# XHC xHB03 & xHB04 INTERNALS

| Characteristics | | | |
|---|---|---|---|
| **MODEL** | **DESCRIPTION** | **DRIVER** | **INITIALLY FOR** |
| WHB03 LHB03 | wireless mpg wired mpg | HID DEVICE VID: 0x10CE PID: 0xEB6E | NC Studio |
| WHB04 LHB04 | wireless mpg wired mpg | HID DEVICE VID: 0x10CE PID: 0xEB70 | Mach 3 |

**Power saving:**

If you use wireless model of this device you need to know what each device will go sleep mode after 30 seconds if no button was pressed.

**HID structure:**

Device has two reports IDs one of it ( Report ID: 04h ) used as expected through writing to USB Endpoint 81h, another one ( Report ID: 06h ) used to HOST->DEVICE communication, but we do not have OUT endpoint in our USB Device Descriptor for it, it was first tricky place, device used HID SET_REPORT request and transfer data via 07h bytes chunks. Take a closer look for these reports.

| Report ID: 0x04 DEVICE->HOST ( 6 bytes ) | | | |
|---|---|---|---|
| **offset** | **size ( bytes )** | **name** | **description** |
| 00h | 1 | ID | report id ( 04h ) |
| 01h | 1 | BUTTON 1 | button key code |
| 02h | 1 | BUTTON 2 | button key code |
| 03h | 1 | WHEEL MODE | axes for wheel and etc |
| 04h | 1 | WHEEL | signed value of wheel speed, sign - direction |
| 05h | 1 | XORED DAY | current day XORed with button 1 key code |

**ID** - it is our report ID according to HID report information.
**BUTTON 1,2** – if we push da button we see our key codes in this fields. We can push TWO buttons simultaneously.

| Key codes | | |
|---|---|---|
| **CODE** | **HB03** | **HB04** |
| 17h | Yes | Reset |
| 16h | No | Stop |
| 01h | Stop | Goto zero |
| 02h | Start/Pause | Start/Pause |

| 03h | Rewind(?) | Rewind |
|------|-----------|--------|
| 04h | Probe-Z | Probe-Z |
| 0Ch | Goto zero | Spindle |
| 06h | X/2= | 1/2= |
| 07h | Y/2= | 0= |
| 08h | Safe-Z | Safe-Z |
| 09h | X=0 | Goto home |
| 0Ah | Y=0 | Macro 1 |
| 0Bh | Z=0 | Macro 2 |
| 05h | Goto home | Macro 3 |
| 0Dh | Step left | STEP++ |
| 0Eh | Step right | MPG/MODE |
| 0Fh | Fine xyz | Macro 6 |
| 10h | Spindle | Macro 7 |

| WHEEL MODE | | |
|------------|---------|---------|
| value | HB03 | HB04 |
| 00h | off state | off state |
| 11h | X direction for hand-wheel | X direction for hand-wheel |
| 12h | Y direction for hand-wheel | Y direction for hand-wheel |
| 13h | Z direction for hand-wheel | Z direction for hand-wheel |
| 15h | adjust Spindle speed | A direction for hand-wheel |
| 14h | adjust Feedrate speed | adjust Spindle speed |
| 18h | adjust Processing speed | adjust Feedrate speed |

**WHEEL** – it is direction and speed of wheel, when we spin the wheel to CW dir it will be positive from 1 to 10 ( this value based on current spin speed ) and when spin wheel to CCW value be negative from -1 to -10 ( based on spin speed too )

**XORED DAY -** very interesting value, it is a crypted DAY OF THE MONTH  this value settable with HID Set Feature and used as key for simple XOR crypt what used for very easy protection of they protocol and solved easily (XORED DAY) XOR (BUTTON 1).

| Report ID: 0x06 HOST->DEVICE ( 8 bytes) | | | |
|--------|----------------|---------------|------------------------------|
| offset | size ( bytes ) | name | description |
| 00h | 1 | ID | report id ( 06h ) |
| 01h | 7 | PAYLOAD CHUNK | chunk of output data report |

**ID** -   it is our report ID according to HID report information.

**PAYLOAD CHUNK** – it is chunk of our payload, if we need to send data not aligned to 7 bytes we need to pad it for 7 bytes.

**Payloads data:**

| HB03 ( 31 bytes ) | | | |
|---|---|---|---|
| offset | size ( bytes ) | name | description |
| 00h | 2 | MAGIC | FDFEh used to detect start sequences |
| 02h | 1 | DAY | day of the month |
| 03h | 2 | WC_X_INT | X workspace position integer |
| 05h | 1 | WC_X_FRAC | X workspace position fractional |
| 06h | 2 | WC_Y_INT | Y workspace position integer |
| 08h | 1 | WC_Y_FRAC | Y workspace position fractional |
| 09h | 2 | WC_Z_INT | Z workspace position integer |
| 0Bh | 1 | WC_Z_FRAC | Z workspace position fractional |
| 0Ch | 2 | MC_X_INT | X machine position integer |
| 0Eh | 1 | MC_X_FRAC | X machine position fractional |
| 0Fh | 2 | MC_Y_INT | Y machine position integer |
| 11h | 1 | MC_Y_FRAC | Y machine position fractional |
| 12h | 2 | MC_Z_INT | Z machine position integer |
| 14h | 1 | MC_Z_FRAC | Z machine position fractional |
| 15h | 2 | FEEDRATE OVR | feedrare override value |
| 17h | 2 | SPINDLE SPEED OVR | spindle speed override value |
| 19h | 2 | FEEDRATE | feedrate value |
| 1Bh | 2 | SPINDLE SPEED | spindle speed value |
| 1Dh | 1 | STEP MUL | step multiplier for hand-wheel |
| 1Eh | 1 | STATE | notify user for something |

| HB04 ( 37 bytes ) | | | |
|---|---|---|---|
| offset | size ( bytes ) | name | description |
| 00h | 2 | MAGIC | FDFEh used to detect start sequences |
| 02h | 1 | DAY | day of the month |
| 03h | 2 | WC_X_INT | X workspace position integer |
| 05h | 2 | WC_X_FRAC | X workspace position fractional |
| 07h | 2 | WC_Y_INT | Y workspace position integer |
| 09h | 2 | WC_Y_FRAC | Y workspace position fractional |
| 0Bh | 2 | WC_Z_INT | Z workspace position integer |
| 0Dh | 2 | WC_Z_FRAC | Z workspace position fractional |
| 0Fh | 2 | MC_X_INT | X machine position integer |
| 11h | 2 | MC_X_FRAC | X machine position fractional |
| 13h | 2 | MC_Y_INT | Y machine position integer |

| 15h | 2 | MC_Y_FRAC | Y machine position fractional |
|-----|---|-----------|------------------------------|
| 17h | 2 | MC_Z_INT | Z machine position integer |
| 19h | 2 | MC_Z_FRAC | Z machine position fractional |
| 1Bh | 2 | FEEDRATE OVR | feedrare override value |
| 1Dh | 2 | SPINDLE SPEED OVR | spindle speed override value |
| 1Fh | 2 | FEEDRATE | feedrate value |
| 21h | 2 | SPINDLE SPEED | spindle speed value |
| 23h | 1 | STEP MUL | step multiplier for hand-wheel |
| 24h | 1 | STATE | notify user for something |

**MAGIC** - magic value indicate what it is a first packet sequence and it is always **0**
**xFDFE DAY** - day of the month, it will read back as XORED DAY ( see before )
**\*_\*C_INT** - it is integer part of position without sign. Example:

| value | result |
|-------|--------|
| 135.17 | 135 |
| -77.89 | 77 |

**\*_\*C_FRAC** – has 8bit for HB03 and 16bit for HB04 it is simple when you look to display max
frac part for HB03 it is .99 and for HB04 it is .9999 this value also have a SIGN bit if value is
negative you must set most significant bit to 1, else drop it. Example:

| value | device | result |
|-------|--------|--------|
| -117.33 | HB03 | 33  OR 80h |
| -220.334 | HB04 | 3340  OR  8000h |

**FEEDRATE OVR** – display our feedrate override value
**SPINDLE SPEED OVR** - display our speed override value
**FEEDRATE** - current feedrate
**SPINDLE SPEED** - current speed
**STEP MUL** - it is step multiplier for our hand-wheel. If we change it in software we can show this
value to user. It divided into two part low and hi nibble.

| Low nibble | | | Hi nibble | |
|------------|-------------|--|-----------|-------------|
| **value** | **description** | | **value** | **description** |
| 0x00 | 0*1x | | 10h | back to origin |
| 0x01 | 1*1x | | 20h | floating on the knife(?) |
| 0x02 | 5*1x | | 50h | back to mechanical origin |
| 0x03 | 10*1x | | 60h | fine adjustment |
| 0x04 | 20*1 x | | | |
| 0x05 | 30*1 x | | | |
| 0x06 | 40*1 x | | | |
| 0x07 | 50*1 x | | | |
| 0x08 | 100*1 x | | | |
| 0x09 | 500*1 x | | | |

| 0x0A | 1000*1 x | |
|------|----------|--|

**STATE** - this is state value, used to notify user of state of our machine or if you need something from user.

| value | description |
|-------|-------------|
| 0x01 | run state blink |
| 0x02 | pause state blink |
| 0x40 ( 0x80 ) | flash yes/no leds ( HB03 only ? ) |

**Changelog:**

09 /01/2014 - initial release

13/01/2014 – add key codes, fix hb04 wheel modes