SBS™
Technologies, Inc.
Modular I/O

IP-Stepper

Dual Stepper Motor
Motion Controller
IndustryPack®

User Manual

Manual Revision: 6 10/27/1999
Hardware Revision: B

IP-Stepper
Dual Stepper Motor

Motion Controller
IndustryPack®

SBS Technologies Modular I/O
181 Constitution Drive
Menlo Park, CA 94025
(650) 327-1200
(650) 327-3808 FAX
http://www.sbs-mio.com

SBS Technologies, Inc. Modular I/O reserves the right to make any changes in the devices or device specifications contained herein at any time and without notice. Customers are advised to verify all information contained in this document.

The electronic equipment described herein generates, uses and may radiate radio frequency energy, which can cause radio interference. SBS Technologies, Inc. Modular I/O assumes no liability for any damages caused by such interference.

SBS Technologies, Inc. Modular I/O's products are not authorized for use as critical components in medical applications such as life support equipment, without the express consent of the president of SBS Technologies, Inc. Modular I/O.

This product has been designed to operate with IndustryPack, PC•MIP or PMC modules or carriers and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage. SBS Technologies, Inc. Modular I/O assumes no liability for any damages caused by such

Table of Contents

List of Figures

Product Description

The IP-Stepper provides two independent motion control channels for stepper systems. Each channel consists of a Nippon Pulse Motor PCL-240MK programmable high speed pulse generator and a U.S. Digital LS7166 Encoder Interface Chip.

The PCL-240MK Provides several advanced features which are supported on the IP-Stepper, including:

- Fully programmable trapezoidal speed profile.
- Individually controllable acceleration and deceleration.
- Continuous constant speed and high speed operation.
- Constant speed and high speed origin return.
- On the fly speed ramp down.
- Halt diagnostic status register.
- Master/Slave operation for multi-axis synchronization.
- 240K pps maximum speed.
- Programmable scaling.

The LS7166 provides additional quadrature decode capability for programmable way-point triggering, indexing, travel limiting and closed loop motion control. Other additional features provided by the LS7166 include:

- Pre-loadable 24 bit Up/Down counter.
- Programmable pre-scalar.
- 24 Bit comparator Register.
- Overflow, Under flow and Match flags.
- Latched counter outputs.

With the marriage of these two ICs in each channel, IP-Stepper can implement powerful aggregate features:

- Programmable "S" curve speed profiles.
- Externally triggerable capture channel configurable as:
    - Gated qualifier for A & B channel inputs or
    - Counter reset or
    - Output latch transfer clock or
    - Counter pre-load or
    - interrupt source.
- Two channel master/slave synchronized movement.
- Immediate emergency stop execution.
- Interrupts generated for Emergency Stop, travel limits, counter overflow.
- Interrupt masking.
- Software selectable counter input.
- Opto isolated inputs configurable for either differential, current loop or TTL signals.
- Software selectable input polarity.
- Programmable interrupt vector.
- Simplified memory mapped register access.
- Fused +5 volt power outputs for driver biasing.

Simplified Block Diagram

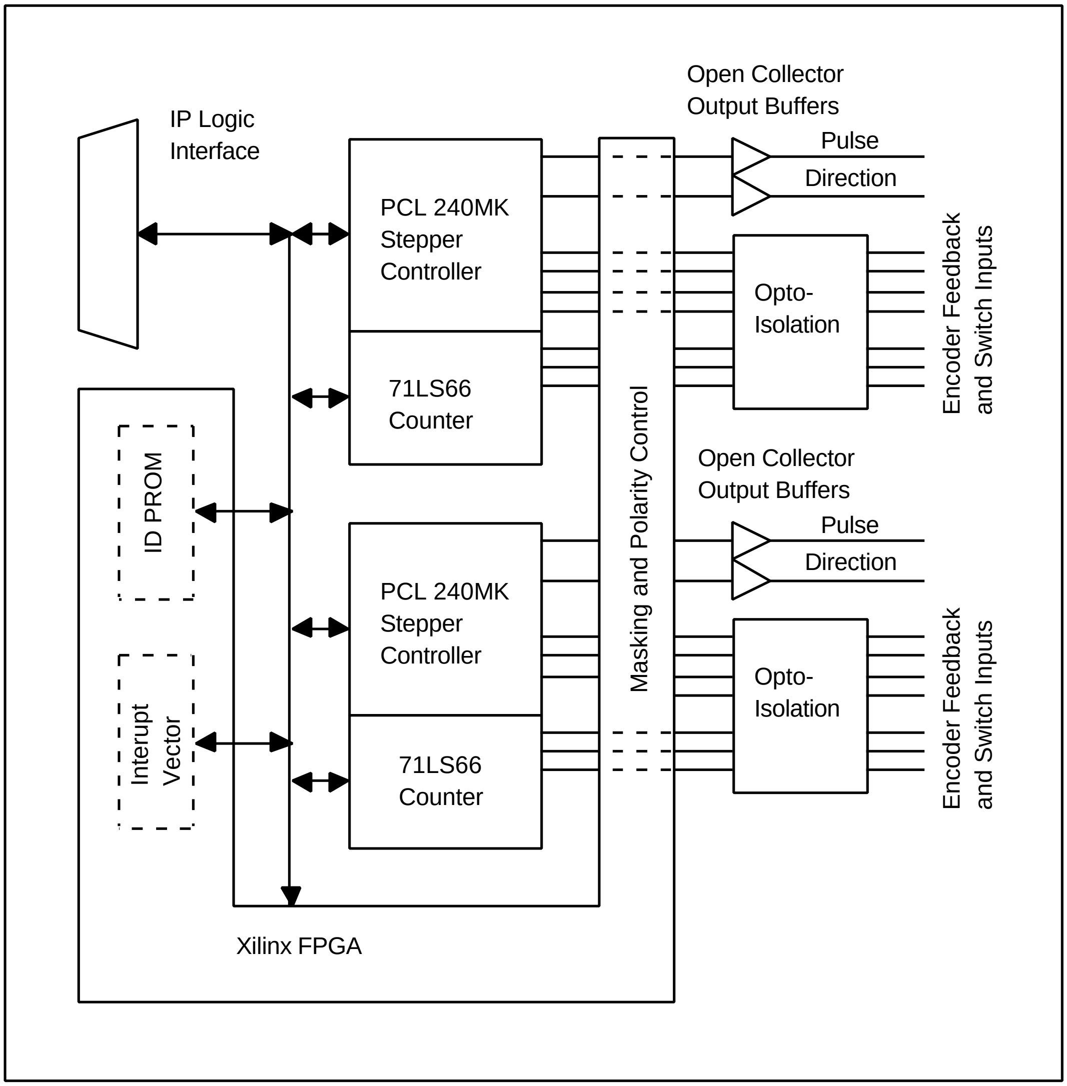


Figure 1 Block Diagram

Applications Guide

The IP-Stepper provides sophisticated and powerful stepper control capabilities. The following information is provided for quick reference only. Complete factory data sheets are provided with the engineering kit which is strongly recommended for first time buyers.

Motion control systems are potentially dangerous. Be sure carefully check signal phasing on the drive signals and encoder feedback. Also, verify the interrupt masking, polarity and motor registers before issuing start commands.

Programming

IP-Stepper integrates one PCL-240MK pulse generator and one LS7166 per channel. All of the PCL-240MK registers are accessible and most of the functions are supported. The hardware way-point inputs SD+ & SD- are not implemented but, both the PCL-240MK and the LS7166 counters generate programmable way-point interrupts. The PCL-240MK control and data registers and the LS7166 counter control and data registers are mapped into IP-I/O space memory locations. System interrupts, interrupt masking, polarity, sync and ID PROM information are all incorporated on the Xilinx FPGA.. The FPGA registers are assessable as IP-I/O space. See the addressing tables for your particular system.

GENERAL

Signals with names having a * suffix are called ACTIVE LOW signals and are TRUE when they are at a 0 logic level which is nominally less than 0.8 Volts. They are FALSE at a 1 logic level which is nominally 1.4 volts or greater. Signals without the suffix are called ACTIVE HIGH signals and their TRUE and FALSE definitions are the OPPOSITE of the ACTIVE LOW definition. For example TERM1* would be set TRUE by writing a 1 to it's bit which would measure as over 1.4 volts on the IP STEPPER board circuitry.

The phrase "Logic Interface" refers to the Electrical/Logical interface between an Industry Pack board like the IP STEPPER and the carrier board on which it is installed.

The phrase "I/O Interface" refers to the OTHER connector on an Industry Pack that carries signals, special power, etc. to the Industry Pack via the carrier board.

All software programmable IP STEPPER board registers should be reinitialized following power up or a Logic Interface reset as the contents are reset to zeros.

Control register bits shown with the same names for both reading and writing have "read back" capability. Other bits have either read or write capability only or a read signal that is functionally related to the write signal of the same bit.

REGARDING PCL-240MK MOTOR CONTROLLER CHIP

Registers with more than 8 bits must be written as though they are 24 bits when using the recommended "collective" access mode. The value of the extra bits beyond the registers specified width is irrelevant. In addition, since the PCL-240MK transfers the value when the least significant byte is written, it must be written LAST. Therefore, write the data in the order of most significant to least significant byte.

It is recommended that "extension" mode be used rather than "standard mode.

The values written to the Start Stop command, Operation Mode Select command and Output Mode Select command can be read back in the R17: Extension Status Buffer. The Register Select command value written can NOT be read back. All other writeable registers can be read back.

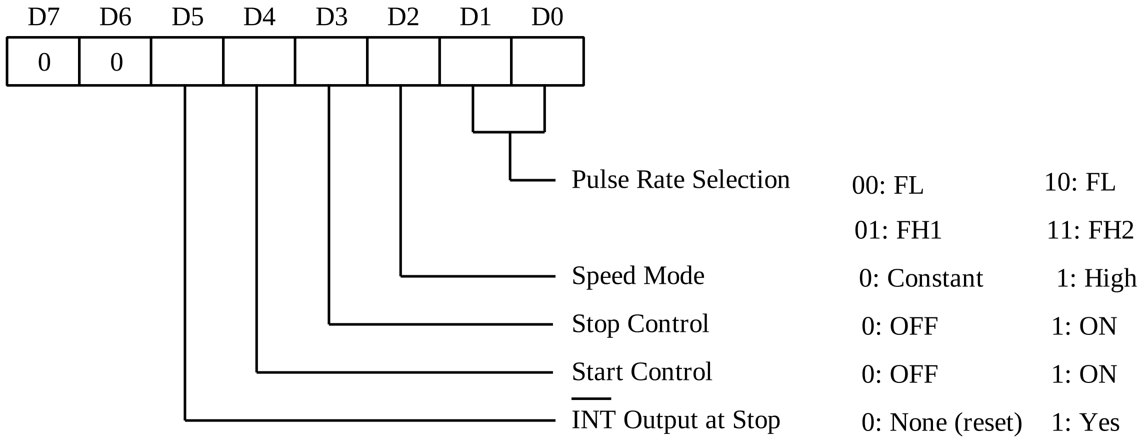
See NIPPON PULSE MOTOR corp. PCL240**MK** data sheet for programming information. Note that you must have the MK data sheet as the AK is a different part.

PCL-240MK Register set summary:

R No.	**Description**	**Bit Length**	**R/W**
S	Start Stop Commands	8	R/W
M	Operation Mode Select Command	8	R/W
R	Register Select Command	8	R/W
O	Output Mode Command	8	R/W
R0	Down counter	24	R/W
R1	FL register	13	R/W
R2	RH1 register	13	R/W
R3	RH2 register	13	R/W
R4	Acceleration rate register	14	R/W
R5	Deceleration rate register	14	R/W
R6	Ramping-down point register	20	R/W
R7	Multiplication register	16	R/W
R10	Current position counter	24	R/W
R11	Current speed monitor	13	R
R12	Extension mode register	16	R/W
R13	Extension mode register	24	R/W
R16	Command buffer monitor	24	R
R17	Extension status buffer	16	R

Figure 2 PCL-240MK register Summary

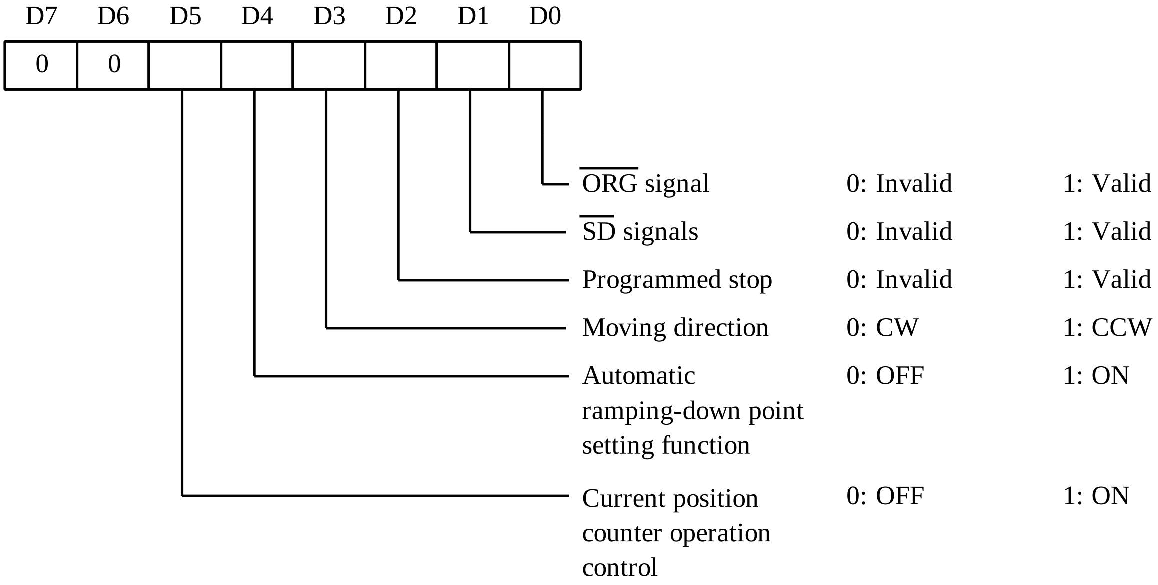
Start-Stop Command Register


D7 D6 D5 D4 D3 D2 D1 D0
0 0
Pulse Rate Selection 00: FL 10: FL
01: FH1 11: FH2
Speed Mode 0: Constant 1: High
Stop Control 0: OFF 1: ON
Start Control 0: OFF 1: ON
$\overline{\text{INT}}$ Output at Stop 0: None (reset) 1: Yes

D7	D6	D5	D4	D3	D2	Dl	D0	Description
0	0	0	1	0	0	0	0	FL-based constant-speed operation (with no INT signal output) The PCL-240MK operates at a speed set in FL register R1.
0	0	1	1	0	0	0	0	FL-based constant-speed operation (with the INT signal output)
0	0	0	1	0	0	0	1	FH1-based constant-speed operation (with no INT signal output) The PCL-240MK operates at a speed set in FH1 register R2.
0	0	0	1	0	0	1	1	FH2-based constant-speed operation (with no INT signal output) The PCL-240MK operates at a speed set in FH2 register R3.
0	0	0	1	0	1	0	1	FH1-based high-speed operation (with no INT signal output) The PCL-240MK starts at a speed set in the FL register, then ramps up to a speed set in the FHI register.
0	0	0	1	0	1	1	1	FH2-based high-speed operation (with no INT signal output) The PCL-240MK starts at a speed set in the FL register, then ramps up to a speed set in the FH2 register.
0	0	0	1	0	1	0	0	Ramping down on the way. The PCL-240MK ramps down from a speed set in the FHI or FH2 register to a speed set in the FL register.
0	0	0	1	1	1	1	1	Deceleration stop. The PCL-240MK ramps down from a speed set in the FHI or FH2 register and stops when the speed reaches what is set in the FL register is reached. (A reset command is required after stop.)
0	0	1	0	1	0	0	0	Immediate stop (with the INT signal output)
0	0	0	0	1	0	0	0	Immediate stop (reset command) (INT output reset)

Figure 3 Example of Stop Start Commands

Operation Mode Select Command Register



D7	D6	D5	D4	D3	D2	D1	D0
0	0						

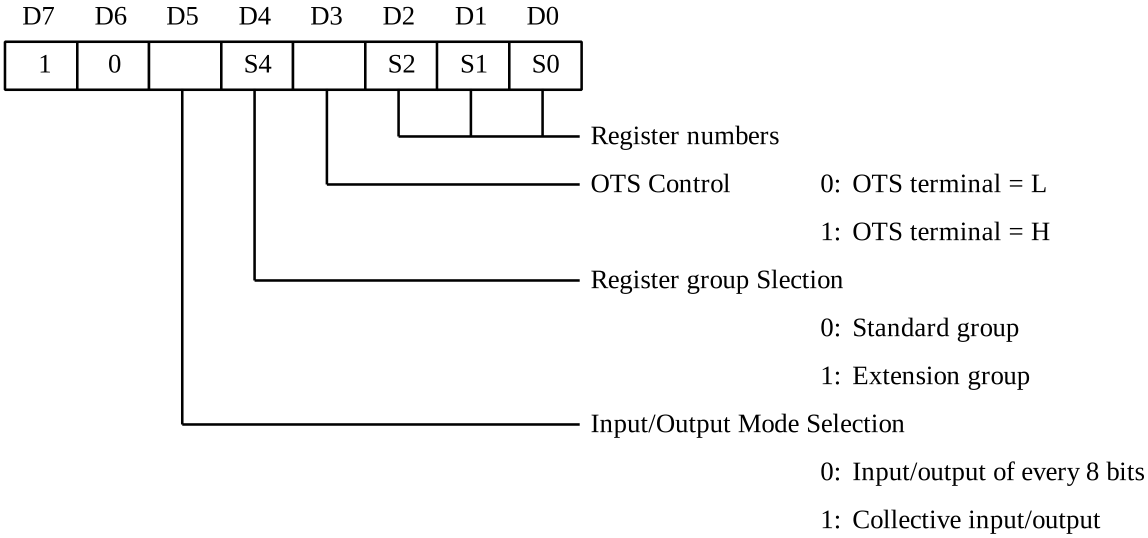
$\overline{\text{ORG}}$ signal	0: Invalid	1: Valid
$\overline{\text{SD}}$ signals	0: Invalid	1: Valid
Programmed stop	0: Invalid	1: Valid
Moving direction	0: CW	1: CCW
Automatic ramping-down point setting function	0: OFF	1: ON
Current position counter operation control	0: OFF	1: ON

Example of Operation Mode Select Command

D7	D6	D5	D4	D3	D2	Dl	D0	Description
0	1	X	X	X	X	X	0	The ORG terminal at low level does not stop pulse output
0	1	X	X	X	X	X	1	The low-level ORG signal to pin 40 stops pulse output.
0	1	X	X	X	X	0	X	The low-level SD signal to pin 23 or 22 does not effect ramping-down.
0	1	X	X	X	X	1	X	The low-level SD signal to pin 23 or 22 effects ramping down.
0	1	X	X	X	0	X	X	Pulse output does not stop with RO=0.
0	1	X	X	X	1	X	X	Pulse output stops with RO=0.
0	1	X	X	0	X	X	X	Moving in CW direction.
0	1	X	X	1	X	X	X	Moving in CCW direction.
0	1	X	0	X	X	X	X	A ramping-down point is to be written in R6.
0	1	X	1	X	X	X	X	A rarnping-down point is to be set automatically.
0	1	0	X	X	X	X	X	The current position counter is not operated.
0	1	1	X	X	X	X	X	The current position counter is operated.
0	1	X	X	X	0	0	0	Manual mode ORG and SD are invalid
0	1	X	X	X	0	X	1	Origin return mode.
0	1	X	X	X	1	1	1	Programmed operation mode ORG and SD are valid also.

Note: Mark X in the above table may be either 0 or 1.

Register Select Command Register



D7	D6	D5	D4	D3	D2	D1	D0
1	0		S4		S2	S1	S0

Register numbers

OTS Control 0: OTS terminal = L

1: OTS terminal = H

Register group Slection

0: Standard group

1: Extension group

Input/Output Mode Selection

0: Input/output of every 8 bits

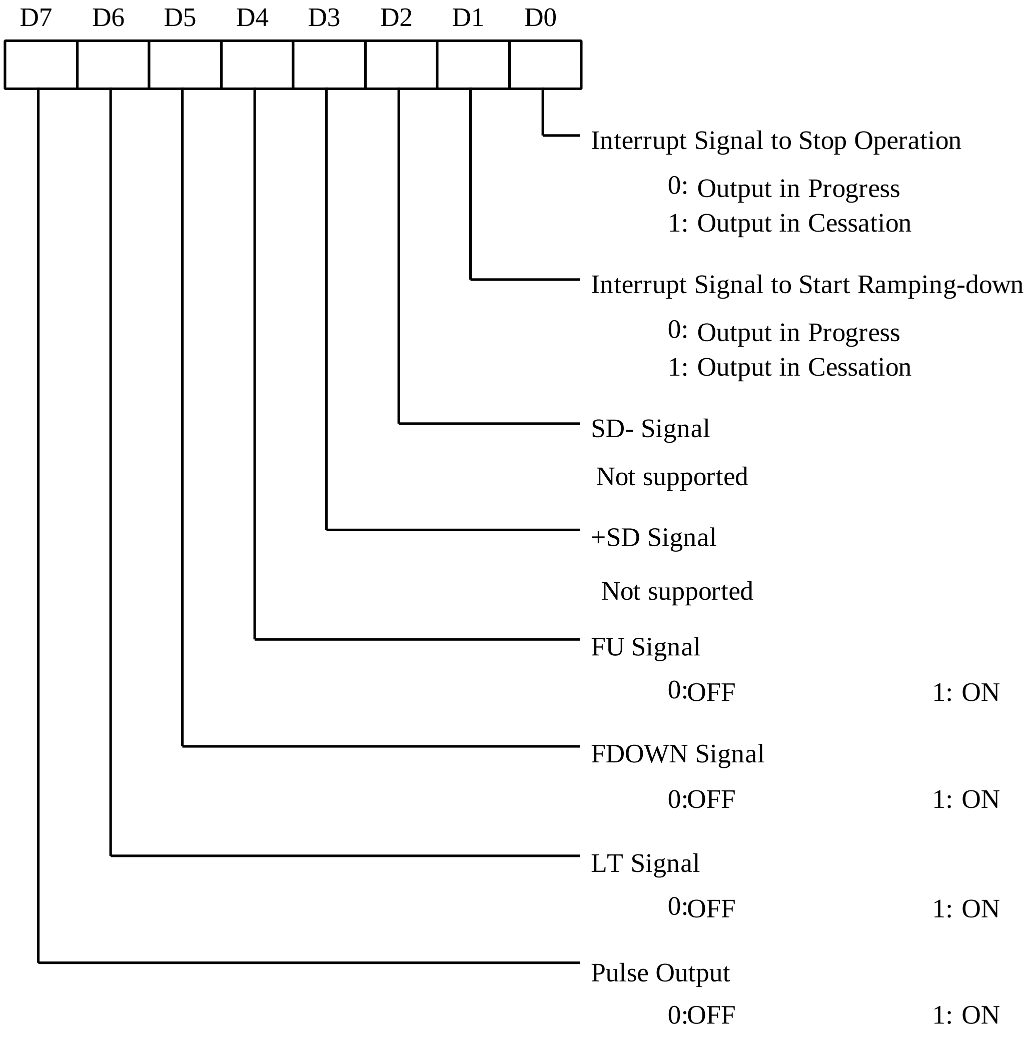
1: Collective input/output

D7	D6	D5	D4	D3	D2	Dl	D0	Description
1	0	X	0	X	0	0	0	Selects register R0.
1	0	X	0	X	0	0	1	Selects register R1.
1	0	X	0	X	0	1	0	Selects register R2.
1	0	X	0	X	0	I	1	Selects register R3.
1	0	X	0	X	1	0	0	Selects register R4.
1	0	X	0	X	1	0	1	Selects register R5.
1	0	X	0	X	1	1	0	Selects register R6.
1	0	X	0	X	1	1	1	Selects register R7.
1	0	X	1	X	0	0	0	Selects register R10.
1	0	X	1	X	0	0	1	Selects register R11.
1	0	X	1	X	0	1	0	Selects register R12.
1	0	X	1	X	0	1	1	Selects register R13.
1	0	X	1	X	1	1	0	Selects register R16.
I	0	X	1	X	1	1	1	Selects register R17.

Figure 4 Example of Register Select Commands

R17: 16-bit Status Buffer Bits D0..D7

The R17 allows the operator to monitor signal status as follows.



D7 D6 D5 D4 D3 D2 D1 D0

D0 – Interrupt Signal to Stop Operation
0: Output in Progress
1: Output in Cessation

D1 – Interrupt Signal to Start Ramping-down
0: Output in Progress
1: Output in Cessation

D2 – SD- Signal
Not supported

D3 – +SD Signal
Not supported

D4 – FU Signal
0:OFF 1: ON

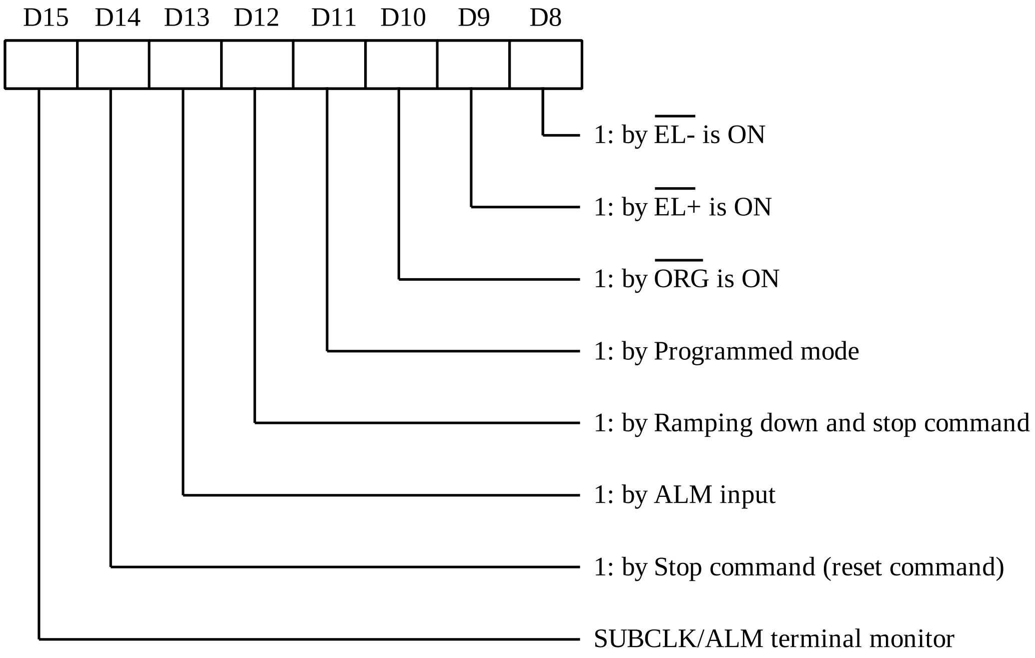
D5 – FDOWN Signal
0:OFF 1: ON

D6 – LT Signal
0:OFF 1: ON

D7 – Pulse Output
0:OFF 1: ON

R17: 16-bit Status Buffer Bits D8..D15

During the extension status, you can and bits 8 ~ 14 to get information of causes why pulse input is stopped.


D15
D14
D13
D12
D11
D10
D9
D8
1: by $\overline{\text{EL-}}$ is ON
1: by $\overline{\text{EL+}}$ is ON
1: by $\overline{\text{ORG}}$ is ON
1: by Programmed mode
1: by Ramping down and stop command
1: by ALM input
1: by Stop command (reset command)
SUBCLK/ALM terminal monitor

XILINX INTERFACE SOFTWARE REGISTER DEFINITIONS

NOTE: All bits are read/write [R/W] unless shown with a **-** in the map and a [R] for read only or a [W] for write only in their definitions.

INTERRUPT REGISTER BIT DEFINITIONS V0.2

Data Bit #	7	6	5	4	3	2	1	0
Write Action	X	X	X	ESTP_ICLR	ESTP_IMSK*	CT_ICLR	CT_IMSK*	MT_IMSK*
Read Action	X	X	X	ESTP_Ipending	ESTP_IMSK*	CT_Ipending	CT_IMSK*	MT_IMSK*

MT_IMSK* [R/W] D0

When set to "0", masks (disables) interrupts from the channel's PCL-240MK motor controller. Has no effect upon the interrupt's generation, rather, it simply prevents the presence of an interrupt being noticed by the interrupt logic on the Industry Pack. Thus enabling (with a "1") when an interrupt has already occurred will immediately precipitate an interrupt to the IP Logic Interface. This bit resets and powers on to a value of "0" which MASKS the interrupt. The interrupt is cleared via a register bit in the PCL-240MK.

CT_IMSK* [R/W] D1

When set to "0", masks (disables) interrupts from the channel's LS7166 Counter with the same behavior as described for MT_IMSK*. The interrupt is cleared via the CT_ICLR bit in this register.

CT_IPENDING [R] D2

CT_ICLR [W] D2

Reading a value of "1" from this bit indicates the presence of a pending LS7166 Counter interrupt. A "0" means no interrupt is pending. Writing a value of "1" to this bit will CLEAR any pending interrupt and prevent any further interrupts until the bit is returned to a "0." Normally a "0" is written to this bit except when clearing an interrupt. The power up default is to a "0", so interrupts are ENABLED, but masked via CT_IMSK*.

ESTP_IMSK* [R/W] D3

When set to "0", masks (disables) interrupts from the channel's Emergency Stop switch input with the same behavior as described for MT_IMSK*. The interrupt is cleared via the ESTP_ICLR bit in this register.

ESTP_IPENDING [R] D4

ESTP_ICLR [W] D4

Reading a value of "1" from this bit indicates the presence of a pending Emergency Stop switch interrupt. A "0" means no interrupt is pending. Writing a value of "1" to this bit will CLEAR any pending interrupt <u>and prevent any further interrupts until the bit is returned to a "0."</u> Normally a "0" is written to this bit except when clearing an interrupt. The power up default is to a "0", so interrupts are ENABLED, but masked via ESTP_IMSK*.

D7 - D5 Not Assigned.

POLARITY REGISTER BIT DEFINITIONS V0.2

Data Bit #	7	6	5	4	3	2	1	0
Write Action	X	STP_Invert	Z_Invert	WP_INVERT	CT_Enable	CP_Invert	CH B Invert	CH A Invert
Read Action	X	STP_Invert	Z_Invert	WP_INVERT	CT_Enable	CP_Invert	CH B Invert	CH A Invert

CH A INVERT [R/W] D0

When set to a "1", the logic levels of the signal to channel A of the channel's LS7166 counter will be inverted prior to reaching the counter. Power up and reset state is NOT inverted.

CH B INVERT [R/W] D1

When set to a "1", the logic levels of the signal to channel B of the channel's LS7166 counter will be inverted prior to reaching the counter. Power up and reset state is NOT inverted.

CP INVERT [R/W] D2

When set to a "1", the logic levels of the CAPTURE signal to the channel's LS7166 counter will be inverted prior to reaching the counter. Power up and reset state is NOT inverted.

CT ENABLE [R/W] D3

When set to a "1", the CAPTURE signal is enabled to affect the channel's LS7166 counter as configured. Power up and reset state is NOT enabled.

NOTE that the counter must also be programmed compatibly with the function selected for proper operation to take

place. This bit resets and powers on to a value of "0".

WP INVERT [R/W] D4

When set to a "1", the logic levels of the channel's way-point signals will be inverted prior to reaching the PCL-240MK. Way-point signals should be configured to be active low ("0" when true) at the PCL-240MK. Power up and reset state is NOT inverted.

Z INVERT [R/W] D5

When set to a "1", the logic level of the channel's Z signal will be inverted prior to reaching the PCL-240MK. Power up and reset state is NOT inverted.

STP INVERT [R/W] D6

When set to a "1", the logic level of the channel's Emergency stop signal will be inverted prior to reaching the Interrupt Logic. The Emergency Stop signal should be configured to be active low ("0" when true) to the interrupt logic. Power up and reset state is NOT inverted.

D7 Not Assigned.

SOURCE REGISTER BIT DEFINITIONS

Data Bit #	7	6	5	4	3	2	1	0
Write Action	X	LDSL1	LDSL0	ABSL1	ABSL0	CBSL0	CASL1	CASL0
Read Action	X	LDSL1	LDSL0	ABSL1	ABSL0	CBSL0	CASL1	CASL0

CASL0 - CASl1 [R/W] D0 - D1

These bits select the source for the Channel A input of the channel's LS7166 counter. Power up and reset values are "0." When using PCL240 step output as input, invert Channel A in the polarity register.

CASL1	**CASL0**	**CHANNEL A SOURCE**
0	0	X input
0	1	PCL240 Step pulse output *(invert Channel A)
1	0	Z signal after optional inversion
1	1	LOGICAL 1, No signal

CBSL0 [R/W] D2

This bit selects the source for the Channel B input of the channel's LS7166 counter. Power up and reset value is "0."

CBSL0	**CHANNEL B SOURCE**
0	Y input
1	PCL240 motor Direction output *(invert Channel B)

ABSL0 - ABSl1 [R/W] D3 - D4

These bits select the source for the ABEN*/RESET* input of the channel's LS7166 counter. Power up and reset values are "0." When using PCL240 motor direction output, set the bit to invert Channel B in the polarity register.

ABSL1	**ABSL0**	**ABEN*/RESET* SOURCE**
0	0	Capture signal after optional inversion
0	1	Stop signal after optional inversion
1	0	Logical OR of above two signals
1	1	LOGICAL 1, No signal

LDSL0 - LDSl1 [R/W] D5 - D6

These bits select the source for the LDCNT*/LDLATCH* input of the channel's LS7166 counter. Power up and reset values are "0."

LDSL1	LDSL0	LDCNT*/LDLATCH* SOURCE
0	0	Capture signal after optional inversion
0	1	Stop signal after optional inversion
1	0	Logical OR of above two signals
1	1	LOGICAL 1, No signal

D7 Not Assigned.

SYNC CONTROL REGISTER

Data Bit #	7	6	5	4	3	2	1	0
Write Action	X	X	X	X	X	X	SUBCLK_EN	SYNC_EN
Read Action	X	X	X	X	X	X	SUBCLK_EN	SYNC_EN

SYNC_EN [R/W] D0

When set to "1" connects the SYNO signal of the channel 1 PCL-240MK to the SYNI input of the channel 2 PCL-240MK. This enables SYNCHRONOUS operation of the two motor controller chips. Power up and reset state is NOT SYNC enabled.

SUBCLK_EN [R/W] D1

When set to "1" connects the SYNO signal of the channel 1 PCL-240MK to the SUBCLK/ALM input of the channel 2 PCL-240MK. This enables   SUBCLK operation of the channel 2 motor controller chip. Power up and reset state is NOT SUBCLK enabled.

NOTE: Only one of the above two options should be enabled at on time.

D7 - D2 Not Assigned .

INTERRUPT VECTOR

Data Bit #	7	6	5	4	3	2	1	0
Write	V7	V6	V5	V4	V3	X	X	X
Read	V7	V6	V5	V4	V3	C2	C1	C0

The IP STEPPER stores any desired value five bit interrupt vector in this register. On a Write cycle, only D7 - D3 are stored. On a read cycle, the previously stored value of D7 - D3 is read back upon the same bits as they were written. In addition, C2 through C0 encode the value of the most significant interrupt source with a pending interrupt. (Channel 2 is the most significant, Channel 1 the least.) This full eight bit vector will be returned as the data of an interrupt vector request cycle (IP Logic Interface INTSEL* true).

VECTOR PRIORITY ENCODING

C2, C1, C0	Interrupt Source
000	E-Stop channel 1 (lowest priority)
001	E-Stop channel 2
010	LS7166 Counter channel 1
011	LS7166 Counter channel 2
100	PCL-240MK channel 1
101	PCL-240MK channel 2 (highest priority)

LS7166 24 BIT COUNTER CHIP REGISTERS

The Data and Control registers in the IP address map provide direct access to the Data and Control registers of the LS7166 counters. The Xilinx FPGA decodes and generates CS* RD* WR* and C/D* signals for the LS7166. Please review the LS7166 register specifications for details of the register functions.

For proper operation the counter's output pin 16 (BW or COMP)should be programmed as active low in the counter's Output Control Register bits 4 & 5. Use 00 for BW* or 11 for COMP*.

None of this device's writeable registers have read back capability. See LSI corp. LS7166 data sheet for programming information.

Be sure to reset the PR/OL address pointer when accessing the Preset and Output latch registers by writing a 1 to bit D0 of the Master control register. Failure to do so might result in garbled data if errant reads or writes are made to these registers.

The two control signals to the LS7166 can provide the following actions depending upon internal register programming of the LS7166.

LS7166 CONTROL FUNCTIONS

LS7166 Input control register bit 4	ABEN*/RESET* TRUE INPUT ACTION
0	Reset the counter
1	Gate the counter's inputs

LS7166 Input control register bit 5	LDCNT*/LDLATCH* TRUE INPUT ACTION
0	Load counter from preset compare register
0	Cause interrupt if 0 in preset register.
1	Transfer counter's count to output latch

Interfacing

The output stepper pulse and direction signals from IP-Stepper are driven by LSTTL open collector buffers. This provides both ruggedness and adaptability. In the standard configuration the output drivers are pulled up to the on board 5 volt supply. This Gives the user TTL compatibility right out of the box. For different power requirements like, 12 or 24 VDC, external pull ups can be used. Additionally, the encoder feedback signals and switch inputs are optio-isolated for system protection and noise immunity. The encoder signals are buffered by special high speed optocouplers for high speed and high resolution quadrature inputs. All optocoupler inputs can be user configured for TTL, Differential or current loop signal levels.

Wiring Options

Normally isolated resistor type resistor packs are installed for current limiting of the optically coupled current loop inputs. Changing these to bussed type resistor packs with +5V applied to the I/O connector's pin corresponding to the resistor pack's pin 1, provides local current sourcing to input signal + inputs. Thus, channel testing can be done by providing only current sinking on the - pins. This provides much easier testing than full current loop interfaces on all pins. Connect as follows.

SPECIAL TESTING WIRING OPTIONS

MOTOR CHANNEL	SIGNALS	I/O PIN to connect to +5 volts
1	X,Y,Z,CAPTURE	2
1	WAY-POINTS, STOP,	4
2	X,Y,Z,CAPTURE	27
2	WAY-POINTS, STOP,	29

VME Addressing

Standard Word I/O Accessing

There are TWO motor control channels each of which has its own PCL240 motor controller and LS7166 counter chips. BOTH channels SHARE the Interrupt Vector and Sync Control registers.

Byte Addresses are shown (A0 is LSB), but board is accessed on 16 bit word boundaries. Values are shown in **HEX**.

MOTOR 1 CONTROL / STATUS	Base + 01
MOTOR 1 DATA BITS 0-7	Base + 03
MOTOR 1 DATA BITS 8-15	Base + 05
MOTOR 1 DATA BITS 16-23	Base + 07
MOTOR 2 CONTROL / STATUS	Base + 09
MOTOR 2 DATA BITS 0-7	Base + 0B
MOTOR 2 DATA BITS 8-15	Base + 0D
MOTOR 2 DATA BITS 16-23	Base + 0F
COUNTER 1 DATA	Base + 11
COUNTER 1 CONTROL	Base + 13
COUNTER 2 DATA	Base + 19
COUNTER 2 CONTROL	Base + 1B
CHANNEL 1 INTERRUPT REGISTER	Base + 21
CHANNEL 1 POLARITY REGISTER	Base + 25
CHANNEL 1 SOURCE REGISTER	Base + 29
CHANNEL 2 INTERRUPT REGISTER	Base + 2D
CHANNEL 2 POLARITY REGISTER	Base + 31
CHANNEL 2 SOURCE REGISTER	Base + 35
SYNC CONTROL REGISTER	Base + 39
INTERRUPT VECTOR REGISTER	Base + 3D

Figure 5   VME Word Access

NuBus Addressing

The formula for conversion from VME to NuBus is:

NuBus address = (VME address x 2) - 1

Standard Word I/O Accessing

There are TWO motor control channels each of which has its own PCL-240MK motor controller and LS7166 counter chips. BOTH channels SHARE the Interrupt Vector and Sync Control registers.

Byte Addresses are shown (A0 is LSB), but board is accessed on 16 bit word boundaries. Values are shown in **HEX**.

MOTOR 1 CONTROL / STATUS	Base + 01
MOTOR 1 DATA BITS 0-7	Base + 05
MOTOR 1 DATA BITS 8-15	Base + 09
MOTOR 1 DATA BITS 16-23	Base + 0D
MOTOR 2 CONTROL / STATUS	Base + 11
MOTOR 2 DATA BITS 0-7	Base + 15
MOTOR 2 DATA BITS 8-15	Base + 19
MOTOR 2 DATA BITS 16-23	Base + 1D
COUNTER 1 DATA	Base + 21
COUNTER 1 CONTROL	Base + 25
COUNTER 2 DATA	Base + 31
COUNTER 2 CONTROL	Base + 35
CHANNEL 1 INTERRUPT REGISTER	Base + 41
CHANNEL 1 POLARITY REGISTER	Base + 41
CHANNEL 1 SOURCE REGISTER	Base + 49
CHANNEL 2 INTERRUPT REGISTER	Base + 51
CHANNEL 2 POLARITY REGISTER	Base + 59
CHANNEL 2 SOURCE REGISTER	Base + 61
SYNC CONTROL REGISTER	Base + 71
INTERRUPT VECTOR REGISTER	Base + 79

Figure 6 NuBus Word Access

ISA (IBM PC-AT) Addressing

Standard Word I/O Accessing

There are TWO motor control channels each of which has its own PCL-240MK motor controller and LS7166 counter chips. BOTH channels SHARE the Interrupt Vector and Sync Control registers.

Byte Addresses are shown (A0 is LSB), but board is accessed on 16 bit word boundaries. Values are shown in **HEX**.

MOTOR 1 CONTROL / STATUS	Base + 00
MOTOR 1 DATA BITS 0-7	Base + 02
MOTOR 1 DATA BITS 8-15	Base + 04
MOTOR 1 DATA BITS 16-23	Base + 06
MOTOR 2 CONTROL / STATUS	Base + 08
MOTOR 2 DATA BITS 0-7	Base + 0A
MOTOR 2 DATA BITS 8-15	Base + 0C
MOTOR 2 DATA BITS 16-23	Base + 0E
COUNTER 1 DATA	Base + 10
COUNTER 1 CONTROL	Base + 12
COUNTER 2 DATA	Base + 18
COUNTER 2 CONTROL	Base + 1A
CHANNEL 1 INTERRUPT REGISTER	Base + 20
CHANNEL 1 POLARITY REGISTER	Base + 24
CHANNEL 1 SOURCE REGISTER	Base + 28
CHANNEL 2 INTERRUPT REGISTER	Base + 2C
CHANNEL 2 POLARITY REGISTER	Base + 30
CHANNEL 2 SOURCE REGISTER	Base + 34
SYNC CONTROL REGISTER	Base + 38
INTERRUPT VECTOR REGISTER	Base + 3C

Figure 7   ISA Word Access

Getting Started

This section is intended to get first time user s started with IP-Stepper, quickly. The following steps will verify IP-Stepper operation and installation in a fewminutes. The procedure uses a low level debugger to program the IP. The debugger will depend on your particular host system. In the example, channel 1 is programmed to execute a constant speed move. And, the Emergency Stop function is verified and its inturrupt is verified and reset. An LED is used to indicate pulse output. This simple load reduces the posibility of power module wiring errors and dangerous mechanical system malfunctions during user orientation. For convience, the LED and resistor are supplied in the IP-Stepper Engineering Kit. A piece of wire (or paper clip) can as the Emergency Stop switch.. This procedure has been verified on several platforms including: GreenSpring VIPC 310 using OS-9 ROMBUG, Motorola MVME 162 running Motorloa's onboard debugger, 162-Bug, and IBM ATs running MS-DOS DEBUG. Please take the time to try this experiment if you are unsure of a new system.

1) In the IP-Stepper Engineering Kit you will find:
    i) an LED,
    ii) a 330 ? resistor,
    iii) a terminal block,
    iv) and a 50 pin ribbon cable.

2) Power down your system and install IP-Stepper in a vacant IP carrier slot. Connect the 50 pin ribbon cable from your carrier to the terminal block. Be sure to observe the connector keying.

3) Install the LED and 330 ? resistor on the STEP_OUT1 output. Connect the LED anode (long lead) to STEP_OUT1, pin 20 on the terminal block. Connect the cathode (short lead) to GND, pin 23. The resistor should be connected between FVCC pin 24 and STEP_OUT1, pin 20.

4) Use a wire to simulate an Emergency Stop switch. Connect one end of the wire to the EM_STOP1* input, pin 25 of the terminal block. Leave the other end free.

5) Power up the system.

6) Determine the correct IP base address for your carrier slot. See your carrier manual for details. For a quick check, read the ID PROM information at IP base address + $80 (hex).

7) Enter the following bytes using your debugger. Be sure to enter register data bytes in the order shown below. All 24 bits of the PCL240MK registers MUST be loaded each time. And they MUST be loaded least significant byte last.

VME	AT/ISA	DATA	READ/	
OFFSET	OFFSET	(HEX)	WRITE	COMMENTS
01	00	08	W	Clear Stop/Start register
01	00	74	W	Set Op Mode register
01	00	C8	W	Set Output Mode register
01	00	A0	W	Select R0, Down Counter
07	06	00	W	R0 Data [D23..D16]
05	04	FF	W	R0 Data [D15..D8]
03	02	FF	W	R0 Data [D7..D0]
01	00	A1	W	Select R1, FL Speed
07	06	00	W	R1 Data [D23..D16]
05	04	00	W	R1 Data [D15..D8]
03	02	01	W	R1 Data [D7..D0]
01	00	A7	W	Select R7, Speed Multiplier
07	06	00	W	R7 Data [D23..D16]
05	04	01	W	R7 Data [D15..D8]
03	02	E8	W	R7 Data [D7..D0]
01	00	B2	W	Select R12, Ext. Mode 1
07	06	00	W	R12 Data [D23..D16]
05	04	00	W	R12 Data [D15..D8]
03	02	40	W	R12 Data [D7..D0]
01	00	BB	W	Select R13, Ext. Mode 2
07	06	13	W	R13 Data [D23..D16]
05	04	20	W	R13 Data [D15..D8]
03	02	00	W	R13 Data [D7..D0]
01	01	10	W	Start command

Figure 8 Programming Example

8) The LED should start blinking at about one pulse per second.

9) Next test the Emergency Stop function by shorting the wire to ground. Touch the free end of the wire to pin 23 of the terminal block. The LED should stop blinking.

10) To verify and clear the Emergency Stop interrupt register bits, peek and poke the following bytes using your debugger. Note that the ESTP_ICLR bit MUST be Re-Enabled after clearing the ESTP_Ipending bit. This is independent of the interrupt masking.

VME	AT/ISA	DATA	READ/	
OFFSET	OFFSET	(Binary)	WRITE	COMMENTS
21	20	???1 0000	R	Interrupt Register Verify ESTP_Ipending bit [D4]
21	20	???1 0000	W	Clear Emergency Stop. The LED should start blinking again.
21	20	???0 0000	W	Re-Enable Emergency Stop.

Figure 9 Emergency Stop Interrupt Example

PCL-240MK Programming Applications Example:

In the following example the PCL-240MK pulse generator is programmed to execute a trapezoidal velocity trajectory. Using displacement as the independent variable is an obvious practical approach to motion control programming. As if by design, transformation of variables from time domain to displacement domain simplifies register value calculations considerably.

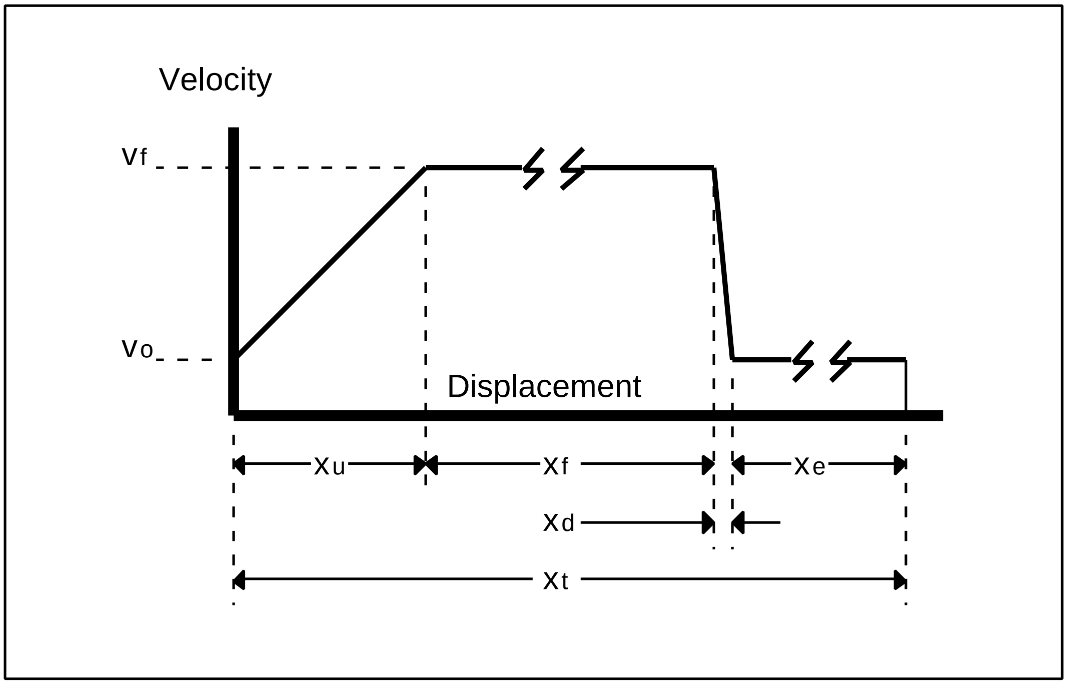


Figure 10 Velocity Trajectory Example

Given parameters:

$$\text{Motor type} = 200 \text{ step / revolution } (1.8^\circ/\text{step})$$
$$x_t = \text{ total displacement} = 10 \text{ revs} = 2000 \text{ pulses}$$
$$x_e = \text{ end displacement (optional)} = 1 \text{ rev} = 200 \text{ pulses}$$
$$v_o = \text{ initial velocity} = 20 \text{ pps}$$
$$v_f = \text{ high speed flight velocity} = 400 \text{ pps}$$
$$t_u = \text{ ramp up time} = 1 \text{ sec}$$
$$t_d = \text{ ramp down time} = 0.1 \text{ sec}$$
$$F_{\text{CLK}} = 4 \text{ Mhz, Standard for IP-Stepper.}$$
$$n = 1 \text{ for ease of calculation}$$

Intermediate variable calculations:

$$a_u = \text{ ramp up acceleration} = \frac{\Delta v}{\Delta t} = \frac{v_f - v_o}{t_u} = \frac{400-20}{1} = 380 \text{ pps}^2$$

$$x_u = \text{ ramp up displacement } = v_o t_u + \tfrac{1}{2} a_u t_u{}^2 = (20)(1) + \tfrac{1}{2}(380)(1)^2 = 210 \text{ pulses}$$

$$a_d = \text{ ramp down acceleration} = \frac{\Delta v}{\Delta t} = \frac{v_o - v_f}{t_d} = \frac{20-400}{0.1} = -3800 \text{ pps}^2$$

$$x_d = \text{ ramp down displacement } = v_f t_d + \tfrac{1}{2} a_d t_d{}^2 = (400)(0.1) - \tfrac{1}{2}(3800)(0.1)^2 = 21 \text{ pulses}$$

$$x_f = \text{ high speed flight dispalcement } = x_t - (x_u + x_d + x_e) = 2000 - (210 + 21 + 200) = 1569 \text{ pulses}$$

Register value calculations:

$$R_0 = 2000 = \$7D0$$

$$R_1 = 20 = \$14$$

$$R_2 = 400 = \$190$$

$$R_4 = \frac{F_{CLK}}{a_u} = \frac{4\text{Mhz}}{380} \approx 10526 = \$291E$$

$$R_5 = \frac{F_{CLK}}{a_d} = \frac{4\text{Mhz}}{3800} \approx 1053 = \$41D$$

$$R_6 = x_d + x_e = 21 + 200 = 221 = \$DD$$

$$R_7 = \frac{F_{CLK}}{n \times 8192} = \frac{4 \text{ Mhz}}{1 \times 8192} = 488 = \$1E8$$

$$S = \$15$$

I/O Pin Wiring

IP-Stepper Pin Assignment. Channel 1, Pins 1..25

PIN NUMBER	MOTOR CHANNEL	PIN NAME	DESCRIPTION / FUNCTION
1	1	**GND**	
2	1	X+	Quadrature input, opto isolated, current in
3	1	X-	Quadrature input, opto isolated, current out
4	1	WPP+	Way-point input, PLUS direction, opto isolated, current in
5	1	WPP-	Way-point input, PLUS direction, opto isolated, current out
6	1	**GND**	
7	1	Y+	Quadrature input, opto isolated, current in
8	1	Y-	Quadrature input, opto isolated, current out
9	1	WPM+	Way-point input, MINUS direction, opto isolated, current in
10	1	WPM-	Way-point input, MINUS direction, opto isolated, current out
11	1	**GND**	
12	1	Z+	Index input, opto isolated, current in
13	1	Z-	Index input, opto isolated, current out
14	1	CAPTURE+	Capture input, opto isolated, current in
15	1	CAPTURE-	Capture input, opto isolated, current out
16	1	**GND**	
17	1	STOP+	Stop input, opto isolated, current in
18	1	STOP-	Stop input, opto isolated, current out
19	1	**GND**	
20	1	STEP	Step output, open collector
21	1	**GND**	
22	1	DIRECTION	Direction output, open collector
23	1	**GND**	
24	1	+5 volts	+5 volts output, fused
25	1	ESTOP*	Emergency Stop input, switch closure to gnd

Figure 11 I/O Pin Assignment Channel 1

IP-Stepper Pin Assignment. Channel 2, Pins 26..50

26	2	**GND**	
27	2	X+	Quadrature input, opto isolated, current in
28	2	X-	Quadrature input, opto isolated, current out
29	2	WPP+	Way-point input, PLUS direction, opto isolated, current in
30	2	WPP-	Way-point input, PLUS direction, opto isolated, current out
31	2	**GND**	
32	2	Y+	Quadrature input, opto isolated, current in
33	2	Y-	Quadrature input, opto isolated, current out
34	2	WPM+	Way-point input, MINUS direction, opto isolated, current in
35	2	WPM-	Way-point input, MINUS direction, opto isolated, current out
36	2	**GND**	
37	2	Z+	Index input, opto isolated, current in
38	2	Z-	Index input, opto isolated, current out
39	2	CAPTURE+	Capture input, opto isolated, current in
40	2	CAPTURE-	Capture input, opto isolated, current out
41	2	**GND**	
42	2	STOP+	Stop input, opto isolated, current in
43	2	STOP-	Stop input, opto isolated, current out
44	2	**GND**	
45	2	STEP	Step output, open collector
46	2	**GND**	
47	2	DIRECTION	Direction output, open collector
48	2	**GND**	
49	2	+5 volts	+5 volts output, fused
50	2	ESTOP*	Emergency Stop input, switch closure to gnd

Figure 12 I/O Pin Assignment Channel 2

IndustryPack Logic Interface Pin Assignment

Figure 13 below gives the pin assignments for the IndustryPack Logic Interface connector on the IP-Stepper. Pins marked n/c below are defined by the specification, but not used on IP-Stepper. See also your User Manual for your IP Carrier board for more information.

GND	GND	1	26
CLK	+5V	2	27
Reset*	R/W*	3	28
D0	IDSel*	4	29
D1	n/c	5	30
D2	MEMSel*	6	31
D3	n/c	7	32
D4	INTSel*	8	33
D5	n/c	9	34
D6	IOSel*	10	35
D7	n/c	11	36
n/c	A1	12	37
n/c	n/c	13	38
n/c	A2	14	39
n/c	n/c	15	40
n/c	A3	16	41
n/c	INTReq0*	17	42
n/c	A4	18	43
n/c	n/c	19	44
n/c	A5	20	45
n/c	n/c	21	46
n/c	n/c	22	47
n/c	Ack*	23	48
+5V	n/c	24	49
GND	GND	25	50

Note 1: The no-connect (n/c) signals above are defined by the IndustryPack Logic Interface Specification, but not used by this IP. See the Specification for more information.

Note 2: The layout of the pin numbers in this table corresponds to the physical placement of pins on the IP connector. Thus this table may be used to easily locate the physical pin corresponding to a desired signal. Pin 1 is marked with a square pad on the IndustryPack.

Figure 13 Logic Interface Pin Assignment

ID PROM

Every IP contains an ID PROM, whose size is at least 32 x 8 bits. The ID PROM aids in software auto configuration and configuration management. The user's software, or a supplied driver, may verify that the device it expects is actually installed at the location it expects, and is nominally functional. The ID PROM contains the manufacturing revision level of the IP. If a driver requires that a particular revision be present, it may check for it directly.

Standard data in the ID PROM on the IP-Stepper is shown in Figure 20 below.
For more information on IP ID PROMs refer to the IndustryPack Logic Interface Specification, available from GreenSpring Computers.

The location of the ID PROM in the host's address space is dependent on which carrier board used. Normally for VMEbus carriers the ID PROM space is directly above the IP's I/O space, or at IP-base + $80. Macintosh drivers use the ID PROM automatically. RM1260 address may be derived from Figure 14 below by multiplying the addresses given by two, then subtracting one. RM1270 addresses may be derived by multiplying the addresses given by two, then adding one.

The ID PROM is equivalent to a Philips (Signetics) 82LS123.

3F 19	(available for user)	
17	CRC	(81)
15	No of bytes used	(0C)
13	Driver ID, high byte	(00)
11	Driver ID, low byte	(01)
0F	reserved	(00)
0D	Revision	(A1)
0B	Model No: IP-Stepper	(40)
09	Manufacturer ID GreenSpring	(F0)
07	ASCII “C”	(43)
05	ASCII “A”	(41)
03	ASCII “P”	(50)
01	ASCII “I”	(49)

Figure 14 ID PROM Data (hex)

Construction and Reliability

IndustryPacks were conceived and engineered for rugged industrial environments. The IP-stepper is constructed out of 0.062 inch thick FR4 V0 material.

The IndustryPack connectors are keyed, shrouded and gold plated on both contacts and receptacles. They are rated at 1 Amp per pin and a minimum of 200 insertion cycles. These connectors make consistent, correct insertion easy and reliable.

The IP is secured to the carrier with four M2 metric stainless steel screws. The heads of the screws are countersunk into the IP. The four screws provide significant protection against shock, vibration, and incomplete insertion. For most applications they are not required.

The IndustryPack provides a low temperature coefficient of 0.89 W/°C for uniform heat. This is based on the temperature coefficient of the base FR4 material of 0.31 W/m-°C, and taking into account the thickness and area of the IP. This coefficient means that if 0.89 Watts is applied uniformly on the component side, that the temperature difference between the component and the solder side is one degree Celsius.

Warranty and Repair

SBS Technologies, Inc. Modular I/O warrants this product to be free from defects in workmanship and materials under normal use and service and in its original, unmodified condition, for a period of one year from the time of purchase. If the product is found to be defective within the terms of this warranty, SBS Technologies, Inc. Modular I/O's sole responsibility shall be to repair, or at SBS Technologies, Inc. Modular I/O's sole option to replace, the defective product. The product must be returned by the original customer, insured, and shipped prepaid to SBS Technologies, Inc. Modular I/O. All replaced products become the sole property of SBS Technologies, Inc. Modular I/O.

SBS Technologies, Inc. Modular I/O's warranty of and liability for defective products is limited to that set forth herein. SBS Technologies, Inc. Modular I/O disclaims and excludes all other product warranties and product liability, expressed or implied, including but not limited to any implied warranties of merchantability or fitness for a particular purpose or use, liability for negligence in manufacture or shipment of product, liability for injury to persons or property, or for any incidental or consequential damages.

SBS Technologies, Inc. Modular I/O's products are not authorized for use as critical components in life support devices or systems without the express written approval of the General Manager of SBS Technologies, Inc. Modular I/O.

Service Policy

Before returning a product for repair, verify as well as possible that the suspected unit is at fault. Then call the Customer Service Department for a RETURN MATERIAL AUTHORIZATION (RMA) number. Carefully package the unit, in the original shipping carton if this is available, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include a return address and the telephone number of a technical contact. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. SBS Technologies, Inc. Modular I/O will not be responsible for damages due to improper packaging of returned items. For service SBS Technologies, Inc. Modular I/O products not purchased directly from SBS Technologies, Inc. Modular I/O, contact your reseller. Products returned to SBS Technologies, Inc. Modular I/O for repair by other than the original customer will be treated as out-of-warranty.

Out of Warranty Repairs

Out of warranty repairs will be billed on a material and labor basis. The current minimum repair charge is $100. Customer approval will be obtained before repairing any item if the repair charges will exceed one half of the quantity one list price for that unit. Return transportation and insurance will be billed as part of the repair and is in addition to the minimum charge.

For Service Contact:

Customer Service Department
SBS Technologies, Inc.
Modular I/O
181 Constitution Drive
Menlo Park, CA 94025
(650) 327-1200
FAX: (650) 327-3808
email: support@sbs-mio.com

Specifications

Logic Interface	IndustryPack logic Interface, 0.7 compatible Single-high size
Number of Channels	Two
Input Isolation	Each line optically isolated
Input Ranges	
Encoder/Counter	EIA 485 differential or 20 mA current loop. User configurable for TTL.
Limit Switches	EIA 485 differential or 20 mA current loop. User configurable for TTL.
Emergency Stop	TTL Compatible. GND = Stop.
Output Isolation	Open collector type outputs.
Output Ranges	TTL Compatible.
Motion Controller IC	Nippon Pulse Motor PCL-240MK
Internal Clock Frequency	4 MHz
Pulse Rate	240,000 pps
Pulse Rate Multiplication	0.01x to 30x
Number of Pulses	1 to 16,777,215 (24-Bit)
Counter IC	U.S. Digital LS 7166
Counter size	24-Bit Preloadable Up/Down
Counter resolution	1x, 2x or 4x
Counter frequency	10 MHz, max.
Input polarity	all inputs have programmable polarity
Input masking	all inputs are maskable
Interrupts	Programmable interrupt vector. Maskable interrupts generated by all hardware inputs and all software programmed testable parameters
Temperature Coefficient	0.89 W/°C for uniform heat, component side to solder side
Dimensions	1.800 by 3.900 by 0.340 inches maximum
Power	650 mW per IP typical (+5 V supply)
Environmental	Operating temperature: 0 to 70°C Humidity: 5 to 95% non-condensing Storage: –10 to +85°C

QuickStart Software Support

QuickStart software provides a simple software functional test of the IP hardware. It is not a driver or an application programming library. Please consult with your GreenSpring sales representative on the availability of QuickPack or DriverPack software for this IP.

QuickStart software is provided in ANSI C source code format. This software has been tested on a Motorola MVME162 running OS-9 v3.0 and compiled with Ultra C v1.1.1 from Microware in ANSI C mode. Some modifications will be necessary to use a different compiler or operating system. The areas specific to OS-9 or Microware software tools have been marked as thoroughly as possible. QuickStart software is supplied as is with no warranty or guarantee. The recipient may reuse and/or modify this software for use with GreenSpring IndustryPacks only.

Function summaries are provided here for customer reference only. This information was current at the time the manual was last revised. This information is not necessarily current or complete manufacturing data, nor is it part of the product specification. All information following is copyright GreenSpring Computers, Inc.

Current code listings, including print outs and floppy disk (DOS 3.5" 1.44 MB format), are available from GreenSpring Computers as part of the Engineering Kit option or from your international distributor.

```
-------------------------------------------------------
NAME
    initLS7166 - initialize the LS7166 as a quadrature decoder

SYNOPSIS
    initLS7166( LS7166 *pCtr )

DESCRIPTION
    initLS7166() initializes the LS7166 chip as a 4X quadrature decoder.

RETURNS
    SUCCESS

-------------------------------------------------------
NAME
    readLS7166 - read the LS7166 quadrature counter

SYNOPSIS
    INT32 readLS7166( LS7166 *pCtr )

DESCRIPTION
    readLS7166() reads the 24 bit counter of the LS7166 and
    returns a 32 bit sign extended integer value.

RETURNS
    24 bit counter value sign extended to 32 bits.

-------------------------------------------------------
NAME
    readDataPCL240MK - read a register (R0 - R17) from the PCL240MK

SYNOPSIS
    UINT32 readDataPCL240MK( PCL240MK *pChip, UINT8 regNum )

DESCRIPTION
    readDataPCL240MK() reads the 24 bit value of the given register
    and returns it as a 32 bit value.  The most significant 8 bits
    are always returned as zero.

RETURNS
    register value

-------------------------------------------------------
NAME
    writeDataPCL240MK - write data to a PCL240MK register (R0 - R17)

SYNOPSIS
    INT writeDataPCL240MK( PCL240MK *pChip, UINT8 regNum, UINT32 data )

DESCRIPTION
    writeDataPCL240MK() writes the lower 24 bits of the value data
    to the given register.

RETURNS
    SUCCESS

-------------------------------------------------------
NAME
    goPCL240MK - move the stepper motor the given number of steps
```

```
SYNOPSIS
    INT goPCL240MK( PCL240MK *pChip, INT32 counts )

DESCRIPTION
    goPCL240MK() causes the motor to move the given number of
    counts.  The motion is programmed for a trapezoidal velocity
    profile.

    A positive count moves the motor in the clockwise direction.
    A negative count moves the motor in the counterclockwise
    direction.

RETURNS
    SUCCESS

-------------------------------------------------------
NAME
    initPCL240MK - initialize the PCL240MK chip

SYNOPSIS
    INT initPCL240MK( PCL240MK *pChip, INT Accel, INT CruiseVel )

DESCRIPTION
    initPCL240MK() initializes the PCL240MK stepper motor
    controller to perform symmetric trapezoidal velocity profile
    moves with the given acceleration and cruise velocity.

RETURNS
    SUCCESS

-------------------------------------------------------
NAME
    resetEStop - clear the emergency stop condition for the given axis

SYNOPSIS
    INT resetEStop( IP_STEPPER_CTRL *pAxisCtrl )

DESCRIPTION
    resetEStop() resets the E-Stop status bit in the IP-Stepper
    control register.

RETURNS
    SUCCESS

-------------------------------------------------------
NAME
    readEStop - read the current status of E-stop

SYNOPSIS
    INT readEStop( IP_STEPPER_CTRL *pAxisCtrl )

DESCRIPTION
    readEStop() returns the value of the E-Stop status bit.

RETURNS
    TRUE if the E-Stop status bit is set (E-Stop pending).
    FALSE if the E-Stop status bit is clear.

-------------------------------------------------------
```

```
NAME
    testPCL240MK - test the stepper axis

DESCRIPTION
    The stepper motor spins clockwise, then counterclockwise.
    The encoder count is printed at each endpoint of motion.
```