Ping200Sr
Interface Control Document

300 Pine Needle Lane
Bigfork, MT 59911

http://www.uavionix.com

support@uavionix.com

Contents

1 Revision History

Revision	Date	Comments
A	11/6/2016	Initial release.
B	12/3/2017	Formatting changes
C	8/22/2018	Update to match User and Installation Guide

2 GDL90

The Ping200SR receives control messages over the Control interface. The interface uses an ASCII-text basis, with an ASCII-encoded hexadecimal checksum. The checksum is algebraic sum of the message byte values. Messages are delimited with a carriage return character.

2.1 Physical Interface

The Control interface uses RS-232 signaling levels. The port is configured for the following characteristics:

- Baud Rate: 57600 baud
- Start Bits: 1
- Data Length: 8
- Stop Bits: 1
- Parity: None
- Flow Control: None
-

2.2 Control Messages

The following table summarizes the Control messages that the Ping200SR receives.

Msg ID	Description	Notes	Ref
^CS	Call Sign	1 min interval or on change	2.2.1
^MD	Operating Mode Message	1 second interval (nominal)	2.2.2

2.2.1 Callsign Message [^CS]

This message provides for a user selectable callsign.

Rate: Every 1 minute or when a change occurs
Message Length: 15 bytes

Byte	Contents	Description
1	'^'	ASCII '^' (0x5E)
2	'C'	ASCII 'C' (0x43)
3	'S'	ASCII 'S' (0x53)
4	' '	ASCII space (0x20)
5-12	dddddddd	ASCII Flight ID (all 8 characters are mandatory, right pad with space)
13-14	dd	Checksum of bytes 1 through 12. In hex ASCII i.e. "FA"
15	'\r'	ASCII carriage return (0x0D)

Example: ^CS UAVIONIX87\r

2.2.2 Mode Message [^MD]

The mode message indicates the current operating mode. It includes the current mode, the Ident status, current squawk code setting and emergency code.

Rate: 1 sec (nominal)
Message Length: 17 bytes

Byte	Contents	Description
1	‘^’	ASCII ‘^’ (0x5E)
2	‘M’	ASCII ‘M’ (0x4D)
3	‘D’	ASCII ‘D’ (0x44)
4	‘ ‘	ASCII space (0x20)
5	m	See mode field table below
6	‘,’	ASCII comma (0x2C)
7	‘I\i’	See Ident field table below
8	‘,’	ASCII comma (0x2C)
9-12	dddd	ASCII squawk code
13	e	See emergency field table below
14	h	Health bit in hex ASCII “1”
15-16	dd	Checksum of bytes 1 through 14. In hex ASCII i.e. “FA”
17	‘\r’	ASCII carriage return (0x0D)

Mode Field

m	Definition	ASCII
O	OFF/STBY	0x4F
A	ON	0x41
C	ALT	0x43
S	ALT	0x53
E	1090ES Only	0x45

Ident Field

i	Definition	ASCII
I	Ident Enabled	0x49
-	Ident is Inactive	0x2D

Emergency Field

e	Definition	ASCII
0	None	0x00
1	General	0x01
2	Medical	0x02
3	Fuel	0x03
4	Com	0x04
5	Hijack	0x05
6	Downed	0x06
7	UAS Lost Link	0x07

The health indication is set to ‘1’ to indicate that everything is operating normally.

Example: ^MD A,I,23540120\r

Mode ON, Ident active, Squawk 2354, No Emergency, Healthy

3 SAGETECH

The Ping200S transponder Control serial interface uses the following message data structure:

Start	Address	Type	ID	Length	Data	Fletcher	Checksum	End
0xA5	0x01	1 Byte	0x00	Byte	0 to 255 Bytes	1 Byte	1 Byte	0x5A

The **Start Byte** precedes all messages and is fixed at 0xA5.

The **Address** is a one byte field that defines the destination. This is always set to 0x01.

The message **Type** is a one byte field that defines the message type.

The message **ID** is set to zero and the message.

The Payload **Length** is a one byte field indicating the number of bytes in the Payload Data field.

The transponder receives data such that the first data bit received is considered the least significant bit of a given Byte. The data format is considered Big Endian as the data bytes themselves within a multi-byte word are received most significant byte first.

The **Fletcher** checksum is a one byte field that holds the 8-bit Fletcher checksum. It is calculated beginning with the start byte through the last byte of the payload (inclusive).

The **Checksum** Byte contains a regular 8-bit arithmetic summation of the message starting with the Start Byte and ending with the last Byte of the Payload Data. The summation is performed assuming all bytes are unsigned, and the result is truncated to the least significant 8 bits.

The Fletcher and Checksum are calculated as follows:

```
uint16_t Fletcher16( uint8_t *data, int payloadLen )
{
    uint8_t sum = 0;
    uint8_t fletcher = 0;
   int index;
   for( index = 0; index <payloadLen+ 5; ++index )
   {
      sum += *data++;
      fletcher+= sum;
   }
   return ((sum <<8) | fletcher);
}
```

3.1 Preflight [0x02]

The preflight data message sets the Flight ID.

Flight ID								Reserved	
00	01	02	03	04	05	06	07	08	09
FD0	FD1	FD2	FD3	FD4	FD5	FD6	FD7	RE0	RE1

Reserved for future use.
Set to zero.

FD0	FD1	FD2	FD3	FD4	FD5	FD6	FD7	Flight ID
0x55	0x41	0x56	0x49	0x4F	0x4E	0x49	0x58	**UAVIONIX**

Flight ID Bytes
Set 64-bit Flight ID. If flight identification is available, this is used in lieu of the aircraft Callsign. Flight ID is used in the flight plan. Otherwise the transponder defaults to the call sign as programmed in the NAV. Up to eight ASCII characters can be entered. Data is sent as unsigned chars and valid ASCII characters are defined below:

Valid ASCII Hex Values

0x20 (Space)
0x30 – 0x39 (0 – 9)
0x41 – 0x5A (A - Z)

The most significant bit is sent first. The ASCII characters are left-justified and the Flight ID may not contain spaces. The Flight ID is padded with space characters on the right.

If Flight ID is not available, fill this field with NULL characters (0x00).

3.2 Operating [0x03]

The Operating Message:

- Sets squawk code (Mode A/4096 Code), altitude data (if desired), transponder mode, power up state and activates the IDENT function.

Squawk Code		Altitude		Mode	Reserved		
0	1	2	3	4	5	6	7
SQU0	SQU1	ALT0	ALT1	MOD0	RES0	RES1	RES2

Reserved for future use.
Set to zero.

SQU0	SQU1	Squawk Code
0x02	0x00	1000
0x02	0x80	1200
0x0e	0x00	7000
...	...	...

Squawk Setting (4096 Code)
[Squawk MSB][Squawk LSB]
Squawk is sent as an unsigned 16 bit Integer. The MSB is sent first. A valid squawk is a string of four 3-bit (octal) numbers. Each 3-bit number can take the value of 0-7 decimal. Valid squawks are 0000 to 7777.
For example, the squawk 1234 would be formatted as:
001 010 011 100
This squawk would be padded with leading zeros and transmitted as"
[0x02][0x9C]
The valid hexadecimal range of squawk is 0x0000 to 0x0FFF. If any of the first four bits are set to '1' the entire message is invalid.

ALT0	ALT1	Altitude
0xFF	0xF4	-1200ft
0x00	0x00	0ft
0x04	0xF3	126,700ft
0x80	0x00	Use Internal Encoder

Altitude
[Altitude MSB][Altitude LSB]
The aircraft barometric altitude is measured in feet with reference to a pressure of 29.9213"Hg for zero feet indication. The MSB is sent first.
The transponder transmits altitude in 100s of feet. Data should be formatted such that 100ft = 0x0001.
The data is a 16 bit signed 2's complement integer, in units of 100s of ft.
Send 0x8000 to configure the transponder to use the FYXnav integrated altitude encoder.
Data values of 0x8001-0xFFF3 and 0x04F4 to 0x7FFF are invalid.

MOD0	Mode
0x00	Mode OFF, IDENT: not Active
0x01	Mode: STBY, IDENT: not Active
0x07	Mode: ALT, IDENT: Active
0x03	Mode: ALT, IDENT: not Active

Mode
Bit 0-1: Transponder Mode

Bit 1	Bit 0	Mode
0	0	OFF
0	1	STBY
1	0	ON
1	1	ALT

Bit 2: IDENT (If enabled: set to 1)
Bit 3-7: Ignored

3.3 Data Request [0x05]

This message is a request for the transponder to send data in a response message. The type of data being requested is specified in the payload of this message, which contains a single byte that specifies the response message type. In response to a valid Data Request Message, the transponder sends an Acknowledge Message immediately followed by a response message.

Requested Messgae Type	Reserved		
00	01	02	03
RMT0	RE0	RE1	RE2

RMT0	Requested Message Type
0x00-0x80	Reserved
0x81	Installation Response
0x82	Preflight Data Response
0x83	Status Response
0x84-0xFF	Reserved

Requested Message Type
[Requested Message Type]
Contains the Message Type of the message being requested.

Reserved
These bytes are reserved for future use. Set to ZERO. All other values are invalid

3.4 GPS Data (Ping200S Only) [0x04]

GPS Data Message: Message type 0x04

The GPS Data Message:

- Provides the transponder with longitude, latitude, ground speed, course over ground, hemisphere, GPS validity data, and time of fix (GPS Time).
- If used, should be sent at regular intervals (between one and five times per second), typically at the nominal update rate of the GPS hardware.
- Requires GPS data (which can be obtained from the NMEA 0183 GPRMC sentence).

Note: If the Navigation Receiver Valid bit (in the Hemisphere byte) is 0, then all data fields in the message are required to be valid, except for Time of Fix (which can be set to all spaces if it is not valid). If the Navigation Receiver Valid bit is 1, all data fields are considered invalid; sending the message with the Navigation Receiver Valid bit set to 1 has the same effect as not sending the message.

GPS Data Message structure:

GPS Longitude (11 bytes)	GPS Latitude (10 bytes)	Speed Over Ground (6 bytes)	Course Over Ground (8 bytes)	Hemisphere (1 bytes)	Time of Fix (GPS Time) (10 bytes)	Reserved (6 Bytes)

3.4.1 GPS Message Payload – GPS Longitude [0x04 bytes 0 – 10]

GPS Longitude										
00	01	02	03	04	05	06	07	08	09	10
LON0	LON1	LON2	LON3	LON4	LON5	LON6	LON7	LON8	LON9	LON10

LON0	LON1	LON2	LON3	LON4	LON5	LON6	LON7	LON8	LON9	LON10	GPS Longitude
0x31	0x32	0x32	0x32	0x39	0x2E	0x37	0x35	0x30	0x30	0x32	122° 19.75002'
0x30	0x35	0x38	0x33	0x33	0x2E	0x39	0x31	0x34	0x38	0x32	58° 33.9142'
0x31	0x32	0x32	0x32	0x30	0x2E	0x39	0x34	0x36	0x30	0x30	122° 20.946'
…	…	…	…	…	…	…	…	…	…	…	

GPS Longitude

Current GPS longitude is sent in the format of degrees, minutes, and fractions of minutes:
dddmm.mmmmm (format of the longitude from GPRMC)
The value is sent as ASCII characters. The longitude in degrees, minutes, and fractions of minutes form is limited to five decimal places. This provides for a resolution up to 0.017 meters.
For example, if the longitude is 58.565247 degress, multiply the number on right of decimal by 60 to get:
58 degress, 33.91482 minutes
Rewritten:
05833.91482 i.e. : dddmm.mmmmm
The packet structure would look like:
[30][35][38][33][33][2E][39][31][34][38][32]

NOTE: The GPS Status Byte contains a bit to declare if the longitude is E or W. The number should not be sent signed

3.4.2 GPS Message Payload – GPS Latitude [0x04 bytes 11 – 20]

GPS Latitude									
11	12	13	14	15	16	17	18	19	20
LAT0	LAT1	LAT2	LAT3	LAT4	LAT5	LAT6	LAT7	LAT8	LAT9

LAT0	LAT1	LAT2	LAT3	LAT4	LAT5	LAT6	LAT7	LAT8	LAT9	GPS Latitude
0x34	0x37	0x33	0x37	0x2E	0x33	0x32	0x34	0x30	0x30	47° 37.224'
0x34	0x39	0x31	0x37	0x2E	0x31	0x31	0x32	0x36	0x36	49° 17.11266'
0x32	0x37	0x35	0x39	0x2E	0x32	0x38	0x33	0x33	0x36	27° 59.28336'
...	...	...	...	...	...	...	...	...	...	

GPS Latitude

Current GPS latitude is sent in the format of degrees, minutes, and fractions of minutes:
dddmm.mmmmm (format of the longitude from GPRMC)

The value is sent as ASCII characters. The latitude in degrees, minutes, and fractions of minutes form is limited to five decimal places. This provides for a resolution up to 0.017 meters.

For example, if the Latitude is 49.285211 Degrees, multiply the number on right of decimal by 60 to get:

49 degrees, 17.11266 Minutes

Rewritten:

4917.11266 i.e. : ddmm.mmmmm

The packet structure would look like:

[34][39][31][37][2E][31][31][32][36][36]

NOTE: The GPS Status Byte contains a bit to declare if the Latitude is N or S. The number should NOT be sent signed.

3.4.3 GPS Message Payload – Speed over Ground and Course over Ground [0x04 bytes 21 – 34]

Speed Over Ground						Course Over Ground							
21	22	23	24	25	26	27	28	29	30	31	32	33	34
SOG0	SOG1	SOG2	SOG3	SOG4	SOG5	CRS0	CRS1	CRS2	CRS3	CRS4	CRS5	CRS6	CRS7

SOG0	SOG1	SOG2	SOG3	SOG4	SOG5	Speed Over Ground
0x31	0x32	0x35	0x2E	0x38	0x30	125.80
0x30	0x36	0x35	0x2E	0x35	0x30	65.50
0x30	0x34	0x38	0x2E	0x33	0x32	48.32
…	…	…	…	…	…	…

Speed Over Ground
Current GPS speed over ground in knots. The data is sent as ASCII characters.
sss.ss (format of data)
For example, if the speed over ground is 62.5 kts, the packet structure would look like:
[30][36][32][2E][35][30]

CRS0	CRS1	CRS2	CRS3	CRS4	CRS5	CRS6	CRS7	Course Over Ground
0x30	0x37	0x37	0x2E	0x35	0x32	0x30	0x30	77.5200
0x31	0x38	0x35	0x2E	0x32	0x30	0x30	0x30	185.2000
0x32	0x37	0x35	0x2E	0x34	0x30	0x30	0x30	275.4000
…	…	…	…	…	…	…	…	…

Course Over Ground
Current GPS course over ground referenced to true north is sent in degrees. The data is sent as ASCII characters. 0 degrees is North, 90 degrees is East, etc.
ccc.cccc (format of data)
For example, if the course over ground is 165.5 degrees, the packet structure would look like:
[31][36][35][2E][35][30][30][30]

3.4.4 GPS Message Payload – Hemisphere and Time of Fix (GPS Time) [0x04 bytes 35 – 51]

Hemisphere	Course Over Ground										Reserved					
35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
GHB0	TOF0	TOF1	TOF2	TOF3	TOF4	TOF5	TOF6	TOF7	TOF8	TOF9	RES0	RES1	RES2	RES3	RES4	RES5

GHB0	Hemisphere
0x00	SW Hemisphere, Data valid
0x01	NW Hemisphere, Data valid
0x82	SE Hemisphere, Data invalid
0x03	NE Hemisphere, Data valid
...	

Hemisphere Byte
The Hemisphere Byte consists of the following information:
Bit 0: N / S Hemisphere indicator. Zero indicates that the latitude is South. One indicates that the latitude is North.

Bit 1: E / W Hemisphere indicator. Zero indicates that the latitude is West. One indicates that the latitude is East.

Bit 2 - 6: Reserved
Bit 7: Navigation receiver status bit. If set to ZERO indicates that the GPS data is valid, if set to ONE GPS data is invalid.

TOF0	TOF1	TOF2	TOF3	TOF4	TOF5	TOF6	TOF7	TOF8	TOF9	Time of Fix (GPS Time)
0x32	0x32	0x33	0x33	0x32	0x33	0x2E	0x30	0x30	0x30	22:33:23.000 UTC
0x31	0x35	0x32	0x34	0x33	0x33	0x2E	0x31	0x31	0x30	15:24:33.110 UTC
0x30	0x38	0x35	0x36	0x30	0x31	0x2E	0x30	0x31	0x30	08:56:01.010 UTC
...	...	...	...	...	...	...	...	...	...	...

Time of Fix
Time of fix in GPS Time.
hhmmss.sss
The value is sent as ASCII characters. The hours, minutes, seconds, and fractions of seconds are sent, indicating the time of fix, relative to midnight UTC.
For example, if the GPS time of fix was 22 hours, 33 minutes, and 23 seconds, the packet structure would look like:
[32][32][33][33][32][33][2E][30][30][30]

If GPS Time of Fix is not available, fill this field with Space characters (0x20).