# PTS/Data Highway Interface Software
# User's Manual

*Issue 2*

*February 1997*

*MAN527*

# Contents



 

# 1. Introduction

This manual relates to the following versions of software in the Mini-PTS and PTS Mark 2 units:

| | |
|---|---|
| Data Highway Interface | Version 1.1 |
| PTS Host software | Version 1.8.2 or later |

This document describes the PTS/Data Highway Interface. This provides considerable flexibility in operating the machine - allowing changes to the program to be made quickly and easily by the user. It also ensures that improvements to the standard PTS product can be readily incorporated into any machine. The PTS units use the Quin Motor Control language which is described fully in the relevant PTS Reference Manual.

Control of the machine is accomplished in the following way:

- The control system communicates with the PTS unit as if it were a PLC. This allows the control system to display information about the running of the machine which it gets from the PTS. In a similar manner the operator can control the machine from the control system.
- The PTS unit controls the running of the motors according to setup data received from the control system. The interface to the control system is via the auxiliary serial port, thus leaving the main serial port available for initial machine setup and diagnostics.
- It is envisaged that most of the digital I/O such as guard switches will be taken care of using a PLC.

 

# 2. Program Functions

The Data Highway interface to the PTS allows the control system to control and monitor the operation of the machine by communicating with the PTS directly over a RS232 or RS422 serial link.

The functions which the PTS/Data Highway interface accomplishes are as follows:

- Monitoring machine status. The control system can interrogate the PTS to determine whether any motor errors have occurred, or to find the current channel status for example. It is also possible to examine the current setting of parameters such as the velocity, position, etc.

- Machine control. The control system can control the PTS by initiating preprogrammed sequences. It is also able to execute most of the normal PTS commands such as setting the velocity, executing a move and so on.

Machine control and monitoring are achieved from the control system by reading from/writing to registers within the PTS in the same way as a PLC would be controlled and monitored. The communication with the PTS is done using the Data Highway DF1 protocol. Registers are 16-bit locations for input and/or output. The PTS interface maps the registers onto standard PTS commands in a predetermined manner thus allowing the operator to accomplish most of the things that are possible using the standard PTS command language.

 

# 3. Programming

## 3.1 General

The Data Highway interface to the PTS is designed to make the PTS look like a PLC as much as possible. To achieve this, the PTS appears as a number of registers or elements which can be read or written as appropriate by the control system using the Data Highway interface. Each PTS channel has its own set of registers to allow random access to most of the data in the PTS. In addition there is a set of registers for system-wide data which is not specific to a particular channel. It is possible to address the channel data in two ways. Using the standard method, consecutive registers access different data items from the same channel. Using the alternative method, consecutive registers access the same data item on different channels.

## 3.2 Register Size

It should be noted that the Data Highway DF1 protocol can only handle 16 bit numbers and therefore the units on the PTS must be set to ensure all data falls within this range. This is accomplished with the SU command.

## 3.3 Files and Messages

The PTS appears to a remote system as a single file which contains all the registers as elements in the file. The PTS ignores the file number contained in incoming messages and will therefore accept messages addressed to any given file.

The PTS recognizes the following PLC-5 messages.

Word Range Write (Block Write) (function code 00)
Word Range Read (Block Read) (function code 01)
Typed Write (function code 67)
Typed Read (function code 68)

The PTS accepts logical Binary addresses or logical ASCII addresses of the form:

`$Nff:eee`

where `ff` is the file number which is currently ignored
and `eee` is the element or register number as descibed in the following pages.

## 3.4 Protocol

The PTS is intended to be connected to the Data Highway network using the KF2 interface module or equivalent. The PTS recogniozes the Data Highway Full Duplex DF1 protocol and can be configured to use either RS232 or RS422 serial port characteristics. The serial line parameters are as follows.

9600 Baud
8 bit characters
1 stop bit
No parity

 

# 4. Configuring the PTS

## 4.1 Software License Key

The software for the Data Highway interface will not operate unless a software key has been entered to enable this option. The software key is different for each PTS and can be obtained from your sales office given the system serial number which can be found by using the SK command as shown below. Note that all options which use the second serial port are mutually exclusive. These options are the standard Operator's panel, Mini Operator's panel, the Modbus interface and the Data Highway interface.

To enable the software the following command should be entered on Port A (the main programming port) in privileged mode. You enter the text in **bold** while the PTS displays something similar to the rest.

```
1> SK
Serial number: 006545
Feature     Version   Key
New feature ? datahw
Version ? 1.1
Key ? abcd
OK
```

Note that the feature name (datahw) must be entered in lower case exactly as shown above. Note also that it is necessary to turn the power off and back on again to run the Data Highway software.

If the software needs to be disabled, first make a note of the software key in case it is needed in the future. Then proceeed as above but simply press the Return key in response to the "Version ? " prompt as follows.

```
1> SK
Serial number: 006545
Feature     Version   Key
datahw        1.1     ABCD
New feature ? datahw
Version ?
Feature datahw removed
```

 

## 4.2 Serial Port

The Data Highway interface uses Port B of the PTS which can be configured to be either RS232 or RS422 levels using a PTS command on the Mini-PTS 2+1, Mini-PTS 3 and PTS Mark 2. To change the serial port configuration the following command should be entered on Port A (the main programming port) in privileged mode. You enter the text in **bold** while the PTS displays something similar to the rest.The example demonstrates changing Port B from RS232 to RS422.

```
1> CF
Port  Configuration
 A    RS232     S/W
 B    RS232     N/A
 D    Stn  1    BCC
Port ? B
Type ? RS422
OK
```

## 4.3 Data Highway Protocol

Certain details of the Data Highway protocol implementation can be configured using the CF command to change the configuration of Port D. These are the station number of the PTS on the Data Highway network and the type of checksum used on messages to and from the PTS. It is important to set these parameters correctly to ensure that the PTS recognizes commands sent to it and reponds correctly to those commands. The following example demonstrates changing the station number to 1 and the checksum type to CRC (the default values).

```
1> CF
Port  Configuration
 A    RS232     S/W
 B    RS485     N/A
 D    Stn  5    BCC
Port ? D
Station number ? 1
Checksum type (BCC or CRC) ? CRC
OK
```

 

# 5. Address Map

The data is organized in read/write registers as shown on the following pages. Each piece of data is accessible at two addresses. The first addressing mode allows different data items on the same channel to be accessed at consecutive addresses. The channel number is shown by the hundreds digit and is represented by "nn" in the tables. Addresses below 100 are used for system-wide data items which are not specific to a particular channel. The second addressing mode is used for addresses above 5000, and allows the same data item on all channels to be accessed at consecutive addresses.

For example the control word (CW) for channel 1 can be accessed as register 106, the control word for channel 3 can be accessed at address 306 and so on. If multiple registers are written starting at register 106, then this will alter the control word (CW), debounce time (DB), zero marker input (DZ) etc. for channel 1. The control word for channel 1 can also be accessed as register 5251, for channel 2 at 5252, for channel 3 at 5253 and so on. The debounce time for channel 1 can be accessed at 5301, for channel 2 at 5302 and so on. This addressing mode is useful because it allows a given data item for all channels to be accessed using a single communications block and so uses the network more efficiently. The register number is given in the tables first for addressing mode 1 (data item consecutive) and secondly for addressing mode 2 (channel consecutive).

It should be noted that certain commands are marked in the tables as write only and will therefore not return a meaningful value when read.

 

## 5.1 System Data

| Address | Command | Contents |
|---|---|---|
| 2 | AS | Start-up sequence number |
| 3 | CD | Character delay |
| 4 | CH | Current channel (Write only) |
| 5 | DW | Display word |
| 6 | GX | Command abort (Write only) |
| 7 | | Reserved |
| 8 | XS | Execute sequence (Write only) |
| 9 | CP | Parallel channel command (Write only) |
| 10 | SY | Set status reporting |
| 11 | $V1 | Variable V1 |
| 12 | $V2 | Variable V2 |
| 13 | $V3 | Variable V3 |
| 14 | $V4 | Variable V4 |
| 15 | $V5 | Variable V5 |
| 16 | $V6 | Variable V6 |
| 17 | $V7 | Variable V7 |
| 18 | $V8 | Variable V8 |
| 19 | $V9 | Variable V9 |
| 20 | $V10 | Variable V10 |
| 21 | $V11 | Variable V11 |
| 22 | $V12 | Variable V12 |
| 23 | $V13 | Variable V13 |
| 24 | $V14 | Variable V14 |
| 25 | $V15 | Variable V15 |
| 26 | $V16 | Variable V16 |
| 27 | $V17 | Variable V17 |
| 28 | $V18 | Variable V18 |
| 29 | $V19 | Variable V19 |
| 30 | $V20 | Variable V20 |
| 31 | $V21 | Variable V21 |
| 32 | $V22 | Variable V22 |
| 33 | $V23 | Variable V23 |
| 34 | $V24 | Variable V24 |
| 35 | $V25 | Variable V25 |

 

**System Data (continued)**

| Address | Command | Contents |
|---|---|---|
| 36 | $V26 | Variable V26 |
| 37 | $V27 | Variable V27 |
| 38 | $V28 | Variable V28 |
| 39 | $V29 | Variable V29 |
| 40 | $V30 | Variable V30 |
| 41 | $V31 | Variable V31 |
| 42 | $V32 | Variable V32 |
| 43 | $V33 | Variable V33 |
| 44 | $V34 | Variable V34 |
| 45 | $V35 | Variable V35 |
| 46 | $V36 | Variable V36 |
| 47 | $V37 | Variable V37 |
| 48 | $V38 | Variable V38 |
| 49 | $V39 | Variable V39 |
| 50 | $V40 | Variable V40 |
| 51 | $V41 | Variable V41 |
| 52 | $V42 | Variable V42 |
| 53 | $V43 | Variable V43 |
| 54 | $V44 | Variable V44 |
| 55 | $V45 | Variable V45 |
| 56 | $V46 | Variable V46 |
| 57 | $V47 | Variable V47 |
| 58 | $V48 | Variable V48 |
| 59 | $V49 | Variable V49 |
| 60 | $V50 | Variable V50 |
| 81 | CM | Compile sequences (Write only) |
| 82 | RD | Read parameters (Write only) |
| 83 | RS | Reset parameters (Write only) |
| 84 | ER | End repeat loop (Write only) |
| 85 | GA | Global abort command (Write only) |
| 86 | GF | Global motor-off command (Write only) |
| 87 | GS | Global stop command (Write only) |
| 88 | GX | Global command abort (Write only) |

 

## 5.2 Channel Data

| Address | | Command | Contents |
|---|---|---|---|
| nn02 | 50mm | AV | Set map base/offset adjustment velocity |
| nn03 | 51nn | BA | Set map base advance |
| nn04 | 51mm | BL | Set backlash compensation distance |
| nn05 | 52nn | BT | Set master speed averaging time constant |
| nn06 | 52mm | CW | Control word |
| nn07 | 53nn | DB | Debounce time |
| nn08 | 53mm | DZ | Define zero marker input on/off |
| nn09 | 54nn | IN | Initialise position (Write only) |
| nn10 | 54mm | IT | Integration time constant |
| nn11 | 55nn | KD | Differential gain |
| nn12 | 55mm | KF | Velocity feed-forward gain |
| nn13 | 56nn | KI | Integral gain |
| nn14 | 56mm | KM | Monitor output gain |
| nn15 | 57nn | KP | Proportional gain |
| nn16 | 57mm | KV | Velocity feedback gain |
| nn17 | 58nn | LH | High position limit |
| nn18 | 58mm | LL | Low position limit |
| nn19 | 59nn | MA | Move to absolute position (Write only) |
| nn20 | 59mm | MB | Map base offset |
| nn21 | 60nn | ME | Motor error sequence |
| nn22 | 60mm | MF | Slave map position offset |
| nn23 | 61nn | ML | Link to master channel for mapping (Write only) |
| nn24 | 61mm | MP | Map bound |
| nn25 | 62nn | MR | Move to relative position (Write only) |
| nn26 | 62mm | MS | Map step |
| nn27 | 63nn | MT | Map timeout |
| nn28 | 63mm | MW | Map options word |
| nn29 | 64nn | OC | Set value on expanded output group (Write only) |
| nn30 | 64mm | OM | Monitor output offset |
| nn31 | 65nn | PA | Set position trigger output advance |
| nn32 | 65mm | PV | Profile velocity |
| nn33 | 66nn | RF | Set reference offset |
| nn34 | 66mm | RL | Set reference repeat length |
| nn35 | 67nn | RT | Registration timeout |

nn = channel number. mm = channel number + 50.

 

**Channel Data (continued)**

| Address | | Command | Contents |
|---|---|---|---|
| nn36 | 67mm | RV | Reference velocity |
| nn37 | 68nn | RW | Reference word |
| nn38 | 68mm | SA | Acceleration |
| nn39 | 69nn | SB | Position overflow bound |
| nn40 | 69mm | SC | Creep distance |
| nn41 | 70nn | SD | Deceleration |
| nn42 | 70mm | SE | Error band |
| nn43 | 71nn | SF | Auxiliary output function |
| nn44 | 71mm | SJ | Deferred adjustment position |
| nn45 | 72nn | SL | Settling time |
| nn46 | 72mm | SM | Map scaling |
| nn47 | 73nn | SR | Maximum reference correction |
| nn48 | 73mm | SS | Creep speed |
| nn49 | 74nn | SU | Set units |
| nn50 | 74mm | SV | Velocity |
| nn51 | 75nn | SW | Window on move endpoint |
| nn52 | 75mm | TM | Transfer map (Write only) |
| nn53 | 76nn | TO | Move timeout |
| nn54 | 76mm | TP | Transfer profile (Write only) |
| nn55 | 77nn | UE | User error sequence |
| nn56 | 77mm | US | Send user signal (Write only) |
| nn57 | 78nn | VC | Enter constant velocity mode (Write only) |
| nn58 | 78mm | VM | Set virtual motor mode |
| nn59 | 79nn | VT | Set velocity averaging time constant |
| nn60 | 79mm | WC | Wait for bound overflow count (Write only) |
| nn61 | 80nn | XM | Execute map (Write only) |
| nn62 | 80mm | XP | Execute profile (Write only) |
| nn63 | 81nn | YA | Wildcard parameter A |
| nn64 | 81mm | YB | Wildcard parameter B |
| nn65 | 82nn | YC | Wildcard parameter C |
| nn66 | 82mm | YD | Wildcard parameter D |
| nn67 | 83nn | YE | Wildcard parameter E |
| nn68 | 83mm | YF | Wildcard parameter F |
| nn69 | 84nn | YG | Wildcard parameter G |

nn = channel number. mm = channel number + 50.

 

**Channel Data (continued)**

| Address | | Command | Contents |
|---|---|---|---|
| nn70 | 84mm | YH | Wildcard parameter H |
| nn71 | 85nn | YI | Wildcard parameter I |
| nn72 | 85mm | YJ | Wildcard parameter J |
| nn73 | 86nn | ZC | Set position counters (Write only) |
| nn74 | | | Reserved |
| nn75 | 87nn | AB | Abort (Write only) |
| nn76 | 87mm | SO/CO | Output lines group 1 |
| nn77 | 88nn | SO/CO | Output lines group 2 |
| nn78 | 88mm | SO/CO | Output lines group 3 |
| nn79 | 89nn | ID | Initialise demand position offset (Write only) |
| nn80 | 89mm | MO | Motor off (Write only) |
| nn81 | 90nn | PC | Position Control (Write only) |
| nn82 | 90mm | RM | Set continuous reference mode on/off |
| nn83 | 91nn | ST | Stop (Write only) |
| nn84 | 91mm | UL | Unlink (Write only) |
| nn85 | 92nn | DA | Display analogue input |
| nn86 | 92mm | DF | Display reference error |
| nn87 | 93nn | DP | Display actual position |
| nn88 | 93mm | DS | Display snapshot position data |
| nn89 | 94nn | DV | Display velocity |
| nn90 | 94mm | DD | Display demand position |
| nn91 | 95nn | | Motor error code |
| nn92 | 95mm | | User error code |
| nn93 | 96nn | | Status word |
| nn94 | 96mm | | Input group 1 |
| nn95 | 97nn | | Input group 2 |
| nn96 | 97mm | | Input group 3 |
| nn97 | 98nn | | Input group 4 |
| nn98 | 98mm | | Input group 5 |

nn = channel number. mm = channel number + 50.

 

# 6. Error Codes

System error codes can be read from channel registers nn91 and nn92. These registers hold respectively the last motor error and last user error which occurred on channel nn. The codes are given in section 5.4 of the relevant PTS Reference Manual.

 

# 7. Register Bit Mapping

## 7.1 Status Word Register

The meaning of the value in the Status word register (channel register nn93) are shown below.

**Status Value**

| Value | Meaning when set |
|---|---|
| 1 | Waiting for time (WT) |
| 2 | Waiting for input line (WI) |
| 3 | Waiting for absolute position (WA) |
| 4 | Waiting for relative position (WR) |
| 5 | Waiting for reference signal (WR) |
| 6 | Waiting for bounds wraparound (WB) |
| 7 | Waiting for bounds counter value (WC) |
| 256 | Constant velocity mode (VC) |
| 512 | Moving (MA/MR) |
| 1024 | Profiling (XP) |
| 2048 | Mapping (XM) |
| 4096 | Stopping (ST) |
| 8192 | Initialising (IN) |
| 16384 | Torque control mode (TQ) |
| 32768 | Motor off |

 

## 7.2 Input/Output Registers

The bits in the output line registers (nn76 etc.) and the input line register (nn94 etc.) reflect the state of the input/output lines as shown below.

**Input/Output Line Bits**

| Bit | Input/Output Line |
|---|---|
| 0 (LSB) | Line 1 |
| 1 | Line 2 |
| 2 | Line 3 |
| 3 | Line 4 |
| 4 | Line 5 |
| 5 | Line 6 |
| 6 | Line 7 |
| 7 | Line 8 |
| 8 .. 15 | Reserved |

 