



SOLUTIONS CUBED

Position Controller Addendum  
(Hidden Commands)

Revision 1  
February 10<sup>th</sup>, 2003

Copyright © 2002 Solutions Cubed

## 1. Hidden Command Sets

### 1.0 Overview

Solutions Cubed makes a great effort to improve our products where possible. Due to customer interaction or internal research and development, additions are sometimes made to functionality or command sets in our products. This addendum lists additions to the command set available to users of our ICON, Mini, and Micro PID position control products.

To see if your product has been upgraded to make use of hidden command sets compare your firmware revision with the revision listed below by product. If your number is equal to or greater than the specified number then you should have access to the commands listed.

### 1.1 Upgrade Policy

Solutions Cubed will upgrade up to 6 products with new firmware (per customer) if the customer pays for shipping expenses (no other costs are incurred). If you have a larger quantity of products that need to be upgraded contact Solutions Cubed to discuss payment options.

### 1.2 Hidden Command Set 1

Product Firmware Revisions		
ICON_PID	MPID	UPID
4	4	2

#### Position Controller Hidden Command Set Revision 1

Command	Description
Write PID Scalar	Allows user to modify the divide-by-2 factor used to reduce the output of the PID filter. May be helpful in controlling low-resolution encoders. By reducing the PID Scalar value you increase the output of the PID filter. The PID Scalar is not stored in EEPROM and resets to its default value on power up.
Read PID Scalar	Reads the current PID Scalar value.
Write USER Register (No EEPROM)	Writes to the USER0 and USER1 registers without storing the values in EEPROM.
Write Active PID Values	Writes PID filter settings to the controller without storing them in EEPROM.
Read Active PID Values	Reads PID filter settings directly from the controller, and does not access EEPROM. This command can be used to verify the values written to the position controller by the Write Active PID Values command.

**1.2.1 Write PID Scalar**

<b>Power-On PID Scalar Settings</b>		
<b>ICON_PID</b>	<b>MPID</b>	<b>UPID</b>
<b>16</b>	<b>16</b>	<b>14</b>

**Description:** After the error signal is run through the PID filter the 32-bit result is shifted 16 (14 for UPID) places to the right. This is the equivalent to dividing the result by two 16 times (or dividing by 65,536). The result of this division is then used to generate a PWM value to drive the H-bridge. In systems with low resolution encoders the PID filter settings may not allow for sufficiently large error signals to survive the divide by 65,536 (all results are 0 PWM). In these instances the PID Scalar may be reduced to allow the PID filter settings to act as “fine-tuning” for the system.

<b>Component of Command</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
Command	246	1	246 (0xF6)
Address	0 to 255	1	1 (0x01)
PID Scalar	0 to 20	1	14 (0x0E)
Checksum	0 to 255	1	05 (0x05)
<b>Component of Response</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
ACK	6	1	6 (0x06)

**Notes:**

**1.2.2 Read PID Scalar**

**Description:** The Read PID Scalar command reads the current value of the PID Scalar.

<b>Component of Command</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
Command	247	1	247 (0xF7)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	236 (0xEC)
<b>Component of Response</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
PID Scalar	0 to 20	1	14 (0x0E)
Checksum	0 to 255	1	175 (0xAF)

**Notes:**

**1.2.3 Write User Registers (No EEPROM)**

**Description:** Write User Registers is used to define the operating mode of the controller. Unlike the normal Write User Registers command this version does not write to EEPROM.

<b>Component of Command</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
Command	248	1	248 (0xF8)
Address	0 to 255	1	1 (0x01)
USER0	0 to 255	1	34 (0x22)
USER1	0 to 255	1	49 (0x31)
Checksum	0 to 255	1	76 (0x4C)
<b>Component of Response</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
ACK	6	1	6 (0x06)

**Notes:**

**1.2.4 Write Active PID Values**

**Description:** The Write Active PID Values command is used to store the “P”, “I”, and “D” constants, and other PID filter associated values. It differs from the Write PID command in that it does not write the values to EEPROM. This can be useful if an application requires constant modifications to the PID settings, as the EEPROM will not exceed its write cycle limitations when updated with this command. In most other respects this command matches the Write PID Values command.

<b>Component of Command</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
Command	250	1	250 (0xFA)
Address	0 to 255	1	1 (0x01)
PID Number	0	1	0 (0x00)
P Constant	0 to 32,767	2	12,000 (0x2EE0)
I Constant	0 to 32,767	2	125 (0x007D)
D Constant	0 to 65,535	2	35,000 (0x88B8)
PID Period	0 to 255	1	116 (0x74)
Error Band	0 to 255	1	10 (0x0A)
Integral Clear Counter	0 to 65,535	2	1000 (0x03E8)
Checksum	0 to 255	1	41 (0x29)
<b>Component of Response</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
ACK	6	1	6 (0x06)

**Notes:**

**1.2.5 Read Active PID Values**

**Description:** Reads the active PID filter settings. Does not read from EEPROM

<b>Component of Command</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
Command	251	1	251 (0xFB)
Address	0 to 255	1	1 (0x01)
PID Number	0 to 2	1	0 (0x00)
Checksum	0 to 255	1	252 (0xFC)
<b>Component of Response</b>	<b>Range of Values</b>	<b>Number of Bytes</b>	<b>Decimal(Hex) Example</b>
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
P Constant	0 to 32,767	2	12,000 (0x2EE0)
I Constant	0 to 32,767	2	125 (0x007D)
D Constant	0 to 65,535	2	35,000 (0x88B8)
PID Period	0 to 255	1	116 (0x74)
Error Band	0 to 255	1	10 (0x0A)
Integral Clear Counter	0 to 65,535	2	1000 (0x03E8)
Checksum	0 to 255	1	213 (0xD5)

**Notes:**