Panasonic Super Hybrid (DPT) Documentation V0.10

TechnicallyObsolete

2023-09-08

Contents

1 Introduction

This document is a collection of notes from reverse engineering the Panasonic Super Hybrid DPT ("Digital Proprietary Telephone") communications interface.
This document aims to cover the following:

- Digital Proprietary Telephones (DPT)
- Digital Voice Processing Systems (VPS)

It will not cover:

- Analogue Proprietary Telephones (APT)
  - These use a very different physical interface
- Direct Station Selection (DSS) consoles
  - These use a similar interface to APTs
- KX-TDA0142 DECT Cell stations
  - These use a very different interface, based around the MicroSemi MT8971 using full duplex communication with echo cancellation.
    Additionally, rather than 2 channels of G.711 PCM as used in the DPT interface, these cell stations transport the raw ADPCM data from the DECT phone.

The KX-TDA0141 DECT cell stations use a DPT style interface, but I don't currently have any for reverse engineering.

2 Physical Layer

Line voltage is typically 40v for modern DPTs, and 15v for older DPTs. The PBX supplies 40v by default, and switches to 15v (if required) after the model ID has been read from the DPT. The KX-TDA30 service manual suggests a maximum of 80mA is available per DPT port (to be split between two phones). This appears only be available after the model ID has been read (20mA is available before this point).

With an analogue voltmeter connected, the voltage can be seen to vary when no telephone is connected, but it is consistent once a DPT has been connected. This line voltage variation is a result of the polling for different types of DPT (/APT).

Tbit = 1.95μs (more accurately, $\frac{1}{512k}$ seconds per bit).
A frame has a total of 64 bits including idle bits, for a total frame time of 125μs.

Voltage level `+2.0V` or `-2.0V` with respect to the idle voltage to encode a 0.
0V with respect to the idle voltage to encode a 1.

Data is encoded using "Alternate Mark Inversion", where every 0 bit sent has the opposite polarity to the previous bit sent. An exception to this happens near to the start of each frame. For example (using `+`, `0` and `-` to represent line polarity), this would look like `++-` or `--+` (this can be seen in the sample packet waveforms in section 10).

3 Bit Assignment

```
PBX -> Phone
Header : 00000  : 5 bits of 0s as a header
MAM    : 000    : 3 additional 0 bits, but with an AMI coding violation
Cch    :        : 1 bit "Control Channel", for identifying the type of phone?
Dch    :        : 2 bits of HDLC coded D channel
Bch1   :        : 8 bits of Bearer Channel 1, MSB first. This is inverted.
Bch2   :        : 8 bits of Bearer Channel 2, MSB first. This is inverted
Parity :        : Ensures the number of 0 bits over the entire frame is even

T/R switch: 1   : always a 1, happens at the handover from PBX driving to phone driving

Phone -> PBX
Header : 00000  : 5 bits of 0s as a header
MAM    : 000    : 3 additional 0 bits, but with an AMI coding violation
Cch    :        : 1 bit "Control Channel", for identifying the type of phone?
Dch    :        : 2 bits of HDLC coded D channel
Bch1   :        : 8 bits of Bearer Channel 1, MSB first. This is inverted
Bch2   :        : 8 bits of Bearer Channel 2, MSB first. This is inverted
Parity :        : Ensures the number of 0 bits over the entire frame is even

Line idle (7 bit times)
```

Note that the first 0 bit of the header will be transmitted with the same polarity as the last 0 bit of the received message. This cancels out the DC offset that is introduced by the AMI coding violation (which has two 0 bits with the same polarity).

Using the same convention identified previously to identify the line state, an example message would look like:
`000...0` (Line idle)
`-+-+-` (Header, 00000)
`++-` (MAM 000, with an AMI coding violation)
`0` (C Channel = 1 in this example)
`+0` (D channel bits = 01 in this example)
`000000-0` (B Channel 1, 0x02 in this example)
`+-0+0-+-` (B Channel 2, 0xD7 in this example - 00101000 before inversion)
`TODO` Parity bit

4 B Channels

Two B channels are present, named here as “B1” and “B2”.

These carry standard 8 bit G.711 encoded data (μ-Law or A-Law, depending on the PBX settings). This data is inverted, and MSB is sent first.

When two DPTs are used in DXDP (“Digital eXtension Device Port”), the phone connected directly to the PBX always uses B1, and the second phone always uses B2. Where only one DPT is present, it always uses B1.

5 C Channel

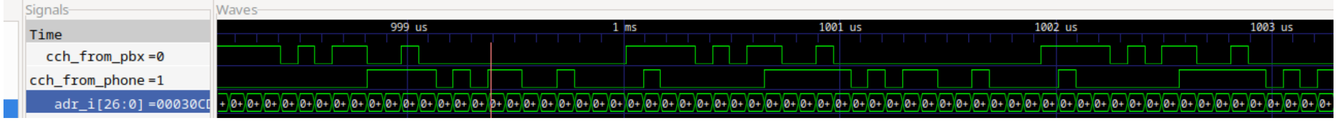
Not much is currently known about the C channel.

In the direction from Phone to PBX, the “New” DPTs (and the VPS I have here, a KX-TVP50) have this bit static 1, and the “Old” DPT I have (a KX-7433, with 3 line LCD) has this set to static 0 after the phone has been initialised.

Before the phone has been initialised, a pattern can be seen.

Data in the PBX to Phone direction shows toggling in the C channel. When plotted, a pattern can be seen in this data:

Figure 1: C Channel Capture



```
PBX         : 000000000000111101011001
Phone (New): 000000111110101100100001
```

6 D Channel (Overview)

The D channel format is very different between the “Old” DPT (operating at 15V) and “New” DPT (operating at 40V)

7 D Channel (New DPT)

Data format is similar (but not identical) to X.25.
The standard X.25 checksum is used, but the control field encoding does not match X.25

The first two bytes of any message are an address, showing the source of the packet and if it is either a data frame or an ACK to a previous frame.

The next two bytes are some kind of sequence counter, and toggle on a packet-by-packet basis.

7.1 Address Bytes

The following addresses have been observed:

Address byte 1	Source	Destination	Notes
0x04	PBX	Phone 1	Reply from PBX
0x06	PBX	Phone 1	Message generated by PBX
0x08	PBX	Phone 2	Reply from PBX
0x0A	PBX	Phone 2	Message generated by PBX
0x00	Phone 1 or 2	PBX	Reply from phone
0x02	Phone 1 or 2	PBX	Message generated by phone

For frames where the source is the PBX, Address 2 is always `0x01`.

For frames where the source is Phone 1, Address 2 is always `0x03`, and for where the source is Phone 2, Address 2 is always `0x05`. The exception to this rule is at startup, where `0x0F` will be seen.

An alternative way to consider the address bits is:

- Address 1 is the destination
  - Bit 0 is clear to indicate destination
  - Bit 1 indicates ACK (=0) or Message (=1)
  - Bits [3:2] indicate the destination is PBX (=0), Phone 1 (=1) or Phone 2(=2)
- Address 2 is the source
  - PBX = 0x01, Phone 1 = 0x03, Phone 2 = 0x05, Broadcast = 0x0F

7.2 Toggle bytes 1 and 2

Toggle byte 1 has been observed to have the following values: `0x00, 0x01, 0x02`.
`0x01` is only ever observed in an ACK.

Toggle byte 2 has been observed to have the values `0x01, 0x03`

A sample packet capture showing the operation of these toggle bytes is as follows:

```
PBX->PHONE 06 01 00 03 40 03 A8 7F
PHONE->PBX 00 03 01 03 F9 38
PBX->PHONE 06 01 02 03 00 04 79 01 7D 00 00 F1
PHONE->PBX 00 03 01 01 EB 1B
PBX->PHONE 06 01 00 03 40 01 BA 5C
PHONE->PBX 00 03 01 03 F9 38
PBX->PHONE 06 01 02 03 00 02 78 01 B8 8E
PHONE->PBX 00 03 01 01 EB 1B
```

In this example, all messages are from the PBX, and all ACKs are from the phone.

Toggle byte 1 in the PBX to Phone direction can be seen to toggle between `0x00` and `0x02` on alternate messages from the PBX.
Toggle byte 2 in the PBX to Phone direction is static, because the phone has not sent any messages back (only ACKs)

Toggle byte 1 in the Phone to PBX direction is always `0x01`, as the phone is only ever sending back ACKs.
Toggle byte 2 in the Phone to PBX direction toggles between `0x01` and `0x03`. If toggle byte 1 of the incoming message was `0x00`, then toggle byte 2 of the ACK will have the value `0x03`. If toggle byte 1 was `0x02`, then toggle byte 2 will have the value `0x01`

A sample packet capture for a Phone to PBX message is as follows:

```
PHONE->PBX 02 03 00 01 01 0B 03 E2 27
PBX->PHONE 04 01 01 03 AD FF
```

Here, toggle byte 1 from the phone was `0x00`, so toggle byte 2 in the ACK from the PBX is `0x03`.

7.3 Messages

This section details the messages sent between the phone and the PBX.
The following conventions are used in the command formats:

```
<A1> Address byte 1 (Destination)
<A2> Address byte 2 (Source)
<T1> Toggle byte 1
<T2> Toggle byte 2
<C1> CRC byte 1
<C2> CRC byte 2
```

Note: The example commands in the following section were captured with two different setups, one of which shows the entire packet (including address, toggle and CRC bytes), and the other setup which includes only the command and data bytes.

The address, toggle and CRC bytes are present in every packet sent.

7.3.1 ACKs

```
04 01 01 01 bf dc (PBX ACK's message from phone 1)
04 01 01 03 ad ff (PBX ACK's message from phone 1)
08 01 01 01 8b 4b (PBX ACK's message from phone 2)
08 01 01 03 99 68 (PBX ACK's message from phone 2)

00 03 01 01 eb 1b (phone 1 ACK's message from PBX)
00 03 01 03 79 38 (phone 1 ACK's message from PBX)
00 05 01 01 32 cd (phone 2 ACK's message from PBX)
00 05 01 03 20 ee (phone 2 ACK's message from PBX)
```

7.3.2 Summary

Command Name	Code	Args	Length Present?
LED Control 7.3.3	0x00	0	Y
Button Press 7.3.5	0x01	2	N
Startup (from Phone) 7.3.6	0x0F	3	N
LCD Update (full line) 7.3.7	0x21	2	Y
LCD Update (multiple line) 7.3.8	0x22	2	Y
LCD Update (Single Char) 7.3.9	0x30	?	?
Audio Path Control 7.3.10	0x40	1	N
Program Mode Status 7.3.11	0x41	1	N
Speaker Enable 7.3.12	0x42	1	N
Remote Audio Connect 7.3.13	0x44	3	N
Volume / Contrast change 7.3.14	0x45	2	N
Headset Mode 7.3.15	0x46	1	N
Unknown	0x47	0	Y
Backlight Control 7.3.17	0x48	2	N
DSP Coefficients 7.3.18	0x50	0	Y
Startup (from PBX) 7.3.18	0x70	0	Y

7.3.3 LED Control

`CMD: 0x00 Args Length: 0 Message length parameter: Present`

The LED control command allows a variable number of LEDs to be updated. Each LED to be updated has an index and a value, and the message length is twice the number of LEDs to be updated.

```
<A1> <A2> <T1> <T2> 0x00 <length> <LED 1 index> <LED 1 value> ... <C1> <C2>
```

A few examples can be seen below:

```
#Update two LEDs: Speakerphone LED static on, auto answer LED off
06 01 00 03 00 04 79 01 7D 00 6F FA

#Auto Answer LED on
06 01 00 01 00 02 7D 01 DE EE
```

A map of LED indexes can be found in Table 2, and LED values can be found in Table 1.

Note that some DPTs do not have a green LED in the message indicator to indicate an incoming call - in these cases, the red LED will be lit instead.

Table 1: LED Values (New DPT)

LED Value	
Off	0x00
Static On	0x01
Slow Flash	0x04
Fast Flash	0x07

Table 2: LED index (New DPT)

LED	Index
Line 1 Red	0x00
Line 2 Red	0x01
Line 3 Red	0x02
Line 4 Red	0x03
Line 5 Red	0x04
Line 6 Red	0x05
Line 7 Red	0x06
Line 8 Red	0x07
Line 1 Green	0x80
Line 2 Green	0x81
Line 3 Green	0x82
Line 4 Green	0x83
Line 5 Green	0x84
Line 6 Green	0x85
Line 7 Green	0x86
Line 8 Green	0x87
Intercom	0x78
Speaker Phone	0x79
Message (Red)	0x7B
Auto Answer	0x7D
Auto Dial	0x7E
Message (Green)	0xFB

7.3.4 Model ID

`CMD: 0x00 Args Length: 4 Message length parameter: Present, non standard`

This command is used to identify the type of phone connected. Note that the command is the same as the LED update command (however, LED update is only sent by a PBX, and model ID is only sent by a phone).

```
KX-T7668: 02 03 00 03 00 00 03 01 01 10 02 02 08 04 03 CE \
          F9 80 01 21 01 23 02 24 03 02 25 04 02 26 0C 02 \
          27 05 02 60 01 7F F2

KX-T7633: 02 03 00 03 00 00 04 01 03 18 00 02 02 18 04 03 FE \
          DF E0 02 04 04 01 20 01 21 01 22 01 23 02 24 04 02 \
          25 04 02 26 0C 02 27 05 01 28 85 88

KX-DT321: 02 03 00 03 00 00 04 01 01 10 00 02 02 08 04 03 CE \
          FF 00 01 21 01 23 02 24 03 02 25 04 02 26 0C 02 27 \
          05 02 60 01 C4 D9
```

The exact encoding of this command is not known, but a few observations can be made on the data section of the packet:

1. For the KX-T7668 with a 1x16 LCD, byte[4] = 0x01 and byte[5] = 0x10
2. For the KX-DT321 with a 1x16 LCD, byte[4] = 0x01 and byte[5] = 0x10
3. For the KX-T7633 with a 3x24 LCD, byte[4] = 0x03 and byte[5] = 0x18

7.3.5 Button Presses

```
CMD: 0x01 Args Length: 0 Message length parameter: Not present
```

Button presses (including lifting and releasing the handset) are sent as a two byte scancode. No length byte is present. The basic format is:

```
<A1> <A2> <T1> <T2> 0x01 <MSB scan code> <LSB scan code> <C1> <C2>
```

A few examples can be seen below:

```
press 1: 02 03 00 01 01 07 01 50 ad
press 2: 02 03 02 01 01 07 02 43 89
press 3: 02 03 00 03 01 07 03 34 b7
```

The complete table of scancode mapping is found in Table 3

Table 3: Button press scancodes (New DPT)

Key	Scancode
0	0x07 0x0A
1	0x07 0x01
2	0x07 0x02
3	0x07 0x03
4	0x07 0x04
5	0x07 0x05
6	0x07 0x06
7	0x07 0x07
8	0x07 0x08
9	0x07 0x09
*	0x07 0x0B
#	0x07 0x0C
Press Flash/Recall	0x03 0x01
Release Flash/Recall	0x03 0x00
Auto Answer / Mute	0x0A 0x00
Speaker Phone	0x0A 0x01
Conference	0x0A 0x02
FWD / DND	0x0A 0x03
Auto Dial / Store	0x0A 0x04
Intercom	0x0A 0x05
Message	0x0A 0x06
Transfer	0x0B 0x00
Redial	0x0B 0x01
Program	0x0B 0x02
Up / Volume Up	0x0B 0x03
Down / Volume Down	0x0B 0x04
Left	0x0B 0x05
Right	0x0B 0x06
Hold	0x0B 0x07
Pause	0x0B 0x08
Enter	0x0B 0x09
Esc	0x0B 0x0A
Line 1	0x0C 0x00
Line 2	0x0C 0x01
Line 3	0x0C 0x02
Line 4	0x0C 0x03
...	
Line 23	0x0C 0x16
Line 24	0x0C 0x17
SoftKey 1	0x0D 0x00
SoftKey 2	0x0D 0x01
SoftKey 3	0x0D 0x02
SoftKey 4	0x0D 0x03
Lift Handset	0x02 0x00
Replace Handset	0x01 0x00

7.3.6 Startup (from Phone)

```
CMD: 0x0F Args Length: 3 Message length parameter: Not Present
```

This command is the first sent by a DPT after power up. The basic format is:

```
<A1> 0F <T1> <T2> 0F <Arg1?> <Arg2?> <Arg3?>

02 0F 03 01 0F 00 01 01 FD 78 32 #Directly connected phone
02 0F 03 01 0F 00 02 01 FD 1C DD #Connected via DXDP
```

This is sent three times by the phone. In reply, the PBX does not ACK, but instead generates the following message:

```
04 01 03 01 0F 00 01 02 03 F3 92 #ACK to directly connected phone
08 01 03 01 0F 00 02 02 05 8E 85 #ACK to phone via DXDP
```

7.3.7 LCD Update (Full Line?)

```
CMD: 0x21 Args Length: 2 Message length parameter: Present
```

This command updates an entire line of text on the LCD. The basic format is:

```
<A1> <A2> <T1> <T2> 0x21 <Start Y> <Start X> <length> <ASCII chars> <C1> <C2>
```

An example can be found below:

```
#Set Display line 0 to "    20 JAN. 07:04PM"
06 01 00 01 21 00 00 10 20 32 30 20 4a 41 4e 2e 20 30 37 3a 30 34 50 4d 5e 2e
```

7.3.8 LCD Update (Multi Line)

```
CMD: 0x22 Args Length: 2 Message length parameter: Present
```

This command is seen on multiline DPTs. It updates multiple lines of the display.

7.3.9 LCD Update (Single Character)

```
CMD: 0x30 Args Length: 4 Message Length parameter: Not Present
```

This command is used to partially update the LCD. The basic format is:

```
<A1> <A2> <T1> <T2> 0x30 <arg1?> <arg2?> <arg3?> <arg4?> <ASCII Character> <C1> <C2>

06 01 00 01 30 07 00 00 10 32 8e 29 #Shift display left, and place a ’2’ on the right
               03 00 00 10 00       #Backspace
```

Arg1 can be the following:

02: Shift display left, don’t append char (also 0x01 and 0x06) 03: Backspace 04: Set display blink 05: Stop display blink 07: Shift left, append character

7.3.10 Audio Path control

```
CMD: 0x40 Args Length: 0 Message length parameter: Not present
```

This command is used to switch on or off an audio path. The basic format is:

```
<A1> <A2> <T1> <T2> 0x40 <Path setting> <C1> <C2>
```

An example can be found below:

```
#Start handset call
06 01 02 01 40 00 FD C1
```

Possible audio path options are noted in Table 4:

Table 4: Audio Path Options

Handset Call	0x00
Speaker phone or headset call	0x01
Handset call, but with audio through the speaker as well	0x11
Speaker phone unmute	0x03

7.3.11 Program Mode Status

```
CMD: 0x41 Args Length: 1 Message Length parameter: Not Present
```

This is sent when entering and exiting program mode.

1. arg1=1 is sent by the PBX when entering program mode
2. arg1=0 is sent when leaving program mode

7.3.12 Speaker Enable (Ring)

```
CMD: 0x42 Args Length: 2 Message Length Parameter: Not Present
```

This appears to be used to enable the speaker on an incoming call (or paging call). Note that the ringing tone is generated on the phone, rather than being sent through the incoming B channel. The basic format is

```
42 <Ring Cadence> <Ring Tone>
```

The Ring Cadence parameter seems to have a similar encoding to the LED control flash patterns. Some examples are as follows:

```
06 01 00 01 42 0C 00 43 D6
Incoming call (ring volume == 0):CMD 0x42 (UNKNOWN   ) 00 00
Incoming call (ring volume > 0) :CMD 0x42 (UNKNOWN   ) 02 00 :Makes ringing noise
Incoming page call              :CMD 0x42 (UNKNOWN   ) 0C 00 :Makes page noise.
```

7.3.13 Remote Audio Connect

```
CMD: 0x44 Args Length: 3 Message Length Parameter: Not Present
```

This appears to be sent once the audio of a call is connected to the remote user.

```
00 00 00 #Sent when audio path connects
01 00 00 #Sent when audio path disconnects
```

7.3.14 Volume / Contrast change

```
CMD: 0x45 Args Length: 2 Message length parameter: Not present
```

This command is used to change the volume level of an audio path, and also to change the LCD contrast. The basic format is:

```
<A1> <A2> <T1> <T2> 0x45 <Index> <Value> <C1> <C2>
```

Some examples can be found below:

```
06 01 00 03 45 00 00 90 CA #Set contrast to 1
06 01 02 01 45 00 01 E7 F4 #Set contrast to 2
06 01 02 03 45 01 02 D2 E6 #Set headset (or handset) volume to 3
```

7.3.15 Headset Mode

```
CMD: 0x46 Args Length: 1 Message length parameter: Not present
```

This command is used to enable or disable the headset path. When the headset is in use, switching to speakerphone mode will instead divert audio to the headset.

```
<A1> <A2> <T1> <T2> 0x46 <Value> <C1> <C2>
```

The two possible values are found below:

```
06 01 02 03 46 00 95 20 #Headset Off
06 01 00 01 46 01 D2 BD #Headset On
```

7.3.16 Unknown (0x47)

7.3.17 Backlight Control

```
CMD: 0x48 Args Length: 2 Message Length parameter: Not Present
```

This command enables or disables the backlight.

```
06 01 02 03 48 03 00 0F 09 #Backlight Off
06 01 02 01 48 02 00 A1 29 #Backlight On
06 01 02 03 48 01 00 BF 3A #Backlight Automatic
```

7.3.18 DSP Coefficients

```
CMD: 0x50 Args Length: 0 Message length parameter: Present
```

This command is not fully understood, but appears to be some kind of DSP coefficient download.

Strings within the KX-TDA30 software refer to a "Gain Table" (which differs for different models of phone), although it is uncertain if all phones use this gain table.

```
<A1> <A2> <T1> <T2> 0x50 <length> <bank> <idx> <val1> <val2>... <C1> <C2>
```

Some examples are as follows:

```
06 01 00 03 50 0A 00 10 00 7F 11 00 04 14 00 73 A4 E4
06 01 02 03 50 0A 01 10 00 01 11 00 08 14 00 79 54 9A
```

The `bank` parameter may be the bank of coefficients to be updated.
This is followed by an `index` and two data bytes per coefficient.

A full example of DSP coefficients can be found in section 11.

7.3.19 Startup (from PBX)

```
CMD: 0x70 Args Length: 3 Message length parameter: Not Present
```

```
<A1> <A2> <T1> <T2> 70 01 00 <C1> <C2>
```

This command is sent by the PBX immediately after it sends the response
(`04 01 03 01 0F 00 01 02 03 F3 92`) to the phone's startup message.

An example is as follows:

```
06 01 00 01 70 01 00 2D 55 #Enable directly connected phone
0A 01 00 01 70 01 00 B7 E4 #Enable phone connected via DXDP (this is sent to phone2)
06 01 00 03 70 03 02 F9 7C #Sent to Phone1, to enable the power switch to phone 2
```

It may be related to the power switching on the phone (I.E. the point at which the DPT is permitted to draw the full operating current). Note that a full description of the startup sequence can be found in subsection 7.4.

7.4 New DPT Startup

The New DPT startup process is as follows.

This should be read in combination with subsubsection 7.3.6 and subsubsection 7.3.19.

Table 5: Startup Sequence (New DPT - Directly Connected)

```
let CCH_PBX   = 000000000000111101011001
let CCH_PHONE = 000000111110101100100001
```

Step	PBX CCH	PBX DCH	Phone CCH	Phone DCH	Notes
1	CCH_PBX				Not connected
2	CCH_PBX		1'b1		Connected
3	CCH_PBX		1'b1	020F03010F000101FD7832	Connected
4	CCH_PBX	040103010F00010203F392	1'b1		PBX ACKs startup
5	CCH_PBX	060100017001002D55	1'b1		PBX sends startup
6	CCH_PBX		1'b1	ACK	Startup Done

Table 6: Startup Sequence (New DPT - Via DXDP)

```
let CCH_PBX   = 000000000000111101011001
```

Step	PBX CCH	PBX DCH	Phone CCH	Phone DCH	Notes
1	CCH_PBX				Not connected
2	CCH_PBX		CCH_PHONE		Connected
3	CCH_PBX		CCH_PHONE	020F03010F000201FD1CDD	Connected
4	CCH_PBX	080103010F000202058E58	1'b1		PBX ACKs startup
5	CCH_PBX	0A010001700100B7E4	1'b1		PBX sends startup
6	CCH_PBX	06010003700302F97C	1'b1		PBX->Phone1
8	CCH_PBX		1'b1	ACK	Startup Done

8 D Channel (Old DPT)

This format is very approximately based on X.25. The standard X.25 checksum is used, but the control fields are used very differently.

Interestingly, although the command encoding is very different to the new DPT protocol, the same command opcodes are kept (for at least a few functions). For example, 0x01 is used to report a button press and 0x20 represents an LCD update in both cases.

8.1 Status Report

Rather than use the ACK packets as in the new DPT protocol, the old protocol is based on a status report. This has the same encoding as a button press, except that is constantly sent.

```
01 00 A0 1A 39 #Status from Phone, on hook
01 00 A2 08 1A #Status from phone, off hook
```

8.2 Messages

8.2.1 LED Control

CMD: 0x00, 0x03 or 0x04

This command is sent by the PBX to update the LED state. The encoding used is the same encoding used by DSS consoles and APTs and is shown in Table 7.

Each LED is controlled with a 4 bit value (I.E. each byte controls two LEDs). The MSB of this value encodes the colour of the LED (0 = Green, 1 = Red), and the three LSBs control the pattern:

Table 7: LED Control options (old DPT)

0	Static Off
1	Static On
2	Slow Flash
3	Double Flash
4	Fast Flash
5	Not Used
6	
7	

LEDs are mapped to data bytes in Table 8 :

Table 8: LED Control options (old DPT)

LED	Prefix	Data Byte Pos
Line 1	00	0 (MSB)
Line 2	00	0 (LSB)
Intercom	00	6 (MSB)
Speaker Phone	03	6 (MSB)?
Auto Answer	04	6 (MSB)
Do Not Disturb	04	6 (LSB)

Some example LED control messages are as follows:

```
00 90 00 00 00 00 00 02 00 CD 12 : Line 1 Static Red
00 40 00 00 00 00 00 02 00 A6 E7 : Line 1 Flash Green
00 09 00 00 00 00 00 00 00 E4 1D : Line 2 Static Red
00 04 00 00 00 00 00 00 00 39 A7 : Line 2 Flash green
```

8.2.2 Button Presses

```
CMD: 0x01
```

This command is sent by the phone when a button is pressed.

```
01 <Scan Code MSB> <Scan Code LSB> <C1> <C2>
```

Some example button presses are as follows:

```
Press 1: 01 11 20 5B 31
Press 2: 01 12 20 33 1B
Press 3: 01 13 20 EB 02
```

The complete table of scancode mapping is found in Table 9

Table 9: Button press scancodes (Old DPT)

Key	Scancode
0	1A 20
1	11 20
2	12 20
3	13 20
4	14 20
5	15 20
6	16 20
7	17 20
8	18 20
9	19 20
*	1B 20
#	1C 20

Recall / Flash	???
Auto Answer / Mute	22 20
Speaker Phone	23 20
Auto Dial / Store	44 20
Intercom	5F 20
Message	46 20
Transfer	20 20
Redial	21 20
Program	24 A0
Hold	41 20
Conference	40 A0
FWD / DND	43 20
Pause	42 A0

Programmable Fn 1	50 20
Programmable Fn 2	51 20
Programmable Fn 3	52 20

...

Programmable Fn 11	5A 20
Programmable Fn 12	5B 20
Programmable Fn 13	90 20
Programmable Fn 14	91 20

...

Programmable Fn 23	9A 20
Programmable Fn 24	9B 20

SoftKey 1	2A 20
SoftKey 2	2B 20
SoftKey 3	2C 20
Shift	27 20

Jog Wheel CW	E1 20
Jog Wheel CCW	F1 A0

8.2.3 LCD Update

```
CMD: 0x20
```

This command updates a maximum of 6 characters on the LCD.

```
20 <start pos> <ASCII Chars> <C1> <C2>
```

The mapping of start pos to physical display position is not known.
A possible encoding is in Table 10

Table 10: LCD position options (Old DPT)

X	Y	Code
0	0	0x0F
6	0	0x09
12	0	0x03
0	1	??
6	1	??
12	1	??
0	2	0x4F
6	2	0x49
12	2	0x43

A few example LCD updates are captured below:

```
20 0F 20 31 36 20 41 55 47 28  _ 16 AUG(
20 09 47 2E 20 31 39 3A D1 C7  _G. 19:__
20 03 33 31 20 20 00 00 BD 05  _31  ____
20 4F 49 4E 46 4F 20 52 71 2F  OINFO Rq/
20 49 4E 47 4F 46 46 20 64 3E  INGOFF d>
20 43 4D 45 4E 55 00 00 38 88  CMENU__8_
```

(Note - the text displayed in this example is 'RNGOFF', rather than 'RINGOFF')

9 D Channel - VPS

This appears to be completely different to both the old and new DPT protocols (but has some slight similarity to the old DPT protocol).

At idle, the following pattern can be observed to loop:

```
VPS->PBX 07 00 00 12 00 00 00 BF B1
PBX->VPS 71 00 00 1E 00 00 00 00 00 C6 69
VPS->PBX 10 00 00 12 00 00 00 6E C3
PBX->VPS 02 00 00 1E 00 00 00 00 00 0F 51
VPS->PBX 21 00 00 12 00 00 00 F1 8A
PBX->VPS 13 00 00 1E 00 00 00 00 00 66 E3
VPS->PBX 32 00 00 12 00 00 00 56 97
VPS->PBX 43 00 00 12 00 00 00 CF 19
```

```
PBX->VPS 24 00 00 1E 00 00 00 00 00 29 02
VPS->PBX 54 00 00 1E 00 00 00 00 00 40 B0
VPS->PBX 65 00 00 12 00 00 00 81 22
PBX->VPS 46 00 00 1E 00 00 00 00 00 89 88
PBX->VPS 57 00 00 1E 00 00 00 00 00 E0 3A
VPS->PBX 76 00 00 12 00 00 00 26 3F
PBX->VPS 60 00 00 1E 00 00 00 00 00 AF DB
VPS->PBX 07 00 00 12 00 00 00 BF B1
```

An incoming call is indicated by the following:

```
PBX -> VPS 71 00 11 26 5F 00 00 00 00 CC BA #Call from 265
PBX -> VPS 73 00 11 26 4F 00 00 00 00 76 95 #Call from 264
PBX -> VPS 64 00 11 12 3F 00 00 00 00 8F 41 #Call from 123
```

When in a call, the status of the VPS changes:

```
VPS->PBX 65 00 00 12 01 00 00 5D 78
```

10 Sample Packet Waveform

Figure 2: Sample packet capture

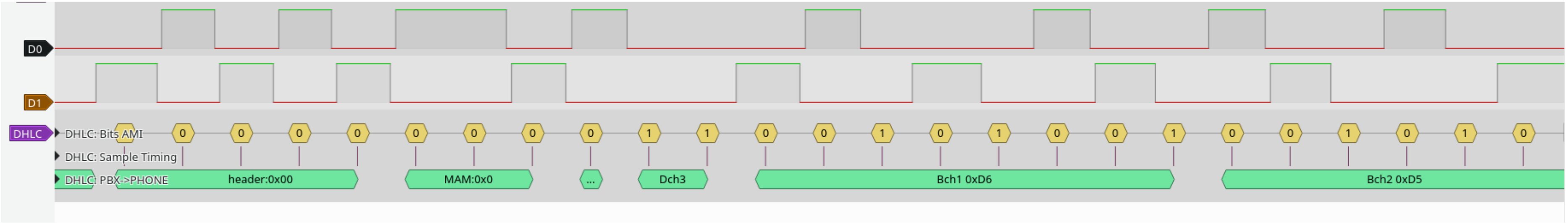
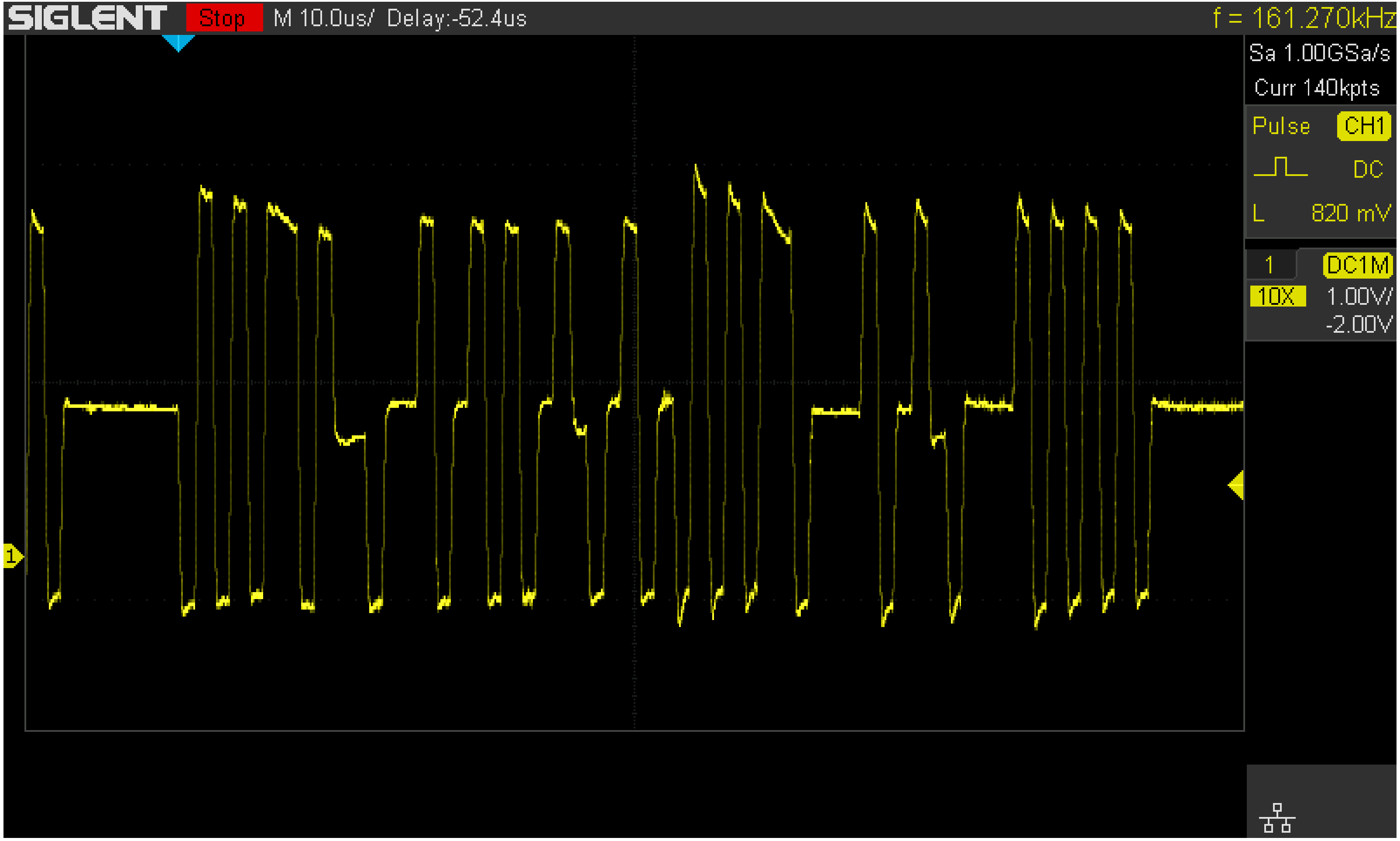


Figure 3: Oscilloscope capture



M 10.0us/ Delay:-52.4us

f = 161.270kHz

Sa 1.00GSa/s

Curr 140kpts

Pulse CH1

DC

L 820 mV

1 DC1M

10X 1.00V/

-2.00V

11 DSP Coefficients

Note: These have been written based on the data section only (they do not contain address, toggle or CRC bytes).

```
0A 00 10 00 7F 11 00 04 14 00 73
0A 01 10 00 01 11 00 08 14 00 79

5B 02 12 00 08 13 00 70 1A 00 72 1B 00 0A 43 40 00 44 07 80 45 00 A3 46 04 00 47 20 00 \
   48 04 00 49 2D 4E 4A 16 B5 4B 20 00 4C 10 00 4D 0F FF 4E 08 00 4F 00 57 50 00 01 51 \
   00 02 52 02 45 53 00 B7 54 02 DB 55 02 06 56 40 00 57 40 00 58 00 00 59 00 04 5A 00 \
   00 5B 00 01 40 00 01

0D 03 12 00 0B 13 00 4D 1A 00 7B 1B 00 4D

61 04 10 00 7F 11 00 04 12 00 08 14 00 73 1A 00 72 1B 00 00 43 40 00 44 07 80 45 00 A3 \
   46 04 00 47 20 00 48 04 00 49 2D 4E 4A 16 B5 4B 20 00 4C 10 00 4D 0F FF 4E 08 00 4F \
   00 57 50 00 01 51 00 00 52 02 45 53 00 B7 54 10 00 55 01 9B 56 40 00 57 40 00 58 00 \
   00 59 00 04 5A 00 00 5B 00 01 40 00 01

13 05 10 00 7F 11 00 4D 12 00 09 14 00 73 1A 00 7B 1B 00 4D

64 06 10 00 7F 11 00 04 12 00 08 13 00 70 14 00 73 1A 00 72 1B 00 0A 43 40 00 44 07 80 \
   45 00 A3 46 04 00 47 20 00 48 04 00 49 2D 4E 4A 16 B5 4B 20 00 4C 10 00 4D 0F FF 4E \
   08 00 4F 00 57 50 00 01 51 00 02 52 02 45 53 00 B7 54 02 DB 55 02 06 56 40 00 57 40 \
   00 58 00 00 59 00 04 5A 00 00 5B 00 01 40 00 01

16 07 10 00 7F 11 00 4D 12 00 0B 13 00 4D 14 00 73 1A 00 7B 1B 00 4D
16 08 13 00 0E 19 00 7A 1A 00 7B C4 B5 C5 C6 0A C4 C7 6D 7A C8 CA C4
0D 09 17 00 77 A2 00 03 C2 00 03 D4 00 03
16 0A 11 00 4D 13 00 4D 18 00 7A 1B 00 4D D2 7F FF D3 65 AB D4 00 03
0A 0B 10 00 4D 16 00 62 D4 00 03
0D 0C 12 00 4D 17 00 6A 1A 00 4D D4 00 03
```

12 Appendix - Change History

- 2023-09-04 V0.11 Update “Startup” and “C Channel” sections for new DPT
- 2023-08-19 V0.10 Major update
  - Split D channel section into “Old DPT” and “New DPT”
  - Add information about frame structure
  - Add detailed information about commands
- 2023-08-10 V0.9 Initial Document version