

DVDO

DVDO-Xtend-TX100

3 x 1 HDBT Wall-plate Transmitter

API Command Set

Version: V1.0.0



RS232 Default Setting

| Parameters | Value |
|--------------|------------|
| Baud Rate | 115200 bps |
| Data bits | 8 bits |
| Parity | None |
| Stop bits | 1 bit |
| Flow control | None |

Command

Take Command **SET SW in out<CR><LF>** as an example:

1. **[SET SW]** denotes command key words, case insensitive.
2. **[in out]** denotes parameters, case insensitive; incorrect parameters number will not be recognized.
3. **<CR><LF>** denotes a carriage return or a line feed; all commands must be ended up with a carriage return or a line feed.

| IDX | Function | Command | Example |
|-----|-------------------------------|---|--|
| 1 | Get UART BAUDRA TE | Command: GET UART_B<CR><LF> Return: UART_B <CR><LF> Description: Get UART BAUDRATE, default value is 115200. | Command: GET UART_B <CR><LF> Return: UART_B 9600<CR><LF> Description: Get UART baudrate 9600 |
| 2 | Set UART BAUDRA TE | Command: SET UART B <i>prm</i> <CR><LF> Return: UART B <i>prm</i> <CR><LF> Parameter: <i>prm</i> = {9600, 19200, 38400, 57600, 115200} Description: Set UART BAUDRATE, default value is 115200. | Command: SET UART B 9600<CR><LF> Return: UART B 9600<CR><LF> Description: Set UART baudrate to 9600 |
| 3 | Reset UART BAUDRA TE | Command: RESET UART_B<CR><LF> Return: UART_B <i>prm</i> <CR><LF> Parameter: <i>prm</i> = {115200} Description: Reset UART BAUDRATE | Command: RESET UART_B <CR><LF> Return: UART_B 115200<CR><LF> Description: Reset UART baudrate to 115200 |

| IDX | Function | Command | Example |
|-----|------------------------|--|---|
| 4 | Get UART ENDCHAR | <p>Command: GET UART_E<CR><LF></p> <p>Return: UART_E <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {null, ,cr,lf,crlf} cr: carriage return, ascii code is 0x0D. lf: line feed, ascii code is 0x0A.</p> <p>Description: Get UART ENDCHAR, default value is crlf.</p> | <p>Command: GET UART E <CR><LF></p> <p>Return: UART_E <i>cr</i><CR><LF></p> <p>Description: UART end Char is cr</p> |
| 5 | Set UART ENDCHAR | <p>Command: SET UART_E <i>prm</i><CR><LF></p> <p>Return: UART_E <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {null, ,cr,lf,crlf} cr: carriage return, ascii code is 0x0D. lf: line feed, ascii code is 0x0A.</p> <p>Description: Set UART ENDCHAR, default value is crlf.</p> | <p>Command: SET UART_E <i>cr</i><CR><LF></p> <p>Return: UART_E <i>cr</i><CR><LF></p> <p>Description: Set UART end Char to cr</p> |

| IDX | Function | Command | Example |
|-----|--------------------------------------|--|---|
| 6 | Reset UART ENDCHAR | <p>Command: RESET UART_E<CR><LF></p> <p>Return: UART_E <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {crlf} cr: carriage return, ascii code is 0x0D. lf: line feed, ascii code is 0x0A.</p> <p>Description: Reset UART ENDCHAR, default value is crlf.</p> | <p>Command: RESET UART E <CR><LF></p> <p>Return: UART_E <i>crlf</i><CR><LF></p> <p>Description: Reset UART end Char to crlf</p> |
| 7 | character uart command edit | <p>Command: SET UART_STR <i>prm1 prm2 prm3</i><CR><LF></p> <p>Return: UART_STR <i>prm1 prm2 prm3</i><CR><LF></p> <p>Parameter: <i>prm1</i> = {poweron, poweroff} //PRM1 is the standard command, newest cmd. <i>prm2</i> = {1,2,3,4,5} //PRM2 is the index of inserting command. <i>prm3</i> = {xxx} xxxxx} //PRM3 is the original command according device guidelines, less than 64 characters.</p> <p>Description: Character uart command edit</p> | <p>Command: SET UART_STR <i>poweron 1</i> <i>xxxx</i><CR><LF></p> <p>Return: UART_STR <i>poweron 1</i> <i>xxxx</i><CR><LF></p> <p>Description: Edit UART command</p> |

| IDX | Function | Command | Example |
|-----|---------------------------------------|---|---|
| 8 | character uart command get | <p>Command: GET UART_STR <i>prm1</i> <i>prm2</i><CR><LF></p> <p>Return: UART_STR <i>prm1</i> <i>prm2</i><CR><LF></p> <p>Parameter: <i>prm1</i> = {poweron, poweroff}//PRM1 is the standard command, newest cmd. <i>prm2</i> = {1,2,3,4,5,all}//PRM2 is the index of inserting command.</p> <p>Description: Get character uart command</p> | <p>Command: GET UART STR <i>poweron 1</i> <CR><LF></p> <p>Return: UART_STR <i>poweron 1</i> xxxx <CR><LF></p> <p>Description: Get UART command</p> |
| 9 | character uart command RESET | <p>Command: RESET UART_STR <i>prm1</i> <i>prm2</i><CR><LF></p> <p>Return: UART_STR <i>prm1</i> <i>prm2</i><CR><LF></p> <p>Parameter: <i>prm1</i> = {poweron, poweroff}//PRM1 is the standard command, newest cmd. <i>prm2</i> = {1,2,3,4,5,all}//PRM2 is the index of inserting command.</p> <p>Description: Character uart command RESET</p> | <p>Command: RESET UART_STR <i>poweron 1</i> <CR><LF></p> <p>Return: UART_STR <i>poweron 1</i> <CR><LF></p> <p>Description: RESET UART command</p> |

| IDX | Function | Command | Example |
|-----|-----------------------|--|---|
| 10 | Hex uart command edit | <p>Command: SET UART_HEX <i>prm1 prm2 hex1 hex2 hex3</i> <CR><LF></p> <p>Return: UART_HEX <i>prm1 prm2 hex1 hex2 hex3</i> <CR><LF></p> <p>Parameter: <i>prm1</i> = {poweron, poweroff} // PRM1 is the standard command, newest cmd. <i>prm2</i> = {1,2,3,4,5} // PRM2 is the index of inserting command. <i>Hex1, hex2</i> = {xx xx xx xx} // hex1, hex2..., is ascii string of hex value. For, string "12", convert to correct format string is "31 32", less than 64 characters.</p> <p>Description: Hex uart command edit</p> | <p>Command: SET UART HEX <i>poweron 1 12 34 56</i> <CR><LF></p> <p>Return: UART_HEX <i>poweron 1 12 34 56</i> <CR><LF></p> <p>Description: Edit UART command with HEX format</p> |
| 11 | Set Long Reach on/off | <p>Command: SET LR_FN <i>prm1</i> <CR><LF></p> <p>Return: LR_FN <i>prm1</i> <CR><LF></p> <p>Parameter: <i>prm1</i> = {on, off};</p> <p>Description: Set long reach on/off, default value is off.</p> | <p>Command: SET LR_FN <i>on</i> <CR><LF></p> <p>Return: LR_FN <i>on</i> <CR><LF></p> <p>Description: Set all hdbt output enable long reach mode</p> |

| IDX | Function | Command | Example |
|-----|-------------------------|--|---|
| 12 | Reset Long Reach on/off | Command: RESET LR_FN<CR><LF> Return: LR_FN <i>prm1</i> <CR><LF> Parameter: <i>prm1</i> = {off}; Description: Reset long reach off, default value is off. | Command: RESET LR_FN<CR><LF> Return: LR_FN <i>off</i> <CR><LF> Description: Reset all hdbt output enable long reach mode status |
| 13 | Get Long Reach on/off | Command: GET LR_FN<CR><LF> Return: LR_FN <i>prm1</i> <CR><LF> Parameter: <i>prm1</i> = {on, off}; Description: Get long reach on/off, default value is off. | Command: GET LR_FN<CR><LF> Return: LR_FN <i>on</i> <CR><LF> Description: Get all hdbt output enable long reach mode status |
| 14 | Factory reset | Command: RESET<CR><LF> Return: RESET<CR><LF> Description: Factory reset | Command: RESET<CR><LF> Return: RESET<CR><LF> Description: Factory reset |

| IDX | Function | Command | Example |
|-----|----------------------|--|---|
| 15 | Get UART CMD | <p>Command: GET UART_CMD<CR><LF></p> <p>Return: <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {SET UART_B <i>prm</i>,GET UART_B,...}</p> <p>Description: Get all RS232 command</p> | <p>Command: GET UARTCMD <CR><LF></p> <p>Return: SET UART_B <i>prm</i> <CR><LF> GET UART B <CR><LF> RESET UART_B <CR><LF> SET UART_E <i>prm</i> <CR><LF> GET UART E <CR><LF> RESET UART_E <CR><LF> ...</p> <p>Description: Get UART CMD</p> |
| 16 | Get software version | <p>Command: GET SW_VERSION<CR><LF></p> <p>Return: <i>prm1</i><CR></p> <p>Parameter: <i>prm1</i> = software version info</p> <p>Description: Get software version</p> | <p>Command: GET SW_VERSION <CR><LF></p> <p>Return: EX0301_N002_E10 V1.0</p> <p>Description: Get software version</p> |

| IDX | Function | Command | Example |
|-----|--------------------------|--|--|
| 17 | Upgrade module | Command: UPG [prm]<CR><LF> Return: UPG [prm]<CR><LF> Parameter: prm= {xxx} Description: Upgrade module | Command: UPG<CR><LF> Return: UPG<CR><LF> Description: Upgrade module |
| 18 | Set CEC POWER Delay Time | Command: SET AUTOCEC_D prm<CR><LF> Return: AUTOCEC_D SET prm<CR><LF> Parameter: prm = {3,4,5,...} // according to the actual time counter, 3 means 3 minute , 4 means 4 minutes, Default wait time is 3 minutes, Max wait time is 60 minutes. Description: AUTOCEC_D is short for CEC auto Power Delay Timing | Command: SET AUTOCEC_D 3 <CR><LF> Return: AUTOCEC_D SET 3<CR><LF> Description: When no active signal outputs, 3 minutes later, the unit will auto power off. |

| IDX | Function | Command | Example |
|-----|---------------------------------|--|---|
| 19 | Get CEC POWER Delay Time Status | <p>Command: GET AUTOCEC_D<CR><LF></p> <p>Return: AUTOCEC_D GET <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {3,4,5,...} // according to the actual time counter, 3 means 3 minute, 4 means 4 minutes, Default wait time is 3 minutes, Max wait time is 60 minutes.</p> <p>Description: AUTOCEC_D is short for CEC auto Power Delay Timing</p> | <p>Command: GET AUTOCEC D <CR><LF></p> <p>Return: AUTOCEC_D GET 3<CR><LF></p> <p>Description: Get output auto power delay time, the result is 3 minutes</p> |
| 20 | Reset CEC POWER Delay Time | <p>Command: RESET AUTOCEC_D<CR><LF></p> <p>Return: AUTOCEC_D RESET <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {3,} // according to the actual time counter, 3 means 3 minute, 4 means 4 minutes, Default wait time is 3 minutes, Max wait time is 60 minutes.</p> <p>Description: AUTOCEC_D is short for CEC auto Power Delay Timing</p> | <p>Command: RESET AUTOCEC_D <CR><LF></p> <p>Return: AUTOCEC_D RESET 3 <CR><LF></p> <p>Description: When no active signal to output, 3 minutes later, the unit will auto power off.</p> |

| IDX | Function | Command | Example |
|-----|---------------------|---|---|
| 21 | Set Auto CEC mode | <p>Command: SET AUTOCEC_M <i>prm</i><CR><LF></p> <p>Return: AUTOCEC_M <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {on,off} //set the auto cec mode, on/off</p> <p>Description: AUTOCEC_M is short for CEC auto mode; default value is on.</p> | <p>Command: SET AUTOCEC M off<CR><LF></p> <p>Return: AUTOCEC_M off<CR><LF></p> <p>Description: Set the auto CEC mode to off.</p> |
| 22 | Get Auto CEC on/off | <p>Command: GET AUTOCEC_M<CR><LF></p> <p>Return: AUTOCEC_M <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {on,off} //get the auto cec mode, on/off;</p> <p>Description: AUTOCEC_M is short for CEC auto mode, default value is on.</p> | <p>Command: GET AUTOCEC_M <CR><LF></p> <p>Return: AUTOCEC_M off<CR><LF></p> <p>Description: The auto CEC mode is off.</p> |

| IDX | Function | Command | Example |
|-----|---------------------------|---|--|
| 23 | Reset Auto CEC on/off | <p>Command: RESET AUTOCEC_M<CR><LF></p> <p>Return: AUTOCEC_M <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {on} // reset the auto cec mode, on;</p> <p>Description: AUTOCEC_M is short for CEC auto mode, default value is on.</p> | <p>Command: RESET AUTOCEC M <CR><LF></p> <p>Return: AUTOCEC_M on<CR><LF></p> <p>Description: The auto CEC mode is on.</p> |
| 24 | Set UART POWER Delay Time | <p>Command: SET AUTOUART_D <i>prm</i><CR><LF></p> <p>Return: AUTOUART_D SET <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {3,4,5,...} // according to the actual time counter, 3 means 3 minute, 4 means 4 minutes, Default wait time is 3 minutes, Max wait time is 60 minutes.</p> <p>Description: AUTOUART_D is short for UART auto Power Delay Timing</p> | <p>Command: SET AUTOUART_D 3 <CR><LF></p> <p>Return: AUTOUART_D SET 3<CR><LF></p> <p>Description: When no active signal to output, 3 minutes later, the unit will auto power off.</p> |

| IDX | Function | Command | Example |
|-----|--|---|---|
| 25 | Get UART POWER Delay Time Status Status. | <p>Command: GET AUTOUART_D<CR><LF></p> <p>Return: AUTOUART_D GET <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {3,4,5,...} // according to the actual time counter, 3 means 3 minute , 4 means 4 minutes, Default wait time is 3 minutes, Max wait time is 60 minutes.</p> <p>Description: AUTOUART_D is short for UART auto Power Delay Timing</p> | <p>Command: GET AUTOUART D <CR><LF></p> <p>Return: AUTOUART_D GET 3<CR><LF></p> <p>Description: Get output auto power delay time, the result is 3 minutes</p> |
| 26 | Reset UART POWER | <p>Command: RESET AUTOUART_D<CR><LF></p> <p>Return: AUTOUART_D RESET <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {3,} // according to the actual time counter, 3 means 3 minute, 4 means 4 minutes, Default wait time is 3 minutes, Max wait time is 60 minutes.</p> <p>Description: AUTOUART_D is short for UART auto Power Delay Timing</p> | <p>Command: RESET AUTOUART_D <CR><LF></p> <p>Return: AUTOUART_D RESET 3 <CR><LF></p> <p>Description: when no active signal to output, 3 minutes later, the unit will auto power off.</p> |

| IDX | Function | Command | Example |
|-----|----------------------|--|---|
| 27 | Set Auto UART mode | <p>Command: SET AUTOUART_M <i>prm</i><CR><LF></p> <p>Return: AUTOUART_M <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {on, off} // set the auto UART mode, on/off</p> <p>Description: AUTOUART_M is short for UART auto mode, default value is on.</p> | <p>Command: SET AUTOUART M off<CR><LF></p> <p>Return: AUTOUART_M off <CR><LF></p> <p>Description: Set the auto UART mode off.</p> |
| 28 | Get Auto UART on/off | <p>Command: GET AUTOUART_M<CR><LF></p> <p>Return: AUTOUART_M <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {on, off} // get the auto UART mode, on/off;</p> <p>Description: AUTOUART_M is short for UART auto mode, default value is on.</p> | <p>Command: GET AUTOUART_M <CR><LF></p> <p>Return: AUTOUART_M off<CR><LF></p> <p>Description: The auto UART mode is off.</p> |

| IDX | Function | Command | Example |
|-----|---------------------------------|---|--|
| 29 | Reset Auto UART on/off | <p>Command: RESET AUTOUART_M<CR><LF></p> <p>Return: AUTOUART_M <i>prm</i><CR><LF></p> <p>Parameter: <i>prm</i> = {on} //reset the autoUART mode, on;</p> <p>Description: AUTOUART_M is short for UART auto mode, default value is on.</p> | <p>Command: RESET AUTOUART M <CR><LF></p> <p>Return: AUTOUART_M on <CR><LF></p> <p>Description: The auto UART mode is on.</p> |
| 30 | Switch Input for Output | <p>Command: SET SW <i>in</i><CR><LF></p> <p>Return: SW <i>in</i><CR><LF></p> <p>Parameter: <i>in</i> = {hdmi1, hdmi2, hdmi3};</p> <p>Description: SW is short for switch Switch one input source for the output sink</p> | <p>Command: SET SW <i>hdmi</i> 1<CR><LF></p> <p>Return: SW <i>hdmi</i> 1<CR><LF></p> <p>Description: Switch hdmi1 for the output sink</p> |
| 31 | Get active input source channel | <p>Command: GET SW<CR><LF></p> <p>Return: <i>in</i><CR><LF></p> <p>Parameter: <i>in</i> = {hdmi 1, hdmi 2, hdmi 3};</p> <p>Description: SW is short for Switch get active input source channel</p> | <p>Command: GET SW<CR><LF></p> <p>Return: hdmi 1<CR><LF></p> <p>Description: Get the input source channel</p> |

| IDX | Function | Command | Example |
|-----|-----------------------------------|--|--|
| 32 | Reset active input source channel | Command: RESET SW<CR><LF> Return: in<CR><LF> Parameter: in = {hdmi 1}; Description: SW is short for Switch Reset active input source channel | Command: RESET SW<CR><LF> Return: hdmi 1<CR><LF> Description: Reset the input source channel |

DVDO
