

## WoW Switch™ (UII) - RS485 / LAN HOST INTERFACE

1	Document intention .....	2
2	Page layouts.....	3
2.1	Page 1 .....	3
2.2	Page 2 .....	3
2.3	Page 3 .....	3
2.4	Page 4 .....	3
2.5	Page 5 .....	4
2.6	Page 6 .....	4
2.7	Page 7 .....	4
2.8	Page 8 .....	4
2.9	Popup page .....	4
3	Commands.....	5
3.1	Ack .....	5
3.2	Nak.....	5
3.3	Panel Brightness.....	6
3.4	Slider Range.....	6
3.5	Sleep.....	6
3.6	Wake.....	6
3.7	Field ON .....	7
3.8	Field OFF .....	7
3.9	Field Pulse .....	7
3.10	Text Write.....	7
3.11	Text Alignment .....	8
3.12	Page Activate .....	8
3.13	Popup Activate .....	8
3.14	Popup Deactivate .....	8
3.15	Bar Graph Write .....	9

10 July 2015

3.16	Special Field Write.....	9
3.16.1	Data format for Special Field 33.....	10
3.16.2	Data format for Special Field 35.....	10
3.16.3	Data format for Special Field 41.....	10
3.16.4	Data format for Special Field 42.....	10
3.17	Touch Event Report.....	11
3.18	Start Up Report.....	12
3.19	Sleep Report.....	12
3.20	Wake Report.....	13
3.21	Temperature.....	13
3.22	Resetting network settings.....	13
3.23	Keep alive.....	13
4	UUI-HOST data transmission.....	13
4.1	Asynchronous Frame Format.....	14
4.2	Transparency.....	14

## 1 Document intention

This document describes the method of communication between the UUI and the HOST. Each UUI manages its own display. The UUI has a series of pre-configured screen layouts 'Pages' and 'Popup Pages'. Pages cannot be edited by the user. Every Page has a series of fields, where each field has one byte address code assigned. The UUI stores the page layouts and the contents for each field for quick redrawing. Each field may contain text or a bar graph.

The HOST sends commands to update the individual fields. The UUI stores the modified contents of the field. If the field is currently being displayed, the screen should be refreshed for the new information to appear.

Popup Pages slide over Pages, normally replacing the bottom two lines of 8px. When hidden the previous page is refreshed.

If the text in any field exceeds the field's size, and cannot be fully displayed, it will be automatically scrolled.

The bar graph is automatically resized to fit the field size.

10 July 2015

## 2 Page layouts

There are 6 pages, and 1 popup page. Each page contains a page number and unique set of fields.

Each field is present by its address, number of symbols it can accept and font size in pixels closed in brackets.

### 2.1 Page 1

Page 1 contains fiend 1

<b>1</b> (32, 32px)
---------------------

### 2.2 Page 2

Page 2 contains fiends 2, 3

<b>2</b> (32,16px)
--------------------

<b>3</b> (32,16px)
--------------------

### 2.3 Page 3

Page 3 contains fiends 4, 5, 6, 7, 8, 9

<b>4</b> (11, 8px)
--------------------

<b>5</b> (21, 8px)
--------------------

<b>6</b> (11, 8px)
--------------------

<b>7</b> (21, 8px)
--------------------

<b>8</b> (11, 8px)
--------------------

<b>9</b> (21, 8px)
--------------------

### 2.4 Page 4

Page 4 contains fields 10, 11, 12

<b>10</b> (32, 16px)
----------------------

<b>11</b> (32, 8px)
---------------------

<b>12</b> (32, 8px)
---------------------

10 July 2015

## 2.5 Page 5

Page 5 contains fields 13,14,15,16

<b>13</b> (32, 8px)
<b>14</b> (32, 8px)
<b>15</b> (32, 8px)
<b>16</b> (32, 8px)

## 2.6 Page 6

Page 6 contains fields 17, 18, 19, 20, 21, 22

<b>17</b> (10, 8px)	<b>18</b> (10, 8px)	
<b>19</b> (3, 8px)	<b>20</b> (20, 8px)	<b>21</b> (3,8px)
<b>22</b> (32, 16px)		

## 2.7 Page 7

Page 7 contains fields 33, 34, 35

<b>33</b> (special field)	<b>34</b> (5,18px)	<b>35</b> (special field)
---------------------------	--------------------	---------------------------

## 2.8 Page 8

Page 8 contains field 36, 37, 38, 39, 40, 41, 42, 42, 44

<b>36</b> (21,8px)								
2	<b>37</b> (4,8px)	6	<b>38</b> (5,8px)	2	4	<b>39</b> (4,8px)	6	<b>40</b> (5,8px)
<b>41</b> (special field)					4	<b>42</b> (9,8px)		6
					4	<b>43</b> (9,8px)		<b>44</b> (sf)

## 2.9 Popup page

The popup page contains fields 23, 24, 25, 26, 27, 28, 29, 30, 31, 32

<b>24</b> (1, 8px)	<b>23</b> (20, 8px)					<b>25</b> (1, 8px)
<b>31</b> (1, 8px)	<b>26</b> (9, 8px)	<b>29</b> (1, 8px)	<b>27</b> (9, 8px)	<b>30</b> (1, 8px)	<b>28</b> (9, 8px)	<b>32</b> (1, 8px)

10 July 2015

## 3 Commands

Communication between UI and HOST are based on a set of commands, described below.

Name	code
Ack	0x0002
Nak	0x0003
PanelBrightness_Cmd	0x0204
SliderRange_Cmd	0x0205
Sleep_Cmd	0x0211
Wake_Cmd	0x0212
Field_ON_Cmd	0x0228
Field_OFF_Cmd	0x0229
Field_Pulse_Cmd	0x0230
Text_Write_Cmd	0x0231
Text_Alignment_Cmd	0x2032
Page_Activate_Cmd	0x0233
PopUp_Activate_Cmd	0x0234
PopUp_Deactivate_Cmd	0x0235
Bar_Write_Cmd	0x0236
Special_Field_Write_Cmd	0x0237
Touch_Event_Rep	0x0221
Startup_Rep	0x0225
Sleep_Rep	0x0226
Wake_Rep	0x0227
Temperature_Read_Cmd	0x0240
Temperature_Biased_Cmd	0x0241
LAN_Reset_Cmd	0x0242
Keep_Alive_Cmd	0x0244

### 3.1 Ack

This is the response to successfully executed command.

The command format is:

data	notes
0x0002	Command code
0xXX	The execution code

### 3.2 Nak

This is the response for unsuccessfully executed command.

The command format is:

data	notes
0x0003	Command code

10 July 2015

### 3.3 Panel Brightness

This command sets the OLED display's brightness in wake and sleep mode to desired value. If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0204	Command code
XX	XX = 0..31; The brightness in wake mode
YY	YY = 0..31; The brightness in sleep mode

### 3.4 Slider Range

This command sets the slider XX to require YYYY degrees to change form min to max value. If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0205	Command code
XX	XXXX = 1..2; The slider ID
YYYY	YYYY = 0..65535 The degrees to change form min to max value (-127 to +127)

### 3.5 Sleep

This command enables the UUI sleep mode.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0211	Command code

### 3.6 Wake

This command disables the UUI sleep mode.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0212	Command code

10 July 2015

### 3.7 Field ON

This command highlights the specified field. If the field is displayed, it is highlighted immediately.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0228	Command code
XX	XX = 1..32; The field number (ID)

### 3.8 Field OFF

This command sets the specified field to it's normal color. If the field is displayed, the color is changed immediately.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0229	Command code
XX	XX = 1..32; The field number (ID)

### 3.9 Field Pulse

This command highlights the specified field for a predefined time interval.

The command affects displayed fields only.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0230	Command code
XX	XX = 1..32; The field number (ID)

### 3.10 Text Write

This command writes a text to the specified field. If the field is displayed, the new text is displayed immediately; else the text is stored in field's buffer. If the text length exceeds the buffer size, the text is truncated.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0231	Command code
XX	XX = 1..32; The field number
Text string	Text string to be stored in field's buffer. (non-unicode text)

10 July 2015

### 3.11 Text Alignment

This command sets the text alignment of the specified field. If the field is displayed, the new alignment is executed immediately, else the alignment is stored in field.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x2032	Command code
XX	XX = 1..32; The field number
XX	XX = 1, 4 or 7 - Left XX = 2, 5 or 8 - Center XX = 3, 6 or 9 - Right

### 3.12 Page Activate

This command activates and displays the specified page.

If the command is successfully executed, UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0233	Command code
XX	XX = 1..6; The page number (ID)

### 3.13 Popup Activate

This command activates and displays the specified popup page.

If the command is successfully executed, UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0234	Command code
XX	XX = 1; The popup page number (ID)

### 3.14 Popup Deactivate

This command deactivates the specified popup page, and displays the previous page.

If the command is successfully executed, UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0235	Command code
XX	XX = 1; The popup page number (ID)



10 July 2015

### 3.15 Bar Graph Write

This command writes the bar graph in the specified field. If the field is displayed, the bar graph is displayed immediately, else the bar graph is stored in the field. The bar graph is deactivated, by writing a new text in this field.

There are two types of bar graph. The first type fills the field to the specified percentage, and second puts a vertical line in the specified position(0 - put the line in the left end of the field, 100 put the line in the right end of the field).

The first type has 1 in most significant bit in the "Bar graph type and value byte" (value = 128 + bar graph size), and second type has 0 in most significant bit in the "Bar graph type and value byte"(value = bar graph position).

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0236	Command code
XX	XX = 1..32; The field number
XX	XX = 128 + 0..100 for the first type bar graph XX = 0..128 for the second type bar graph The bar graph type and value

### 3.16 Special Field Write

This command writes data in the specified special field. If the field is displayed, the field's data are processed and displayed immediately by special field display function, else the data are stored in the field.

If the command is successfully executed UUI returns Ack, else returns Nak.

The command format is:

data	notes
0x0237	Command code
XX	XX the field number
XX .. XX	XX the data bytes as many, as required by the specified field

10 July 2015

### 3.16.1 Data format for Special Field 33

The data format is:

data	notes
XX	XX = 0...20 XX = 0 - clear the special field XX = 1..20 - number of displayed picture
XX	Bit 0 = 0 - do not display error marker Bit 0 = 1 - display error marker

### 3.16.2 Data format for Special Field 35

The data format is:

data	notes
XX	XX = 0...7 XX = 0 - clear the special field XX = 1..7 - flow level
XX	Bit 0 = 0 - show empty flow picture Bit 0 = 1 - show full flow picture Bit 1 = 0 - do not display "ECO" Bit 1 = 1 - display "ECO"

### 3.16.3 Data format for Special Field 41

The data format is:

data	notes
XX	XX = 0...4 XX = 0 - clear the special field XX = 1..4- number of displayed picture
XX	

### 3.16.4 Data format for Special Field 42

The data format is:

data	notes
XX	XX = 0...1 XX = 0 - clear the special field XX = 1 - display picture
XX	

10 July 2015

## 3.17 Touch Event Report

Touch event report gives to HOST information containing the states and changes of all defined buttons, round sliders and linear slider. The information is present by structure Event\_QT\_Send, described below.

```
typedef struct _Event_QT_Send //Touch event parameters for sending to host
{
    char Pressed_Buttons[4];           // All pressed buttons
    char Changed_Buttons[4];          // All changed buttons
    signed char Slider_1_Positon_Change; //slider 1 change -128 +128
    signed char Slider_2_Positon_Change; //slider 2 change -128 +128
    signed char Slider_3_Positon_Change; //slider 3 change -128 +128
    char Slider_L_X_Position_Change[2]; //linear slider X coord. change -255
+255
    char Slider_L_Y_Position_Change[2]; //Linear slider Y coord. change -127
+127
} Event_QT_Send_t;
```

The first sent byte Pressed\_Buttons[3] contains the state of buttons 31 – 24 .

The second sent byte Pressed\_Buttons[2] contains the state of buttons 23 – 16.

The fourth sent byte Pressed\_Buttons[1] contains the state of buttons 15 – 8.

The fifth sent byte Pressed\_Buttons[0] contains the state of buttons 7 – 0.

Each bit in bytes represents the state of the corresponded button. If the bit is 1, the button is pressed,

If the bit is 0, the button is released, The bit 7 of Pressed\_Buttons[3] represents the button 31 and bit 0 of Pressed\_Buttons[0] represent the button 0;

The first sent byte Changed\_Buttons[3] contains change of buttons 31 – 24 .

The second sent byte Changed\_Buttons[2] contains the change of buttons 23 – 16.

The fourth sent byte Changed\_Buttons[1] contains the change of buttons 15 – 8.

The fifth sent byte Changed\_Buttons[0] contains the change of buttons 7 – 0.

Each bit in bytes represents the change of the corresponded button. If the bit is 1, the button is changed,

If the bit is 0, the button is not changed, The bit 7 of Pressed\_Buttons[3] represents button 31 and bit 0 of Pressed\_Buttons[0] represent s the button 0.

Slider\_1\_Positon\_Change contains the position change of round slider 1 from -127 to +127

Slider\_2\_Positon\_Change contains the position change of round slider 1 from -127 to +127

Slider\_3\_Positon\_Change contains the position change of round slider 1 from -127 to +127

The first sent byte Slider\_L\_X\_Position\_Change[1] contains the MS byte of the Linear slider's X coordinate change.

The second sent byte Slider\_L\_X\_Position\_Change[0] contains the LS byte of the Linear slider's X coordinate change .

The coordinate change is in range -255 +255.

10 July 2015

The first sent byte Slider\_L\_Y\_Position\_Change[1] contains the MS byte of the Linear slider's Y coordinate change.

The second sent byte Slider\_L\_Y\_Position\_Change[0] contains the LS byte of the Linear slider's Y coordinate change.

The coordinate change is in range -255 +255.

The report format is:

data	notes
0x0221	Command code
Pressed_Buttons [3]	States of the buttons 31..24
Pressed_Buttons [2]	States of the buttons 23..16
Pressed_Buttons [1]	States of the buttons 15..8
Pressed_Buttons [0]	States of the buttons 7..0
Changed_Buttons [3]	Changes of the buttons 31..24
Changed_Buttons [2]	Changes of the buttons 23..16
Changed_Buttons [1]	Changes of the buttons 15..8
Changed_Buttons [0]	Changes of the buttons 7..0
Slider_1_Positon_Change	-127 to +127 Change of position of Round slider 1
Slider_2_Positon_Change	-127 to +127 Change of position of Round slider 2
Slider_3_Positon_Change	-127 to +127 Change of position of Round slider 3
Slider_L_X_Position_Change	-255 to +255 Change of X position of Linear slider
Slider_L_Y_Position_Change	-255 to +255 Change of Y position of Linear slider

### 3.18 Start Up Report

This report is send to HOST, after UUI is powered up and ready to communicate with HOST.

The report format is:

data	notes
0x0225	Command code

### 3.19 Sleep Report

This report is send to HOST, just before UUI is set in low power (sleep) mode.

The report format is:

data	notes
0x0226	Command code

10 July 2015

### 3.20 Wake Report

This report is send to HOST, when UUI is set in wake mode after detecting touch or receiving command. The report format is:

data	notes
0x0227	Command code

### 3.21 Temperature

Temperature reading can be obtained from units fitted with temperature sensor

The report format is:

data	notes
0x0240	Temperature Read Command Code
0x0241	Temperature Biased Command Code

### 3.22 Resetting network settings

This command reset the network settings to default. The report format is:

data	notes
0x0242	LAN Reset Command Code

### 3.23 Keep alive

On receiving keep alive command from HOST UUI returns the same indicating it is connected and ready. The report format is:

data	notes
0x0244	Keep Alive Command Code

## 4 UUI-HOST data transmission

The used data frame format is:

data	notes
Source address	When receiving, this is the HOST address (0) When transmitting, this is the UUI address (1..255)
Length	The length of the command
Destination address	When receiving, this is the UUI address (1..255) When transmitting, this is the HOST address(0)
Command	The command data (payload)
Checksum	The frame checksum

The Checksum includes all data in the packet added as signed byte and discarding the sign bit overflow and then negating the byte to get the checksum.

As the data exchanged between UUI and HOST are in binary format, the frame delimitation of KISS protocol is used, as described below.

10 July 2015

## 4.1 Asynchronous Frame Format

The "asynchronous packet protocol" spoken between the host and TNC is very simple, since its only function is to delimit frames. Each frame is both preceded and followed by a special FEND (Frame End) character, analogous to an HDLC flag. No CRC or checksum is provided. In addition, no RS-232C handshaking signals are employed.

The special characters are:

Abbreviation	Description	Hex value
FEND	Frame End	C0
FESC	Frame Escape	DB
TFEND	Transposed Frame End	DC
TFESC	Transposed Frame Escape	DD

The reason for both preceding and ending frames with FENDs is to improve performance when there is noise on the asynch line. The FEND at the beginning of a frame serves to "flush out" any accumulated garbage into a separate frame (which will be discarded by the upper layer protocol) instead of sticking it on the front of an otherwise good frame. As with back-to-back flags in HDLC, two FEND characters in a row should not be interpreted as delimiting an empty frame.

## 4.2 Transparency

Frames are sent in 8-bit binary; the asynchronous link is set to 8 data bits, 1 stop bit, and no parity. If a FEND ever appears in the data, it is translated into the two byte sequence FESC TFEND (Frame Escape, Transposed Frame End). Likewise, if the FESC character ever appears in the user data, it is replaced with the two character sequence FESC TFESC (Frame Escape, Transposed Frame Escape).

As characters arrive at the receiver, they are appended to a buffer containing the current frame. Receiving a FEND marks the end of the current frame. Receipt of a FESC puts the receiver into "escaped mode", causing the receiver to translate a following TFESC or TFEND back to FESC or FEND, respectively, before adding it to the receive buffer and leaving escaped mode. Receipt of any character other than TFESC or TFEND while in escaped mode is an error; no action is taken and frame assembly continues. A TFEND or TESC received while not in escaped mode is treated as an ordinary data character.

This procedure may seem somewhat complicated, but it is easy to implement and recovers quickly from errors. In particular, the FEND character is never sent over the channel except as an actual end-of-frame indication. This ensures that any intact frame (properly delimited by FEND characters) will always be received properly regardless of the starting state of the receiver or corruption of the preceding frame.

This asynchronous framing protocol is identical to "SLIP" (Serial Line IP), a popular method for sending ARPA IP datagrams across asynchronous links. It could also form the basis of an asynchronous amateur packet radio link protocol that avoids the complexity of HDLC on slow speed channels.

To communicate over TCP/IP please connect to TCP port 21826 and use the same protocol to send and receive messages.

To read temperature connect to port 22826

To communicate to multiple UUI units on RS485 via the main please connect to port 23826