OE RANGE
BY IMENCO

☐ CAMPUS 1
ABERDEEN INNOVATION PARK
BALGOWNIE ROAD
BRIDGE OF DON
ABERDEEN
AB22 8GT

Tel: (01224) 226500

☒ AIRPORT INDUSTRIAL ESTATE
WICK
CAITHNESS
KW1 4QS

Tel: (01955) 603606

TITLE:

Software Protocol Document

For the OE10-104.

Serial Pan and Tilt Unit

IMENCO UK	DATE	2C	17790	03/23
DRAWN: S Wilson	08/16	2B	17453	03/23
CHECKED:		2A	16960	06/20
APPROVED:		2	16464	12/18
ORIGIN DOC No:		ISS	MOD	DATE
DRAWING No: 10-104-5035		SHEET 1 of 15		

SM21/3A

TABLE OF CONTENTS

2C	17790	03/23
Iss	Mod	Date
Drawing No. **10-104-5035**		Sheet 2

SM21/3A

RECORD OF ISSUE AND REVISION

COPY NO: **ISSUED TO:**

LOCATION: **DATE:**

DATE	REV NO.	CRN	DETAILS	AUTHORISED
08/16	1	-	First Issue	SW
05/17	1A	15771	ID 0x03	RB
05/17	1A	15798	ED Command	RB
01/18	1C	16193	AS: to confirm id	SW
11/18	1D	16123	Example added	RB
12/18	2	16464	Changed to Imenco Format	RB
06/20	2A	16960	Remove Camera Commands	IS
03/23	2B	17453	Change ID Data	RB
03/2023	2C	17790	Change to "AS" command	SW

Authorised amendments shall be notified by or under cover of this page of the Document and shall be issued as complete pages, which shall replace the equivalent unamended pages. The amended pages shall carry the Document reference and Amendment number.

Amendments are to be embodied by the authorised holders of the Document and superseded pages shall be withdrawn and destroyed. The Amendment Record above shall be completed.

A Change Request Note (CRN) shall be used to request and document any change.

2C	17790	03/23
Iss	Mod	Date
Drawing No. **10-104-5035**		Sheet 3

SM21/3A

1. SUMMARY

This document lists the commands that have been implemented in the OE10-104 Serial Pan and Tilt Unit. For a more complete list of the standard Imenco commands, port settings and timings please refer to document 0350-5020.

2. RELATED DOCUMENTS

OE0350-5020 Imenco Standard Network Command Protocol for Imenco Camera.

10-104-5020 Product Specification for the OE10-104 Serial Pan and Tilt Unit

2C	17790	03/23
Iss	Mod	Date
Drawing No. 10-104-5035		Sheet 4

3. PACKET DATA STRUCTURE

The packet data structure is a combination of bytes ranging from 0x00 to 0x7F except for the data section which may contain bytes outside of this range, and will adopt the following structure:

<to:from:length:command:data_1 data2 data_3 ... data_n:checksum:checksum ind>

Where:

To = 1 Byte in the range 0x01 to 0xFF. (0x00 is a reserved address and cannot be used)
> The id of the device the packet is intended for. This may be a peripheral device or a controller. The controllers ID is reserved as 0x01.

From = 1 Byte in the range 0x01 to 0xFF
> The id of the device that has transmitted the packet. This may be a peripheral device or a controller. The controllers ID is reserved as 0x01.

Length = 1 Byte in the range 0x00 to 0xFF.
> The combined length of the command section, plus the data section, plus the colon joining them, e.g. **<to:from:06:TP:034:checksum:checksum ind>.**
> If the length of the command section is greater than 99 bytes then the Length will be set to 99 and the user must assume that the data is correct. Command = variable length with a maximum length of 2 Bytes in the range 0x00 to 0x7F. The command that is to be sent to the peripheral or controller.

Data = Variable length
> The data that is either sent with the command or data that is returned as part of a status check etc. Data elements are separated with space (0x20) delimiters.

Checksum = 1 Byte
> The packet checksum is an XOR of all the bytes (including delimiters such as ':' but excluding the < at the beginning and the :checksum:checksum ind> section at the end.) in the packet.
> If the checksum when calculated, equals 0x3C then the Checksum byte will be set to 0xFF. If the checksum when calculated, equals 0x3E then the Checksum byte will be set to 0xFF.

Checksum Ind = 1 Byte
> If the checksum when calculated, equals 0x3C then the Checksum Ind byte will be set to ASCII 0. If the checksum when calculated, equals 0x3E then the Checksum Ind byte will be set to ASCII 1. For all other values of Calculated checksum the Checksum Ind byte will be set to ASCII G.

Example:

For a standard broadcast status packet ***<to:from:length:command:checksum:checksum ind>***
When converted to hexadecimal the following can be sent:
3C FF 3A 01 3A 03 3A 53 54 3A 3A FA 3A 47 3E

2C	17790	03/23
Iss	Mod	Date
Drawing No.		Sheet
10-104-5035		5

4. FEATURE CONTROL COMMANDS

Following is a list of standard command packets that have been implemented in the OE10-107 Serial Rotator Unit. All additions must be included into this document through the companies change control procedure.

NOTE: Unless otherwise specified, in the commands shown and examples given, the section Byte* represent one(1) eight bit byte. For example in the Tilt Up Command the command sent to the peripheral is:

<to:from:03:TU::checksum:checksum ind>

And the peripheral will reply with:

<to:from:08:ACK:TUByte1Byte2Byte3:checksum:checksum ind>

Where ACK = The ASCII non-printable ACK character = 0x06
Byte1 = The number of hundreds of degrees
Byte2 = The number of tens of degrees
Byte3 = The number of units of degrees
So to reply with an angle of 276 degrees, the reply would be:

<to:from:08:ACK:TU276:checksum:checksum ind>

Where a byte corresponds to a non-printable character in the ASCII codes it will be shown in the command and its reply as its Hexadecimal value except for the non-printable characters "ACK"(0x06) and "NAK"(0x15) which will be shown explicitly. If the byte is a printable character it will be shown in its printable form not as its hexadecimal value.

Check Status command:

Sent to peripheral device :

      \<to:from:03:ST::checksum:checksum ind\>

Returned from peripheral device:

      \<to:from:0D:ACK:STByte1Byte2Byte3…Byte9:checksum:checksum ind\>

    Byte1 = Status information of the camera.

     Bit0 = N/A Reads as '0'
     Bit1 = N/A Reads as '0'
     Bit2 = N/A Reads as '0'
     Bit3 = Pan supported
     Bit4 = Tilt supported
     Bit5 = N/A Reads as '0'
     Bit6 = N/A Reads as '0'
     Bit7 = N/A Reads as '0'

    Byte2 bit 5 is used as an error flag. If not set then there are no errors. If set to 1 then there is an error with the device. If an error is raised the specific error can be found by interrogating the unit with the error diagnosis command.

    Byte2 = Status information of the product.

     Bit0 = N/A Reads as '0'
     Bit1 = N/A Reads as '0'
     Bit2 = N/A Reads as '0'
     Bit3 = N/A Reads as '0'
     Bit4 = N/A Reads as '0'
     Bit5 = Error Flag
     Bit6 = Reserved
     Bit7 = Reserved

    Byte3 = N/A for the OE10-104

    Byte4 to Byte6 = Pan position if applicable.
      Where Byte4 = Hundreds of Degrees,
      Where Byte5 = Tens of Degrees,
      Where Byte6 = Single Degrees,
      transmitted as ASCII characters in the range 0x30h to 0x39h
      (e.g. 000 = 0x30h 0x30h 0x30h , 270 = 0x32h 0x37h 0x30h)

    Byte7 to Byte9 = Tilt position if applicable.
      Where Byte7 = Hundreds of Degrees,
      Where Byte8 = Tens of Degrees,
      Where Byte9 = Single Degrees,
      transmitted as ASCII characters in the range 0x30h to 0x39h
      (e.g. 000 = 0x30h 0x30h 0x30h , 270 = 0x32h 0x37h 0x30h)

2C	17790	03/23
Iss	Mod	Date
Drawing No.		Sheet
10-104-5035		7

Change ID command:

Sent to peripheral device :

        <to:from:04:SI:*:checksum:checksum ind>

Where * = the new peripheral id. The id must be 1 byte in the range 0x02 to 0xFE. (0x00, 0x01 and 0xFF are reserved addresses and cannot be used)

Returned from peripheral device:

        <to:from:04:ACK:SI:checksum:checksum ind>

When the peripheral receives the Change ID command it will CHANGE to the NEW id and transmit the return packet with the NEW id in the "from" section of the packet. This means that the controller will have to swap to the new id to receive the peripherals returned packet. The controller MUST then reply to the peripheral with a status command or a pan and tilt status command. This allows the peripheral to tell whether the controller is receiving on the peripherals new id. If no status command of either type, is received from the controller within 1 second the peripheral will swap back to the old id and wait for commands from the controller. The controller will revert back to the original address if no response to the status command is received.

Request Protocol Version Command:

Send to peripheral command:

        <to:from:03:PV::checksum:checksum ind>

Returned from peripheral device:

        <to:from:06:ACK:PVXY:checksum:checksum ind>

Where X and Y are the ASCII version number of this document at time of software issue (e.g. _B or 1A where "_" is a space character).

Request Software Version command:

Send to peripheral command:

        <to:from:03:CV::checksum:checksum ind>

Returned from peripheral device:

        <to:from:0A:ACK:CVByte1Byte2…Byte 5Byte6:checksum:checksum ind>

Byte1 = Major Change 1st Digit Hex Value
Byte2 = Major Change 2nd Digit Hex Value
Byte3 = Minor Change 1st Digit Hex Value
Byte4 = Minor Change 2nd Digit Hex Value
Byte5 = Revision Change 1st Digit Hex Value
Byte6 = Revision Change 2nd Digit Hex Value

An example of the data bytes would be
0x30 0x31 0x30 0x34 0x32 0x38 = software version 01 04 28.

2C	17790	03/23
Iss	Mod	Date
Drawing No.		Sheet
10-104-5035		8

Pan Left Command:
Sent to peripheral device :
    <to:from:03:PL::checksum:checksum ind >

Returned from peripheral device:
    <to:from:07:ACK:PLByte1Byte2Byte3:checksum:checksum ind >

    Byte1 to Byte3 = Current Pan position if applicable.
        Where Byte1 = Hundreds of Degrees,
        Where Byte2 = Tens of Degrees,
        Where Byte3 = Single Degrees,
        Transmitted as ASCII characters in the range 0x30h to 0x39h
        (e.g. 000 = 0x30h 0x30h 0x30h , 270 = 0x32h 0x37h 0x30h)

Pan Right command:
Sent to peripheral device :
    <to:from:03:PR::checksum:checksum ind >

Returned from peripheral device:
    <to:from:07:ACK:PRByte1Byte2Byte3:checksum:checksum ind >

    Byte1 to Byte3 = Current Pan position if applicable.
        Where Byte1 = Hundreds of Degrees,
        Where Byte2 = Tens of Degrees,
        Where Byte3 = Single Degrees,
        transmitted as ASCII characters in the range 0x30h to 0x39h
        (e.g. 000 = 0x30h 0x30h 0x30h , 270 = 0x32h 0x37h 0x30h)

Pan Stop command:
Sent to peripheral device :
    <to:from:03:PS::checksum:checksum ind >

Returned from peripheral device:
    <to:from:07:ACK:PSByte1Byte2Byte3:checksum:checksum ind >

    Byte1 to Byte3 = Current Pan position if applicable.
        Where Byte1 = Hundreds of Degrees,
        Where Byte2 = Tens of Degrees,
        Where Byte3 = Single Degrees,
        transmitted as ASCII characters in the range 0x30h to 0x39h
        (e.g. 000 = 0x30h 0x30h 0x30h , 270 = 0x32h 0x37h 0x30h)

Go to Pan Position:
Sent to peripheral device :
    <to:from:06:PP:*…*:checksum:checksum ind >

Returned from peripheral device:
    <to:from:07:ACK:PP*…*:checksum:checksum ind >

Where *…* = the position to go to in degrees ranging from 000 to peripherals maximum pan position transmitted as ASCII characters in the range 0x30h to 0x39h
(e.g. 000 = 0x30h 0x30h 0x30h , 270 = 0x32h 0x37h 0x30h)

2C	17790	03/23
Iss	Mod	Date
Drawing No.		Sheet
10-104-5035		9

<u>Tilt Up command:</u>
Sent to peripheral device:
  <to:from:03:TU::checksum:checksum ind>

Returned from peripheral device:
  <to:from:07:ACK:TUByte1Byte2Byte3:checksum:checksum ind>

  Byte1 to Byte3 = Current Tilt position if applicable.
  Where Byte1 = Hundreds of Degrees,
  Where Byte2 = Tens of Degrees,
  Where Byte3 = Single Degrees,
  transmitted as ASCII characters in the range 0x30 to 0x39
  (e.g. 000 = 0x30 0x30 0x30   270 = 0x32 0x37 0x30)

<u>Tilt Down command:</u>
Sent to peripheral device:
  <to:from:03:TD::checksum:checksum ind>

Returned from peripheral device:
  <to:from:07:ACK:TDByte1Byte2Byte3:checksum:checksum ind>

  Byte1 to Byte3 = Current Tilt position if applicable.
  Where Byte1 = Hundreds of Degrees,
  Where Byte2 = Tens of Degrees,
  Where Byte3 = Single Degrees,
  transmitted as ASCII characters in the range 0x30 to 0x39
  (e.g. 000 = 0x30 0x30 0x30   270 = 0x32 0x37 0x30)

<u>Tilt Stop command:</u>
Sent to peripheral device :
  <to:from:03:TS::checksum:checksum ind>

Returned from peripheral device:
  <to:from:07:ACK:TSByte1Byte2Byte3:checksum:checksum ind>

  Byte1 to Byte3 = Current Tilt position if applicable.
  Where Byte1 = Hundreds of Degrees,
  Where Byte2 = Tens of Degrees,
  Where Byte3 = Single Degrees,
  transmitted as ASCII characters in the range 0x30 to 0x39
  (e.g. 000 = 0x30 0x30 0x30   270 = 0x32 0x37 0x30)

<u>Go to Tilt Position command:</u>
Sent to peripheral device :
  <to:from:06:TP:*…*:checksum:checksum ind>

Returned from peripheral device:
  <to:from:07:ACK:TP*…*:checksum:checksum ind>

  Where *…* = the position to go to in degrees ranging from 000 to peripherals maximum tilt
  position transmitted as ASCII characters in the range 0x30 to 0x39
  (e.g. 000 = 0x30 0x30 0x30   270 = 0x32 0x37 0x30)

2C	17790	03/23
Iss	Mod	Date
Drawing No.		Sheet
10-104-5035		10

Proportional control command:

Sent to peripheral device :

<to:from:07:PC:Byte1Byte2Byte3Byte4:checksum:checksum ind>

Where Byte1 = Commands to action:

Pan Bits   0 & 1: 00_2=Stop, 01_2=Pan Left, 10_2 = Pan Right

Tilt Bits   2 & 3: 00_2=Stop, 01_2=Tilt up, 10_2 = Tilt down

Focus Bits   4 & 5: 00_2=Stop

Zoom Bits   6 & 7: 00_2=Stop

Byte2 = Pan Speed in the range 0x00 - 0x64 where 0x00 = stopped and 0x64 = maximum speed

Byte3 = Tilt Speed in the range 0x00 - 0x64 where 0x00 = stopped and 0x64 = maximum speed

Byte4 = N/A

Returned from peripheral device:

<to:from:04:ACK:PC:checksum:checksum ind>

Proportional Control with Feedback command:

Sent to peripheral device :

<to:from:07:PF:Byte1Byte2Byte3Byte4:checksum:checksum ind>

Where Byte1 = Commands to action:

Pan Bits   0 & 1: 00_2=Stop, 01_2=Pan Left, 10_2 = Pan Right

Tilt Bits   2 & 3: 00_2=Stop, 01_2=Tilt up, 10_2 = Tilt down

Focus Bits   4 & 5: 00_2=Stop

Zoom Bits   6 & 7: 00_2= Stop

Byte2 = Pan Speed in the range 0x00 - 0x64 where 0x00 = stopped and 0x64 = maximum speed

Byte3 = Tilt Speed in the range 0x00 - 0x64 where 0x00 = stopped and 0x64 = maximum speed

Byte4 = N/A

Returned from peripheral device:

<to:from:0E:ACK:PFByte1…Byte10:checksum:checksum ind>

Where   Byte1 = Pan Speed

Where   Byte2 = Tilt Speed

Where   Byte3 = Tilt Position hundreds   (Ascii char in range 0x30 to 0x39)

Where   Byte4 = Tilt Position tens   (Ascii char in range 0x30 to 0x39)

Where   Byte5 = Tilt Position units   (Ascii char in range 0x30 to 0x39)

Where   Byte6 = Pan Position hundreds   (Ascii char in range 0x30 to 0x39)

Where   Byte7 = Pan Position tens   (Ascii char in range 0x30 to 0x39)

Where   Byte8 = Pan Position units   (Ascii char in range 0x30 to 0x39)

Where   Byte9 = Pan endstops enable   (0x30 = enabled, 0x31 = disabled)

Where   Byte10 = Tilt endstops enable   (0x30 = enabled, 0x31 = disabled)

SM21/3A

Set pan anti-clockwise soft end stop:
Send to peripheral device:
<to:from:03:AW::checksum:checksum ind >

Returned from peripheral device:
<to:from:04:ACK:AW:checksum:checksum ind >

Set pan clockwise soft end stop:
Send to peripheral device:
<to:from:03:CW::checksum:checksum ind >

Returned from peripheral device:
<to:from:04:ACK:CW:checksum:checksum ind >

Set tilt up soft end stop:
Send to peripheral device:
<to:from:03:UT::checksum:checksum ind >

Returned from peripheral device:
<to:from:04:ACK:UT:checksum:checksum ind >

Set tilt down soft end stop:
Send to peripheral device:
<to:from:03:DT::checksum:checksum ind >

Returned from peripheral device:
<to:from:04:ACK:DT:checksum:checksum ind >

Set Pan Speed command:
Sent to peripheral device:
<to:from:04:DS:*:checksum:checksum ind >
Where * = data representing the new pan speed and will be in the range 0x00h to 0x64h where
0x00h = stopped and 0x64h = maximum speed

Returned from peripheral device:
<to:from:04:ACK:DS:checksum:checksum ind >

Set Tilt Speed command:
Sent to peripheral device:
<to:from:04:TA:*:checksum:checksum ind >
Where * = data representing the new pan speed and will be in the range 0x00h to 0x64h where
0x00h = stopped and 0x64h = maximum speed

Returned from peripheral device:
<to:from:04:ACK:TA:checksum:checksum ind >

Use Endstops command:
Sent to peripheral device:
> **< to:from :04:ES:*:checksum:checksum ind>**
>
> where * = 0x30h for don't use end stops and * = 0x31h for use end stops.

Returned from peripheral device:

> **< to:from :05:ACK:ES*:checksum:checksum ind>**
>
> where * = 0x30h for not using end stops and * = 0x31h for using end stops.

Pan and Tilt Status command:
Sent to peripheral device:
> **< to:from :03:AS::checksum:checksum ind>**

Returned from peripheral device:
> **<to:from:0E:ACK:ASByte1…Byte10:checksum:checksum ind >**
> Where Byte1 = Pan Speed
> Where Byte2 = Tilt Speed
> Where Byte3 = Pan Position hundreds (Ascii char in range 0x30 to 0x39)
> Where Byte4 = Pan Position tens (Ascii char in range 0x30 to 0x39)
> Where Byte5 = Pan Position units (Ascii char in range 0x30 to 0x39)
> Where Byte6 = Tilt Position hundreds (Ascii char in range 0x30 to 0x39
> Where Byte7 = Tilt Position tens (Ascii char in range 0x30 to 0x39)
> Where Byte8 = Tilt Position units (Ascii char in range 0x30 to 0x39)
> Where Byte9 = Pan endstops enable (0x30 = enabled, 0x31 = disabled)
> Where Byte10 = Tilt endstops enable (0x30 = enabled, 0x31 = disabled).

Go to Location command:
Sent to peripheral device :
> **<to:from:09:GL:*…*:checksum:checksum ind >**

Returned from peripheral device:
> **<to:from:0A:ACK:GL*…*:checksum:checksum ind >**

> Where *…* = the position to go to in degrees ranging from 000 to the peripherals maximum axis position transmitted as ASCII characters in the range 0x30h to 0x39h
> (e.g. 000 = 0x30h 0x30h 0x30h , 270 = 0x32h 0x37h 0x30h).

> The position to go to will be sent as a group of six ACSII characters with the pan position transmitted first followed by the tilt position
> (e.g. go to pan position 20 deg and tilt position 65 deg would be encoded as:
> > **<to:from:09:GL:020065:checksum:checksum ind >**
> with reply:　　**<to:from:0A:ACK:GL020065:checksum:checksum ind >**)

> Note: If the position requested is within the dead band the device is to reply to the command with the relevant position bytes set to 999.

SM21/3A

Set/Clear/Enquiry Termination command:
Sent to peripheral device:
<to:from:04:TR:*:checksum:checksum ind >

Where　　* = ascii value representing required status of the termination.
　　　　　* = 0x30　equals no termination switched in.
　　　　　* = 0x31　equals termination switched in.
　　　　　* = 0x32　equals termination state enquiry.

Returned from peripheral device**:**
<to:from:05:ACK:TR*:checksum:checksum ind >

Where　　* = ascii value representing required status of the termination.
　　　　　* = 0x30　equals no termination switched in.
　　　　　* = 0x31　equals termination switched in.

Error Diagnosis:
Sent to peripheral device :
<to:from:03:ED::checksum:checksum ind >

Returned from peripheral device:
<to:from:ByteCount:ACK:EDByte1Byte2…Byte_n:checksum:checksum ind >

ByteCount can be a variable value depending on the location of the Error. It is not necessary to send all Error Bytes after the desired error Byte if the remaining Bytes = 0x00.

Where　　Byte1 = Error information from the peripheral.
　　　　　Bit0 = Over Temperature Error
　　　　　Bit1 = Low Oil Level Error
　　　　　Bit2 = Moisture Ingress Error
　　　　　Bit3 = Over Current Error
　　　　　Bit4 = Tilt Stall Error
　　　　　Bit5 = Pan Stall Error
　　　　　Bit6 = Reserved
　　　　　Bit7 = Reserved
　　　　　Byte2 to …Byte_n = Error information from the peripheral.

SM21/3A

ACK command:
Sent/Received by peripheral device :
<to:from:*:ACK:*..*:checksum:checksum ind >

The actual data in the field is dependent on the initial command that was sent to the peripheral as specified above.

NAK command:
Sent by peripheral device :
<to:from:03:NAK:CommandByte1:checksum:checksum ind >

Where Command = Command that was initially sent to the device and caused the NAK reply.

Byte1 = Error code returned by the peripheral. The error code will consist of one(1) byte of data where each bit represents a specific error.

Byte1 = Error Byte.
Bit0 = 1   Error: Device under control of another controller
Bit1   N/A Reads as '0'
Bit2   N/A Reads as '0'
Bit3 = 1   Error: Command not available for device
Bit4 = 1   Error: Command not recognized
Bit5 = 1   Error: Device timed out
Bit6 = 1   Error: User undefined
Bit7 = 1   Error: User undefined

Example:
Command sent to the Device **<to:from:03:FN::checksum:checksum ind >**
NAK reply: **<to:from:05:NAK:FN2:checksum:checksum ind >**

5. DEFAULT/BROADCAST ADDRESS:

There is also a default/broadcast address. All peripherals connected to the network will enact the specified command. The broadcast address is fixed and cannot be changed. This is characterized by the TO section being set to 0xFF with all other sections being as standard for the required command

e.g.**<FF:from:length:command:data_1 data_2 data_3…….data_n:checksum:checksum ind>**

Care MUST be taken when using the broadcast format when operating on a multi-peripheral network to avoid corrupting id settings etc in the peripherals. It is up to the programmer to ensure that care is taken, with relevant warnings, when using the broadcast packet on a network as the peripheral will not conduct any checks out with its usual checks.

2C	17790	03/23
Iss	Mod	Date
Drawing No.		Sheet
10-104-5035		16