard Scale & Supply Company
313-255-6700 * dawright@standardscale.com

Table of Contents

GSE

1.1 PROFIBUS BASICS - GSE SCALE 60 SERIES INDICATORS (PROFIBUS SLAVES).

GSE 60 Series Indicators have passed conformance testing for Profibus.

1.1.1 Profibus Master:

The device that requests information from or sends information to the slave.

1.1.2 Profibus Slaves

Equipment on a Profibus Network that provides information to and or are controlled by the Master. Profibus architecture allows for 32 stations per network segment, 126 total if multiple segments are connected using repeaters. The GSE 60 Series Indicators perform as Slaves on the Profibus Network.

1.1.3 GSD File

A computer file of information that configures the master for interfacing with a slave device. Not all Masters have the capability of using an GSD File and depend on the person programming the master for configuration.
PTO Approved GSE GSD Files are available at www.profibus.com.

1.1.4 I/O Module

The I/O module is selected at the master while creating the slave device. The GSE Indicator has sixteen possible modules to select from. Which module gets selected will depend on how many scales are attached to the unit, and whether string data will be communicated. While the largest module could always be chosen, this will cause the transfer of more data bytes than is required, needlessly using network bandwidth.

Profibus has two main methods of communicating information.

I/O Assemblies and Explicit Messaging.

1.1.5 I/O Assemblies

The main block of information that is regularly communicated to the Master. If, for example, four fields of information need to be rapidly and regularly supplied to the Master, they would be included in the I/O Assembly. In this case, a single transmission supplies four answers. The assembly being used can be changed at anytime, all that is required is the data fit within the I/O module that is being used. Large I/O assemblies that provide unneeded information decrease system performance.

1.1.6 Explicit Messaging

Any other information that is available through the GSE Profibus Interface may be read or written in this way. It is information that is generally set or requested as one piece of information at a time, and is piggy backed onto the end of the i/o assembly data. This type of communication is suited to information that is not rapidly changing.

1.1.7 Overview of the GSE Profibus Option

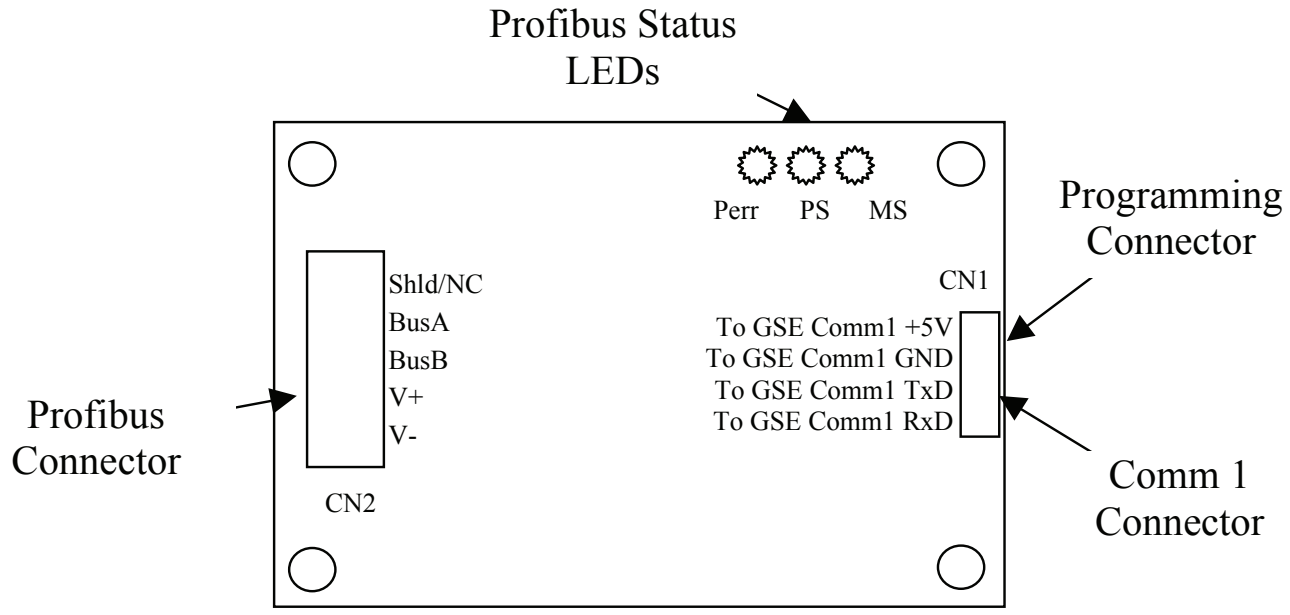
The Profibus Option for GSE 60-Series Instruments operates as a slave on the Profibus network. It utilizes the Master/Slave Connection Set, as defined in the Profibus Specifications. It supports Explicit and I/O Assemblies.

The GSE Profibus Option communicates with the GSE 60-Series Instrument on the Comm 1 serial port.

1.1.7.1 GSE Profibus Option Board

The Profibus Option mounts inside the enclosure of a GSE 60-Series Instrument on 4 mounting posts (See Quick Start Guide). A 4-pin connector (CN1) is wired to the Instrument to provide power and communications to the GSE Profibus Option Board. A 5-pin Open Pluggable connector (CN2) provides an optically-isolated connection to the Profibus network. Three LEDs provide status (see Figure 1.1).

Figure 1-1 Profibus Communications Module



1.1.7.2 Profibus and the GSE 60 Series Indicators

Each GSE 60 Series Indicator has a built in Profibus setup script, which puts the indicator into a known setup state with selectable I/O assemblies and most scale parameters available through Explicit Messages.

WARNING* Do not change the length of the strings of Var# 15, 30 ,45 and 60 or change a Numeric Var to a String. Doing so will shift all data stored above the Var and make the registers above this Var Profibus registers unuseable by Profibus. See "Advanced Profibus Setup and Modification".**

The programmability of the 60 Series Indicators is still available for you to collaborate with the programmer of the Master to customize an I/O Assembly that will efficiently supply virtually any combination of scale information from weight, future weight and rate to blocks of relay status. The number of bytes in the I/O assemblies must remain the same but the information that is contained in these bytes can be easily changed for a specific application.

1.2 INITIAL SETUP OF INDICATOR

IMPORTANT* The 66x Series Indicators require the installation of an additional 8K FRam (EEProm on older units) to provide enough memory to run this setup.**

Detailed installation information is provided in the **"Quick Start Guide".**

1	Install any Multi-Scale Options and the Profibus Option per "Quick Start Guide". Power up the indicator with the [CLR] key held. After it states that macros are disabled, a prompt will appear for disabling comm1. Press [ENTER].
2	Go into setup and enable the scales to be used.
3	Go to parameter 65002. "Deflt – Cal"
4	Press the Tare Key until "Profibus" is displayed.
5	Press Enter, read the prompts and Press Enter the required number of times for your model. a. The scale will load its Profibus Setup Script. b. Any scale that was disabled will have been changed to saved. c. The Indicator is now configured to communicate on Profibus and Port 1 is not available for other communication.
6	To test the complete interface, leave the unused scales "Saved" This will cause zero values to be returned for the "Saved" scales. For normal operation, disabling the "Saved" scales will cause the interface to return resource unavailable for the disabled scales, a more reasonable response for scales that don't exist.

1.3 SERIAL PORT ONE SETTINGS

Serial port number one is used for profibus communications. The port is automatically configured when the built-in profibus script is run. Once the script has run, P200 and P209 can be changed if necessary.

Once the Receive (P205) is set to Profibus "253 [ENTER]", changes that are made to any other Port #1 Parameter that are Identified as PFbus on the lower 2x5 Display will be saved but suppressed until P205 is changed to another Receive choice.		
P200	Baud	Profibus Baud – Auto, 9.6K, 19.2K, 31.2K, 45.5K, 93.7K, 187K, 500K, 1.5M, 3M, 6M, 12M (auto will detect from: 19.2K, 187.5K, 500K, 1.5M, 3M, 6M, 12M)(Curr UnKwn flashes if not connected)
P201	Data Bits	Suppressed
P202	Parity	Suppressed
P203	Stop Bits	Suppressed
P204	Flow Control	Suppressed
P205	ReceiveMode	Profibus = 253
P206	Full	Suppressed
P207	Tx Buffer	Suppressed
P208	Rx Buffer	Suppressed
P209	PFad	This is the address that the Master refers to to communicate with this device. All devices on the network must have a unique address. (Address choices 2-125/126) (address 126 indicates the Master should assign a value)
P210	MbMd	Suppressed
P211	Word Order	Suppressed
P212	I/O Assy	Informational only, indicates which i/o assembly is being used.
P213	Ptime	You may slow the updates of the I/O Assys by setting this value above 16 ms (actual value based on indicator model). This could be used when the Indicator is running a large CPU intensive program. It sets how often the I/O Assy data is updated to free some CPU time. Profibus response time is the same.

1.4 INITIAL PROFIBUS MASTER SETUP

Every master is different, so the exact steps required will vary.

Place the GSD and DIB files supplied by GSE into the folders appropriate for your master. Select one of the sixteen defined I/O Modules, this will instruct the master and slave how many data bytes will be exchanged between them.

The following graphics are for use by the master software for pictorially representing the GSE Indicator.

1.5 THE I/O MODULES

I/O modules define how many bytes are exchanged between the master and slave device. I/O modules do not give any meaning to the bytes. Which module to use is selected when setting up slaves in the master, once setup in the master this value will probably never be changed (it will probably require deleting and recreating the slave device).

GSE indicators support sixteen I/O modules. Modules #1-8 do not support transferring string data. Module #1 supports polled data for one scale, #2 supports polled data for two scales, etc. Modules #9-16 support transferring string data. Module #9 supports polled data for one scale, #10 supports polled data for two scales, etc.

Which module you select will be based on the configuration of the indicator and how it is being used. Make sure chosen module is large enough to transfer the amount of data required by your setup.

TABLE 1.5

I/O Module	I/O Assembly Numbers usable in this I/O module	Comment
1 8 Bytes Out, 16 Bytes In	1, 9-10	No Strings
2 8 Bytes Out, 24 Bytes In	1-2, 9-12	
3 8 Bytes Out, 32 Bytes In	1-3, 9-14	
4 8 Bytes Out, 40 Bytes In	1-4, 9-16	
5 8 Bytes Out, 48 Bytes In	1-5, 9-16	
6 8 Bytes Out, 56 Bytes In	1-6, 9-16	
7 8 Bytes Out, 64 Bytes In	1-7, 9-16	
8 8 Bytes Out, 72 Bytes In	1-8, 9-16	
9 22 Bytes Out, 30 Bytes In	1, 9-10	Supports Strings
10 22 Bytes Out, 38 Bytes In	1-2, 9-12	
11 22 Bytes Out, 46 Bytes In	1-3, 9-14	
12 22 Bytes Out, 54 Bytes In	1-4, 9-16	
13 22 Bytes Out, 62 Bytes In	1-5, 9-16	
14 22 Bytes Out, 70 Bytes In	1-6, 9-16	
15 22 Bytes Out, 78 Bytes In	1-7, 9-16	
16 22 Bytes Out, 86 Bytes In	1-8, 9-16	

1.6 THE I/O ASSEMBLIES

The I/O Assemblies are used to communicate the most frequently required data. The assembly requested must fit within the I/O module that was setup in the master. Any data beyond the I/O module length will be lost. Any data less than the I/O module length will be zero filled.

The information is arranged in logical blocks in the indicator. The register numbers identify the order of available data. Table 1.6 shows the logical I/O Assembly blocks in order from Assy 1-to Assy-8 and from Assy-9 to Assy-16. For example, Assembly 3 would include all the information from Assy 1, 2 and 3. Assembly 12 would include Assy 9, 10, 11 and 12.

I/O assemblies 1 to 8 are in blocks of 8 Bytes. Two bytes for each of three 16 bit integers and two bytes for the status word. I/O assemblies 9 to 16 are in blocks of four Bytes. Four bytes for each 32 bit IEE floating point number.

One of the following I/O Assemblies will be selected, at run time, by the Master. Indicators with less than eight scale capability will have the highest available scale data duplicated in the I/O assemblies in place of higher scale information. See section 1.8 for how to change the assembly data sent.

TABLE 1.6 SCALE REGISTERS FOR I/O ASSEMBLIES

I/O Assys 1 - 8

Assembly Numbers	Scale Parm	Registers	Scaled Y/N	Type	Scale Value
8 7 6 5 4 3 2 1	6001	40001	Y	INT16	Gross Scale 1
8 7 6 5 4 3 2 1	6002	40002	Y	INT16	Future Gross 1
8 7 6 5 4 3 2 1	6003	40003	Y	INT16	Rate Scale 1
8 7 6 5 4 3 2 1	6004	40004	N	High/ Low Byte	Scaling/ Stat Scale 1
8 7 6 5 4 3 2	6005	40005	Y	INT16	Gross Scale 2
8 7 6 5 4 3 2	6006	40006	Y	INT16	Future Gross 2
8 7 6 5 4 3 2	6007	40007	Y	INT16	Rate Scale 2
8 7 6 5 4 3 2	6008	40008	N	High/Low Byte	Scaling/Stat Scale 2
8 7 6 5 4 3	6009	40009	Y	INT16	Gross Scale 3
8 7 6 5 4 3	6010	40010	Y	INT16	Future Gross 3
8 7 6 5 4 3	6011	40011	Y	INT16	Rate Scale 3
8 7 6 5 4 3	6012	40012	N	High/ Low Byte	Scaling/ Stat Scale 3
8 7 6 5 4	6013	40013	Y	INT16	Gross Scale 4
8 7 6 5 4	6014	40014	Y	INT16	Future Gross 4
8 7 6 5 4	6015	40015	Y	INT16	Rate Scale 4
8 7 6 5 4	6016	40016	N	High/Low Byte	Scaling/Stat Scale 4
8 7 6 5	6017	40017	Y	INT16	Gross Scale 5
8 7 6 5	6018	40018	Y	INT16	Future Gross 5
8 7 6 5	6019	40019	Y	INT16	Rate Scale 5
8 7 6 5	6020	40020	N	High/ Low Byte	Scaling/ Stat Scale 5
8 7 6	6021	40021	Y	INT16	Gross Scale 6
8 7 6	6022	40022	Y	INT16	Future Gross 6
8 7 6	6023	40023	Y	INT16	Rate Scale 6
8 7 6	6024	40024	N	High/Low Byte	Scaling/Stat Scale 6
8 7	6025	40025	Y	INT16	Gross Scale 7
8 7	6026	40026	Y	INT16	Future Gross 7
8 7	6027	40027	Y	INT16	Rate Scale 7
8 7	6028	40028	N	High/ Low Byte	Scaling/ Stat Scale 7
8	6029	40029	Y	INT16	Gross Scale 8
8	6030	40030	Y	INT16	Future Gross 8
8	6031	40031	Y	INT16	Rate Scale 8
8	6032	40032	N	High/Low Byte	Scaling/Stat Scale 8

See Section 02 for Bit Definition of Scale Parameter 96 Scaling / Stat

I/O Assys 9-16

Assembly Numbers	Scale Parm	Registers	Scaled Y/N	Type	Scale Value
16 15 14 13 12 11 10 9	6033	40033 / *40034*	N	IEE FLOAT 32	Rounded Gross Sc 1
16 15 14 13 12 11 10	6034	40035 / *40036*	N	IEE FLOAT 32	Rounded Gross Sc 2
16 15 14 13 12 11	6035	40037 / *40038*	N	IEE FLOAT 32	Rounded Gross Sc 3
16 15 14 13 12	6036	40039 / *40040*	N	IEE FLOAT 32	Rounded Gross Sc 4
16 15 14 13	6037	40041 / *40042*	N	IEE FLOAT 32	Rounded Gross Sc 5
16 15 14	6038	40043 / *40044*	N	IEE FLOAT 32	Rounded Gross Sc 6
16 15	6039	40045 / *40046*	N	IEE FLOAT 32	Rounded Gross Sc 7
16	6040	40047 / *40048*	N	IEE FLOAT 32	Rounded Gross Sc 8

***See Section 1.7 for Explicit Message Addressing.

1.7 EXPLICIT MESSAGING

Explicit messages are used to set or retrieve specific data, set or check the status of a coil, run a macro or see if it is currently running.

Tables 1.6 and 1.7 show the predefined register map for the data and controls that can be accessed using explicit messages. Note that the data in Table 1.6 is already being polled constantly by the master.

In the tables, the register number is used to reference the data item, and the get/set field indicates how the data may be accessed.

TABLE 1.7 EXPLICIT MESSAGING ADDRESS CHART

Explicit Messaging						
6000 Param	**Registers**	**Scaled Y/N**	**Type**	**Get/Set**	**Parameter Name**	**Parm Number**
These Six are common to all enabled scales						
6041	40049	N	IEE Float	Get/Set	APW	34 Common
	40050					
6042	40051	Y	INT32	Get	Scales	Common
6043	40053	Y	INT32	Get	Combined Net of All Scales	41 Common
6044	40055	Y	INT32	Get	Combined Tare of All Scales	42 Common
6045	40057	N	INT32	Get	Scales	Common
The following block is common to All Series						
6046	40059	Y	INT16	Get	Gross	0 Scale 1
6047	40060	Y	INT16	Get	Net	1 Scale 1
6048	40061	Y	INT16	Get/Set	Tare	2 Scale 1
6049	40062	Y	INT16	Get	Average Gross	15 Scale 1
6050	40063	Y	INT16	Get	Average Net	16 Scale 1
6051	40064	Y	INT16	Get	Peak Gross	18 Scale 1
6052	40065	Y	INT16	Get	Peak Net	19 Scale 1
6053	40066	.01 Sec.	INT16	Get/Set	Free Fall	24 Scale 1
6054	40067	Y	INT16	Get	Future Gross	25 Scale 1
6055	40068	Y	INT16	Get	Future Net	26 Scale 1
6056	40069	N	INT16	Get	Count (Qty)	30 Scale 1
6057	40070	Y	INT32	Get/Set	Gross Total	3 Scale 1
6058	40072	Y	INT32	Get/Set	Net Total	6 Scale 1
6059	40074	Y	INT32	Get/Set	# Accumulations	9 Scale 1
6060	40076	N	INT32	Get	Count Total (Qty)	31 Scale 1
6061	40078	Y	INT16	Get	Gross	0 Scale 2
6062	40079	Y	INT16	Get	Net	1 Scale 2
6063	40080	Y	INT16	Get/Set	Tare	2 Scale 2
6064	40081	Y	INT16	Get	Average Gross	15 Scale 2
6065	40082	Y	INT16	Get	Average Net	16 Scale 2
6066	40083	Y	INT16	Get	Peak Gross	18 Scale 2
6067	40084	Y	INT16	Get	Peak Net	19 Scale 2
6068	40085	.01 Sec.	INT16	Get/Set	Free Fall	24 Scale 2
6069	40086	Y	INT16	Get	Future Gross	25 Scale 2
6070	40087	Y	INT16	Get	Future Net	26 Scale 2
6071	40088	N	INT16	Get	Count (Qty)	30 Scale 2
6072	40089	Y	INT32	Get/Set	Gross Total	3 Scale 2
6073	40091	Y	INT32	Get/Set	Net Total	6 Scale 2
6074	40093	Y	INT32	Get/Set	# Accumulations	9 Scale 2
6075	40095	N	INT32	Get	Count Total (Qty)	31 Scale 2
		Setpoints set to act never deact				
6076	40097	Y	INT16	Get/Set	SP1 Target	Var 80.1
6077	40098	Y	INT16	Get/Set	SP2 Target	Var 80.2
6078	40099	Y	INT16	Get/Set	SP3 Target	Var 80.3
6079	40100	Y	INT16	Get/Set	SP4 Target	Var 80.4
6080	40101	Y	INT16	Get/Set	SP5 Target	Var 80.5
6081	40102	Y	INT16	Get/Set	SP6 Target	Var 80.6
6082	40103	Y	INT16	Get/Set	SP7 Target	Var 80.7

6083	40104	Y	INT16	Get/Set	SP8 Target	Var 80.8
6084	40105	Y	INT16	Get/Set	Var # 9	Var 80.9
6085	40106	Y	INT16	Get/Set	Var # 10	Var 80.10
6086	40107	Y	INT16	Get/Set	Var # 11	Var 80.11
6087	40108	Y	INT32	Get/Set	Var # 12	Var 80.12
6088	40110	Y	INT32	Get/Set	Var # 13	Var 80.13
6089	40112	N	IEE Float	Get/Set	Var # 14	Var 80.14
6090	40114	N	String 18 Char=9 Reg	Get/Set	Var # 15	Var 80.15
6091	40123	N	16 Bits	Get	SetPoint 1 - 16 Status	
********** END OF 460 SERIES**********						
The following block is common to 560 and 660 Series						
6092	40124	Y	INT16	Get	Gross	0 Scale 3
6093	40125	Y	INT16	Get	Net	1 Scale 3
6094	40126	Y	INT16	Get/Set	Tare	2 Scale 3
6095	40127	Y	INT16	Get	Average Gross	15 Scale 3
6096	40128	Y	INT16	Get	Average Net	16 Scale 3
6097	40129	Y	INT16	Get	Peak Gross	18 Scale 3
6098	40130	Y	INT16	Get	Peak Net	19 Scale 3
6099	40131	.01	INT16	Get/Set	Free Fall	24 Scale 3
6100	40132	Y	INT16	Get	Future Gross	25 Scale 3
6101	40133	Y	INT16	Get	Future Net	26 Scale 3
6102	40134	N	INT16	Get	Count (Qty)	30 Scale 3
6103	40135	Y	INT32	Get/Set	Gross Total	3 Scale 3
6104	40137	Y	INT32	Get/Set	Net Total	6 Scale 3
6105	40139	Y	INT32	Get/Set	# Accumulations	9 Scale 3
6106	40141	N	INT32	Get	Count Total (Qty)	31 Scale 3
6107	40143	Y	INT16	Get	Gross	0 Scale 4
6108	40144	Y	INT16	Get	Net	1 Scale 4
6109	40145	Y	INT16	Get/Set	Tare	2 Scale 4
6110	40146	Y	INT16	Get	Average Gross	15 Scale 4
6111	40147	Y	INT16	Get	Average Net	16 Scale 4
6112	40148	Y	INT16	Get	Peak Gross	18 Scale 4
6113	40149	Y	INT16	Get	Peak Net	19 Scale 4
6114	40150	.01 Sec.	INT16	Get/Set	Free Fall	24 Scale 4
6115	40151	Y	INT16	Get	Future Gross	25 Scale 4
6116	40152	Y	INT16	Get	Future Net	26 Scale 4
6117	40153	N	INT16	Get	Count (Qty)	30 Scale 4
6118	40154	Y	INT32	Get/Set	Gross Total	3 Scale 4
6119	40156	Y	INT32	Get/Set	Net Total	6 Scale 4
6120	40158	Y	INT32	Get/Set	# Accumulations	9 Scale 4
6121	40160	N	INT32	Get	Count Total (Qty)	31 Scale 4
6122	40162	Y	INT16	Get/Set	SP Target 9	Var 80.16
6123	40163	Y	INT16	Get/Set	SP Target 10	Var 80.17
6124	40164	Y	INT16	Get/Set	SP Target 11	Var 80.18
6125	40165	Y	INT16	Get/Set	SP Target 12	Var 80.19
6126	40166	Y	INT16	Get/Set	SP Target 13	Var 80.20
6127	40167	Y	INT16	Get/Set	SP Target 14	Var 80.21
6128	40168	Y	INT16	Get/Set	SP Target 15	Var 80.22
6129	40169	Y	INT16	Get/Set	SP Target 16	Var 80.23
6130	40170	Y	INT16	Get/Set	Var # 24	Var 80.24
6131	40171	Y	INT16	Get/Set	Var # 25	Var 80.25
6132	40172	Y	INT16	Get/Set	Var # 26	Var 80.26
6133	40173	Y	INT32	Get/Set	Var # 27	Var 80.27
6134	40175	Y	INT32	Get/Set	Var # 28	Var 80.28

6135	40177	N	IEE Float	Get/Set	Var # 29	Var 80.29
6136	40179	N	String 18 Char=9 Reg	Get/Set	Var # 30	Var 80.30
6137	40188	Y	INT16	Get/Set	SP Target 17	Var 80.31
6138	40189	Y	INT16	Get/Set	SP Target 18	Var 80.32
6139	40190	Y	INT16	Get/Set	SP Target 19	Var 80.33
6140	40191	Y	INT16	Get/Set	SP Target 20	Var 80.34
6141	40192	Y	INT16	Get/Set	SP Target 21	Var 80.35
6142	40193	Y	INT16	Get/Set	SP Target 22	Var 80.36
6143	40194	Y	INT16	Get/Set	SP Target 23	Var 80.37
6144	40195	Y	INT16	Get/Set	SP Target 24	Var 80.38
6145	40196	Y	INT16	Get/Set	Var # 39	Var 80.39
6146	40197	Y	INT16	Get/Set	Var # 40	Var 80.40
6147	40198	Y	INT16	Get/Set	Var # 41	Var 80.41
6148	40199	Y	INT32	Get/Set	Var # 42	Var 80.42
6149	40201	Y	INT32	Get/Set	Var # 43	Var 80.43
6150	40203	N	IEE Float	Get/Set	Var # 44	Var 80.44
6151	40205	N	String 18 Char=9 Reg	Get/Set	Var # 45	Var 80.45
6152	40214	Y	INT16	Get/Set	SP Target 25	Var 80.46
6153	40215	Y	INT16	Get/Set	SP Target 26	Var 80.47
6154	40216	Y	INT16	Get/Set	SP Target 27	Var 80.48
6155	40217	Y	INT16	Get/Set	SP Target 28	Var 80.49
6156	40218	Y	INT16	Get/Set	SP Target 29	Var 80.50
6157	40219	Y	INT16	Get/Set	SP Target 30	Var 80.51
6158	40220	Y	INT16	Get/Set	SP Target 31	Var 80.52
6159	40221	Y	INT16	Get/Set	SP Target 32	Var 80.53
6160	40222	Y	INT16	Get/Set	Var # 54	Var 80.54
6161	40223	Y	INT16	Get/Set	Var # 55	Var 80.55
6162	40224	Y	INT16	Get/Set	Var # 56	Var 80.56
6163	40225	Y	INT32	Get/Set	Var # 57	Var 80.57
6164	40227	Y	INT32	Get/Set	Var # 58	Var 80.58
6165	40229	N	IEE Float	Get/Set	Var # 59	Var 80.59
6166	40231	N	String 18 Char=9 Reg	Get/Set	Var # 60	Var 80.60
6167	40240	N	16 Bits	Get	SetPoint 17 - 32 Status	
			********** END OF 560 SERIES **********			
			The following block applys to the 660 Series			
6168	40241	Y	INT16	Get	Gross	0 Scale 5
6169	40242	Y	INT16	Get	Net	1 Scale 5
6170	40243	Y	INT16	Get/Set	Tare	2 Scale 5
6171	40244	Y	INT16	Get	Average Gross	15 Scale 5
6172	40245	Y	INT16	Get	Average Net	16 Scale 5
6173	40246	Y	INT16	Get	Peak Gross	17 Scale 5
6174	40247	Y	INT16	Get	Peak Net	18 Scale 5
6175	40248	.01 Sec.	INT16	Get/Set	Free Fall	19 Scale 5
6176	40249	Y	INT16	Get	Future Gross	20 Scale 5
6177	40250	Y	INT16	Get	Future Net	21 Scale 5
6178	40251	N	INT16	Get	Count (Qty)	22 Scale 5
6179	40252	Y	INT32	Get/Set	Gross Total	23 Scale 5
6180	40254	Y	INT32	Get/Set	Net Total	24 Scale 5
6181	40256	Y	INT32	Get/Set	# Accumulations	25 Scale 5
6182	40258	N	INT32	Get/Set	Count Total (Qty)	26 Scale 5
6185	40260	Y	INT16	Get	Gross	0 Scale 6

6186	40261	Y	INT16	Get	Net	1 Scale 6
6187	40262	Y	INT16	Get/Set	Tare	2 Scale 6
6188	40263	Y	INT16	Get	Average Gross	15 Scale 6
6189	40264	Y	INT16	Get	Average Net	16 Scale 6
6190	40265	Y	INT16	Get	Peak Gross	17 Scale 6
6191	40266	Y	INT16	Get	Peak Net	18 Scale 6
6192	40267	.01 Sec.	INT16	Get/Set	Free Fall	19 Scale 6
6193	40268	Y	INT16	Get	Future Gross	20 Scale 6
6194	40269	Y	INT16	Get	Future Net	21 Scale 6
6195	40270	N	INT16	Get	Count (Qty)	22 Scale 6
6196	40271	Y	INT32	Get/Set	Gross Total	23 Scale 6
6197	40273	Y	INT32	Get/Set	Net Total	24 Scale 6
6198	40275	Y	INT32	Get/Set	# Accumulations	25 Scale 6
6199	40277	N	INT32	Get/Set	Count Total (Qty)	26 Scale 6
6198	40279	Y	INT16	Get	Gross	0 Scale 7
6199	40280	Y	INT16	Get	Net	1 Scale 7
6200	40281	Y	INT16	Get/Set	Tare	2 Scale 7
6201	40282	Y	INT16	Get	Average Gross	15 Scale 7
6202	40283	Y	INT16	Get	Average Net	16 Scale 7
6203	40284	Y	INT16	Get	Peak Gross	17 Scale 7
6204	40285	Y	INT16	Get	Peak Net	18 Scale 7
6205	40286	.01 Sec.	INT16	Get/Set	Free Fall	19 Scale 7
6206	40287	Y	INT16	Get	Future Gross	20 Scale 7
6207	40288	Y	INT16	Get	Future Net	21 Scale 7
6208	40289	N	INT16	Get	Count (Qty)	22 Scale 7
6209	40290	Y	INT32	Get/Set	Gross Total	23 Scale 7
6210	40292	Y	INT32	Get/Set	Net Total	24 Scale 7
6211	40294	Y	INT32	Get/Set	# Accumulations	25 Scale 7
6212	40296	N	INT32	Get/Set	Count Total (Qty)	26 Scale 7
6213	40298	Y	INT16	Get	Gross	0 Scale 8
6214	40299	Y	INT16	Get	Net	1 Scale 8
6215	40300	Y	INT16	Get/Set	Tare	2 Scale 8
6216	40301	Y	INT16	Get	Average Gross	15 Scale 8
6217	40302	Y	INT16	Get	Average Net	16 Scale 8
6218	40303	Y	INT16	Get	Peak Gross	17 Scale 8
6219	40304	Y	INT16	Get	Peak Net	18 Scale 8
6220	40305	.01 Sec.	INT16	Get/Set	Free Fall	19 Scale 8
6221	40306	Y	INT16	Get	Future Gross	20 Scale 8
6222	40307	Y	INT16	Get	Future Net	21 Scale 8
6223	40308	N	INT16	Get	Count (Qty)	22 Scale 8
6224	40309	Y	INT32	Get/Set	Gross Total	23 Scale 8
6225	40311	Y	INT32	Get/Set	Net Total	24 Scale 8
6226	40313	Y	INT32	Get/Set	# Accumulations	25 Scale 8
6227	40315	N	INT32	Get/Set	Count Total (Qty)	26 Scale 8
6228	40317	Y	INT16	Get/Set	SP Target 33	Var 80.61
6229	40318	Y	INT16	Get/Set	SP Target 34	Var 80.62
6230	40319	Y	INT16	Get/Set	SP Target 35	Var 80.63
6231	40320	Y	INT16	Get/Set	SP Target 36	Var 80.64
6232	40321	Y	INT16	Get/Set	SP Target 37	Var 80.65
6233	40322	Y	INT16	Get/Set	SP Target 38	Var 80.66
6234	40323	Y	INT16	Get/Set	SP Target 39	Var 80.67
6235	40324	Y	INT16	Get/Set	SP Target 40	Var 80.68
6236	40325	Y	INT16	Get/Set	Var # 69	Var 80.69

6237	40326	Y	INT16	Get/Set	Var # 70	Var 80.70
6238	40327	Y	INT16	Get/Set	Var # 71	Var 80.71
6239	40328	Y	INT16	Get/Set	Var # 72	Var 80.72
6240	40329	Y	INT32	Get/Set	Var # 73	Var 80.73
6241	40331	Y	INT32	Get/Set	Var # 74	Var 80.74
6242	40333	N	IEE Float	Get/Set	Var # 75	Var 80.75
6243	40335	Y	INT16	Get/Set	SP Target 41	Var 80.76
6244	40336	Y	INT16	Get/Set	SP Target 42	Var 80.77
6245	40337	Y	INT16	Get/Set	SP Target 43	Var 80.78
6246	40338	Y	INT16	Get/Set	SP Target 44	Var 80.79
6247	40339	Y	INT16	Get/Set	SP Target 45	Var 80.80
6248	40340	Y	INT16	Get/Set	SP Target 46	Var 80.81
6249	40341	Y	INT16	Get/Set	SP Target 47	Var 80.82
6250	40342	Y	INT16	Get/Set	SP Target 48	Var 80.83
6251	40343	Y	INT16	Get/Set	Var # 84	Var 80.84
6252	40344	Y	INT16	Get/Set	Var # 85	Var 80.85
6253	40345	Y	INT16	Get/Set	Var # 86	Var 80.86
6254	40346	Y	INT16	Get/Set	Var # 87	Var 80.87
6255	40347	Y	INT32	Get/Set	Var # 88	Var 80.88
6256	40349	Y	INT32	Get/Set	Var # 89	Var 80.89
6257	40351	N	IEE Float	Get/Set	Var # 90	Var 80.90
6258	40353	N	16 Bits	Get	SetPoint 33 - 48 Status	
6259	40354	Y	INT16	Get/Set	SP Target 49	Var 80.91
6260	40355	Y	INT16	Get/Set	SP Target 50	Var 80.92
6261	40356	Y	INT16	Get/Set	SP Target 51	Var 80.93
6262	40357	Y	INT16	Get/Set	SP Target 52	Var 80.94
6263	40358	Y	INT16	Get/Set	SP Target 53	Var 80.95
6264	40359	Y	INT16	Get/Set	SP Target 54	Var 80.96
6265	40360	Y	INT16	Get/Set	SP Target 55	Var 80.97
6266	40361	Y	INT16	Get/Set	SP Target 56	Var 80.98
6267	40362	Y	INT16	Get/Set	Var # 99	Var 80.99
6268	40363	Y	INT16	Get/Set	Var # 100	Var 80.100
6269	40364	Y	INT16	Get/Set	Var # 101	Var 80.101
6270	40365	Y	INT16	Get/Set	Var # 102	Var 80.102
6271	40366	Y	INT32	Get/Set	Var # 103	Var 80.103
6272	40368	Y	INT32	Get/Set	Var # 104	Var 80.104
6273	40370	N	IEE Float	Get/Set	Var # 105	Var 80.105
6274	40372	Y	INT16	Get/Set	SP Target 57	Var 80.106
6275	40373	Y	INT16	Get/Set	SP Target 58	Var 80.107
6276	40374	Y	INT16	Get/Set	SP Target 59	Var 80.108
6277	40375	Y	INT16	Get/Set	SP Target 60	Var 80.109
6278	40376	Y	INT16	Get/Set	SP Target 61	Var 80.110
6279	40377	Y	INT16	Get/Set	SP Target 62	Var 80.111
6280	40378	Y	INT16	Get/Set	SP Target 63	Var 80.112
6281	40379	Y	INT16	Get/Set	SP Target 64	Var 80.113
6282	40380	Y	INT16	Get/Set	Var # 114	Var 80.114
6283	40381	Y	INT16	Get/Set	Var # 115	Var 80.115
6284	40382	Y	INT16	Get/Set	Var # 116	Var 80.116
6285	40383	Y	INT16	Get/Set	Var # 117	Var 80.117
6286	40384	Y	INT32	Get/Set	Var # 118	Var 80.118
6287	40386	Y	INT32	Get/Set	Var # 119	Var 80.119
6288	40388	N	IEE Float	Get/Set	Var # 120	Var 80.120
6289	40390	N	16 Bits	Get	SetPoint 49 - 64 Status	
6290	40391	Y	INT16	Get/Set	SP Target 65	Var 80.121
6291	40392	Y	INT16	Get/Set	SP Target 66	Var 80.122

6292	40393	Y	INT16	Get/Set	SP Target 67	Var 80.123
6293	40394	Y	INT16	Get/Set	SP Target 68	Var 80.124
6294	40395	Y	INT16	Get/Set	SP Target 69	Var 80.125
6295	40396	Y	INT16	Get/Set	SP Target 70	Var 80.126
6296	40397	Y	INT16	Get/Set	SP Target 71	Var 80.127
6297	40398	Y	INT16	Get/Set	SP Target 72	Var 80.128
6298	40399	Y	INT16	Get/Set	Var # 129	Var 80.129
6299	40400	Y	INT16	Get/Set	Var # 130	Var 80.130
6300	40401	Y	INT16	Get/Set	Var # 131	Var 80.131
6301	40402	Y	INT16	Get/Set	Var # 132	Var 80.132
6302	40403	Y	INT32	Get/Set	Var # 133	Var 80.133
6303	40405	Y	INT32	Get/Set	Var # 134	Var 80.134
6304	40407	N	IEE Float	Get/Set	Var # 135	Var 80.135
6305	40409	Y	INT16	Get/Set	SP Target 73	Var 80.136
6306	40410	Y	INT16	Get/Set	SP Target 74	Var 80.137
6307	40411	Y	INT16	Get/Set	SP Target 75	Var 80.138
6308	40412	Y	INT16	Get/Set	SP Target 76	Var 80.139
6309	40413	Y	INT16	Get/Set	SP Target 77	Var 80.140
6310	40414	Y	INT16	Get/Set	SP Target 78	Var 80.141
6311	40415	Y	INT16	Get/Set	SP Target 79	Var 80.142
6312	40416	Y	INT16	Get/Set	SP Target 80	Var 80.143
6313	40417	Y	INT16	Get/Set	Var # 144	Var 80.144
6314	40418	Y	INT16	Get/Set	Var # 145	Var 80.145
6315	40419	Y	INT16	Get/Set	Var # 146	Var 80.146
6316	40420	Y	INT16	Get/Set	Var # 147	Var 80.147
6317	40421	Y	INT32	Get/Set	Var # 148	Var 80.148
6318	40423	Y	INT32	Get/Set	Var # 149	Var 80.149
6319	40425	N	IEE Float	Get/Set	Var # 150	Var 80.150
6320	40127	N	16 Bits	Get	SetPoint 65 - 80 Status	
6321	40428	Y	INT16	Get/Set	SP Target 81	80.151
6322	40429	Y	INT16	Get/Set	SP Target 82	80.152
6323	40430	Y	INT16	Get/Set	SP Target 83	80.153
6324	40431	Y	INT16	Get/Set	SP Target 84	80.154
6325	40432	Y	INT16	Get/Set	SP Target 85	80.155
6326	40433	Y	INT16	Get/Set	SP Target 86	80.156
6327	40434	Y	INT16	Get/Set	SP Target 87	80.157
6328	40435	Y	INT16	Get/Set	SP Target 88	80.158
6329	40436	Y	INT16	Get/Set	Var # 151	80.159
6330	40437	Y	INT16	Get/Set	Var # 152	80.160
6331	40438	Y	INT16	Get/Set	Var # 153	80.161
6332	40439	Y	INT16	Get/Set	Var # 154	80.162
6333	40440	Y	INT32	Get/Set	Var # 155	80.163
6334	40442	Y	INT32	Get/Set	Var # 156	80.164
6335	40444	N	IEE Float	Get/Set	Var # 157	80.165
6336	40446	Y	INT16	Get/Set	SP Target 89	80.166
6337	40447	Y	INT16	Get/Set	SP Target 90	80.167
6338	40448	Y	INT16	Get/Set	SP Target 91	80.168
6339	40449	Y	INT16	Get/Set	SP Target 92	80.169
6340	40450	Y	INT16	Get/Set	SP Target 93	80.170
6341	40451	Y	INT16	Get/Set	SP Target 94	80.171
6342	40452	Y	INT16	Get/Set	SP Target 95	80.172
6343	40453	Y	INT16	Get/Set	SP Target 96	80.173
6344	40454	Y	INT16	Get/Set	Var # 174	80.174
6345	40455	Y	INT16	Get/Set	Var # 175	80.175
6346	40456	Y	INT16	Get/Set	Var # 176	80.176
6347	40457	Y	INT16	Get/Set	Var # 177	80.177

6348	40458	Y	INT32	Get/Set	Var # 178	80.178
6349	40460	Y	INT32	Get/Set	Var # 179	80.179
6350	40462	N	IEE Float	Get/Set	Var # 180	80.180
6351	40464	N	16 Bits	Get	SetPoint 81 - 96 Status	
6352	40465	Y	INT16	Get/Set	SP Target 97	80.181
6353	40466	Y	INT16	Get/Set	SP Target 98	80.182
6354	40467	Y	INT16	Get/Set	SP Target 99	80.183
6355	40468	Y	INT16	Get/Set	SP Target 100	80.184
6356	40469	Y	INT16	Get/Set	SP Target 101	80.185
6357	40470	Y	INT16	Get/Set	SP Target 102	80.186
6358	40471	Y	INT16	Get/Set	SP Target 103	80.187
6359	40472	Y	INT16	Get/Set	SP Target 104	80.188
6360	40473	Y	INT16	Get/Set	Var # 189	80.189
6361	40474	Y	INT16	Get/Set	Var # 190	80.190
6362	40475	Y	INT16	Get/Set	Var # 191	80.191
6363	40476	Y	INT16	Get/Set	Var # 192	80.192
6364	40477	Y	INT32	Get/Set	Var # 193	80.193
6365	40479	Y	INT32	Get/Set	Var # 194	80.194
6366	40481	N	IEE Float	Get/Set	Var # 195	80.195
6367	40483	Y	INT16	Get/Set	SP Target 105	80.196
6368	40484	Y	INT16	Get/Set	SP Target 106	80.197
6369	40485	Y	INT16	Get/Set	SP Target 107	80.198
6370	40486	Y	INT16	Get/Set	SP Target 108	80.199
6371	40487	Y	INT16	Get/Set	SP Target 109	80.200
6372	40488	Y	INT16	Get/Set	SP Target 110	80.201
6373	40489	Y	INT16	Get/Set	SP Target 111	80.202
6374	40490	Y	INT16	Get/Set	SP Target 112	80.203
6375	40491	Y	INT16	Get/Set	Var # 204	80.204
6376	40492	Y	INT16	Get/Set	Var # 205	80.205
6377	40493	Y	INT16	Get/Set	Var # 206	80.206
6378	40494	Y	INT16	Get/Set	Var # 207	80.207
6379	40495	Y	INT32	Get/Set	Var # 208	80.208
6380	40497	Y	INT32	Get/Set	Var # 209	80.209
6381	40499	N	IEE Float	Get/Set	Var # 210	80.210
6382	40501	N	16 Bits	Get	SetPoint 97 - 112 Status	
6383	40502	Y	INT16	Get/Set	SP Target 113	80.211
6384	40503	Y	INT16	Get/Set	SP Target 114	80.212
6385	40504	Y	INT16	Get/Set	SP Target 115	80.213
6386	40505	Y	INT16	Get/Set	SP Target 116	80.214
6387	40506	Y	INT16	Get/Set	SP Target 117	80.215
6388	40507	Y	INT16	Get/Set	SP Target 118	80.216
6389	40508	Y	INT16	Get/Set	SP Target 119	80.217
6390	40509	Y	INT16	Get/Set	SP Target 120	80.218
6391	40510	Y	INT16	Get/Set	Var # 219	80.219
6392	40511	Y	INT16	Get/Set	Var # 220	80.220
6393	40512	Y	INT16	Get/Set	Var # 221	80.221
6394	40513	Y	INT16	Get/Set	Var # 222	80.222
6395	40514	Y	INT32	Get/Set	Var # 223	80.223
6396	40516	Y	INT32	Get/Set	Var # 224	80.224
6397	40518	N	IEE Float	Get/Set	Var # 225	80.225
6398	40520	Y	INT16	Get/Set	SP Target 121	80.226
6399	40521	Y	INT16	Get/Set	SP Target 122	80.227
6400	40522	Y	INT16	Get/Set	SP Target 123	80.228
6401	40523	Y	INT16	Get/Set	SP Target 124	80.229
6402	40524	Y	INT16	Get/Set	SP Target 125	80.230

Run Macro	Number to pass	Also Simulates Front Panel Keypress	ON / READ	
1	301	[F1]	**ON- This will queue the macro on the stack. It will run when it reaches the top of the stack. Immediately if no other macros are running.(or queued ahead of it)**	**READ Status-** Whenever the macro is called with this method, **reading the instance will return "on" if the macro has not yet run or is running. It will return "off" if the macro has completed.**
2	302	[F2]		
3	303	[F3]		
4	304	[F4]		
5	305	[F5]		
6	306	[Start]		
7	307	[Stop]		
8	308	[Setup]		
9 - 250	309-550	N/A	Add 300 to the macro number you wish to run or get status on (I.e. Pass 301 for macro 1).	

Scale Specific Commands

Zero Scale 1	**All of these functions are affected by the Motion Inhibit Settings. When one of these functions is called, reading the instance will return true until the function completes. In addition to this, the <u>motion inhibited status bit,</u> in the scale status word, is set until all motion inhibited functions have completed.**
Zero Scale 2	
Zero Scale 3	
Zero Scale 4	
Zero Scale 5	
Zero Scale 6	
Zero Scale 7	
Zero Scale 8	
Tare Scale 1	
Tare Scale 2	
Tare Scale 3	
Tare Scale 4	
Tare Scale 5	
Tare Scale 6	
Tare Scale 7	
Tare Scale 8	
Accumulate Scale 1	
Accumulate Scale 2	
Accumulate Scale 3	
Accumulate Scale 4	
Accumulate Scale 5	
Accumulate Scale 6	
Accumulate Scale 7	
Accumulate Scale 8	

The following do not require preassigned Data Registers

Number to pass	Scale Specific Commands
601	Zero Scale 1
602	Zero Scale 2
603	Zero Scale 3
604	Zero Scale 4
605	Zero Scale 5
606	Zero Scale 6
607	Zero Scale 7
608	Zero Scale 8
621	Tare Scale 1
622	Tare Scale 2
623	Tare Scale 3
624	Tare Scale 4
625	Tare Scale 5
626	Tare Scale 6
627	Tare Scale 7
628	Tare Scale 8
641	Accumulate Scale 1
642	Accumulate Scale 2
643	Accumulate Scale 3
644	Accumulate Scale 4
645	Accumulate Scale 5
646	Accumulate Scale 6
647	Accumulate Scale 7
648	Accumulate Scale 8

All of these functions are affected by the Motion Inhibit Settings. When one of these functions is called, reading the instance will return true until the function completes. In addition to this, the motion inhibited status bit, in the scale status word, is set until all motion inhibited functions have completed. Zero scale is macro #601-608. Tare scale is macro #621-628. Accumulate scale is macro #641-648.

1.8 ADVANCED PROFIBUS SETUP AND MODIFICATION

1.8.1 How to Modify the Profibus Register Map

Care must be taken when modifying the register map. All data structures above an incorrect change will be shifted. The best case in this situation is that the Indicator will return an error. Worst case is that it will return the wrong information.

Each of the data registers (identified as 40001 and up) hold two bytes of data. Parameters can use one, two, or nine registers, depending on the data type.

A 16 bit integer, a scale status word and a block of setpoint status are each 2 bytes long and each would require one data register for storage. i.e. The first two bytes of I/O Assy #1, **40001** is a 16 bit integer representation of the gross weight on scale #1 in graduations. (engineering units)

A 32 bit integer and an IEEE Float are 4 bytes long and require 2 consecutive data registers for storage. The data registers in the chart in ***Italic*** indicate the second register used by 32 bit values. i.e. The four bytes of I/O Assy #9, **40033 and 40034** are a 32 bit floating point representation of the gross weight on scale #1.

Any 2 byte value can replace another 2 byte value and any 4 byte value can replace another 4 byte value and 6xxx (scale param 6xxx) **will still match the chart above.**

I/O Assys Only: **Two** 2 byte values can replace **one** 4 byte value and **one** 4 byte value can replace **two** 2 byte values. It takes two 16 bit 6xxx entries to replace one 32 bit 6xxx entry and it takes one 32 bit 6xxx entry to replace two 16 bit 6xxx entries. *** 6xxx (scale param) **WILL NO LONGER MATCH** the chart above as a different number of 6xxx entries will be used to complete the change.
Any combination of data that **exactly** fills the number of bytes in **the I/O Assembly that will be used** would be an acceptable configuration. Example of using I/O Assy 3 (24 Bytes total): One 20 Char String and one 4 Byte Number (Float or 32 bit Int) would correctly fill the I/O Assy. In this case I/O Assys #1 and #2 would no longer be useable by Profibus as the data crosses the assembly boundaries.

*****Make changes to the I/O Assemblies only in cooperation with the programmer of the Master. Changes made otherwise are useless at best!**

1.8.1.1 Examples of replacing Register Values

A. Example of replacing a 2 byte value with another 2 byte value.

This example will replace the 6002 / 40002 Value. This is part of I/O assemblies 1 – 8.

	DO	RESULT / COMMENT
1	Do 6002 [SELECT]	"FutGr"
2	[CLEAR]	The scale is always in the insert mode for these parameters. You must press [CLEAR] to delete values when replacing them. Failure to do this will cause all higher values to shift upward in the registers and will be in unknown locations.
3	[ENTER]	"Parm=Gross"
4	KEY IN THE PARAMETER NUMBER (WE WILL USE 1) [ENTER]	"Okay? Net"
5	[ENTER]	"Inst 0 Net"
6	Key in the instance (we will use scale 1) [ENTER]	"Okay1 Net"
7	[ENTER]	'Set Format" then "Float" (the type of variable) "Int16"(the type to convert to for the Profibus)

8	0 [ENTER] 16 Bit Integer (we will enter this) 1 [ENTER] 32 Bit Integer 2 [ENTER] IEEE Float – If the scale internally treats a value as an integer, the IEEE float selection will not be accepted. (**Note: Count is treated as a Float) 3 [ENTER] String – only accepted if parameter is a string type	"Float Int16"
9	[ENTER]	"PFbus Net"
	You have finished the entry.	
*	Press the Down Arrow Key.	"40002 Int16"
*	Press [SELECT] until the 4xxxx value is 40049.	"40049 Int16"
*	Press the Down Arrow Key.	"PFbus APW" ***Whenever you complete a change, you should always go to this Register Number and make sure that 40049 and APW match. This is an indication that you have used the correct number of registers. The indicator will not allow you to exit setup if an I/O Boundary is incorrect.***

B. Example of replacing a 4 byte value with another 4 byte value

This example will replace the 6034 / 40035 and 40036 Value. This is part of I/O assemblies 10 – 16.

	Do	Result / Comment
1	Do 6034 [SELECT]	"RndGr"
2	[CLEAR] In this case you will be deleting 2 registers. 40035 and 40036. **The deletion will not be obvious because the next register is also set up as Rounded Gross and the display will look the same.**	The scale is always in the insert mode for these parameters. **You must press [CLEAR] to delete values when replacing them. Failure to do this will cause all higher values to shift upward in the registers and will be in unknown locations.**
3	[ENTER]	"Parm=Gross"
4	Key in the parameter number (we will use 21) [ENTER]	"Okay? RndNt"
5	[ENTER]	"Inst 0 RndNt"
6	Key in the instance (we will use scale 1) [ENTER]	"Okay1 RndNt"
7	[ENTER]	'Set Format" then "Float" (the type of variable) "Int16"(the type to convert to for the Profibus)
8	1 [ENTER] 32 Bit Integer **(enter this only)** 0 [ENTER] 16 Bit Integer 2 [ENTER] IEEE Float – If the scale internally treats a value as an integer, the IEEE float selection will not be accepted. (**Note: Count is treated as a Float) 3 [ENTER] String – only accepted if parameter is a string type	"Float Int32"
9	[ENTER]	"PFbus RndNt"
	YOU HAVE FINISHED THE ENTRY.	
*	Press the Down Arrow Key.	"40035 Int32"
*	Press [SELECT] until the 4xxxx value is 40049.	"40049 Int16"
*	Press the Down Arrow Key.	"PFbus APW" ***Whenever you complete a change, you should always go to this Register Number and make sure that 40049 and APW match. This is an indication that you have used the correct number of registers.*** ***The indicator will not allow you to exit setup if an I/O Boundary is incorrect.***

***** IMPORTANT:** The **above two procedures may** also be used to replace data in the higher numbered registers used for "Explicit Messaging". **You may not use the following two** examples to change data used for "Explicit Messaging" *(boundary changes would make some data unavailable to Profibus*).

C. Example of replacing a 4 byte value with Two 2 byte values (I/O Assemblies Only)

This example will replace the 6033 / 40033 and 40034 Value. This is part of I/O assemblies 9 – 16.
We will replace the 32 Bit IEE Float of Rounded Gross with a 16 Bit Gross and 16 Bit Net.

	Do	**Result / Comment**
1	Do 6033 [SELECT]	"RndGr"
2	[CLEAR] In this case you will be deleting 2 registers. 40033 and 40034. **The deletion will not be obvious because the next register is also set up as Gross and the display will look the same.**	The scale is always in the insert mode for these parameters. **You must press [CLEAR] to delete values when replacing them. Failure to do this will cause all higher values to shift upward in the registers and will be in unknown locations.**
3	[ENTER]	"Parm=Gross"
4	Key in the parameter number (we will use 0) [ENTER]	"Okay? Gross"
5	[ENTER]	"Inst 0 Gross"
6	Key in the instance (we will use scale 1) [ENTER]	"Okay1 Gross"
7	[ENTER]	'Set Format" then "Float" (the type of variable) "Int16"(the type to convert to for the Profibus)
8	0 [ENTER] 16 Bit Integer	"Float Int16"
9	[ENTER]	"PFbus Gross"
10	[ENTER]	"Parm=Gross"
11	Key in the parameter number (we will use 1) [ENTER]	"Okay? Net"
12	[ENTER]	"Inst 0 Gross"
13	Key in the instance (we will use scale 1) [ENTER]	"Okay1 Net"
14	[ENTER]	'Set Format" then "Float" (the type of variable) "Int16"(the type to convert to for the Profibus)
15	0 [ENTER] 16 Bit Integer	"Float Int16"
16	[ENTER]	"PFbus Net"
	You have finished the entry.	
*	Press the Down Arrow Key.	"40034 Int16"
*	Press [SELECT] until the 4xxxx value is 40049.	"40049 Int16"
*	Press the Down Arrow Key.	"PFbus APW" ***Whenever you complete a change, you should always go to this Register Number and make sure that 40049 and APW match. This is an indication that you have used the correct number of registers. The indicator will not allow you to exit setup if an I/O Boundary is incorrect.***
**	**Once you have replaced 4 byte data with two 2 byte pieces or two 2 byte pieces with one 4 byte, the 6xxx numbers will no longer match the chart. The register number of the first piece of information at each I/O Assy choice will always be the same if no errors are made.**	

D. Example of replacing Two 2 byte values with One 4 byte value (I/O Assemblies Only)

This example will replace the 6009-10 / 40009-10 Value. This is part of I/O assemblies 3 – 8.
We will replace 16 Bit Gross and 16 Bit Future Gross with IEE Float of Rounded Net.

	Do	**Result / Comment**
1	Do 6009 [SELECT]	"Gross"

2	**[CLEAR][CLEAR]** We press clear twice to remove two 16 bit registers.	The scale is always in the insert mode for these parameters. **You must press [CLEAR] to delete values when replacing them. Failure to do this will cause all higher values to shift upward in the registers and will be in unknown locations.**
3	[ENTER]	"Parm=Gross"
4	Key in the parameter number (we will use 21) 1 [ENTER]	"Okay? RndNt"
5	[ENTER]	"Inst 0 RndNt"
6	Key in the instance (we will use scale 1) 1 [ENTER]	"Okay1 RndNt"
7	[ENTER]	'Set Format" then "Float" (the type of variable) "Int16"(the type to convert to for the Profibus)
8	**2 [ENTER] IEEE Float (we will enter this)** – *If the scale internally treats a value as an integer, the IEEE float selection will not be accepted. (**Note: Count is treated as a Float)* (You would enter a 1 for a 32 Bit Integer)	"Float FltIE"
9	[ENTER]	"PFbus RndNt"
	You have finished the entry.	
*	Press the Down Arrow Key.	"40009 Int32"
*	Press [SELECT] until the 4xxxx value is 40049.	"40049 Int16"
*	Press the Down Arrow Key.	"PFbus APW" ***Whenever you complete a change, you should always go to this Register Number and make sure that 40049 and APW match. This is an indication that you have used the correct number of registers. The indicator will not allow you to exit setup if an I/O Boundary is incorrect.***
**	**Once you have replaced 4 byte data with two 2 byte pieces or two 2 byte pieces with one 4 byte, the 6xxx numbers will no longer match the chart. <u>The register number of the first piece of information at each I/O Assy choice will always be the same if no errors are made.</u>**	

***Make changes to the I/O Assemblies only in cooperation with the programmer of the Master. Changes made otherwise are useless at best!

1.8.2 Reclaiming Setup Memory

You may reduce the scale memory requirements for a specific Profibus application to free more memory for scale programming providing the needed Profibus information is not removed.

1. **You may not remove any I/O Assembly Registers**.
2. Look in Section 1.7 Explicit Messaging Addressing to find the highest numbered Explicit Message Register that would be required by your application.
3. Go to one register higher than this register in the scales P6xxx settings.
4. Hold down the clear key until the scale displays clear all then press clear one more time.
5. Note the Var # of the Explicit Message Register chosen in the Table (or the var# closest to the register chosen) . You may change the number of Vars at Param 680 down to this number or any value higher than this number.
6. You **should** disable any unused setpoints.

1.9 SETPOINTS

The following information describes how the Profibus Script sets up the Profibus Setpoint interfaces. You will need an understanding of this to modify the setpoint compare parameters for systems that are not using the gross weight of scale one to control setpoint output. After the initial Profibus Setup, the setpoints may be reconfigured however needed, including as inputs and Disabled. Profibus will still be able to read their status.

All of these setpoints are configured as "Never On", "Deactive Above" and compare to "Gross of Scale #1". The number of the setpoints and their target variables follows: (options are required to physically connect to these).

TABLE 1.9 SETPOINT TARGET VARIABLES

Model	Setpoint Numbers	Target Variables	Target Variable Names
46x	1-8	80.1 – 80.8	Trg1 – Trg8
56x ***Plus the Above 4xx Setpoints***	9-16 17 – 24 25 – 32	80.16 –80.23 80.31 – 80.38 80.46 – 80.53	Trg9 – Trg16 Trg17 – Trg24 Trg25 – Trg32
66x ***Plus the Above 4xx Setpoints and the 5xx Setpoints***	33 – 40 41 – 48 49 – 56 57 – 64 65 – 72 73 – 80 81 – 88 89 – 96 97 – 104 105 – 112 113 – 120 121 – 128 131 - 138	80.61 – 80.68 80.76 – 80.83 80.91 – 80.98 80.106 – 80.113 80.121 – 80.128 80.136 – 80.143 80.151 – 80.158 80.166 – 80.173 80.181 – 80.188 80.196 – 80.203 80.211 – 80.218 80.226 – 80.233 80.243 – 80.250	Trg33 – Trg40 Trg41 – Trg48 Trg49 – Trg56 Trg57 – Trg64 Trg65 – Trg72 Trg73 – Trg80 Trg81 – Trg88 Trg89 – Trg96 Trg97 – Trg99, Tg100 – Tg104 Tg105 – Tg112 Tg113 – Tg120 Tg121 – Tg128 Tg131 – Tg138 (PDIO Outputs) ***Note: 138 will NOT automatically be enabled as an output as its chanel is used to drive 66x **LCD** indicators when Profibus is Enabled. You may enable it manually if the Display is NOT Liquid Crystal.

WARNING* Do not change the length of the strings of Var# 15, 30 ,45 and 60 or change a Numeric Var to a String. Doing so will shift all data stored above the Var and make the registers above this Var Profibus registers unuseable by Profibus. See "Advanced Profibus Setup and Modification".**

1.10 INFORMATIONAL PARAMETERS FOR PROFIBUS

1.10.1 P60000 FRam (E^2) Available

The top 2x5 prompt for a unit with FRam will be FrInst and FrAvl, E2Ins and E2Avl for a unit with E^2 .

Indicator	46x	56x	66x
P60000	4096	8192	8192 expandable to 16384, expansion is required for Profibus (8176, 16352 for E2)
FRam Used for Profibus	627 Default + **1182 for Profibus** = 1809	721 Default **+ 3340 for Profibus** = 4061	1045 Default + **11461 for Profibus** = 12056
P60001 Fram Available	4131	2287	3878 (3846 E2)

The Memory requirement to integrate a custom program into a Profibus setup can be calculated by first loading your custom program and reading the ram available parameter, 60001. Add the additional Profibus memory required **for Profibus** listed in the above table. You will undoubtably have some overlap (var and setpoint setups in common would use less memory) but could safely assume that manually integrating your program into a Profibus script will fit into the memory calculated. Refer to 0 for information about reclaiming Setup Memory.

The 660 Series Indicators only, require a second 8 K Fram (EEProm on older units) to be installed to provide enough memory to complete this setup.

1.10.2 Profibus Specific

P60300	Network Interface Adapter Version & Serial Number P60300 only exists if a Profibus or Device Net network is installed and setup. On initial selection it displays "PFbus Ser #". After a second, the display will be updated. The top line of the 2 x 5 display will then show Vnnnn. Where nnnn is the version of the firmware in the Network Interface Adapter. The second line will show #0000.
P60301	PFbus Map Check. If this display shows STD then all original register setups are in the scripted Profibus configuration and the scale has the standard register setup. **IF IT DOESN'T SAY STD:** Press the enter key and the scale will momentarily display Reg Check and either STD or a number. If this displays STD, the register boundaries are correct and the scale will operate with data that has been replaced with acceptable data. A number will indicate that register boundaries have been changed in some way but not necessarily in a bad way. See "Using P60301 Reg Check" to verify correct setup changes.

1.10.2.1 Using P60301 Reg Check

Acceptable Reasons Reg Check would indicate a number.	1. Information in the data registers has been replaced by other correctly sized information. Two bytes for two bytes, four bytes for four bytes, two 2 byte registers for one four byte or one 4 byte registers for two 2 byte. 2. The register setup has been trimmed down to recapture setup memory. Unneeded Registers higher than the I/O Assys may be removed. **This may only be done starting with the highest register, working downward**. If any register is removed from the middle of the setup, all higher registers will be shifted and incorrect data or an error will be returned.
Unacceptable Reasons Reg Check would indicate a number.	1. Incorrectly sized data has been placed in the data setup. 2. A register has been deleted from the middle of the data. All higher data is now shifted downward. 3. A register has been inserted into the middle of the data. All higher registers are now shifted upward.
*****Important**	**Incorrect modification of the *I/O Assemblies*. This is an unacceptable error and the scale will not allow you to save this setup configuration.**

WARNING* Do not change the length of the strings of Var# 15, 30 ,45 and 60 or change a Numeric Var to a String. Doing so will shift all data stored above the Var and make the registers above this Var Profibus registers unuseable by Profibus. See "Advanced Profibus Setup and Modification"**

1.11 New Weigh Mode Parameter 95 – Network Status, Option Board LEDs

This parameter is accessed from the Weigh Mode. Press 95 [SELECT].
It allows viewing of network status information without opening the enclosure to see the Option Board Status LEDs or putting the scale into Setup. It is possible that an error could occur that would not update this display. To force a re-initialization with the network: 1. Press [CLR] – the display will change to 00000. 2. Wait for the numbers to change again – This may take several seconds, up to the setting of P213 (Ptime).

Table 1.11 Defines the meaning of each digit displayed at parameter 95 "Profibus Stat".

New Scale Parameter 95 (95 Select from the weigh mode)		
I/O Module		
000 **00** PFbus Stat		
Values from 1 to 16		
Network Status		
00 **0** 00 PFbus Stat	**LEDs**	
0=No Connection to Host	Off	
1=Connected to Host	Green	
Network Error Status		
0 **0** 000 PFbus Stat	**LEDs**	
0=No Error	Off	
1=Error Detected	Flashing Red	
Module Status		
0 0000 PFbus Stat	**LEDs**	
0=non-existant (no power or not running, can't report this)	Off	
1=Configuring	Solid Green	
2=Operational	Flashing Green	Talking to indicator, steady flash. Not talking to indicator (also in setup), flashing with pauses.
3=Error	Red	

1.12 NEW SCALE PARAMETER P96

This parameter is not a selectable parameter. It is however accessible through macros and over the profibus network.

TABLE 1.12 BIT DEFINITION FOR SCALE PARAMETER 96

	Scale Status and Scaling (16 Bits)				
LSB	Overload				
2	Underload				
3	Motion				
4	Center of Zero				
5	Motion Delayed Command Running				
6	Neg. Gross Polarity				
7	Negative Net Polarity				
8	A / D Error				
9..13	**Scale Graduation Size** (Engineering Units)				
0	0.00001	9	0.01	18	10
1	0.00002	10	0.02	19	20
2	0.00005	11	0.05	20	50
3	0.0001	12	0.1	21	100
4	0.0002	13	0.2	22	200
5	0.0005	14	0.5	23	500
6	0.001	15	1		
7	0.002	16	2		
8	0.005	17	5		
14..15	Future				
MSB	Future				

INTEGERS:
DEFAULT SETUP: With the exceptions of Count, #Accums, Status (Int / UInt) and free-fall time (always .01 sec.), All Integer / UInt values are returned as the number of graduations based on the calibration setting for scale #1 (P111). (this default setup can be modified)

1.13 COMMUNICATING WITH THE DEVICE

Communications between the master and device involves sending defined requests for data and receiving defined responses. The following sections show the format of these requests and responses, as well as a few examples.

1.13.1 OUTPUT DATA

The Profibus Output Data consists of 8 bytes or 22 bytes:

a) A byte with a value of 1-16 that specifies which I/O Assembly to send.
b) A byte that acts as a Command Byte with the following two nibbles:
   i) Low Nibble
      (1) 0 Non command, clear out explicit message area
      (2) 1 for a read register
      (3) 2 for a write register
      (4) 3 for read coil or get macro status
      (5) 4 for force coil or run macro
      (6) 5 Identify Indicator Model Number Index
   ii) High Nibble
      (1) Number of words to send
         (a) only values 1-9 are allowed for Read Register or Write Register
         (b) only values 1-2 are allowed for Identify Indicator Model Index, Force Coil, or Read Coil. (a 1 means the number is a word value, and a 2 means the number is a Motorola format long)
c) A word to specify the register number or coil number or macro number. Registers start at 40001, Coils at 1, and Macros at 301 or 601. See the predefined register map (Table 1.7).
d) The Explicit Message Data area (Motorola format) consisting of :
   i) A Int 16/32 containing the data to be used with a write register, force coil or run macro command. Int 16 are to be located in bytes 5 & 6 with 7 & 8 zeroed.
   ii) Eighteen bytes (14 bytes in addition to the four bytes from item i) above). Unused bytes will be filled with zeroes.

TABLE 1.13.1 OUTPUT DATA FORMAT

<table>
<tr><th colspan="6">Output Data (8 or 22 bytes)</th></tr>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3 & 4</th><th>Bytes 5 & 6</th><th>Bytes 7 & 8</th><th>Bytes 9-22</th></tr>
<tr><td rowspan="3">Requested I/O Assembly</td><td rowspan="3">Command</td><td rowspan="3">Register or Coil (word)</td><td>Int 16</td><td>Zero</td><td rowspan="2">Filled with zeroes</td></tr>
<tr><td colspan="2">Int 32 Motorola Format</td></tr>
<tr><td colspan="3">String (unused bytes zeroed)</td></tr>
</table>

NOTE: To send the same explicit command repeatedly, a different command (like command zero) must be sent between each occurance.

1.13.2 INPUT DATA

The Profibus Input Data will consist of two sections:

a) The Status & Explicit Data Section. This can be either 8 or 22 bytes
   i) A byte with a value of 1-16 that echoes the I/O Assembly that is currently being sent (0=>none). The I/O Assembly is being continuously updated by the Interface Adapter.
   ii) A byte that acts as a Command Byte with the following values (when the Interface Adapter has the new data):
      (1) 0 Ready for next command (required when one wants to reissue the same explicit command)
      (2) 1 in the lower nibble and 1-9 in the upper nibble - for a Read Register completed.
      (3) 2 in the lower nibble and 1-9 in the upper nibble - for a Write Register completed.
      (4) 3 in the lower nibble and 1-2 in the upper nibble – for a Force Coil or macro run completed.
      (5) 4 in the lower nibble and 1-2 in the upper nibble - for Read Coil or check macro running completed.
      (6) 5 in the lower nibble and 1-2 in the upper nibble - Identify Scale Model Index (1=660, 2=5500, 3=560, 4=460)
      (7) 255 for an error
   iii) A word to specify the register number, coil number or macro number. See the predefined register map used (Table 1.7).
   iv) The Explicit Message Data area (Motorola format). If the Output Command Byte does not require data from the Interface Adapter the Explicit Data area of the Output data will be echoed

(1) A Motorola Int 16/32 containing the data to received. Int 16 are to be located in bytes 5 & 6 with 7 & 8 zeroed. If a error exists the command byte (item ii above) is set to 255 and the Int 32 will consist of:
   (a) MS word: The Output command byte that produced the error (both nibbles)
   (b) LS word: The actual error code (See item 1.13.4)
(2) Eighteen bytes (14 bytes in addition to the four bytes from item i) above). Unused bytes will be filled with zeroes.

b) The I/O Assembly Section which can be one of eight sizes consisting of 16, 24, 32, 40, 48, 56, 64 or 72 bytes. The data in this section will be stored in Motorola format. The unneeded data bytes will filled with zeroes and if the I/O Assembly requested is too large for the Module size selected then it will simply be truncated.

TABLE **1.13.2 INTPUT DATA FORMAT**

<table>
<tr><th colspan="6">Input Data</th></tr>
<tr><th colspan="6">Status & Explicit Data Section (8 or 22 bytes corresponding to output data)</th></tr>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3 & 4</th><th>Bytes 5 & 6</th><th>Bytes 7 & 8</th><th>Bytes 9-22</th></tr>
<tr><td rowspan="3">Requested I/O Assembly</td><td rowspan="3">Command</td><td rowspan="3">Register or Coil (word)</td><td>Int 16</td><td>Zero</td><td rowspan="2">Filled with zeroes</td></tr>
<tr><td colspan="2">Int 32 Motorola Format</td></tr>
<tr><td colspan="3">String (unused bytes zeroed)</td></tr>
<tr><th colspan="6">I/O Assembly Data Section</th></tr>
<tr><td colspan="6">I/O Assembly Data (balanced filled with zeroes or truncated dependent on module size selected) (8, 16, 24, 32, 40, 48, 56, or 64 bytes dependent on Module selected in GDS)</td></tr>
</table>

NOTE: The I/O Assembly and command requested in the output data will be echoed in the input data if they are accepted. Bytes 3-8 may also be echoed.

1.13.3 COMMUNICATION EXAMPLES

1.13.3.1 Get I/O Assembly data for one scale

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0	0	
3	0	0	
4	0	0	
5	0	0	
6	0	0	
7	0	0	
8	0	0	
9		Gw	**I/O DATA** Word, gross weight
10		Gw	
11		Fg	Word, future gross
12		Fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

Note: Issuing a command of zero causes all bytes between the Assembly# to the I/O Data to be returned as zero. This effectively clears out the explicit message area.

Note: see table 1.6 and table 1.12 for polled data information

1.13.3.2 Get I/O Assembly data for two scales

(using i/o module 2 (8 out, 24 in), and i/o assembly 2)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x02	0x02	Byte, Assembly#
2	0	0	
3	0	0	
4	0	0	
5	0	0	
6	0	0	
7	0	0	
8	0	0	
9		Gw	**I/O DATA** Word, gross weight
10		Gw	
11		Fg	Word, future gross
12		Fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	
17		Gw	**I/O DATA** Word, gross weight
18		Gw	
19		Fg	Word, future gross
20		Fg	
21		R	Word, rate
22		R	
23		Scal/stat	Word, bit structure, scale & status
24		Scal/stat	

1.13.3.3 Request model number of indicator

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0x15	0x15	Byte, Cmd
3	0	0	
4	0	0	
5	0	0x00	Word, model#, 660
6	0	0x01	
7	0	0	
8	0	0	
9		Gw	**I/O DATA** Word, gross weight
10		Gw	
11		Fg	Word, future gross
12		Fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.4 Read a Register (word)

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0x11	0x11	Byte, Cmd
3	0x9c	0x9c	Word, register#
4	0x41	0x41	
5	0	0xVV	Word, register value
6	0	0xVV	
7	0	0	
8	0	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.5 Read a Register (long)

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0x21	0x21	Byte, Cmd
3	0x9c	0x9c	Word, register#
4	0x41	0x41	
5	0	0xVV	Long, register value
6	0	0xVV	
7	0	0xVV	
8	0	0xVV	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.6 Write a Register (word)

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0x12	0x12	Byte, Cmd
3	0x9c	0x9c	Word, register#
4	0x41	0x41	
5	0xVV	0	Word, register value
6	0xVV	0	

7	0	0	
8	0	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.7 Write a Register (long)

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x22	0x22	Byte, Cmd
3	0x9c	0x9c	Word, register#
4	0x41	0x41	
5	0xVV	0	Long, register value
6	0xVV	0	
7	0xVV	0	
8	0xVV	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.8 Read a Register (string)

(using i/o module 9 (22 out, 30 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x91	0x91	Byte, Cmd
3	0x9c	0x9c	Word, register#
4	0xb2	0xb2	
5	0	0xVV	Up to 9 words, String value
6	0	0xVV	
7	0	0xVV	
8	0	0xVV	
9	0	0xVV	
10	0	0xVV	
11	0	0xVV	
12	0	0xVV	
13	0	0xVV	
14	0	0xVV	
15	0	0xVV	
16	0	0xVV	
17	0	0xVV	
18	0	0xVV	

19	0	0xVV	
20	0	0xVV	
21	0	0xVV	
22	0	0xVV	
23		gw	**I/O DATA** Word, gross weight
24		gw	
25		fg	Word, future gross
26		fg	
27		R	Word, rate
28		R	
29		Scal/stat	Word, bit structure, scale & status
30		Scal/stat	

1.13.3.9 Write a Register (string)

(using i/o module 9 (22 out, 30 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x92	0x92	Byte, Cmd
3	0x9c	0x9c	Word, register#
4	0xb2	0xb2	
5	0xVV	0	Up to 9 words, String value
6	0xVV	0	
7	0xVV	0	
8	0xVV	0	
9	0xVV	0	
10	0xVV	0	
11	0xVV	0	
12	0xVV	0	
13	0xVV	0	
14	0xVV	0	
15	0xVV	0	
16	0xVV	0	
17	0xVV	0	
18	0xVV	0	
19	0xVV	0	
20	0xVV	0	
21	0xVV	0	
22	0xVV	0	
23		gw	**I/O DATA** Word, gross weight
24		gw	
25		fg	Word, future gross
26		fg	
27		R	Word, rate
28		R	
29		Scal/stat	Word, bit structure, scale & status
30		Scal/stat	

1.13.3.10 Turn a coil on

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x14	0x14	Byte, Cmd

3	0x00	0x00	Word, coil#
4	0x01	0x01	
5	0x00	0	Word, coil on or off
6	0x01	0	
7	0	0	
8	0	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.11 Turn a coil off

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x14	0x14	Byte, Cmd
3	0x00	0x00	Word, coil#
4	0x01	0x01	
5	0x00	0	Word, coil on or off
6	0x00	0	
7	0	0	
8	0	0	
9		Gw	**I/O DATA** Word, gross weight
10		Gw	
11		Fg	Word, future gross
12		Fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.12 Read coil status

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x13	0x13	Byte, Cmd
3	0x00	0x00	Word, coil#
4	0x01	0x01	
5	0	0xSS	Word, coil status
6	0	0xSS	
7	0	0	
8	0	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	

13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.13 Run a user created macro

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x14	0x14	Byte, Cmd
3	0x01	0x01	Word, macro# + 300
4	0x2d	0x2d	
5	0x00	0	Word, macro on
6	0x01	0	
7	0	0	
8	0	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

NOTE: user macros are 300 + macro#, which gives #301-550.

1.13.3.14 Check if user created macro running

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	**Output Data (request)**	**Input Data (response)**	**Meaning**
1	0x01	0x01	Byte, Assembly#
2	0x13	0x13	Byte, Cmd
3	0x01	0x01	Word, macro# + 300
4	0x2d	0x2d	
5	0	0xSS	Word, macro status
6	0	0xSS	
7	0	0	
8	0	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.3.15 Run a built-in macro

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0x14	0x14	Byte, Cmd
3	0x02	0x02	Word, macro# + 600
4	0x59	0x59	
5	0x00	0	Word, macro on
6	0x01	0	
7	0	0	
8	0	0	
9		Gw	**I/O DATA** Word, gross weight
10		Gw	
11		Fg	Word, future gross
12		Fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

NOTE: built-in macros are 600 + macro#, which gives #601-608 (zero), 621-628 (tare), 641-648 (accumulate).

1.13.3.16 Check if built-in macro running

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0x13	0x13	Byte, Cmd
3	0x02	0x02	Word, macro# + 600
4	0x59	0x59	
5	0	0xSS	Word, macro status
6	0	0xSS	
7	0	0	
8	0	0	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.13.4 RETURNED ERROR CODES

The Indicator can send the following codes:

a) ILLEGAL_FUNCTION = 1: Used when unsupported function is called
b) ILLEGAL_DATA_ADDRESS =2: Used when partial parm data set, exceed number of setpoints, or bad register starting address
c) ILLEGAL_DATA_VALUE=3: used when command contains invalid data when issuing force single coil or trying set a Indicator parameter that is not settable
d) SLAVE_DEVICE_FAILURE=4: Could not get the value of a selected parameter. This can be caused because a scale has been disabled or saved instead of enabled. It can be caused if the requested parameter no longer exists. It can also occur if an invalid translation type is requested for the type of parameter requested.

e) SLAVE_DEVICE_BUSY=6: Used when we enter weigh mode in response to all functions codes except 17, 18, & 19 until we receive at least one function 17
f) LENGTH_INVALID=0x100: The number of words requested (byte 2, upper nibble) is not valid. (1-9 for read register and write register, 1–2 for Identify Scale Model Index, Force or Read coil.)
g) COMMAND_INVALID=0x101: The command send (byte 2) is not valid. Only 0, 1,2,3,4, & 5 are valid.
h) REGISTER_INVALID=0x102: The requested register (byte 3&4) is not valid (only 40000-49999 are valid) or the requested coil number is not valid (only 1-999 are valid).

1.13.4.1 ERROR Trying to Read a Register (word)

The following shows an error response to an invalid explicit message request for data.

(using i/o module 1 (8 out, 16 in), and i/o assembly 1)

Byte	Output Data (request)	Input Data (response)	Meaning
1	0x01	0x01	Byte, Assembly#
2	0x11	0xff	Byte, ERROR
3	0x80	0x80	Word, register#
4	0x00	0x00	
5	0	0x00	Word, ERROR CMD
6	0	0x11	
7	0	0x00	Word, ERROR NUMBER
8	0	0x02	
9		gw	**I/O DATA** Word, gross weight
10		gw	
11		fg	Word, future gross
12		fg	
13		R	Word, rate
14		R	
15		Scal/stat	Word, bit structure, scale & status
16		Scal/stat	

1.14 TROUBLESHOOTING AND PROFIBUS RELATED ERROR MESSAGES

1.14.1 Framing or other Comm Port Errors when attempting Setup (Script or Upload).

The Profibus interface board attempts to initiate conversation with the main board at power up. This communication will cause errors in the loading of the Script or Upload. If the Profibus Board is installed and you have not yet run the Profibus Setup Script or uploaded one, Hold the [CLR] key down during power up and answer yes to the prompt to disable comm 1. You will also need to perform this before uploading a setup file through one of the other Comm Ports.

1.14.2 Error Codes ***See the 60 Series Tech Manual Appendix E for the description of other Codes

Error Code	Prompt	Description
24	NVRam Full	The scale is out of setup memory. The 66x indicators require the installation of an additional 8K FRam (or E2), for a total of 16K, to load its internal Profibus Script. If the indicator does not have this installed (go to Scale Parameter 60000 to view the installed Setup Memory), the load of the Profibus script will fail when its Setup memory is filled. The display will indicate NV Ram Full and the script will stop loading.
50	Port Clash	The indicator is attempting some other form of communication through the Profibus Port. e.g. Macro doing a write to port 1. Custom Transmit initiated to Port 1.
55	Check Parm Map	The 6000 entries for the Profibus register map has an unacceptable setup. 1. An I/O Assembly boundary has been violated. 2 An Explicit message boundary has been violated. 3. There are more registers set up in the map than the scripted amount. See 1.10.2 and 0.