User Manual

XCDX
Four Axis Controller/ Driver
Limited Warranty

Nanomotion Ltd. (hereinafter NM) warrants the product (other than software) manufactured by it to be free from defects in material and workmanship for a period of time of one year (except those parts normally considered as consumable/expendable components such as motor conditioning brushes). The warranty commences thirty (30) days from the date of shipment.

NM warrants those parts replaced under warranty for a period equal to the remaining warranty coverage of the original part.

NM's sole and exclusive obligation under this warranty provision shall be to repair, or at its sole option exchange defective products or the relevant part or component, but only if: (i) the Purchaser reports the defect to NM in writing and provides a description of the defective product and complete information about the manner of its discovery within ten (10) days of its discovery; (ii) NM has the opportunity to investigate the reported defect and to determine that the defect arises from faulty material, parts or workmanship; and (iii) the Purchaser returns the affected product to a location designated by NM. These provisions constitute the exclusive remedy of the Purchaser for product defects or any other claim of liability in connection with the purchase or use of NM products.

This warranty policy applies only to NM products purchased directly from NM or from an authorized NM distributor or representative.

This warranty shall not apply to (i) products repaired or altered by anyone other than those authorized by NM; (ii) products subjected to negligence, accidents or damage by circumstances beyond NM control; (iii) product subjected to improper operation or maintenance (i.e. operation not in accordance with NM Installation Manuals and/or instructions) or for use other than the original purpose for which the product was designed to be used.

NM shall not in any event have obligations or liabilities to the Purchaser or any other party for loss of profits, loss of use or incidental, increased cost of operation or delays in operation, special or consequential damages, whether based on contract, tort (including negligence), strict liability, or any other theory or form of action, even if NM has been advised of the possibility thereof, arising out of or in connection with the manufacture, sale, delivery, use, repair or
performance of the NM products. Without limiting the generality of the preceding sentence, NM shall not be liable to the Purchaser for personal injury or property damages.

**Patents**

Nanomotion products are covered under one or more of the following registered or applied for patents.

5,453,653; 5,616,980; 5,714,833; 111597; 5,640,063; 6,247,338; 6,244,076; 6,747,391; 6,661,153; 69836991.3; 6,384,515; 7,119,477; 7,075,211; 69932359.5; 1186063; 7,211,929; 69941195.5; 1577961; 4813708; 6,879,085; 6,979,936; 7,439,652; 7061158; 1800356; 1800356; 1800356; 2007-533057 (pending); 2011-093431 (pending); 7,876,509; 10-2007-7009928 (pending); 200780019448.6; 7713361.9 (pending); 12/294,926 (pending); GB200800004178 (pending); GB200900003796 (pending); 12/398,216 (pending); GB2446428; 12/517,261 (pending); 08702695.1 (pending); 10-2009-7017629 (pending); 12/524,164 (pending); 12/581,194 (pending)

**Contact Information**

Website: www.nanomotion.com

**Customer Service**

Contact your local distributor or email Nanomotion Ltd. Technical Support Department at techsupport@nanomotion.com, with detailed problem description, additions, corrections or suggestions.

**Nanomotion Ltd. Worldwide Headquarters**

Mordot HaCarmel Industrial Park
HaYetsira Street, PO Box 623
Yokneam 20692
Tel: +972-73-249-8000
Fax: +972-73-249-8099
Email: nano@nanomotion.com

**Nanomotion Inc - US Headquarters**

1 Comac Loop, Suite 14B2
Ronkonkoma
NY 11779
Tel: +1-800-8216266
Fax: +1-631-5851947
Email: nanoUS@nanomotion.com
**Revision History**

The following table shows the last three revisions to this document.

<table>
<thead>
<tr>
<th>ECO</th>
<th>Doc Rev</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>May 2016</td>
<td>Release</td>
</tr>
</tbody>
</table>
# Table of Contents

**Introduction** ................................................................. 8
  Conventions used in this manual ......................................... 8
  Related .............................................................................. 9

**Overview & Specification** .................................................... 10
  Specifications. ..................................................................... 10
  Front and Rear Panel Layout .................................................. 11
  Power .................................................................................. 12
  Motor Interface ..................................................................... 13
  Encoder Interface ................................................................... 13
  I/O Connectors. ..................................................................... 14
  Isolated signals ...................................................................... 17
  Digital triggers ....................................................................... 19
  Position Pulse Incremental (ppi) .............................................. 19
  Coordinated moves .................................................................. 20
  Emergency Stop (EMO) ........................................................... 20
  Joystick ................................................................................ 20

**Installation and Operation** .................................................. 21
  Electrical Connections. ............................................................ 21
  Operating the XCDX Controller/Driver ..................................... 21
    System Setup ....................................................................... 22
    XCDX Commander ................................................................ 23
    XCDX Test Panel ................................................................. 25
    Commands tab ...................................................................... 25
    Raster tab .......................................................................... 27
    Log Settings ........................................................................ 28
    Joystick .............................................................................. 30
    Flash ................................................................................. 30
    Upgrade ............................................................................... 32
    User Flash .......................................................................... 32

**XCDX Communications Protocol** ............................................ 34
  COM Settings. ....................................................................... 34
  Destination Address ................................................................ 34
  Communication Protocol ......................................................... 35
    General USB Protocol .......................................................... 36
    Ethernet Protocol ................................................................. 36
    DHCP protocol ..................................................................... 36
    Static address ...................................................................... 36
    AutoIP. ............................................................................... 36
    MAC Address ....................................................................... 36
    UDP ports ........................................................................... 37
    Command and reply structure .............................................. 37
    Discovery protocol ............................................................... 40
  Data types. ........................................................................... 41
  XCDX Commands ................................................................. 41
Set IP ................................................................. 41
Add Vectors .......................................................... 42
Get Version .......................................................... 42
Axis enable .......................................................... 42
Get IP ................................................................. 43
Set License .......................................................... 43
Get License .......................................................... 43
Get IP ................................................................. 44
GET MAC Address .................................................... 44
Home ................................................................. 44
Get Status ............................................................ 45
Set Origin ............................................................ 45
Kill ..................................................................... 46
Set Trigger Polarity ................................................... 46
Joystick ............................................................... 46
Center Joystick ....................................................... 47
Raster Data ............................................................ 47
Raster Mode .......................................................... 48
Move ................................................................. 48
Read User Flash ....................................................... 48
  Write User Flash ................................................... 49
Erase User Flash ..................................................... 49
Special Operations .................................................. 49
  Raster operation .................................................. 49
  Raster Data ........................................................ 50
  Rectangular raster ............................................... 52
  Non-rectangular raster .......................................... 53
  Pause, Abort and Single Vector .................................. 53
  Step-Repeat Operation .......................................... 54
Example Scripts ...................................................... 54
  Setting IO State ................................................ 54
  Creating Square Pulse on IO_2 .................................. 54

Programming the XCDX Controller/Driver ......................... 56
  Programming the XCDX MCU and the XCD-HR MCUs .......... 56
  PREPARE THE UPGRADE ...................................... 56
  Writing the Script .............................................. 56
  TO EXECUTE UPGRADE: ....................................... 57
1  INTRODUCTION

The manual provides operating information about the Nanomotion Ltd XCDX™ Controller/Driver. The XCDX Controller/Driver can control up to 4 axes on a mechanical stage using either programmed operations or a joystick.

This manual contains:

- **Overview & Specification on page 10**
- **Installation and Operation on page 21**
- **XCDX Communications Protocol on page 34**
- **Programming the XCDX Controller/Driver on page 56**

The XCDX Box contains two firmware and software programs. The XCD-HR™ and EDGE™ Controller/Drivers are programmed with XCD Motion Script™ (XMS). for detailed information about this firmware refer to the latest XCD Firmware user manual.

1.1 CONVENTIONS USED IN THIS MANUAL

Throughout this manual commands are shown in **BOLD** and parameter values are shown in *italics*.

**NOTE**: Notes provide additional information that is not included in the normal text flow.

**CAUTION**: Caution provides information about actions that will adversely affect system performance.
Related

Best Known Methods: Provides additional detailed information about operations and methods.

BKM

Danger: Indicates operations or activities that may cause damage to equipment or injury to personnel.

1.2 RELATED

The following table lists Nanomotion products which may have this software version. Refer to your user manual to verify installed software version.

<table>
<thead>
<tr>
<th>Product</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCD HR1™ Controller Driver</td>
<td>XCD-HR1-BD-04</td>
</tr>
<tr>
<td>XCD HR2™ Controller Driver</td>
<td>XCD-HR2-BD-04</td>
</tr>
<tr>
<td>XCD HR4™ Controller Driver</td>
<td>XCD-HR4-BD-04</td>
</tr>
<tr>
<td>XCD HR8™ Controller Drive</td>
<td>XCD-HR8-BD-04</td>
</tr>
<tr>
<td>XCD EDGE™ Controller</td>
<td>XCD-EDGE-BD-03</td>
</tr>
<tr>
<td>XCD™ Component</td>
<td>IC000028</td>
</tr>
</tbody>
</table>
The XCDX motherboard has an FPGA and MCU that provide control of up to four axes with any of the following motors:

- EDGE
- HR1
- HR2
- HR4
- HR8
- HR16MCU

Control of the axes is by open loop control from the XCDX Commander, an external joystick, or a XMS program saved to each of the XCD Controller/Divers.

**System Features**

- Usable with a 3 axis Joystick
- Trigger composition from the 4 axis drivers
- general I/O outputs (software controlled)
- Analog/Digital I/O on rear panel
- USB and ethernet communication (MAC Address)
- Power switch on the Back panel
- Front panel LEDs show XCDX power and XCD-HR status
- Communication between the XCDX and the host computer is through the XCDX Commander (refer to XCDX Commander, page 23).

### 2.1 Specifications

All communications with the computer are via USB or Ethernet connection.

- USB2.0 (USB type B connector, VCP implemented)
- Ethernet 10/100 Mbit
2.1.1 **FRONT AND REAR PANEL LAYOUT**

The XCDX front panel has LEDs to indicate the status of the XCDX box and each of the installed axes.

![Figure 1: Front Panel](image1)

- **XCDX Power** - power to the XCDX box
- **Axis indicators**
  - **Power** - power to the axis Controller/Driver
  - **Enable** - the axis motor is enabled
  - **Fault** - indicates an error in axis (Controller/Driver, motor, or interconnection)

![Figure 2: Rear Panel](image2)
Specifications

- Motor connectors - refer to Table 2
- Encoder connectors - refer to Table 3
- I/O connectors - refer to Table 4 and Table 5

2.1.2 POWER

The XCDX Controller/Driver is factory configured for either AC or DC power based on customer request. The XCDX Box meets EMI/RFI compatibility according to CE class B. The front panel has LED power indicators for the XCDX Box and the XCD Controller//Driver for each configured axis.

AC Power

AC power supply is 110/220VAC using a standard US or European mains connection.

Table 1: 6 pin Male, DIN453322

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24 VDC</td>
</tr>
<tr>
<td>2</td>
<td>Return</td>
</tr>
<tr>
<td>3</td>
<td>Return</td>
</tr>
<tr>
<td>4</td>
<td>Return</td>
</tr>
<tr>
<td>5</td>
<td>24 VDC</td>
</tr>
<tr>
<td>6</td>
<td>24 VDC</td>
</tr>
<tr>
<td>outer shell</td>
<td>GND</td>
</tr>
</tbody>
</table>
Specifications

2.1.3 MOTOR INTERFACE

XCDX have 4x Motor connectors corresponding to each axis.

Table 2: Motor connection - NM standard  D-SUB 9pin plug

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Motor Power</td>
<td>Motor White - High Voltage</td>
</tr>
<tr>
<td>4</td>
<td>Motor Power</td>
<td>Motor Common</td>
</tr>
<tr>
<td>5</td>
<td>Motor Power</td>
<td>Motor Red - High Voltage</td>
</tr>
<tr>
<td>1</td>
<td>Protection</td>
<td>Motor Connected Common</td>
</tr>
<tr>
<td>6</td>
<td>Protection</td>
<td>Motor Connected</td>
</tr>
<tr>
<td>Case</td>
<td>Shield</td>
<td>Cable shield</td>
</tr>
</tbody>
</table>

Safety hazard

Pins 1 and 6 are a safety connection that disables the driver when the motor is disconnected. They must be shorted together on the plug connected to the motor.

2.1.4 ENCODER INTERFACE

The XCDX Controller/Driver supports up to 4 digital Differential encoders (Renishaw), each encoder corresponding to specific axis and XCD-HR Controller/Driver connected. When configured with XCD-EDGE Controller/Drivers, only single-ended encoders are supported.

Table 3: Encoder Connector Pinout

<table>
<thead>
<tr>
<th>Function</th>
<th>Output Type</th>
<th>Signal</th>
<th>Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td></td>
<td>5 V</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 V</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 V</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 V</td>
<td>9</td>
</tr>
<tr>
<td>Incremental signal</td>
<td>RS422A</td>
<td>A+</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Digital</td>
<td>A-</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B+</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b-</td>
<td>5</td>
</tr>
<tr>
<td>Reference Mark</td>
<td>RS422A</td>
<td>Z+</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Digital</td>
<td>Z-</td>
<td>4</td>
</tr>
<tr>
<td>Limits</td>
<td>Open</td>
<td>P</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>collector</td>
<td>Q</td>
<td>10</td>
</tr>
</tbody>
</table>
2.1.5 **I/O CONNECTORS**

The back panel has two 37 pin connectors to provide monitoring and use of various internal signals including GPIOs.

---

![Figure 3: I/O Connector Layout](image)

---

**Table 3: Encoder Connector Pinout**

<table>
<thead>
<tr>
<th>Function</th>
<th>Output Type</th>
<th>Signal</th>
<th>Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>E-</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Set-up</td>
<td>X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Shield</td>
<td>inner shield</td>
<td>no connection</td>
<td></td>
</tr>
<tr>
<td></td>
<td>outer shield</td>
<td>case</td>
<td></td>
</tr>
</tbody>
</table>

The Limits, Alarm, and Set-up signals are not connected

Are these still not enabled?

---

**Analog input**

- Connected directly to corresponding axis controller
- Input range ±10V
- Absolute maximum range ±12V
Specifications

- Input impedance >7kΩ

**Analog output**

- Connected directly to corresponding axis controller
- Output voltage range 0-10V
- Maximum load 5kΩ (1% voltage drop)

Example external power supply 24V+10% Max

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Direction</th>
<th>Description</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>GPIO 1</td>
<td>input</td>
<td>T-Axis digital GPIO1</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A3</td>
<td>GPIO 2</td>
<td>output</td>
<td>T-Axis digital GPIO2</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A4</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>GPIO 5</td>
<td>input</td>
<td>Z-Axis digital GPIO1</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A6</td>
<td>GPIO 6</td>
<td>output</td>
<td>Z-Axis digital GPIO2</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A7</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td>GPIO 9</td>
<td>input</td>
<td>Y-Axis digital GPIO1</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A9</td>
<td>GPIO 10</td>
<td>output</td>
<td>Y-Axis digital GPIO2</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A10</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A11</td>
<td>GPIO 13</td>
<td>input</td>
<td>X-Axis digital GPIO1</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A12</td>
<td>GPIO 14</td>
<td>output</td>
<td>X-Axis digital GPIO2</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A13</td>
<td>24V</td>
<td></td>
<td>Do not use</td>
<td>24V</td>
</tr>
<tr>
<td>A14</td>
<td>Limit 2</td>
<td>input</td>
<td>T-Axis Right Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A15</td>
<td>Limit 3</td>
<td>input</td>
<td>Z-Axis Left Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A16</td>
<td>Limit 5</td>
<td>input</td>
<td>Y-Axis Left Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A17</td>
<td>NC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A18</td>
<td>Limit 8</td>
<td>input</td>
<td>X-Axis Right Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A19</td>
<td>GND</td>
<td></td>
<td>Do not use</td>
<td></td>
</tr>
<tr>
<td>A20</td>
<td>GPIO 0</td>
<td>output</td>
<td>T-Axis PPI digital output</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A21</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A22</td>
<td>GPIO 3</td>
<td>output</td>
<td>T-Axis digital GPIO3</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A23</td>
<td>GPIO 4</td>
<td>output</td>
<td>Z-Axis PPI digital output</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A24</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A25</td>
<td>GPIO 7</td>
<td>output</td>
<td>Z-Axis digital GPIO3</td>
<td>TTL 5V</td>
</tr>
</tbody>
</table>
### Table 4: J5A- GPIO, Connector 2

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Direction</th>
<th>Description</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A26</td>
<td>GPIO 8</td>
<td>output</td>
<td>Y-Axis PPI digital output</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A27</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A28</td>
<td>GPIO 11</td>
<td>output</td>
<td>Y-Axis digital GPIO3</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A29</td>
<td>GPIO 12</td>
<td>output</td>
<td>X-Axis PPI digital output</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A30</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>A31</td>
<td>GPIO 15</td>
<td>output</td>
<td>X-Axis digital GPIO3</td>
<td>TTL 5V</td>
</tr>
<tr>
<td>A32</td>
<td>Limit 1</td>
<td>input</td>
<td>T-Axis Left Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A33</td>
<td>NC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A34</td>
<td>Limit 4</td>
<td>input</td>
<td>Z-Axis Right Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A35</td>
<td>Limit 6</td>
<td>input</td>
<td>Y-Axis Right Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A36</td>
<td>Limit 7</td>
<td>input</td>
<td>X-Axis Left Limit</td>
<td>Pull to GND</td>
</tr>
<tr>
<td>A37</td>
<td>ES</td>
<td>input</td>
<td>Emergency Stop</td>
<td>Pull to 24V</td>
</tr>
</tbody>
</table>

### Table 5: J5B - Analog IO, Connector 1

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Direction</th>
<th>Description</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td>AI 2</td>
<td>input</td>
<td>T-Axis Analog Input 2</td>
<td>+/-10V</td>
</tr>
<tr>
<td>B3</td>
<td>AO 1</td>
<td>output</td>
<td>X-Axis AO 2</td>
<td>0V-10V</td>
</tr>
<tr>
<td>B4</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>B5</td>
<td>AI 4</td>
<td>input</td>
<td>Z-Axis Analog Input 2</td>
<td>+/-10V</td>
</tr>
<tr>
<td>B6</td>
<td>AO 3</td>
<td>output</td>
<td>Y-Axis AO 2</td>
<td>0V-10V</td>
</tr>
<tr>
<td>B7</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>B8</td>
<td>AI 6</td>
<td>input</td>
<td>Y-Axis Analog Input 2</td>
<td>+/-10V</td>
</tr>
<tr>
<td>B9</td>
<td>AO 5</td>
<td>output</td>
<td>Z-Axis AO 2</td>
<td>0V-10V</td>
</tr>
<tr>
<td>B10</td>
<td>GND</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>B11</td>
<td>AI 8</td>
<td>input</td>
<td>X-Axis Analog Input 2</td>
<td>+/-10V</td>
</tr>
<tr>
<td>B12</td>
<td>AO 7</td>
<td>output</td>
<td>T-Axis AO 2</td>
<td>0V-10V</td>
</tr>
<tr>
<td>B13</td>
<td>INS_OI_</td>
<td></td>
<td>Do not use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B14</td>
<td>INS_OUT</td>
<td></td>
<td>Do not use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B15</td>
<td>INS_OUT</td>
<td></td>
<td>Do not use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Specifications

2.1.6 ISOLATED SIGNALS

The XCDX Box provides internally isolated inputs and outputs.

- Limits
- INS - isolated IO inputs
- PLD_INS_IN
- PLD_INS_OUT

Each of the following images shows the isolated circuit.
Specifications

Figure 4: Limit and EMO Switch Input Connections

Figure 5: Isolated Output Connection
2.1.7  **DIGITAL TRIGGERS**

The XCDX Controller/Driver supports a TTL level digital trigger output signal (selectable polarity) that can be programmed to provide a pulse at various times during the execution of the stored program. The pulse is available on a BNC connector (shell is ground). This signal can provide triggers from all 4 axes. The XCDX Motherboard combines (ORs) the 4 signals to create the output signal. The trigger is programmable (Level, Axis in trigger, Pulse width)

Maximum delay from each axis output trigger to the XCDX output trigger is less than 1μs. This corresponds to 100ns jitter.

**Trigger generation**

- Standard - a pulse is generated on position compare
- Extended - the output goes high when Position 1 is reached, and goes low when Position 2 (> Position 1) is reached.

2.1.7.1  **POSITION PULSE INCREMENTAL (PPI)**

The Incremental Position Pulse (ppi) allows for a position compare function for an axis or between axes. This function provides for synchronizing motor/axis position. The function is supported in XCD-HR hardware with a related delay of 0.1μsecond.
Specifications

The function can be activated using either the XMS command `ppi` or Host command `POSITION PULSE INCREMENTAL (33)`. Three parameters are specified in the command:

- **start** - when the start position is reached the Controller generates the first pulse.
- **increment** - specifies the pulse timing in mm; the next position for compare is set as previous position plus increment. The increment is rounded to integer number of encoder pulses.
- **count** - specifies the number of pulses. If count is set to one a single pulse is produced at the start point.

For a detailed explanation of the Position Compare (`ppi`) refer to the XCD-HR Firmware user manual, XCD0458002-00

2.1.7.2 **COORDINATED MOVES**

The XCDX can coordinate moves between the X and Y axes when operating in Raster mode.

To minimize turnaround time, the secondary axis must start motion as soon as possible after completing last trigger position in the primary axis.

2.1.8 **EMERGENCY STOP (EMO)**

The user can use the EMO to stop stage operation. Refer to Figure 4, page 18

2.1.9 **JOYSTICK**

The XCDX can be controlled by a 3 axis 2 button joystick when configured for 3 axis operation. The default setting at power up is joystick. The XCDX SW can lockout the joystick if desired. The joystick interface connector is a mini DIN 8 pin.

The joystick is calibrated to its middle position at beginning of operation
3 INSTALLATION AND OPERATION

This section provides a description of the process of connecting the system components, starting the system, and launching and using the SW. The following components are supplied:

- Controller/Driver
- Simulation SW for Controller/Driver and XCDX Commander. These components are supplied for testing and integration purposes only.

In addition the following components must be added:

- USB cable type B
- joystick (optional)

Before installing and operating the software ensure that the host computer has Microsoft® .NET 4 installed.

3.1 ELECTRICAL CONNECTIONS

The XCDX Controller/Driver can be configured to operate 4 independent axes of a mechanical stage. Each axis is connected to a Nanomotion XCD™ or EDGE™ motor (refer to Figure 1, page 22). For closed-loop operation an encoder can be connected from the stage axis back to the XCDX box.
3.2 OPERATING THE XCDX CONTROLLER/DRIVER

The XCDX Box can control a stage with up to 4 axes. Control is either through a motion profile saved on the XCD-HR and EDGE Controller/Drivers, or with a joystick that can control up to 3 axes.

The XCD-HR and EDGE Controller/Drivers are factory programmed. The user can modify some of the parameters and include a motion profile to control stage movement. Refer to the XCD FW ver 1.5.0.7 User manual, XCD0458002-00 for detailed information.
3.2.1 **SYSTEM SETUP**

1. Install the XCDX Commander on the host computer.

---

The XCDX Commander and sample test scripts was supplied with the XCDX Box. For additional copies contact your field service representative.

---

Ensure that the Controller/Driver’s power switch is in the off position and connect to the power (AC or DC).

Connect each axis of the stage, motor and encoder, to the appropriate connector on the rear panel of the Controller Driver.

Connect the USB cable from the Controller/Driver to the host computer.

(Optional) Connect the joystick to the Controller/Driver.

On the host computer launch the XCDX Commander.

Turn on the Controller/Driver.

---

3.2.2 **XCDX COMMANDER**

The XCDX Commander provides communication with each of the Nanomotion XCD Controller/Drivers and the XCDX motherboard. The XCDX Commander's main window has four main panels:

- Communication
- Motion program
- Setup
- Command interface
Operating the XCDX Controller/Driver

Communication

The Communications panel has two dropdown menus and an INFO field.

- **Port** - selects the port on the host computer connected to the XCDX Box
- **Address** - Selects which subassembly in the XCDX Box to communicate with.

<table>
<thead>
<tr>
<th>Address</th>
<th>Controller/Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X axis</td>
</tr>
<tr>
<td>2</td>
<td>Y axis</td>
</tr>
<tr>
<td>3</td>
<td>Z axis</td>
</tr>
<tr>
<td>4</td>
<td>T axis</td>
</tr>
<tr>
<td>80</td>
<td>Motherboard</td>
</tr>
</tbody>
</table>

- **Info** - reads the details of the Controller/Driver. Details include FW version and board configuration.

Motion Program
Operating the XCDX Controller/Driver

Allows opening, editing, and executing a motion profile program (XMS). For details on motion programs refer to the XCD Firmware user manual for the programmed FW.

Setup

The panel contains four buttons that provide access to dialogs for programming and servo control actions. For detailed information on using first three button refer to the XCD FW ver 1.5.0.7 User manual, XCD0458002-00.

- Configure Servo Loop
- Tune Servo Loop
- Manage Flash Data

In current FW release XCDXF1000000000 there is a known bug that prevents Saving to Flash when connected using the IP address. Saving to Flash must be done with a direct link between the host computer and the XCDX box.

- XCDX Test - refer to the XCDX Test Panel section below

Command Interface

Provides the ability to send individual commands or read variable parameters from the selected XCD Controller/Driver or the motherboard.

3.2.3 XCDX TEST PANEL

This section describes the panels in the XCDX Test Panel.

The Test Panel has been modified. These paragraphs and images must be reviewed.

3.2.3.1 COMMANDS TAB

The Command tab provides initiation a range of functions:

- **Set Stage Initialization** - allows selecting one of four homing modes.
  - Positive hard stop and index pulse (Default)
  - Negative hard stop and index pulse
  - Positive hard stop
  - Negative hard stop
- **Get Status**
- **Set Stage Zero Origin** - after moving the stage to a new position select this function to reset the zero origin.
- **Get Limits** - displays the stage limits in the lower pane.
Operating the XCDX Controller/ Driver

- **Kill** - aborts current stage operation.
- **Set Trigger Polarity** - options are Active High and Active Low.
- **Joystick Enable** - enables the joystick. Before selecting and sending this function, set the Init Joystick center option.
- **Init Joystick center** - provides a zero setting for the joystick. This must be selected and sent before selecting Joystick Enable.
- **Set Position** - allows setting a position for the 3 axes.
- **Set Power Supplies** - enables the power supplies for each of the XCD Controller/Drivers.
- **Led Test** - Tests the LEDs on the front panel. Report value (Hex) in bottom field.
- **SEND** - allows sending a command to anyone of the XCD Drivers. Enter the command code, parameters and select the component to send to and click SEND.

![Figure 3: Serial Port Tab (for reference only)](image)
3.2.3.2 RASTER TAB
The Raster data must be sent before selecting and sending Raster mode selected.

- **Raster Mode** - provides selection of any of four modes: Start, Pause, Resume, Abort, Send/Repeat.
- **Raster Data** - provides selection of parameters for the Raster.
  - Vector Count - count on the secondary axis.
  - Speed - set in mm/sec.
  - Initial X
  - Initial Y
  - Initial Speed - Provides a selection for faster movement at the start.
  - Primary Axis - Provides selection of the X or Y axis as primary.
  - Trigger Resolution - set in mm.
  - Dwell - Not functional in this release
  - Shape - provides selection of the raster shape. Option include: Rectangle, Horizontal Line, Vertical Line, Triangle.
3.2.3.3 **LOG SETTINGS**

The Log Settings tab is used for integration only.

- **CLEAR** - deletes data in lower field
- **Command 26 reply** determines the display format. (Binary and Float are not operational)
Operating the XCDX Controller/Driver

Figure 5: Log Settings Tab
3.2.3.4 JOYSTICK
The three fields read the voltage on each axis of the joystick.

![Joystick Tab]

Figure 6: Joystick Tab

3.2.3.5 FLASH
- IP Address (refer to Ethernet Protocol, page 36)
- Net Mask
- License
- Active XCDs

⚠️ This function is set at the factory. Do not change.
By default X, Y, Z are active. It can be modified from the commander under the Flash tab:

- 0 = All LEDs are off
- 1 = X is active
- 2 = X,Y is active
- 3 = X,Y,Z is active
- 4 = X,Y,Z,T is active

Changes will apply after restart.
### 3.2.3.6 Upgrade

The Upgrade tab allows running the upgrade script directly from the XCDX Commander. Refer to *Programming the XCDX Controller/Driver*, page 56 for details of preparing the upgrade script.

![Upgrade Tab](image)

**Figure 8: Upgrade Tab**

### 3.2.3.7 User Flash

The Flash pane provides random access to the 128KB storage for free entry data. Each Read/Write operation has a maximum of 32Bytes in the Length field. The Offset field provides entry to the remaining data area.

---

In order to overwrite data it is necessary to delete the entire 128KB before writing new data.

---

The Read function requires entering the offset and length of the desired data. To read the Flash using a command refer to *Read User Flash*, page 48, and *Write User Flash*, page 49.
Operating the XCDX Controller/Driver

Figure 9: User Flash
4 XCDX COMMUNICATIONS PROTOCOL

4.1 COM SETTINGS

XCDX Commander automatically configures the selected COM channel.

If the customer creates his own program to communicate with XCDX, the COM channel should be configured as follows:

- Baud Rate: 115200
- Number of Data Bits: 8
- Parity: None
- Number of Stop bits: 1
- Flow Control: None

4.2 DESTINATION ADDRESS

Each command sent to XCDX has a field of destination address. Destination address defines which controller processes the command. The following destination addresses are recognized:

- 1 - XCD 1 (X)
- 2 - XCD 2 (Y)
- 3 - XCD 3 (Z)
- 4 - XCD 4 (T)
- 128 (0x80) - MCU

Commands sent to channels 1/2/3 follow the standard XCD protocol; see XCD User Manual for full specification. The total length of the commands should not exceed 50 bytes.

Commands sent to channel 128 (0x80) are extended commands executed in MCU; see XCDX Commands for details. The total length of the commands is up to 80 bytes.
**4.3 Communication Protocol**

The controller is a communication client and plays a passive role. Other side (customer processor or PC) is a communication host and plays an active role.

The communication is performed in a ping-pong manner. Each communication session includes two events:

- The host initiates communication by sending a command.
- The controller sends reply; in many cases, the reply is simply a prompt, which reports whether the command is accepted or rejected.

Each host command consists of the following parts:

### Table 1: XCDX Host Command Format

<table>
<thead>
<tr>
<th>Part</th>
<th>Byte offset</th>
<th>Size in bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command prefix</td>
<td>0</td>
<td>2</td>
<td>Synchronization sequence 0xE4, 0xA5 (228, 165)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>Destination address 0 (X), 1 (Y), 2 (Z), or 128 (MCU)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>Length of the command body in bytes</td>
</tr>
<tr>
<td>Command body</td>
<td>4</td>
<td>1</td>
<td>Command ID</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>N</td>
<td>Parameters, see specific command for details</td>
</tr>
</tbody>
</table>

Controller reply has similar parts:

### Table 2: XCDX Controller/Driver Reply Format

<table>
<thead>
<tr>
<th>Part</th>
<th>Byte offset</th>
<th>Size in bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply prefix</td>
<td>0</td>
<td>2</td>
<td>Synchronization sequence 0xE4, 0xA5 (228, 165)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>Controller address 1 (X), 2 (Y), 3 (Z), or 128 (MCU)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>Length of the reply body in bytes</td>
</tr>
<tr>
<td>Reply body</td>
<td>4</td>
<td>1</td>
<td>Command ID (copied from the command)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>Result: 1 - command accepted, 2 - command rejected</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>N</td>
<td>Extension. Reply to most commands does not require extension (N is zero).</td>
</tr>
</tbody>
</table>
4.3.1 **GENERAL USB PROTOCOL**

Communication of the XCDX box with the host Host computer is provided through USB cable with B-type device connector.

The Host computer should run WINDOWS XP/7 with Standard standard FTDI driver is required to communicate with the XCDX box.

FTDI driver enumerates connected XCDX box as one COM channel. Use XCDX Commander to find COM channel. XCDX Commander shows all available COM channels in the drop-down list of the IP combo-box; select XCDX-related channel to open communication:

4.3.2 **ETHERNET PROTOCOL**

Each XCDX box has its own 4-byte IP address. In a dotted form, IP address consists of four numbers, like 172.21.234.168.

The XCDX box determines its IP address at start up, and then accepts only commands sent to this address, or broadcasted commands.

The XCDX box tries three methods to determine IP address:

4.3.2.1 **DHCP PROTOCOL**

This is standard method if the XCDX is connected to local network that provides DHCP service. If the XCDX obtains IP address from DHCP server, it does not try other options.

4.3.2.2 **STATIC ADDRESS**

The method is available only if the customer previously saved predefined IP address to the XCDX flash memory using command 12.

This is standard method if the XCDX is connected directly to PC, or to a network where DHCP service is not available.

4.3.2.3 **AUTOIP**

The method is used only if DHCP server is not available and no static address was saved in the XCDX flash.

In this case, the XCDX tries to determine its address once it receives Discover message from the Host.

4.3.3 **MAC ADDRESS**

A unique MAC address is programmed at the factory and is printed on the back panel of the XCDX Box.
To access the MAC address refer to GET MAC Address, page 44.

4.3.4 UDP PORTS
XCDX uses the following UDP ports:
- 0x1745 (5957) is used for receiving regular XCDX commands and sending replies
- 0x1E27 (7719) is used for receiving Discover command

4.3.5 COMMAND AND REPLY STRUCTURE
XCDX Ethernet protocol is based on UDP protocol. Each message sent to the XCDX and from XCDX is a UDP datagram complying with IPv4 standard.

<table>
<thead>
<tr>
<th>Ethernet header</th>
<th>IP header</th>
<th>UDP header</th>
<th>XCDX Command block</th>
<th>Ethernet checksum</th>
</tr>
</thead>
</table>

Figure 1: UDP Protocol

UDP header should specify destination port 0x1745 (5957).

One command UDP datagram can comprise one or more XCDX commands (number in parenthesis specifies the field size in bytes):
Similarly, one reply UDP datagram can comprise one or more XCDX replies (number in parenthesis specifies the field size in bytes):

**Figure 2: UDP Command Structure**

**Figure 3: UDP Reply Structure**
Communication Protocol

Numbers in XCDX command and reply block are sent in little-endian format, i.e. the least significant byte first.

Each XCDX command block consists of the following parts:

**Table 3: XCDX command Block**

<table>
<thead>
<tr>
<th>Part</th>
<th>Byte offset</th>
<th>Size in bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block prefix</td>
<td>0</td>
<td>2</td>
<td>Constant 0x18FC (byte 0 is 0xFC, byte 1 is 0x18)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>Length of the whole block in bytes, including the block prefix</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>ID assigned by the Host</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2</td>
<td>Format code 0x0002 (byte 6 is 0x02, byte 7 is 0x00)</td>
</tr>
<tr>
<td>Command 1 prefix</td>
<td>8</td>
<td>1</td>
<td>Destination 0 (X), 1 (Y), 2 (Z), or 128 (MCU)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>Length of the command body in bytes</td>
</tr>
<tr>
<td>Command 1 body</td>
<td>10</td>
<td>1</td>
<td>Command code, see XCDX Commands</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>N</td>
<td>Parameters, see specific command for details</td>
</tr>
<tr>
<td>Command 2 prefix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command 2 body</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command N prefix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command N body</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Controller reply has similar parts; the number of replies is the same as the number of commands in the corresponding command block:

### Table 4: Controller Reply

<table>
<thead>
<tr>
<th>Part</th>
<th>Byte offset</th>
<th>Size in bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block prefix</td>
<td>0</td>
<td>2</td>
<td>Constant 0x18FC (byte 0 is 0xFC, byte 1 is 0x18)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>Length of the whole block in bytes, including the block prefix</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>ID copied from the command</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2</td>
<td>Error: 0 - no errors, &gt;0 - number of rejected commands, 0x00F0 - command format error</td>
</tr>
<tr>
<td>Reply 1 prefix</td>
<td>8</td>
<td>1</td>
<td>Source 0 (X), 1 (Y), 2 (Z), or 128 (MCU)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>Length of the reply body in bytes</td>
</tr>
<tr>
<td>Reply 1 body</td>
<td>10</td>
<td>1</td>
<td>Command code copied from the corresponding command</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1</td>
<td>Result: 1 - command accepted, 2 - command rejected</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>N</td>
<td>Extension. Reply to most commands does not require extension (N is zero).</td>
</tr>
</tbody>
</table>

### 4.3.5.1 Discovery Protocol

Discovery protocol is used to find out one or more XCDX boxes connected to a PC or to network segment.

Unlike other XCDX commands, Discover command is sent with broadcast IP address; each node in the LAN segment receives the command. The XCDX responds to Discovery command submitting to the host its IP address.

Once the responses have been received, the host knows IP addresses of all connected XCDX boxes and is able to communicate with any of them.

- Discover command is sent by Host as a UDP datagram formatted as follows:
4.4 DATA TYPES

Parameters field in a command and Extension field in a reply can contain a number of numerical values. Each value is a binary of specific type in little-endian format (the least significant byte appears first).

The following formats are used:

<table>
<thead>
<tr>
<th>Format</th>
<th>Number of bytes</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int8</td>
<td>1</td>
<td>-128 to +127</td>
</tr>
<tr>
<td>Int16</td>
<td>2</td>
<td>-32768 to +32767</td>
</tr>
<tr>
<td>Int32</td>
<td>4</td>
<td>-2147483648 to +2147483647</td>
</tr>
<tr>
<td>Real</td>
<td>4</td>
<td>-3.4<em>10^38 to +3.4</em>10^38 approximately (complying with IEEE 754)</td>
</tr>
<tr>
<td>ID</td>
<td>2</td>
<td>0 to 65535</td>
</tr>
</tbody>
</table>

4.5 XCDX COMMANDS

This section includes available XCDX commands.

4.5.1 SET IP

The command sets current IP address and network mask and stores specified values in the flash memory. The flash values will be retrieved and used at the next start up (power up). After setting IP address/mask, restart is recommended to re-initialize network operation.

- IP Address - four bytes that specify desired address in the processor byte order. E.g., address 198.162.0.103 (0xC6.0xA2.0x00.0x67) should be specified as 0x6700A2C6.
Subnet Mask - four bytes that specify desired network mask in the processor byte order. E.g., mask 255.255.255.0 (0xFF.0xFF.0xFF.0x00) should be specified as 0x00FFFFFF.

<table>
<thead>
<tr>
<th>Code</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ID</td>
<td>IP Address</td>
<td>Int32 (4)</td>
</tr>
<tr>
<td></td>
<td>IP Mask</td>
<td>Int32 (4)</td>
</tr>
</tbody>
</table>

### 4.5.2 ADD VECTORS

Adds vectors to the raster area. The command adds the specified vectors to the raster area; maximum 6 vectors can be specified in one command. The command should be used in case of non-rectangular raster area; in this case, the raster data (see command 108) should specify bit 12 in the Option parameter and non-zero Count parameter.

Each vector is specified with two values:
- \( \text{Count}_n \) - the number of trigger points on the vector
- \( X_n \) - start X coordinate of the vector

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Count1 X1</td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td>Count2 X2...</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Real (4)</td>
</tr>
</tbody>
</table>

### 4.5.3 GET VERSION

Request MCU Firmware version. The XCDX reply contains 10-byte Extension structured as follows:
- Bytes 0-3: four bytes of Firmware version
- Bytes 4-7: four bytes of Serial number
- Bytes 8-9: two bytes of Application code (not used)

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### 4.5.4 AXIS ENABLE

Data Byte = High (1) power supply enabled
- bit 0 = X
- bit 1 = Y
XCDX Commands

- bit 2 = Z
- bit 3 = T

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>axis</td>
<td>Int8(1)</td>
</tr>
</tbody>
</table>

4.5.5 **GET IP**
Request IP address and network mask stored in the flash memory. The XCDX returns IP address and network mask stored in the flash memory; the values may differ from active XCDX address and network mask. The XCDX reply contains 8-byte Extension structured as follows:
- Bytes 0-3: four bytes of IP address
- Bytes 4-7: four bytes of network mask

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

4.5.6 **SET LICENSE**
Set License data. The XCDX stores license data in the flash memory.
- Zero - one byte, should be zero.
- License - up to 60 bytes of arbitrary data.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>Zero</td>
<td>Int8(1)</td>
</tr>
<tr>
<td></td>
<td>License</td>
<td>Int8(0º59)</td>
</tr>
</tbody>
</table>

4.5.7 **GET LICENSE**
Get License data. The XCDX returns License data stored in the flash memory. Extension field of the XCDX reply contains up to 60 bytes of the License data.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
XCDX Commands

4.5.8 GET IP

The command has no parameters. The controller response contains 8 bytes:

- four bytes contain current IP address of the XCDX
- next four bytes contain network mask.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>74</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.5.9 GET MAC ADDRESS

The command has no parameters. The controller response contains 6 bytes that specify the XCDX MAC address.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.5.10 HOME

Initiate a Homing operation for all selected axes. All parameters are optional. XCDX stores internal table of default parameters for each axis. Therefore, if no parameters are specified XCDX executes homing process specific for each selected axis.

However, if a parameter is specified, it replaces the default value for all selected axes.

- **Selector (optional)** - a bitwise mask that defines which coordinates are involved (bit 0 - X, bit 1 - Y, bit 2 - Z, bit 3 - T). If the parameter is omitted, all axes assumed.
- **Method** (optional) defines sequence of homing operations. See XCD Software User Manual for detailed information.
- **Origin** (optional) defines the position at the homing point.
- **Velocity1** (optional) defines the first stage velocity.
- **Velocity2** (optional) defines the second stage velocity.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Selector (opt) Method (opt) Origin (opt) Vel 1 (opt) Vel 2 (opt)</td>
<td>Int8(1) Int8 (1) Real(4) Real(4) Real (4)</td>
</tr>
</tbody>
</table>
### 4.5.11 GET STATUS

Request XCDX status.

The XCDX reply contains 28-byte Extension field structured as follows:

- Bytes 0-3: bitwise MCU state:
  - Bit 0: X axis power; 1 - powered, 0 - not powered
  - Bit 1: Y axis power; 1 - powered, 0 - not powered
  - Bit 2: Z axis power; 1 - powered, 0 - not powered
  - Bit 3: T axis power, 1 - powered, 0 - not powered
  - Bit 8: X joystick; 1 - enabled, 0 - disabled
  - Bit 9: Y joystick; 1 - enabled, 0 - disabled
  - Bit 10: Z joystick; 1 - enabled, 0 - disabled
  - Bit 16: raster; 1 - in progress, 0 - disabled
- Bytes 4-7: X status (variable 900)
- Bytes 8-11: X program status (variable 901)
- Bytes 12-15: Y status (variable 900)
- Bytes 16-19: Y program status (variable 901)
- Bytes 20-23: Z status (variable 900)
- Bytes 24-27: Z program status (variable 901)

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### 4.5.12 SET ORIGIN

Set coordinates in the current point. If all parameters are omitted, the command sets origin (zero point) in the current point.

- Selector (optional): a bitwise mask that defines which coordinates are set (bit 0 - X, bit 1 - Y, bit 2 - Z, bit 3 - T). If the parameter is omitted, all axes assumed.
- X (optional): defines X value. If omitted, X is set to zero in the current point.
- Y (optional): defines Y value. If omitted, Y is set to zero in the current point.
- Z (optional): defines Z value. If omitted, Z is set to zero in the current point.
XCDX Commands

- T (optional): defines T value. If omitted, T is set to zero in the current point.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>Selector (opt)</td>
<td>Int8 (1)</td>
</tr>
<tr>
<td></td>
<td>X (opt)</td>
<td>Real (4)</td>
</tr>
<tr>
<td></td>
<td>Y (opt)</td>
<td>Real (4)</td>
</tr>
<tr>
<td></td>
<td>Z (opt)</td>
<td>Real (4)</td>
</tr>
<tr>
<td></td>
<td>T (opt)</td>
<td>Real (4)</td>
</tr>
</tbody>
</table>

4.5.13 **Kill**

Kill motion. The command causes deceleration of the axis at KDEC in all axes or in the specified axes.

Selector (optional) - a bitwise mask that defines which coordinates are involved (bit 0 - X, bit 1 - Y, bit 2 - Z, bit 3 - T). If the parameter is omitted, all axes assumed.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>Selector (opt)</td>
<td>Int8 (1)</td>
</tr>
</tbody>
</table>

4.5.14 **Set Trigger Polarity**

Set trigger polarity.

The parameter defines polarity of the trigger pulse:

- 0 - active low
- 1 - active high

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>Polarity</td>
<td>Int8 (1)</td>
</tr>
</tbody>
</table>

4.5.15 **Joystick**

Enable / disable joystick operation. If joystick is enabled, three analog signals from 3D joystick are transformed into X/Y/Z velocity. If joystick is disabled, the signals are ignored.

- Selector: a bitwise mask that defines which coordinates are involved (bit 0 - X, bit 1 - Y, bit 2 - Z).
**Enable** defines required state: 0 - disabled, 1 - enabled.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>107</td>
<td>Selector</td>
<td>Int8 (1)</td>
</tr>
<tr>
<td></td>
<td>Enable</td>
<td>Int8 (1)</td>
</tr>
</tbody>
</table>

**4.5.16 CENTER JOYSTICK**

Set joystick center. The command measures joystick analog inputs and stores their values as zero level. If the joystick is enabled, analog input greater than the measured level will cause motion in positive direction; analog input lower than the measured level will cause motion in negative direction.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>Selector</td>
<td>Int8(1)</td>
</tr>
<tr>
<td>(opt)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**4.5.17 RASTER DATA**

Set Raster data. The command prepares XCDX for raster motion. Refer to [Raster operation, page 49](#) for more information.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>108</td>
<td>Options</td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td>Count</td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td>Speed</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Acceleration</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Start X</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Start Y</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Increment X</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Increment Y</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Overrun</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Dwell Pulse</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Width Pulse</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Count Vector</td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td>Count</td>
<td>Int32(4)</td>
</tr>
</tbody>
</table>
4.5.18 **RASTER MODE**

Set Raster mode. The command sets raster mode. Parameter Mode can specify one of the following values:

- 1 - Run
- 2 - Pause; the motion stops at the end of current vector and waits for the next command
- 3 - Resume; the motion exits from Pause mode and continues from the next vector
- 4 - Abort; the raster operation stops immediately
- 5 - Single Vector; the motion goes through one vector and switches to Pause mode
- 9 - Step-repeat; the motion stops in each trigger point, waits for Dwell milliseconds, and then goes to the next trigger point

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>109</td>
<td>Mode</td>
<td>Int8 (1)</td>
</tr>
</tbody>
</table>

4.5.19 **MOVE**

Moves the specified axis to specified point

- Selector - a bitwise mask that defines which axes move (bit 0 - X, bit 1 - Y, bit 2 - Z). One, two, or three axes can be selected.
- X - defines X target point.
- Y (optional) - defines Y target point.
- Z (optional) - defines Z target point.

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>Selector</td>
<td>Int8(1)</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Y (opt)</td>
<td>Real(4)</td>
</tr>
<tr>
<td></td>
<td>Z (opt)</td>
<td>Real (4)</td>
</tr>
</tbody>
</table>

4.5.20 **READ USER FLASH**

The controller response contains up to 60 extension bytes that are read from the requested user memory.

- ID of the user flash (should be 1),
- offset in the user flash, from 0 to 128K even values only
Special Operations

- number of bytes to read, from 2 to 60, even values only

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>140</td>
<td>ID</td>
<td>Int8 (1)</td>
</tr>
<tr>
<td></td>
<td>offset</td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td>count</td>
<td>Int32(4)</td>
</tr>
</tbody>
</table>

4.5.21 WRITE USER FLASH

The controller writes the specified bytes to user flash.

- ID of the user flash (should be 1),
- offset in the user flash, from 0 to 128K even values only
- number of bytes to write, from 2 to 60, even values only
- Data to be written. K must be equal to the count

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>141</td>
<td>ID</td>
<td>Int8 (1)</td>
</tr>
<tr>
<td></td>
<td>offset</td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td>count</td>
<td>Int32(4)</td>
</tr>
<tr>
<td></td>
<td>Data</td>
<td>K x Int8(1)</td>
</tr>
</tbody>
</table>

4.5.22 ERASE USER FLASH

The controller erases user flash (sector 11).

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>142</td>
<td>ID</td>
<td>Int8 (1)</td>
</tr>
</tbody>
</table>

The operation takes long time; set long timeout for controller response, 5 sec at least.

4.6 SPECIAL OPERATIONS

4.6.1 RASTER OPERATION

Raster operation is complex operation that combines scanning motion with Position Pulse generation in trigger points. The trigger points build up a grid of either rectangular or non-rectangular form. The scanning motion is
implemented in vectors oriented along either X or Y axis. The following state diagram describes Raster operation:

![Figure 4: Raster Operation Flow Chart](image)

To initiate Raster operation, first send to XCDX Raster Data command. In case of non-rectangular raster, the Raster Data command must be followed with a set of Add Vectors commands to specify individual vectors. To start physical operation, send Raster Mode command with parameter 1 (Run).

Executing Raster operation can be interrupted by sending Raster Mode command with parameter 2 (Pause) or 4 (Abort). In case of Pause, the operation stops after finishing the current vector and waits for the next command. Usually, the next command is Raster Mode with parameter 3 (Resume) that continues execution from the next vector. In case of Abort, the operation stops immediately after decelerating current motion to zero velocity.

### 4.6.2 RASTER DATA

Raster Data command should be sent before raster operation to initialize internal data.

The Raster data command defines the following parameters
**Change this formatting**

**Table 6: Raster Data Commands Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Format</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>Int32 (4)</td>
<td>Bitwise mask of raster options. The following bits are meaningful: Bit 0 - Orientation: 0 - X is primary axis 1 - Y is primary axis Bit 12 - Area: 0 - Rectangular 1 - Arbitrary (vector definitions required)</td>
</tr>
<tr>
<td>Count</td>
<td>Int32 (4)</td>
<td>If bit 12 of Options is 1, the parameter defines the number of vectors to be specified in the following Add Vectors commands. If bit 12 of Options is 0, the parameter should be zero; the number of vectors should be defined by Vector Count parameter.</td>
</tr>
<tr>
<td>Speed</td>
<td>Real (4)</td>
<td>Velocity in mm/sec.</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Real (4)</td>
<td>Acceleration in mm/sec/sec.</td>
</tr>
<tr>
<td>Start X</td>
<td>Real (4)</td>
<td>Primary axis start position. The parameter is meaningful only if Options bit 12 is zero; otherwise, the parameter is ignored, and individual start position is defined with each vector.</td>
</tr>
<tr>
<td>Start Y</td>
<td>Real (4)</td>
<td>Secondary axis start position of the first vector.</td>
</tr>
<tr>
<td>Increment X</td>
<td>Real (4)</td>
<td>Distance between trigger pulses on a vector. Along with Pulse Count, the parameter define vector length: Vector Length = Increment X * (Pulse Count - 1)</td>
</tr>
<tr>
<td>Increment Y</td>
<td>Real (4)</td>
<td>Distance between vectors.</td>
</tr>
<tr>
<td>Overrun</td>
<td>Real (4)</td>
<td>Acceleration / deceleration distance. The distance is added to the beginning and to the end of each vector. The addition is required to provide triggering at constant velocity. If the parameter is zero, XCDX automatically calculates the distance using Velocity and Acceleration parameters.</td>
</tr>
</tbody>
</table>
**Special Operations**

**Table 6: Raster Data Commands Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Format</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dwell</td>
<td>Real (4)</td>
<td>Dwell in milliseconds in a trigger point. The parameter is only used in Step-repeat operation; in normal operation, the parameter is ignored.</td>
</tr>
<tr>
<td>Pulse Width</td>
<td>Real (4)</td>
<td>Width of trigger pulse in milliseconds.</td>
</tr>
<tr>
<td>Pulse Count</td>
<td>Int32 (4)</td>
<td>The number of trigger pulses on one vector. The parameter is meaningful only if Options bit 12 is zero; otherwise, the parameter is ignored, and individual pulse count is defined with each vector.</td>
</tr>
<tr>
<td>Vector Count</td>
<td>Int32 (4)</td>
<td>The number of vectors. The parameter is meaningful only if Options bit 12 is zero; otherwise, the parameter is ignored.</td>
</tr>
</tbody>
</table>

**4.6.3**  

**RECTANGULAR RASTER**

The following drawing describes the operation in case of rectangular raster (Options bit 12 is zero). The drawing also implies Options bit 0 is zero; if the bit is one, axes X and Y change their roles.
4.6.4 **NON-RECTANGULAR RASTER**

The following drawing describes the operation in case of non-rectangular raster (Options bit 12 is one). In this case, Start X and Pulse Count values from Raster Data are ignored; instead, they are specified for each vector individually.

![Non-rectangular Raster Diagram](image)

**Figure 6: Non-rectangular Raster**

4.6.5 **PAUSE, ABORT AND SINGLE VECTOR**

Raster Mode command with parameter 2 (Pause) does not stop the motion immediately. The operation continues until the end of the current vector, and only then stops. This way, all trigger points are processed at required velocity. Then XCDX holds XY position until the next Raster Mode command; further operation depends on the Raster Mode parameter:

- Raster Mode with parameter 3 (Resume) continues raster operation from the next vector.
- Raster Mode with parameter 4 (Abort) terminates the current raster operation. If the next Raster Mode command has parameter 1 (Run), the operation restarts from the first vector.

If Raster Mode with parameter 4 (Abort) is issued when raster motion is in progress, the motion is interrupted at any point, and the motion stops after deceleration to zero.
Example Scripts

If raster motion is started by Raster Mode with parameter 5 (Single vector), the motion stops at the end of the vector and waits for the next Raster Mode command as described above.

### 4.6.6  Step-Repeat Operation

Raster Mode command with parameter 9 starts raster operation in a Step-repeat mode.

In this case, the operation does not hold constant velocity at a vector, but stops in each trigger point. Dwell parameter in the Raster Data defines milliseconds to stay in each trigger point; and the time has elapsed, the motion goes to the next trigger point or to the next vector if the current vector is through.

If Raster Mode command with parameter 2 (Pause) is issued within Step-repeat operation, the motion stops in the next trigger point and waits for the next Raster Mode command. If the next command specifies Resume, the axis moves to the next trigger point and continues operation in Step-repeat mode.

### 4.7  Example Scripts

#### 4.7.1  Setting IO State

//This script shows how to use IO at XCD
// Set IO_2 to logical 1
IO_2 = 1
// Set IO_2 to logical 0
IO_2 = 0

#### 4.7.2  Creating Square Pulse on IO_2

//Infinite loop to produce square pulse at IO_2 at 20 and 10 Hz
while 1
    // wait 25 msec for 20Hz
delay 25
    // check if IO_1 is high
if IO_1 = 1
// wait additional 25 msec for 10Hz
delay 25
end

// Toggle IO_2
if IO_2
    IO_2 = 0
else
    IO_2 = 1
end
end
5 Programming the XCDX Controller/Driver

5.1 Programming the XCDX MCU and the XCD-HR MCUs

The XCDX MCU and assembled XCD-HR firmware can be upgraded using the following procedure.

5.1.1 Prepare the Upgrade

1. Create separate upgrade directory. All required files should be placed into this directory.

2. Copy the following files to the upgrade directory:
   - SBOOT_28035.b00 - secondary bootloader for XCD
   - XCDX.HEX - MCU firmware
   - LCCD_28035CLA.hex - XCD firmware
   - .S19 file(s) - configuration file(s) for XCD's, e.g. XCD_HR8-16_1-5_3E800.s19. If XCD's require different configuration files, copy several S19 files to the directory.

3. Prepare upgrade script with extension *.SF and place it to the same upgrade directory.

5.1.2 Writing the Script

The script should contain one line for each XCD and MCU to be upgraded. Each line starts with a command :XCD1, :XCD2, :XCD3, or :MCU that defines the target within the XCDX box. This is followed by one or more file names separated by space(s). The file names define the binary files to be programmed into the target defined at the beginning of the line. Specify the file
Programming the XCDX MCU and the XCD-HR MCUs

name with extension only. Do not include the directory path. The upgrade always looks for binary files in the same directory where the script file resides.

To insert comments into script file, start the comment with // (double slash).

Here is example of script file:

```// This script upgrades XCD1/2/3 to version 1.5.0.7 and MCU to version 1.1.1.2
//Example :<target> <filename>.S19 <firmware>.hex //
:XCD1 XCD_HR8-16_1-5_3E800.s19 LCCD_28035CLA.hex //
Upgrade HR8 to v.1.5.0.7
:XCD2 XCD_HR8-16_1-5_3E800.s19 LCCD_28035CLA.hex //
Upgrade HR8 to v.1.5.0.7
:XCD3 XCD_HR8-16_1-5_3E800.s19 LCCD_28035CLA.hex //
Upgrade HR8 to v.1.5.0.7
:MCU XCDX.HEX // Upgrade MCU to v.1.1.1.2```

5.1.3 TO EXECUTE UPGRADE:

1. Copy the upgrade directory (see above) to your PC.
2. Launch the XCDX Commander and establish communication with XCDX box.
3. Select **XCDX TEST PANEL**, and select **UPGRADE** tab.
4. Click **BROWSE**, navigate to the upgrade directory, and select the script file.
5. Click **UPGRADE**, confirm the message box, and watch upgrade progress in the bottom pane.
INDEX

A
analog I/O
examples ........................................14

C
closed-loop .....................................21
command
add vector ....................................42
center joystick ..............................47
get IP ...........................................43
get license ....................................43
get status .....................................45
get version ..................................42
home ...........................................44
joystick .......................................46
kill .............................................46
move ..........................................48
raster data ...................................47
raster mode ..................................48
set IP ..........................................41
set license ....................................43
set origin ....................................45
set trigger polarity ......................46
command body .................................35
command prefix ..............................35
connections
axis ...........................................21
electrical ....................................21

D
data
format ..........................................41
type ............................................41

E
code
interface ........................................13

F
features .........................................10
FPGA ...........................................10

G
get limits ......................................25
get status ......................................25

I
init joystick center ............................26
input
analog ........................................14
Installation ....................................23
IO connector ..................................14
IO connector 1
output .......................................18

J
joystick ........................................21
joystick enable ...............................26

K
kill .............................................26

L
LED test ..........................................26
limit switch
input connection .............................18

M
MCU ...........................................10, 34, 35, 39, 42, 45
motor
connector .....................................13
moves
coordinated ..................................20
joystick ........................................20
raster mode ..................................20

N
non-rectangular raster ......................53

O
output
analog ........................................15
## INDEX

### P
- PPI ..............................................19

### R
- raster
  - non-rectangular ..................53
  - rectangular ......................52
- raster data ............................27, 50
- raster data acceleration ..........51
- raster data count ......................51
- raster data parameter
  - option ..................................51
  - start X ................................51
- raster data speed ......................51
- raster mode ............................20, 27
- raster mode command
  - abort ..................................53
  - pause ..................................53
  - single ..................................53
  - step repeat ..........................54
- raster operation ......................49
- rectangular raster ....................52
- reply body ................................35
- reply prefix ................................35

### S
- send ...........................................26
- set position ................................26
- set power supplies ....................26
- set stage initialization ..............25
- set stage zero origins ...............25
- set trigger polarity ....................26
- specification
  - communication .......................10
  - power ..................................12
- step-repeat raster operation ........54

### T
- trigger .......................................19
  - generation ................................19

### U
- Upgrade ......................................32

### X
- XCDX Commander .......................23
- XCDX controller reply format .......35
- XCDX host command format ..........35
- XCDX Test Panel .......................23
- XCDX test panel
  - commands tab .........................25
  - command 26 reply ....................28
  - get limits .............................25
  - get status .............................25
  - init joystick center .................26
  - joystick enable .......................26
  - kill .....................................26
  - LED test ................................26
  - raster data ............................27
  - raster mode ............................27
  - send .....................................26
  - set position ...........................26
  - set power supplies ...................26
  - set stage initialization ..........25
  - set stage zero origin ...............25
  - set trigger polarity .................26
  - flash tab ...............................30
  - joystick tab ............................30
  - log settings tab .....................28
  - clear ....................................28
  - raster tab ..............................27