2100-XX Communications Messages

Data Format and Baud Rates
Data format = 8 bits, no parity, 1 stop bit.
Baud Rates 2400,4800,9600 (default) set by dip switches/links on the station.

Single Master, Multi Slave, the stations will only send messages when they have received a message from the Master (i.e computer or PLC device).

Characters used and their ASCII values

@	40h	Start Char
:	3Ah	End of message frame
<sp>	20h	Data Separator (space)
<CR>	0dh	End of Message Indicator
NN	30-39h,41-46h	Station Number BCD
YY	30-39h,41-46h	BCC Checksum Value HEX
XX	30-39h,41-46h	2 Digit Data Value HEX (MSB first)
XXXX	30-39h,41-46h	4 Digit Data Value HEX (MSB first)
FFFFFFFF	30-39h,41-46h	IEEE Floating Point Number
,	2Ch	Data Separator

Message format
Start Char = @
Station Number = nn (BCD number from 00 to 64)
Message Contents (Command)
End of message marker : (colon)
BCC 2 digit hex number giving checksum of message
End of message char = CR (0dh)

Example – read digital inputs from station 1
@01EX<sp>DI:XX<cr>

Response
@01EX<sp>DI<sp>0010<sp>0000<sp>0000:yy<cr>

The Station number is set using the Station Program Address dialog box. All stations are default to station 1 at the factory. If more than one station is on the loop they must all be given unique addresses before reading data out of the stations.

BCC calculation range and calculation method.
The BCC is calculated by adding the ascii characters in the data stream from the start of the station number upto and including the colon character ":" (8 bit addition, carry ignored). The BCC is generated in the station and appended to the response message in the same way.

The station number is programmed into the unit using the station programming dialog boxes. Each station on the data loop uses a unique station number.

Station LEDS.

RX	Flashes when the station receives serial characters
TX	Flashes indicating the station has understood the message and is sending a reply.
TXE	Flashes indicating the RS422 or 485 transmitter has turned on. There may be delays between the receipt of the RX data and the lighting of the TXE and TX lights depending on the setting of the TX and TXE delays in the station's advanced programming dialog boxes.

Individual Message Descriptions.

EX DO – Write Digital Outputs

Writes States of Relay Outputs

Send Message
@NNEX<sp>DO<sp>XXXX<sp>XXXX:YY<CR>

Param1 – digital outputs on station
XXXX=$b_{15}b_{14}b_{13}b_{12}$ $b_{11}b_{10}b_9b_8$ $b_7b_6b_5b_4$ $b_3b_2b_1b_0$

b_0=Relay 1	A16,A4,A4e,AO,2100-D
b_1=Relay 2	A16,A4,A4e,AO,2100-D
b_2=Relay 3	A4,A4e,2100-D
b_3=Relay 4	A4,A4e,2100-D
b_4=Relay 5	A4e,2100-D
b_5=Relay 6	A4e,2100-D
b_6=Relay 7	A4e,2100-D
b_7=Relay 8	A4e,2100-D
b_8=Relay 9	2100-D
b_9=Relay 10	2100-D
b_{10}=Relay 11	2100-D
b_{11}=Relay 12	2100-D

All other bits zero

The States of the onboard relays of the station can be read using the EX DI command, param1.

Param2 – digital outputs on 2100-R
XXXX=$b_{15}b_{14}b_{13}b_{12}$ $b_{11}b_{10}b_9b_8$ $b_7b_6b_5b_4$ $b_3b_2b_1b_0$

b_0	Relay 1 State	b_1	Relay 2 State
b_2	Relay 3 State	b_3	Relay 4 State
b_4	Relay 5 State	b_5	Relay 6 State
b_6	Relay 7 State	b_7	Relay 8 State
b_8	Relay 9 State	b_9	Relay 10 State
b_{10}	Relay 11 State	b_{11}	Relay 12 State
b_{12}	Relay 13 State	b_{13}	Relay 14 State
b_{14}	Relay 15 State	b_{15}	Relay 16 State

The States of the 2100-R relays of the station can be read using the EX DI command, param3.

Both params are specified even if a 2100-R is not fitted.

Reply Message
@NNOK:YY<CR>
Indicating successful write

To Write individual bits, a cached copy of the current states of the digital outputs must be read from the station using the EX DI command and then the bits are modified accordingly and written out using the EX DO command.

Any internal function on the station will override the states of the relay written by the EX DO command. i.e Comms time out function on will always set Relay 2 when a valid comms command has been received. Station controllers will write the relay states every time the controller setpoints are checked (includes manual override modes).

A16 REV1.3 (Update 31/8/2007)

Add extra param for 2nd bank 2100-R
@NNEX<sp>DO<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

Param1 – digital outputs on station
Param2 – digital outputs on 2100-R
Param3 - digital outputs on 2100-R #2
XXXX=b15b14b13b12 b11b10b9b8 b7b6b5b4 b3b2b1b0

b0	Relay 1 State	b1	Relay 2 State
b2	Relay 3 State	b3	Relay 4 State
b4	Relay 5 State	b5	Relay 6 State
b6	Relay 7 State	b7	Relay 8 State
b8	Relay 9 State	b9	Relay 10 State
b10	Relay 11 State	b11	Relay 12 State
b12	Relay 13 State	b13	Relay 14 State
b14	Relay 15 State	b15	Relay 16 State

The States of the 2100-R relays of the station can be read using the EX DI command, param4.
All three params are specified even if a 2100-R is not fitted.

A16 REV1.3 will work with both

@NNEX<sp>DO<sp>XXXX<sp>XXXX:YY<CR>
@NNEX<sp>DO<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

if the short form is sent param1, and param2 the output values to 2100-R#2 will not be written.

EX DI – Read Digital outputs and digital outputs

Reads States of Digital Inputs and Digital Outputs

Send Message
@NNEX<sp>DI:YY<CR>

A16,A4,A4e,AO Reply Message
@NNEX<sp>DI<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

2100-D replies with
@NNEX<sp>DI<sp>XXXX<sp>XXXX:YY<CR>

Param1 – state of digital outputs
XXXX=$b_{15}b_{14}b_{13}b_{12}$ $b_{11}b_{10}b_{9}b_{8}$ $b_{7}b_{6}b_{5}b_{4}$ $b_{3}b_{2}b_{1}b_{0}$

b_0=Relay 1	A16,A4,A4e,AO,2100-D
b_1=Relay 2	A16,A4,A4e,AO,2100-D
b_2=Relay 3	A4,A4e,2100-D
b_3=Relay 4	A4,A4e,2100-D
b_4=Relay 5	A4e,2100-D
b_5=Relay 6	A4e,2100-D
b_6=Relay 7	A4e,2100-D
b_7=Relay 8	A4e,2100-D
b_8=Relay 9	2100-D
b_9=Relay 10	2100-D
b_{10}=Relay 11	2100-D
b_{11}=Relay 12	2100-D

All other bits zero

Param2 – state of digital inputs
XXXX=$b_{15}b_{14}b_{13}b_{12}$ $b_{11}b_{10}b_{9}b_{8}$ $b_{7}b_{6}b_{5}b_{4}$ $b_{3}b_{2}b_{1}b_{0}$

b_0	Digital Input 1 State
b_1	Digital Input 2 State
b_2	Digital Input 3 State
b_3	Digital Input 4 State (AO,A4, A16 Rev1 Only, A16 Rev0 returns 0)
b_4	Digital Input 5 State (AO,2100-D only)
b_5	Digital Input 6 State (AO,2100-D only)
b_6	Digital Input 7 State (AO,2100-D only)
b_7	Digital Input 8 State (AO,2100-D only)
b_8	Digital Input 9 State (AO,2100-D only)
b_9	Digital Input 10 State (AO,2100-D only)
b_{10}	Digital Input 11 State (AO,2100-D only)
b_{11}	Digital Input 12 State (AO,2100-D only)

All other bits zero

Param3 – state of 2100-R digital outputs.
XXXX=$b_{15}b_{14}b_{13}b_{12}$ $b_{11}b_{10}b_{9}b_{8}$ $b_{7}b_{6}b_{5}b_{4}$ $b_{3}b_{2}b_{1}b_{0}$

All three Params are returned even if a 2100-R is not fitted.

b_0	Relay 1 State	b_1	Relay 2 State
b_2	Relay 3 State	b_3	Relay 4 State
b_4	Relay 5 State	b_5	Relay 6 State
b_6	Relay 7 State	b_7	Relay 8 State
b_8	Relay 9 State	b_9	Relay 10 State
b_{10}	Relay 11 State	b_{11}	Relay 12 State
b_{12}	Relay 13 State	b_{13}	Relay 14 State

b_{14}	Relay 15 State	b_{15}	Relay 16 State

A Bit Set/ Reset has to be synthesised by retaining the last value written and ANDING(reset) or ORING(set) a new value to create the bit set or reset.

A16 REV1.3   (Update 31/8/2007)

Add extra param for 2nd bank 2100-R

@NNEX<sp>DI<sp>XXXX<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

Param4 - state of 2100-R#2 digital outputs.
XXXX=b15b14b13b12 b11b10b9b8 b7b6b5b4 b3b2b1b0

All four Params are returned even if a 2100-R is not fitted.

b0	Relay 1 State	b1	Relay 2 State
b2	Relay 3 State	b3	Relay 4 State
b4	Relay 5 State	b5	Relay 6 State
b6	Relay 7 State	b7	Relay 8 State
b8	Relay 9 State	b9	Relay 10 State
b10	Relay 11 State	b11	Relay 12 State
b12	Relay 13 State	b13	Relay 14 State
b14	Relay 15 State	b15	Relay 16 State

A16 prior to rev1.3 returns
@NNEX<sp>DI<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>
A16 rev1.3 and after returns
@NNEX<sp>DI<sp>XXXX<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

This can be used to identify if the A16 is a rev1.3 or not.

RCn - Read Counter Values

Reads counter values of digital inputs as a group of four values

Send Message
@NNRCn:YY<CR>

n=1, inputs 1-4	2100-A16, 2100-AO,2100-A4,2100-A4e
n=2, inputs 5-8	2100-D Only
n=3, inputs 9-12	2100-D Only

i.e @01RC1:YY<CR>

Reply Message
@NNRCn<sp>QQ<sp>AAAA<sp>BBBB<sp>CCCC<sp>DDDD:YY<CR>

QQ=power up read flag
01= this is the first time this string has been read from the station on power up
00=the string has been read from the station more than once after powerup
This flag is maintained by the station, and is set to 1 on power up and cleared after the RC string has been read for the first time. This flag can be used to indicate the station has just powered up, and the delta count value calculation should be started again. i.e dont do a count calculation on this value, save this value and start the calculations again from the next value.

	RC1	RC2	RC3
AAAA	Count Value 1	Count Value 5	Count Value 9
BBBB	Count Value 2	Count Value 6	Count Value 10
CCCC	Count Value 3	Count Value 7	Count Value 11
DDDD	Count Value 4	Count Value 8	Count Value 12

AAAA,BBBB,CCCC,DDDD=0000-3FFF count value, use an AND to destroy bits 14,and 15.
i.e AND 0x3FFF. These bits are used by the Micro Scan and should be destroyed.

Use Delta count maths to calculate the number of pulses to occur.
i.e Read 1 = 200, Read 2 = 500 . Therefore pulse count =300 over the time period between read 1 and read 2, therefore the rate can be calculated. The counts must be read often enough so that less 16384 pulses occur between each read of the counter value.

The counts are not reset between each read.

The 2100-AO has 12 digital inputs, however counts are only supported on channels 9,10,11,12 which is read by the RC1 message. AAAA=input 9, BBBB=input 10 etc.

EX E5 – Read analogue inputs

Reads the States of the station analogue inputs (A16,A4,A4e,AO only)

Send Message

@NNEX<sp>E5<sp>XX:YY<CR> read analogue inputs

@NNEX<sp>E5<sp>00:YY<CR> read analogue inputs 1-4
@NNEX<sp>E5<sp>01:YY<CR> read analogue inputs 5-8
@NNEX<sp>E5<sp>02:YY<CR> read analogue inputs 9-12 (A16 only)
@NNEX<sp>E5<sp>03:YY<CR> read analogue inputs 13-16 (A16 only)

Reply Message
@NNEX<sp>E5<sp>00<sp>ffffffff<sp>ffffffff<sp>ffffffff<sp>ffffffff:YY<CR>
@NNEX<sp>E5<sp>01<sp>ffffffff<sp>ffffffff<sp>ffffffff<sp>ffffffff:YY<CR>
@NNEX<sp>E5<sp>02<sp>ffffffff<sp>ffffffff<sp>ffffffff<sp>ffffffff:YY<CR>
@NNEX<sp>E5<sp>03<sp>ffffffff<sp>ffffffff<sp>ffffffff<sp>ffffffff:YY<CR>

ffffffff=IEEE float number giving value of input, ranged according to the input type selected. i.e
DCX range value = 0-100 %
RTD1 range value = 0-25 Deg C (plus slight overrange at both ends)
RTD10 range value = -200-250 Deg C (plus slight overrange at both ends)

IEEE Floating point format.
First ff = Byte 3
Second ff = Byte 2
Third ff = Byte 1
Fourth ff = Byte 0

Byte 3 Byte 2 Byte 1 Byte 0
31 30 29 28 27 26 25 24.23 22 21 20 19 18 17 16.15 14 13 12 11 10 9 8.7 6 5 4 3 2 1 0
s exponent mantissa
s=sign = bit 31
exponent=bits 22-30
mantissa=bits 0-21

Note:
The values returned by the EX E5 commands are the last values read from the inputs. Issuing a EX E5 command does not force the A16 to read in the channels requested at the time the command is issued.

EX E6 – Read Ambient Sensor & misc

Send Message

@NNEX<sp>E6:YY<CR> read data

Reply Message
@NNEX<sp>E6<sp>ffffffff<sp>XX<sp>XX<sp>XXXX<sp>XX<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

Param1 ffffffff Ambient onboard sensor for cold junction compensation (See EX E5 fp format)
Param 2 XX Current Input being read (0-0f h)
Param 3 XX Current Muxltiplexer channel being read.
Param 4 XXXX reserved
Param 5 XX modeswitch (0-3fh)
Param 6 XXXX reserved
Param 7 XXXX reserved
Param 8 XXXX current rtx channel

EX EN – Read Multiplexer Channels

EX En reads the 16 channels of each Multiplexer

@NNEX<sp>E1:YY<CR> Read Multiplexer 1
@NNEX<sp>E2:YY<CR> Read Multiplexer 2
@NNEX<sp>E3:YY<CR> Read Multiplexer 3
@NNEX<sp>E4:YY<CR> Read Multiplexer 4

Reply Message

@NNEX<sp>EN<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX<sp>XXX:YY<CR>

i.e EN=E1,E2,E3,E4.

Param1-Param 16 Values of channels 1-16, XXX = Hex 000-FFF 12 bit number.

EX RO – Read Analogue Output Channels

The EX RO command reads the values of the analogue output channels.

@NNEX<sp>RO:YY<CR>

Reply Message

@NNEX<sp>RO<sp>XXXX<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

Param1..Param4 values of analogue output channels 1-4 (three and four are not used).
XXXX=12 bit value 0000-0FFF, leading zero is set to zero.

EX RO=O=Letter O

EX AO – Write Analogue Output Channels

The EX AO command writes the values of the analogue output channels.

@NNEX<sp>RO<sp>XXXX<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

Reply Message

@NNOK:YY<CR>

Param1..Param4 values of analogue output channels 1-4 (three and four are not used).
XXXX=12 bit value 0000-0FFF, leading zero should be set to zero.

A cached copy of the output states must be kept for individual write to the analogue outputs.
If the Analogue outputs are assigned specific functions in the station other than general SCADA outputs then the values written by this command will be overwritten by the internal function in the station.

EX WA – Write Single Analogue Output Channel 2100-AO only

The EX AO command writes the values of a single output channel.

@NNEX<sp>WA<sp>XX<sp>XXXX:YY<CR>

Reply Message

@NNOK:YY<CR>

Param 1 is output index 00=output1,01=output2 to 07=output 8
Param 2 value of analogue output channel.
XXXX=12 bit value 0000-0FFF, leading zero should be set to zero.

If the Analogue outputs are assigned specific functions in the station other than general SCADA outputs then the values written by this command will be overwritten by the internal function in the station.

EX R1 – Read Analogue Output Channels 2100-AO only

The EX R1 command reads the values of the analogue output channels 5-8

@NNEX<sp>R1:YY<CR>

Reply Message

@NNEX<sp>R1<sp>XXXX<sp>XXXX<sp>XXXX<sp>XXXX:YY<CR>

Param1..Param4 values of analogue output channels 5-8.
XXXX=12 bit value 0000-0FFF, leading zero is set to zero.

EX R1=1=Number 1

PS – Read Controller Data

This message reads the setup of the station controllers.
Number of controllers.

Station	2100-R not fitted	2100-R fitted
A16	2 Single Action, or 1 Dual Action controller	16 Single Action or 8 Dual Action Controller **REV 1.3** – 16 Dual Action with 2100-R #2
A4	4 Single Action or two Dual Action controllers.	4 Single Action or 4 Dual Action controllers.
A4e	4 Single Action or four Dual Action controllers.	4 Single Action or 4 Dual Action controllers.
AO	2 Single Action, or 1 Dual Action controller	N/A

Send Message

@NNPS<sp>XX:YY<CR> read data

Reply Message

@NNPS<sp>XX,XXXXffffffffffffffff:YY<CR>
XX=Controller index, =10d*controller number-1
i.e 00=controller 1, 0a=controller 2, 14=controller 3

Param1=XXXX controller flags
Param2=ffffffff setpoint (IEEE Number)
Param3=ffffffff differential (IEEE Number)

A16 REV1.3 Scaled Integers DA controllers.

USING THIS MESSAGE WITH ANYTHING OTHER THAN A16 REV1.3 WILL HAVE UNDEFINED EFFECTS.

@NNPS<sp>XX,XXXXAAAABBBBCCCCDDDD:YY<CR>
XX=Controller index, =10d*controller number-1
i.e 00=controller 1, 0a=controller 2, 14=controller 3

Param1=XXXX controller flags
Param2=AAAA setpoint
Param3=BBBB cool differential
Param4=CCCC deadband
Param5=DDDD heat differential

Scaled integers are integer values, but the value is 10 times what it should be to imitate one decimal point.

Controller Value	Scaled Integer (Message Value)
0	0
0.5	5
1.0	10
25.3	253
-22.6	-226

These two messages are the same length, but the scaled integer bit (b7) of controller flags determines what the packing of the data is.

Controller flags
SA= Single Action, DA= Dual Action. Set by the Controller mode in the Station Advanced Dialog box.

Bit	Function	SA/DA Mode	Value = 0	Value =1
b_0	Enable	Both	Controller Disabled	Controller Enabled
b_1	Manual Override	Both	Auto Mode	Manual Mode. Output State Specified by Bit 2 (SA) or bits 5,6 (DA).
b_2	Manual State	SA	Output Off	Output On
b_3	Reverse Action	Both	Heating Action (Relay is on when heating required i.e when temp is below setpoint)	Cooling Action (Relay is on when cooling is required i.e when temp is above setpoint).
b_4	Heat Cool Mode	DA	Single action control with either one of Heating or Cooling control specified by Reverse Action Bit	Dual Action mode with both Heat and Cool relays working.
b_5	Manual Heat On	DA	Heat Relay Off when manual override set	Heat Relay On when manual override set. (Only one of bit 5 and bit 6 should be set at one time)
b_6	Manual Cool On	DA	Cool Relay Off when manual override set	Cool Relay On when manual override set. (Only one of bit 5 and bit 6 should be set at one time)
b7	Scaled Integer **A16 REV1.3 ONLY**	DA	Settings are float values (2 floats per message)	Settings are DA Scaled Integers (4 words per message)

Other bits are reserved and should be set to zero.
If a controller is active and then deactivated by clearing the the controllers enable bit, the controllers relay state will be left in the last remaining state. It is up to the software to turn the relay to the desired state.

Controller Modes
Single Action/Dual Action.
The Single Action/Dual Action mode is specified in the Station’s Advanced dialog box. Details of relay allocations are specified in the manual supplied with the station. In the Single Action Mode the behaviour of the relays is defined by the control action chosen. In the Dual Action mode, the functions of the relays are fixed, but will only operate according to the control action chosen.

Dual Action Mode Relays

Relay	Function	Heating	Cooling	Heat/Cool
Relay 1	Heat Action	Active	Off	Active
Relay 2	Cool Action	Off	Active	Active

This is chosen so the user can switch between Heat/Cool and Heat only or Cool only and have the relay numbers stay the same.

A16
For the Single Action control mode there are 16 controllers.
For the Dual Action mode, there are 8 controllers. With the setup for the extra parameters needed for the controller coming from the controller index+8.

A16 Dual Action Summary (1 x 2100-R)

Controller	Index	Controller Flags	Setpoint	Differential
1	00	As Required	Controller 1 Setpoint	Controller 1 Cool Differential
2	0A	As Required	Controller 2 Setpoint	Controller 2 Cool Differential

3	14	As Required	Controller 3 Setpoint	Controller 3 Cool Differential
4	1E	As Required	Controller 4 Setpoint	Controller 4 Cool Differential
5	28	As Required	Controller 5 Setpoint	Controller 5 Cool Differential
6	32	As Required	Controller 6 Setpoint	Controller 6 Cool Differential
7	3C	As Required	Controller 7 Setpoint	Controller 7 Cool Differential
8	46	As Required	Controller 8 Setpoint	Controller 8 Cool Differential
9	50	0	Controller 1 Deadband	Controller 1 Heat Differential
10	5A	0	Controller 2 Deadband	Controller 2 Heat Differential
11	64	0	Controller 3 Deadband	Controller 3 Heat Differential
12	6E	0	Controller 4 Deadband	Controller 4 Heat Differential
13	78	0	Controller 5 Deadband	Controller 5 Heat Differential
14	82	0	Controller 6 Deadband	Controller 6 Heat Differential
15	8C	0	Controller 7 Deadband	Controller 7 Heat Differential
16	96	0	Controller 8 Deadband	Controller 8 Heat Differential

Index=(Controller number-1)*10d

A16 Dual Action Summary (2 x 2100-R Dual Action Mode) A16 REV 1.3 ONLY

Unlike support for DA 8 controllers which used 2 messages per controller, DA 16 controllers uses 1 message per controller and the values are specified as scaled integers instead of floats.

Controller	Index	Controller Flags	Word 1 AAAA	Word 2 BBBB	Word 3 CCCC	Word 4 DDDD
1	00	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
2	0A	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
3	14	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
4	1E	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
5	28	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
6	32	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
7	3C	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
8	46	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
9	50	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
10	5A	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
11	64	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
12	6E	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
13	78	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
14	82	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
15	8C	As Required	Setpoint	Cool Differential	Deadband	Heat Differential
16	96	As Required	Setpoint	Cool Differential	Deadband	Heat Differential

A4/A4e

For the Single Action control mode there are 4 controllers.
For the Dual Action mode, there are 4 controllers. With the setup for the extra parameters needed for the controller coming from the controller index+4.

A4 Dual Action Summary

Controller	Index	Controller Flags	Setpoint	Differential
1	00	As Required	Controller 1 Setpoint	Controller 1 Cool Differential
2	0A	As Required	Controller 2 Setpoint	Controller 2 Cool Differential
3	14	As Required	Controller 3 Setpoint	Controller 3 Cool Differential
4	1E	As Required	Controller 4 Setpoint	Controller 4 Cool Differential
5	28	0	Controller 1 Deadband	Controller 1 Heat Differential
6	32	0	Controller 2 Deadband	Controller 2 Heat Differential
7	3C	0	Controller 3 Deadband	Controller 3 Heat Differential
8	46	0	Controller 4 Deadband	Controller 4 Heat Differential

Index=(Controller number-1)*10d

PS – Write Controller Data

Send Message

@NNPS<sp>XX,XXXXffffffffffffffff:YY<CR>   send data

XX=Controller index, =10d*controller number-1
i.e 00=controller 1, 0a=controller 2, 14=controller 3

Param1=XXXX controller flags
Param2=ffffffff setpoint (IEEE Number)
Param3=ffffffff differential (IEEE Number)

Reply Data
@NNPS<sp>XX,XXXXffffffffffffffff:YY<CR>
The reply data is the same as one for a read of the controller data and can be used to verify the data has been written correctly.

A16 REV1.3 Scaled Integers DA controllers.

USING THIS MESSAGE WITH ANYTHING OTHER THAN A16 REV1.3 WILL HAVE UNDEFINED EFFECTS.

Send Message

@NNPS<sp>XX,XXXXAAAABBBBCCCCDDDD:YY<CR>
XX=Controller index, =10d*controller number-1
i.e 00=controller 1, 0a=controller 2, 14=controller 3

Param1=XXXX controller flags (b7 =1)
Param2=AAAA setpoint
Param3=BBBB cool differential
Param4=CCCC deadband
Param5=DDDD heat differential

Reply Data
@NNPS<sp>XX,XXXXAAAABBBBCCCCDDDD:YY<CR>
The reply data is the same as one for a read of the controller data and can be used to verify the data has been written correctly.

The Message is the same as the Read Controller Data message, but instead of the ":" indicating the end of message after the controller index, the "," indicates write data is to follow.
Note
1. Always check controller write data with the supporting dialog boxes in the Station Setup to check the data has been written to the correct locations.
2. Be aware of the EEPROM endurance limit of 10,000 writes per eeprom word. Try to restrict changes to controller setpoints if possible. EEPROM performance after 10,000 writes is not guaranteed (HC11E9 EEPROM limitation).

Sample C Routines
Convert a floating point number to a string

```
//For Writing a float value to the station

//put the float value into the buffer string
void ConvertFloatToIEEE(float fvalue,char *buffer)
{
unsigned char *ptr;

//point to the beginning of the float value
ptr=(unsigned char *)&fvalue;

//get the data bytes from the float value and put them in the string,
//starting with byte 3
sprintf(buffer,"%02X%02X%02X%02X",*(ptr+3)&255,*(ptr+2)&255,*(ptr+1)&
255,*(ptr+0)&255);
}
```

Convert the string to a floating point number

```
//for converting a float value in the string into a float number
float ConvertIEEEToFloat(char * pData)
{
float fvalue;
int count;
int i;
unsigned char *cptr;

//error check
//dont let the data if all chars in the string are 'F', it is an
//invalid number
for (i=0,count=0;i<8;i++)
    if (*(pData+i)=='F')
      count++;
    else break;

if (count==8)
    return 0;

fvalue=0;
//point to the float value
cptr=(unsigned char *)&fvalue;
//start with byte 3
cptr+=3;
//four bytes to di
count=4;
while(count)
    {
    //get the data from the string
    sscanf(pData,"%02X",&i);
    //store it
    *cptr=i&255;

    //move to the next byte
    count--;
    lpData+=2;
    cptr--;
    }
return (float)fvalue;
}
```

Calculate a BCC value and append it to a message

Message =@NN…….:

```
void AddBCCToMessage(char *message)
{
PSTR pData;
int bccvalue;
char endtext[10];

bccvalue=0;
pData=message+1;

while(*pData!='\0')
    bccvalue+=(unsigned char)(*pData++)&255;

bccvalue&=255;
sprintf(endtext,"%02X\x00d",bccvalue);
strcat(message,endtext);
}
```

Note on Serial Communications

In order to accomplish successful communications the following is recommended.

Generate a BCC on the sent data.
Always check the BCC on the reply data and throw the message away if the calculated value is different from the value in the receive string.

If any framing,overrun or other receive errors occur, throw the message away and send a new message.

Do not assume the first replied character will be a @. Rubbish characters can always occur, so start assembling a receive string when the @ is detected.

Decode the reply message when the <CR> has arrived, don't use the : as an end of message marker.

Check the reply data is the same as you have sent i.e if sending EX DI, check the reply string has EX DI in it before decoding the data. I.e the send string is EX<sp>DI: and the reply is EX<sp>DI<sp>, the colon changes to a space in the reply.

Also check the station number in the reply string is correct in case another station is replying with data.

Make the time to wait for a reply message adjustable for use with radio installations if the reply message will take several seconds.

When writing data, use the Micro Scan station programming boxes to verify the data has been programmed to the correct locations. The Micro Scan Programming Station Address dialog box must be used to give each station a unique station number if more than one station is on the loop.

Revisions
Add RCn Counter Message
A16 REV1.3 EX DI Message Changes
A16 REV1.3 EX DO Message Changes
A16 REV1.3 PS Message Changes