XPRESS 3.6 Addendum

Version 3.6 of the HCS II firmware and XPRESS language primarily adds support for the Answer MAN network module. The Answer MAN offers many interfacing options such as digital and analog I/O, frequency in, totalizer in, PWM waveform generation, iButton serial number reading, and LCD display and keypad support. Note that not all options are available simultaneously and some may require additional hardware.

Upgrading

Upgrading your HCS II is simply a matter of installing the new EPROM, recompiling your XPRESS program with the new compiler, and loading your recompiled program into the HCS II. The version numbers of the EPROM and the compiler must match or the HCS will refuse to accept your program upload. Therefore, it is important to remove all old copies of the XPRESS compiler when upgrading.

Configuring the Answer MAN

The Answer MAN module must be configured before it can be used with an HCS II network. It is not possible to configure an Answer MAN from the HOST screen or while it is installed in the HCS network. The Answer MAN must be removed from the network, connected to a PC or serial terminal, put into configuration mode, and set up directly. Please refer to the Answer MAN datasheet for more information on configuring the module.

When using the LCD and keypad support of the Answer MAN, be sure to configure the module address as TERM0, TERM1, and so forth up to TERM7. The TERM*x* address is the default network address of the LCD-Link. The Answer MAN is designed to emulate the LCD-Link directly, so your XPRESS program should access the Answer MAN as if it were an LCD-Link.

When using any of the other Answer MAN features, be sure to address the module as MAN0, MAN1, on up to MAN7. The HCS automatically polls each Answer MAN to find out how it is set up and will respond to only those features for which the module is configured. Note that it may take up to a minute for the HCS to poll the Answer MAN for its setup and to adapt to that setup.

Finally, in all cases, be sure to issue a CD 32 command to set a reply delay of 50 ms.

ADIO-Link and Channel Numbers

ADIO-Link support is removed to make way for Answer MAN support. ADIO-Link netbits are redefined as shown in the following table. Netbits and analog I/O are accessed using the Netbit(), ADC(), and DAC() XPRESS

commands. See the *XPRESS Reference Manual* for more details on the use of those commands.

Answer MAN Number	Input Netbits	Output Netbits	Analog Input Channels	Analog Output Channels
0	96–103	104–111	16–21	0,1
1	112–119	120–127	24–29	4,5
2	128–135	136–143	32–37	8,9
3	144–151	152–159	40–45	12,13
4	160–167	168–175	48–53	16,17
5	176–183	184–191	56–61	20,21
6	192–199	200–207	64–69	24,25
7	208–215	216–223	72–77	28,29

Analog inputs and outputs are likewise redefined. Note that only six analog inputs are supported on each Answer MAN, but eight slots have been reserved. The highest two analog input numbers on each module are not used at this time. Similarly, only two analog outputs are supported on each module, but four slots each have been reserved.

XPRESS Configuration

Your XPRESS program must be modified slightly to inform the Supervisory Controller that one or more Answer MAN modules are present on the network. At the top of your program, include the following line:

Config AMAN-Link = *xx*

where *xx* is the number of Answer MAN modules you have on the network.

New Commands

While most of the Answer MAN features can be accessed using existing XPRESS commands, there are several new commands that take advantage of some of the modules new features. The following section describes these new commands.

PWMtotal — Set the total PWM period value

Syntax: PWMtotal(n) = t

where: n = Answer MAN module number (0–7)
t = total period count (0020–7FFF)

See the Answer MAN SP command for details on how to calculate values. Setting this value to zero shuts off the PWM output.

Set the high PWM period value — PWMhigh

Syntax: PWMhigh(n) = h

where: n = Answer MAN module number (0–7)
h = high period count (0008–7FF0)

See the Answer MAN SP command for details on how to calculate values. Setting this value to zero shuts off the PWM output.

Period of frequency being fed to Answer MAN module — Frequency

Syntax: f = Frequency(n)

where: f = period of input frequency (0–32767)
n = Answer MAN module number (0–7)

To calculate frequency from the period returned, use the following equation:

11184 / (Frequency(1) / 10)

Number of pulses received by the Answer MAN since last clear — Total

Syntax: t = Total(n)

where: t = total number of pulses received (0–32767)
n = Answer MAN module number (0–7)

Note that the count is *not* cleared when read. Use ClearTotal to clear the count.

Clear the Answer MAN totalizer count — ClearTotal

Syntax: ClearTotal(n)

where: n = Answer MAN module number (0–7)

Configure the I/O port bit directions — BitDirection

Each Answer MAN digital I/O bit can be set up as either an input or an output. Each of the eight I/O bits corresponds to a bit in a direction byte. When a bit in the direction byte is set to 0, its I/O bit is configured as an

output. When a bit in the direction byte is set to 1, the I/O bit is set up as an input. This command is used to configure the directions of all eight bits.

Note that since the HCS doesn't support hexadecimal numbers, the direction byte must be converted to decimal to be used in an XPRESS program.

Syntax: BitDirection(n) = b

where: n = Answer MAN module number (0–7)
b = decimal equivalent of the hexadecimal value that defines each bit as an input or output

KeyDigit Read a single keypad press

When the Answer MAN module is configured as an LCD-Link (module name TERMn), it supports a scanned matrix keypad. The Answer MAN will buffer up to eight keypresses, so the user can press the keys quickly without losing any data. The KeyDigit function is used to get the next keypress from the buffer.

The KeypadTimeout function is used to set how long the KeyDigit function will wait for a keypress. A negative value is returned if no key is available before the end of the timeout period.

Only works in the sequential section.

Syntax: KeyDigit(n)

where: n = Answer MAN module number (0–7)

The value returned is 0–9 (0–9), 10 (*), 11 (#), or 12–15 (A–D).

KeyNumber Read up to four keypad presses as a number

When the Answer MAN module is configured as an LCD-Link (module name TERMn), it supports a scanned matrix keypad. The Answer MAN will buffer up to eight keypresses, so the user can press the keys quickly without losing any data. The KeyNumber function is used to get up to four keypresses from the buffer and returns them as a single large numeric value. The function waits until four keys have been pressed or a nonnumeric key is pressed (for fewer than four keypresses).

The KeypadTimeout function is used to set how long the KeyNumber function will wait for a keypress. A negative value is returned if no key is available before the end of the timeout period. Each keypress restarts the

timeout timer, so it's not necessary to press all four keys within a single timeout period.

Only works in the sequential section.

Syntax: KeyNumber(n)

where: n = Answer MAN module number (0–7)

The value returned ranges 0000–9999.

Set the timeout value for a keypad read — KeypadTimeout

When the keypad is read, sequential section processing pauses until one or more keypad keys are pressed or a timeout occurs. This function is used to set the timeout for the keypad functions.

Only works in the sequential section.

Syntax: KeypadTimeout(n) = t

where: n = Answer MAN module number (0–7)
t = timeout value in hundreds of milliseconds (0–255 = 0 to 25.5 seconds)

Read a Dallas Semiconductor iButton serial number — iButton

All Dallas Semiconductor iButton devices include a unique serial number that may be read independent of any other functions the device supports. The Answer MAN may be set up to read this serial number and pass it along to the HCS.

The byte sequence consists of an iButton family code, followed by several bytes of the serial number, followed by several zeros, and terminated with a CRC. If the byte sequence printed on the device consists of a single byte followed immediately by several zeros, reverse the order of the bytes before using the sequence.

When the Answer MAN detects an iButton device, it reads the serial number and stores it in a buffer. The HCS then polls the Answer MAN for that number and stores it in its own buffer. If a new serial number comes in before the original one was tested by an XPRESS program, the new number overwrites the old one.

Note that iButton values may only be tested within an IF statement and must

be compared to constants. iButton serial numbers may *not* be stored in variables nor may they be compared to values stored in variables. However, an iButton serial number remains available for testing until a new value is read. That is, it may be tested multiple times within an XPRESS program without being automatically cleared by the system.

Syntax: iButton(n) = s1,s2,s3,...,s8

where: n = Answer MAN module number (0–7)
s1–s8 = eight decimal bytes of an iButton serial number

Example:
```
DEFINE FredKey = 2,70,242,0,0,0,0,152
DEFINE GeorgeKey =

BEGIN
IF iButton(0)=FredKey THEN
  Console = “Fred’s key detected”
END
IF iButton(0)=GeorgeKey THEN
  Console = “George’s key detected”
END
```

Answer MAN Versions

The majority of features of all Answer MAN modules since V 1.0 work fine with XPRESS 3.5. The exception is 12-bit analog I/O. You must be using V 1.11 or later of the Answer MAN in order to use 12-bit analog I/O with the HCS.

To find out what version Answer MAN you have, connect a terminal to the Answer MAN, ground the CFG pin, and power it up in configuration mode. The signon banner displayed on reset indicates the Answer MAN version number.