PCIe-8158

8-Axis Servo/Stepper
Motion Control Card

User's Manual

Manual Rev.: 2.00

Revision Date: Nov. 18, 2016

Part No: 50-11262-1000

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Advance Technologies; Automate the World.

Revision History

Revision	Release Date	Description of Change(s)
2.00	Nov. 18, 2016	Initial Release

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Preface

Copyright 2016 ADLINK Technology, Inc.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

Disclaimer

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Environmental Responsibility

ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encouraged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.

Trademarks

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Conventions

Take note of the following conventions used throughout this manual to make sure that users perform certain tasks and instructions properly.



Additional information, aids, and tips that help users perform tasks.



Information to prevent ***minor*** physical injury, component damage, data loss, and/or program corruption when trying to complete a task.



Information to prevent ***serious*** physical injury, component damage, data loss, and/or program corruption when trying to complete a specific task.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table of Contents

Preface iii

List of Tables ix

List of Figures xi

1 Introduction 1

1.1 Features 3
1.2 Specifications 4
1.3 Supported Software 6
Programming Library 6
MotionCreatorPro 6
1.4 Available Terminal Boards 6
1.5 PCB Layout 7
1.6 CN2 Pin Assignments: Main Connector 8
1.7 K1/K2 Pin Assignments: Simultaneous Start/Stop 11
1.8 P1 Manual Pulse Generator 11
1.9 CN5 Pin Assignments: TTL I/O 12

2 Getting Started 15

2.1 Package Contents 15
2.2 PCIe-8158 Hardware Installation 15
Hardware Configuration 15
PCIe Slot Selection 15
Installation Procedures 16
Troubleshooting: 16
2.3 Software Driver Installation 16
2.4 J1 to J16 Jumper Setting for Pulse Output 17
2.5 SW1 Card Index Selection 18
2.6 Signal Connections 18
2.6.1 Pulse Output Signals OUT and DIR on CN2 18

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.6.2 Encoder Feedback Signals EA, EB and EZ 22
2.6.3 Origin Signal ORG 26
2.6.4 End-Limit Signals PEL and MEL 27
2.6.5 In-Position Signal INP 29
2.6.6 Alarm Signal ALM 30
2.6.7 Deviation Counter Clear Signal ERC 31
2.6.8 General Purpose Signal SVON 32
2.6.9 General-purpose Signal RDY 33
2.6.10 Multi-Functional Output Pin: DO/CMP 34
2.6.11 Multi-Functional Input Pin: DI/LTC/SD/PCS/CLR/EMG 35
2.6.12 Manual Pulse Generator Input Signals PA and PB 36
2.6.13 Simultaneous Start/Stop Signals STA and STP 37
2.6.14 General Purpose TTL I/O EDI And EDO 39

A Appendix: MotionCreatorPro 41
A.1 About MotionCreatorPro 41
A.2 Initiating MotionCreatorPro 41
A.3 MotionCreatorPro Introduction 42
Main Menu 42
Select Menu 43
Card Information Menu 44
Configuration Menu 45
Single Axis Operation Menu 49
Two-Axis and Four-Axis Operation Menu 57
2D_Motion Menu 62
Help Menu 67

B Appendix: Function Library Reference 69
B.1 Data Types 69
Function Naming 69
B.2 List of Functions 70

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

B.3 System and Initialization 74
B.4 Pulse Input/Output Configuration 78
B.5 Velocity mode motion 80
B.6 Single Axis Position Mode 83
B.7 Linear Interpolated Motion 87
B.8 Circular Interpolation Motion 95
B.9 Helical Interpolation Motion 101
B.10 Home Return Mode 105
B.11 Manual Pulse Generator Motion 108
B.12 Motion Status 111
B.13 Motion Interface I/O 112
B.14 Interrupt Control 120
B.15 Position Control and Counters 123
B.16 Position Compare and Latch 129
B.17 Continuous motion 133
B.18 Multiple Axes Simultaneous Operation 135
B.19 General-Purpose DIO 138
B.20 Soft Limit 140
B.21 Backlash Compensation / Vibration Suppression 142
B.22 Speed Profile Calculation 144
B.23 Extended General Purpose TTL Input/Output 147
B.24 Return Code 149

C Appendix: Connection Example 151
C.1 General Description of Wiring 151
C.2 Terminal Board User Guide 151

Important Safety Instructions 153

Getting Service 157

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

List of Tables

Table 1-1: PCB Layout Legend 7
Table 1-2: P1 Manual Pulse Generator 12
Table 2-1: SW1 Card Index 18
Table 2-2: Pulse Output Signals on CN2 20
Table 2-3: OUT or DIR Output by Jumper 20
Table 2-4: EA, EB, and EZ Pin Assignments 24
Table 2-5: Device/Encoder/Power Connection 25
Table 2-6: ORG0-ORG7 Pin Assignments 26
Table 2-7: End-Limit Signal Pin Assignment 28
Table 2-8: INP Signal Connection 29
Table 2-9: Alarm Signal Connection 30
Table 2-10: ERC Connection 31
Table 2-11: SVON Connection 32
Table 2-12: RDY Signal Connection 33
Table 2-13: DO/CMP Connection 34
Table 2-14: DI/LTC/SD/PCS/CLR/EMG Connection 35
Table 2-15: Manual Pulse Generator Input Signal Connection . 36

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

List of Figures

Figure 1-1: PCIe-8158 Block Diagram 2
Figure 1-2: PCB Layout 7
Figure 1-3: IDE 44-pin Connector Assignment 12
Figure 1-4: DSUB 37-pin Connector Assignment 13
Figure 2-1: J1 to J16 Jumper Settings 17
Figure 2-2: OUT and DIR Axis Signals 21
Figure 2-3: OUT/DIR Connection for Open-Collector Wiring 22
Figure 2-4: EA, EB, and EZ Input Circuits 24
Figure 2-5: Line Driver Connection Output Circuit 25
Figure 2-6: Device/Encoder Connection Circuit 26
Figure 2-7: ORG Input Circuit 27
Figure 2-8: End-Limit Signals Circuit 28
Figure 2-9: INP Signal Circuit 29
Figure 2-10: Input Alarm Circuit 30
Figure 2-11: ERC Circuit 32
Figure 2-12: SVON Circuit 33
Figure 2-13: RDY Circuit 34
Figure 2-14: DO/CMP Circuit 35
Figure 2-15: DI/LTC/SD/PCS/CLR/EMG Circuit 36
Figure 2-16: Manual Pulse Generator Input Signal Circuit 37
Figure 2-17: STA & STP Connection 38
Figure 2-18: STA & STP Connection With External Start/Stop 38
Figure 2-19: EDI And EDO Circuit 39

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1 Introduction

The PCIe-8158 is an advanced, modulized 8-axis motion controller card with a PCIe interface. It can generate high frequency pulses (6.55MHz) to drive steppers or servomotors, and as a motion controller, it provides 8-axis linear, circular, and continuous interpolation for continuous velocity. Also, position/speed change on the fly is available with a single axis operation. Multiple PCIe-8158 cards can be used in one system, and incremental encoder interfaces on all eight axes allow correction of positioning errors generated by inaccurate mechanical transmissions.

The PCIe-8158 carrier board's 8-axis pulse train output control channels allow, for additional functions such as high-speed triggering or distributed I/O control, addition of a daughter board as needed. Position comparison is provided for situations such as line scanning in which the application requires the motion controller to generate high speed triggering pulse and gain high resolution images. In such situations, adoption of a DB-8150 can extend PCIe-8158 function. In addition to motion control, sensor and actuator elements in machine automation can require the presence of I/O access for integration. Accordingly, ADLINK provides distributed I/O connection of these devices via employment of a daughter board. The cost-effective configuration reduces wiring efforts and physical controller footprint.

As shown in the following functional block diagram, motion control functions include trapezoidal and S-curve acceleration/deceleration, linear and circular interpolation between two axes and continuous motion positioning, and 13 home return modes, all performed internally by the ASIC, thus reducing CPU loading.

The PCIe-8158 also offers the following functions for enhanced ease of use.

1. Card Index Setting:

   The card index value of the PCIe-8158 can be set with a DIP switch to a value between 0 and 15, useful for machine makers running exceptionally large control systems

2. Emergency Input

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

An emergency input pin can be wired to a control to allow immediate interruption of pulse output

3. Software Security Protection

To secure applications, a 16-bit value can be set in the EEPROM to prevent access to custom programs.

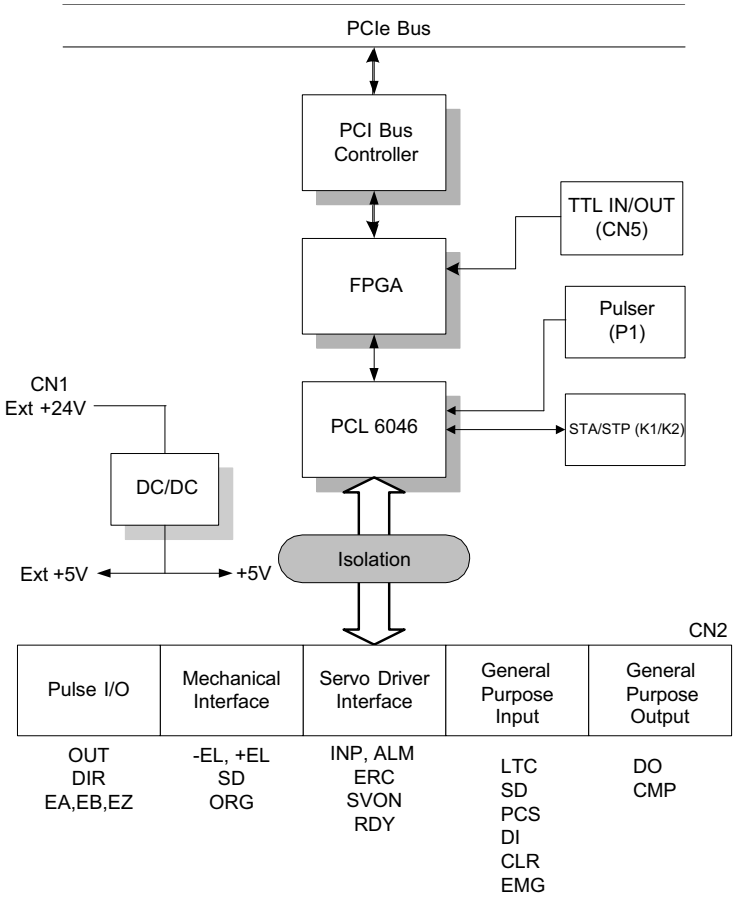


Figure 1-1: PCIe-8158 Block Diagram

MotionCreatorPro and *MotionCreatorPro 2* are Windows-based application development software packages included with the PCIe-8158. They are useful for debugging motion control systems

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

during the design phase. The display lists information for all installed axes and I/O signal status of the PCIe-8158.

Windows programming libraries are also provided for C++ compilers. Sample programs are provided to illustrate the operations of the functions.

1.1 Features

The following list summarizes the main features of the PCIe-8158 motion control system.

- PCIe bus Plug-and-Play (Universal)
- 8 axes of step and direction pulse output for controlling stepping or servomotor
- Maximum output frequency of 6.55MPPS
- Pulse output options: OUT/DIR, CW/CCW, AB phase
- Pulse input options: CW/CCW, AB phase x1, x2, x4
- Maximum pulse input frequency of 3.2Mhz in CW/CCW or AB phase X1 mode (AB phase x4 can reach 6.5Mhz).
- Programmable acceleration and deceleration time for all modes
- Trapezoidal and S-curve velocity profiles for all modes
- 2 to 4 axes linear interpolation
- 2 axes circular interpolation
- Continuous interpolation for contour following motion
- Change position and speed on the fly
- 13 home return modes with auto searching
- Hardware backlash compensator and vibration suppression
- 2 software end-limits for each axis
- 28-bit up/down counter for incremental encoder feedback
- Home switch, index signal (EZ), positive, and negative end limit switches interface on all axes
- 8-axis high speed position latch input
- 8-axis position compare and trigger output
- All digital input and output signals are 2500V_{rms} isolated

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- Programmable interrupt sources
- Simultaneous start/stop motion on multiple axes
- Manual manual pulse generator input interface
- Card index selection
- Security protection on EERPOM
- Dedicated emergency input pin for wiring
- Software supports a maximum of up to 12 PCIe-8158 cards operation in one system
- Compact PCB design
- Includes MotionCreatorPro, a Microsoft Windows-based application development software
- PCIe-8158 libraries and utilities for Windows 7 and 8.1

1.2 Specifications

Applicable Motors:

- Stepping motors
- AC or DC servomotors with pulse train input servo drivers

Performance:

- Number of controllable axes: 8
- Maximum pulse output frequency: 6.55MPPS, linear, trapezoidal, or S-Curve velocity profile drive
- Internal reference clock: 19.66MHz
- 28-bit up/down counter range: 0-268, 435, 455 or –134, 217, 728 to +134, 217, 727
- Position pulse setting range (28-bit): -134, 217, 728 to +134, 217, 728
- Pulse rate setting range (Pulse Ratio = 1: 65535):
  - 0.1 PPS to 6553.5 PPS. (Multiplier = 0.1)
  - 1 PPS to 65535 PPS. (Multiplier = 1)
  - 100 PPS to 6553500 PPS. (Multiplier = 100)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

I/O Signals:

- Input/Output signals for each axis
- All I/O signal are optically isolated with 2500Vrms isolation voltage
- Command pulse output pins: OUT and DIR
- Incremental encoder signals input pins: EA and EB
- Encoder index signal input pin: EZ
- Mechanical limit/home signal input pins: ±EL, ORG
- Composite pins: DI / LTC (Latch) / SD (Slow-down) / PCS (Position Change Signal) / CLR (Clear) / EMG (Emergency Input)
- Servomotor interface I/O pins: INP, ALM, and ERC
- General-purposed digital output pin: SVON, DO
- General-purposed digital input pin: RDY, GDI
- Pulse signal input pin: PA and PB (with Isolation)
- Simultaneous Start/Stop signal: STA and STP
- 16 TTL level DO and 16TTL level DI

General Specifications

- Connectors: 100-pin SCSI-type connector
- Operating Temperature: 0°C - 50°C
- Storage Temperature: -20°C - 80°C
- Humidity: 5 - 85%, non-condensing

Power Consumption

- Slot power supply (input): +12V DC ±5%, 400mA max
- External power supply (input): +24V DC ±5%, 500mA max
- External power supply (output): +5V DC ±5%, 300mA, max

PCIe-8158 Dimensions (PCB size):

- 185 (L) X 98.4 (H) mm

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.3 Supported Software

Programming Library

Windows 7/8.1 DLLs are provided for the PCIe-8158. These function libraries are shipped with the board.

MotionCreatorPro

The pre-loaded Windows-based utility sets up cards, motors, and systems, and can aid in debugging hardware and software problems. Users can set I/O logic parameters to be loaded in their own program.

For more information, please see “MotionCreatorPro” on page 41.

1.4 Available Terminal Boards

ADLINK provides steppers with the DIN-100S pin-to-pin terminal board. For servo users, the DIN-814-GP, DIN-814M-J3A, DIN-814Y and DIN-814P-A4 are provided, with suitable servos as follows:

General Purpose Servos	DIN-814-GP
Mitsubishi J3A	DIN-814M-J3A
Yaskawa Sigma II	DIN-814Y
Panasonic MINAS A4	DIN-814P-A4

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.5 PCB Layout

NOTE:

All dimensions shown are in millimeters (mm) unless otherwise stated.

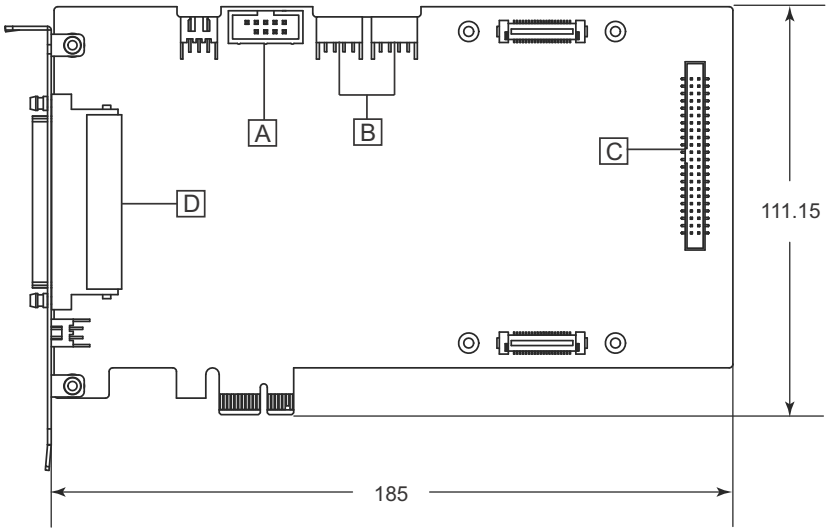


Figure 1-2: PCB Layout

	Connector	Function
A	P1	Manual pulse generator
B	K1/K2	Simultaneous start/stop
C	CN5	TTL I/O
D	CN2	I/O signal (200-pin)

Table 1-1: PCB Layout Legend

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.6 CN2 Pin Assignments: Main Connector

CN2 is the main connector for motion control I/O signals.

Pin	Name	I/O	Function
1	VDD	O	+5V power supply output
2	EXGND	COM	Ext. power ground
3	OUT0+	O	Pulse signal (+)
4	OUT0-	O	Pulse signal (-)
5	DIR0+	O	Dir. signal (+)
6	DIR0-	O	Dir. signal (-)
7	SVON0	O	Servo on/off
8	ERC0	O	Dev. ctr, clr. signal
9	ALM0	I	Alarm signal
10	INP0	I	In-position signal
11	RDY0	I	Multi-purpose Input signal
12	EXGND	COM	Ext. power ground
13	EA0+	I	Encoder A-phase (+)
14	EA0-	I	Encoder A-phase (-)
15	EB0+	I	Encoder B-phase (+)
16	EB0-	I	Encoder B-phase (-)
17	EZ0+	I	Encoder Z-phase (+)
18	EZ0-	I	Encoder Z-phase (-)
19	VDD	O	+5V power supply output
20	EXGND	COM	Ext. power ground
21	OUT1+	O	Pulse signal (+)
22	OUT1-	O	Pulse signal (-)
23	DIR1+	O	Dir. signal (+)
24	DIR1-	O	Dir. signal (-)
25	SVON1	O	Servo on/off
26	ERC1	O	Dev. ctr, clr. signal
27	ALM1	I	Alarm signal
28	INP1	I	In-position signal
29	RDY1	I	Multi-purpose Input signal
30	EXGND	COM	Ext. power ground

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Pin	Name	I/O	Function
31	EA1+	I	Encoder A-phase (+)
32	EA1-	I	Encoder A-phase (-)
33	EB1+	I	Encoder B-phase (+)
34	EB1-	I	Encoder B-phase (-)
35	EZ1+	I	Encoder Z-phase (+)
36	EZ1-	I	Encoder Z-phase (-)
37	PEL0	I	End limit signal (+)
38	MEL0	I	End limit signal (-)
39	GDI0	I	DI/LTC/PCS/SD/CLR0
40	DO0	O	General Output 0
41	ORG0	I	Origin signal
42	EXGND		Ext. power ground
43	PEL1	I	End limit signal (+)
44	MEL1	I	End limit signal (-)
45	GDI1	I	DI/LTC/PCS/SD/CLR1
46	DO1	O	General output 1
47	ORG1	I	Origin signal
48	EXGND	COM	Ext. power ground
49	EXGND	COM	Ext. power ground
50	EXGND	COM	Ext. power ground
51	VDD	O	+5V power supply output
52	EXGND	COM	Ext. power ground
53	OUT2+	O	Pulse signal (+)
54	OUT2-	O	Pulse signal (-)
55	DIR2+	O	Dir. signal (+)
56	DIR2-	O	Dir. signal (-)
57	SVON2	O	Servo On/Off
58	ERC2	O	Dev. ctr, clr. signal
59	ALM2	I	Alarm signal
60	INP2	I	In-position signal
61	RDY2	I	Multi-purpose Input signal
62	EXGND	COM	Ext. power ground
63	EA2+	I	Encoder A-phase (+)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Pin	Name	I/O	Function
64	EA2-	I	Encoder A-phase (-)
65	EB2+	I	Encoder B-phase (+)
66	EB2-	I	Encoder B-phase (-)
67	EZ2+	I	Encoder Z-phase (+)
68	EZ2-	I	Encoder Z-phase (-)
69	VDD	O	+5V power supply output
70	EXGND	COM	Ext. power ground
71	OUT3+	O	Pulse signal (+)
72	OUT3-	O	Pulse signal (-)
73	DIR3+	O	Dir. signal (+)
74	DIR3-	O	Dir. signal (-)
75	SVON3	O	Servo on/off
76	ERC3	O	Dev. ctr, clr. signal
77	ALM3	I	Alarm signal
78	INP3	I	In-position signal
79	RDY3	I	Multi-purpose Input signal
80	EXGND	COM	Ext. power ground
81	EA3+	I	Encoder A-phase (+)
82	EA3-	I	Encoder A-phase (-)
83	EB3+	I	Encoder B-phase (+)
84	EB3-	I	Encoder B-phase (-)
85	EZ3+	I	Encoder Z-phase (+)
86	EZ3-	I	Encoder Z-phase (-)
87	PEL2	I	End limit signal (+)
88	MEL2	I	End limit signal (-)
89	GDI2	I	DI/LTC/PCS/SD/CLR2
90	DO2	O	General output 2
91	ORG2	I	Origin signal
92	EXGND		Ext. power ground
93	PEL3	I	End limit signal (+)
94	MEL3	I	End limit signal (-)
95	GDI3	I	DI/LTC/PCS/SD/CLR3
96	DO3	O	General output 3

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Pin	Name	I/O	Function
97	ORG3	I	Origin signal
98	EXGND	COM	Ext. power ground
99	E_24V	COM	Isolation power Input, +24V
100	E_24V	COM	Isolation power Input, +24V

1.7 K1/K2 Pin Assignments: Simultaneous Start/ Stop

K1 and K2 are for simultaneous start/stop signals for multiple axes or multiple cards.

No.	Name	Function
1	N/C	
2	STA	Simultaneous start signal input/output
3	STP	Simultaneous stop signal input/output
4	GND	PCIe bus power ground

GND pins are powered by PCIe bus.

1.8 P1 Manual Pulse Generator

#	Name	Function (Axis)
1	VDD	Isolated Power +5V
2	PA+	Manual Pulse Generator A+ phase signal input
3	PA-	Manual Pulse Generator A- phase signal input
4	PB+	Manual Pulse Generator B+ phase signal input
5	PB-	Manual Pulse Generator B- phase signal input
6	EXGND	External Ground
7	N/A	Not Available
8	N/A	Not Available

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

#	Name	Function (Axis)
9	N/A	Not Available

Table 1-2: P1 Manual Pulse Generator



NOTE:

Refer to the contents of this chapter before wiring any cable between the PCIe-8158 and any motor driver

1.9 CN5 Pin Assignments: TTL I/O

#	Name	I/O	Function	#	Name	I/O	Function
1	DGND	N/A	Power ground	2	DGND	N/A	Power ground
3	EDI0	I	Digital Input 0	4	EDI1	I	Digital Input 1
5	EDI2	I	Digital Input 2	6	EDI3	I	Digital Input 3
7	EDI4	I	Digital Input 4	8	EDI5	I	Digital Input 5
9	VCC	O	Power +3.3V	10	DGND	N/A	Power ground
11	EDI6	I	Digital Input 6	12	EDI7	I	Digital Input 7
13	EDI8	I	Digital Input 8	14	EDI9	I	Digital Input 9
15	EDI10	I	Digital Input 10	16	EDI11	I	Digital Input 11
17	DGND	N/A	Power ground	18	DGND	N/A	Power ground
19	EDI12	I	Digital Input 12	20	EDI13	I	Digital Input 13
21	EDI14	I	Digital Input 14	22	EDI15	I	Digital Input 15
23	EDO0	O	Digital Output 0	24	EDO1	O	Digital Output 1
25	EDO2	O	Digital Output 2	26	EDO3	O	Digital Output 3
27	DGND	N/A	Power ground	28	DGND	N/A	Power ground
29	EDO4	O	Digital Output 4	30	EDO5	O	Digital Output 5
31	EDO6	O	Digital Output 6	32	EDO7	O	Digital Output 7
33	EDO8	O	Digital Output 8	34	EDO9	O	Digital Output 9
35	DGND	N/A	Power ground	36	VCC	O	Power +3.3V
37	EDO10	O	Digital Output 10	38	EDO11	O	Digital Output 11
39	EDO12	O	Digital Output 12	40	EDO13	O	Digital Output 13
41	EDO14	O	Digital Output 14	42	EDO15	O	Digital Output 15
43	DGND	N/A	Power ground	44	DGND	N/A	Power ground

Figure 1-3: IDE 44-pin Connector Assignment

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

#	Name	I/O	Function	#	Name	I/O	Function
1	DGND	N/A	Power ground	20	DGND	N/A	Power ground
2	EDI0	I	Digital Input 0	21	EDO0	O	Digital Output 0
3	EDI1	I	Digital Input 1	22	EDO1	O	Digital Output 1
4	EDI2	I	Digital Input 2	23	EDO2	O	Digital Output 2
5	EDI3	I	Digital Input 3	24	EDO3	O	Digital Output 3
6	EDI4	I	Digital Input 4	25	EDO4	O	Digital Output 4
7	EDI5	I	Digital Input 5	26	EDO5	O	Digital Output 5
8	EDI6	I	Digital Input 6	27	EDO6	O	Digital Output 6
9	EDI7	I	Digital Input 7	28	EDO7	O	Digital Output 7
10	EDI8	I	Digital Input 8	29	EDO8	O	Digital Output 8
11	EDI9	I	Digital Input 9	30	EDO9	O	Digital Output 9
12	EDI10	I	Digital Input 10	31	EDO10	O	Digital Output 10
13	EDI11	I	Digital Input 11	32	EDO11	O	Digital Output 11
14	EDI12	I	Digital Input 12	33	EDO12	O	Digital Output 12
15	EDI13	I	Digital Input 13	34	EDO13	O	Digital Output 13
16	EDI14	I	Digital Input 14	35	EDO14	O	Digital Output 14
17	EDI15	I	Digital Input 15	36	EDO15	O	Digital Output 15
18	DGND	N/A	Power ground	37	DGND	N/A	Power ground
19	VCC	O	Power +3.3V	N/A	N/A	N/A	N/A

Figure 1-4: DSUB 37-pin Connector Assignment

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2 Getting Started

2.1 Package Contents

In addition to this *User's Guide*, the package also includes the following items:

- PCIe-8158: advanced 8-axis servo/stepper motion control card
- General-purpose TTL I/O accessory cable (for CN5)
- Quick Start Guide



The terminal board is an optional accessory and is not included in the PCIe-8158 package.

If any of these items are missing or damaged, contact the dealer. Save the shipping materials and carton to ship or store the product in the future.

2.2 PCIe-8158 Hardware Installation

Hardware Configuration

The PCIe-8158 is fully Plug-and-Play compliant. Hence, memory allocation (I/O port locations) and IRQ channel of the PCIe card are assigned by the system BIOS. Addresses are assigned on a board-by-board basis for all PCIe cards in the system.

PCIe Slot Selection

The PCIe-8158 can be installed in any PCIe slot.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Installation Procedures

1. Read through the manual and set up the jumpers according to your needs
2. Turn off the computer and all connected accessories (printer, modem, monitor, etc.). Remove the cover.
3. Select a PCIe expansion slot.
4. Before handling the PCIe-8158, discharge any static buildup by touching the metal case of the computer. Hold the edge of the card and do not touch the components.
5. Position the board into the selected PCIe slot.
6. Secure the card in place at the rear panel of the system unit using screws removed from the slot.

Troubleshooting:

If the system doesn't boot or erratic operations occur with the PCIe board in place, an interrupt conflict is likely. Please consult the BIOS documentation.

Ensure the Control Panel lists the card. If not, check PCIe settings in BIOS or change PCIe slots.

2.3 Software Driver Installation

1. Run PCIe-8158 SDK.
2. Follow the procedures as instructed.
3. After setup installation is completed, restart Windows.



NOTE:

Ensure the latest software version from the ADLINK website is installed

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4 J1 to J16 Jumper Setting for Pulse Output

J1-J16 set the type of output pulse signals (DIR and OUT), which can either be differential line driver or open collector output. Please See “Pulse Output Signals OUT and DIR on CN2” on page 18. for details. The default setting is differential line driver mode. Mapping is as follows:

J1 & J2	Axis 0	**J9 & J10**	Axis 4
J3 & J4	Axis 1	**J11 & J12**	Axis 5
J5 & J6	Axis 2	**J13& J14**	Axis 6
J7 & J8	Axis 3	**J15 & J16**	Axis 7

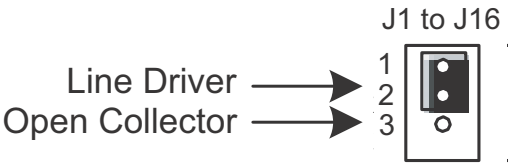


Figure 2-1: J1 to J16 Jumper Settings

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.5 SW1 Card Index Selection

The SW1 switch is used to set the card index. For example, if 1 is set to ON and the others are OFF, that card index is 1. The index value can be from 0 to 15. Refer to the following table for details.

Card ID	Switch Setting (ON=1)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Table 2-1: SW1 Card Index

2.6 Signal Connections

2.6.1 Pulse Output Signals OUT and DIR on CN2

With 8 axis pulse output signals on the PCIe-8158, each axis uses two pairs of OUT and DIR differential signals to transmit the pulse train and indicate the direction. The OUT and DIR signals can also be programmed as CW and CCW signal pairs. Each signal consists of a pair of differential signals. For example, OUT0 consists of OUT0+ and OUT0- signals.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

CN2 Pin	Signal	Description	Axis #
3	OUT0+	Pulse signal (+)	0
4	OUT0-	Pulse signal (-)	0
5	DIR0+	Dir. signal (+)	0
6	DIR0-	Dir. signal (-)	0
21	OUT1+	Pulse signal (+)	1
22	OUT1-	Pulse signal (-)	1
23	DIR1+	Dir. signal (+)	1
24	DIR1-	Dir. signal (-)	1
53	OUT2+	Pulse signal (+)	2
54	OUT2-	Pulse signal (-)	2
55	DIR2+	Dir. signal (+)	2
56	DIR2-	Dir. signal (-)	2
71	OUT3+	Pulse signal (+)	3
72	OUT3-	Pulse signal (-)	3
73	DIR3+	Dir. signal (+)	3
74	DIR3-	Dir. signal (-)	3
103	OUT4+	Pulse signal (+)	4
104	OUT4-	Pulse signal (-)	4
105	DIR4+	Dir. signal (+)	4
106	DIR4-	Dir. signal (-)	4
121	OUT5+	Pulse signal (+)	5
122	OUT5-	Pulse signal (-)	5
123	DIR5+	Dir. signal (+)	5
124	DIR5-	Dir. signal (-)	5
153	OUT6+	Pulse signal (+)	6
154	OUT6-	Pulse signal (-)	6
155	DIR6+	Dir. signal (+)	6
156	DIR6-	Dir. signal (-)	6
171	OUT7+	Pulse signal (+)	7
172	OUT7-	Pulse signal (-)	7
173	DIR7+	Dir. signal (+)	7

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

CN2 Pin	Signal	Description	Axis #
174	DIR7-	Dir. signal (-)	7

Table 2-2: Pulse Output Signals on CN2

The output of the OUT or DIR signals can be configured by jumpers as either differential line drivers or open collector output. Users can select the output mode either by jumper wiring between 1 and 2 or 2 and 3 of jumpers J1 to J8 as follows:

Output Signal	For differential line driver output, close 1 and 2 on	For open-collector output, close 2 and 3 on
OUT0+	J1	J1
DIR0+	J2	J2
OUT1+	J3	J3
DIR1+	J4	J4
OUT2+	J5	J5
DIR2+	J6	J6
OUT3+	J7	J7
DIR3+	J8	J8
OUT4+	J9	J9
DIR4+	J10	J10
OUT5+	J11	J11
DIR5+	J12	J12
OUT6+	J13	J13
DIR6+	J14	J14
OUT7+	J15	J15
DIR7+	J16	J16

Table 2-3: OUT or DIR Output by Jumper

The default setting of OUT and DIR is differential line driver mode.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

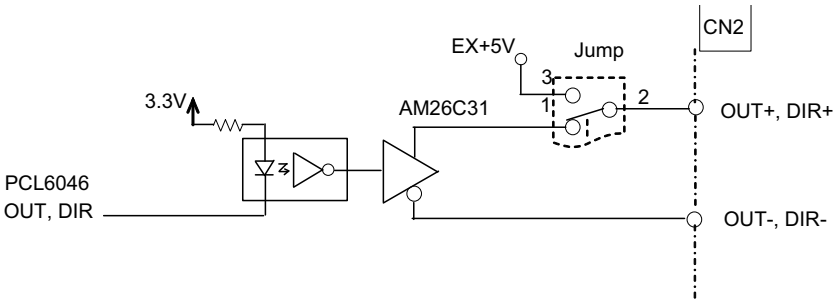


Figure 2-2: OUT and DIR Axis Signals

NOTE:

If the pulse output is set to open collector output mode, OUT- and DIR- transmit OUT and DIR signals, with sink current not exceeding 20mA on the OUT- and DIR- pins, and default setting 1-2 shorted

OUT-/DIR- can connect to a 470Ω pulse input interface COM as shown.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

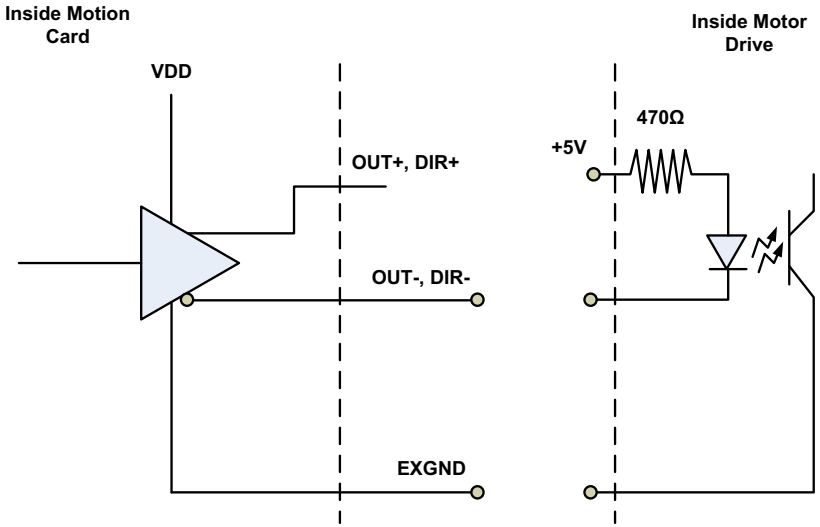


Figure 2-3: OUT/DIR Connection for Open-Collector Wiring

CAUTION:

Sink current exceeding 20mA will damage the 26LS31.

2.6.2 Encoder Feedback Signals EA, EB and EZ

The encoder feedback signals include EA, EB, and EZ. Every axis has six pins for three differential pairs of phase-A (EA), phase-B (EB), and index (EZ) inputs. EA and EB are used for position counting, and EZ is used for zero position indexing. Its relative signal names, pin numbers, and axis numbers are as follows.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

CN2 Pin	Signal	Description	Axis #
3	OUT0+	Pulse signal (+)	0
4	OUT0-	Pulse signal (-)	0
5	DIR0+	Dir. signal (+)	0
6	DIR0-	Dir. signal (-)	0
21	OUT1+	Pulse signal (+)	1
22	OUT1-	Pulse signal (-)	1
23	DIR1+	Dir. signal (+)	1
24	DIR1-	Dir. signal (-)	1
103	OUT4+	Pulse signal (+)	4
104	OUT4-	Pulse signal (-)	4
105	DIR4+	Dir. signal (+)	4
106	DIR4-	Dir. signal (-)	4
121	OUT5+	Pulse signal (+)	5
122	OUT5-	Pulse signal (-)	5
123	DIR5+	Dir. signal (+)	5
124	DIR5-	Dir. signal (-)	5
53	OUT2+	Pulse signal (+)	2
54	OUT2-	Pulse signal (-)	2
55	DIR2+	Dir. signal (+)	2
56	DIR2-	Dir. signal (-)	2
71	OUT3+	Pulse signal (+)	3
72	OUT3-	Pulse signal (-)	3
73	DIR3+	Dir. signal (+)	3
74	DIR3-	Dir. signal (-)	3
153	OUT6+	Pulse signal (+)	6
154	OUT6-	Pulse signal (-)	6

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

CN2 Pin	Signal	Description	Axis #
155	DIR6+	Dir. signal (+)	6
156	DIR6-	Dir. signal (-)	6
171	OUT7+	Pulse signal (+)	7
172	OUT7-	Pulse signal (-)	7
173	DIR7+	Dir. signal (+)	7
174	DIR7-	Dir. signal (-)	7
53	OUT2+	Pulse signal (+)	2

Table 2-4: EA, EB, and EZ Pin Assignments

The input circuit of the EA, EB, and EZ signals is as shown.

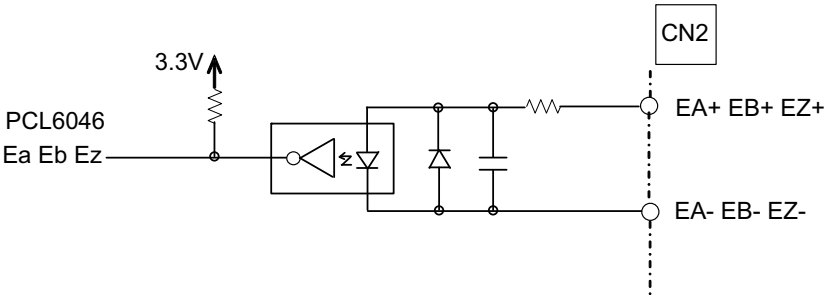


Figure 2-4: EA, EB, and EZ Input Circuits

Please note that the voltage across each differential pair of encoder input signals (EA+, EA-), (EB+, EB-), and (EZ+, EZ-) should be at least 3.5V. Therefore, the output current must be observed when connecting to the encoder feedback or motor driver feedback as not to over drive the source. The differential signal pairs are converted to digital signals EA, EB, and EZ; then feed to the motion control ASIC.

Examples of connecting the input signals with an external circuit include the input circuit connected to an encoder or motor driver if it is equipped with: (1) a differential line driver or (2) an open collector output.

Connection to Line Driver Output

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

To drive the PCIe-8158 encoder input, the driver output must provide at least 3.5V across the differential pairs with at least 8mA driving capacity. The grounds of both sides must be tied together. The maximum frequency is 3Mhz or more depends on wiring distance and signal conditioning.

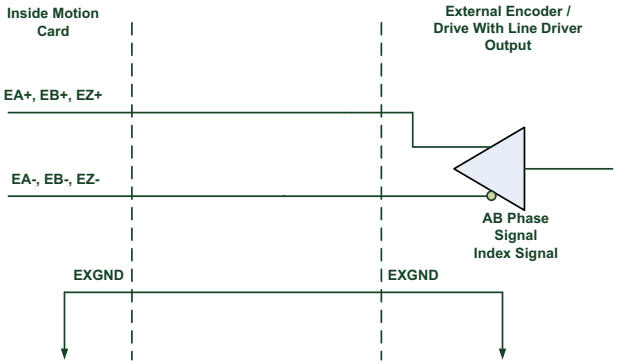


Figure 2-5: Line Driver Connection Output Circuit

Connection to Open Collector Output

To connect with an open collector output, an external power supply is necessary. Some motor drivers can provide the power source. The connection between the PCIe-8158, encoder, and the power supply is as shown. Note that an external current limiting resistor R is necessary to protect the PCIe-8158 input circuit.

Encoder Power (V)	External Resistor R
+5V	0Ω (None)
+12V	1.5kΩ
+24V	3.0kΩ

Table 2-5: Device/Encoder/Power Connection

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

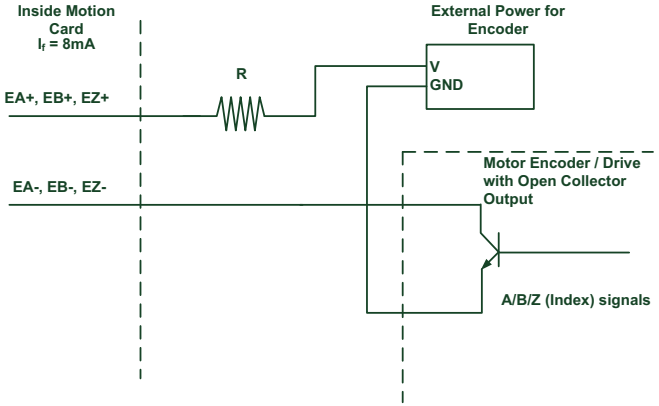


Figure 2-6: Device/Encoder Connection Circuit

2.6.3 Origin Signal ORG

The origin signals (ORG0-ORG7) are used as input signals for the origin of the mechanism. Signal names, pin numbers, and axis numbers are as follows:

CN2 Pin	Signal	Description	Axis #
41	ORG0	Origin signal	0
47	ORG1	Origin signal	1
91	ORG2	Origin signal	2
97	ORG3	Origin signal	3
141	ORG4	Origin signal	4
147	ORG5	Origin signal	5
191	ORG6	Origin signal	6
197	ORG7	Origin signal	7

Table 2-6: ORG0-ORG7 Pin Assignments

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

With the input circuit of the ORG signals, a limit switch is normally used to indicate the origin on one axis. The specifications of the limit switch should have contact capacity of +24V @ 6mA minimum. An internal filter circuit filters out any high frequency spikes, which may cause errors in the operation.

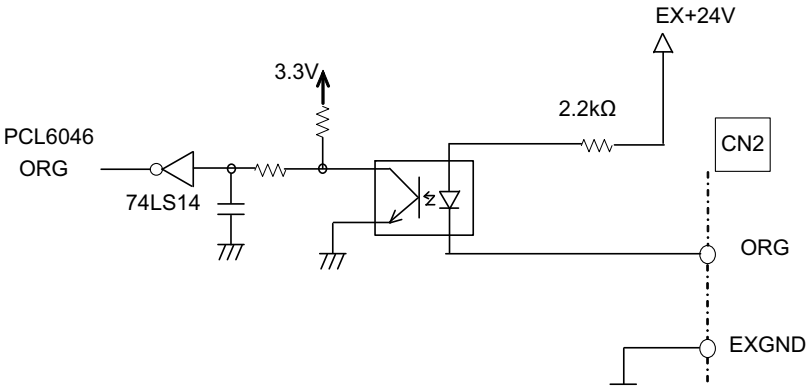


Figure 2-7: ORG Input Circuit

When the motion controller is operated in the home return mode, the ORG signal is used to inhibit the control output signals (OUT and DIR).

2.6.4 End-Limit Signals PEL and MEL

The end-limit signals for each axis, can be in the plus direction (PEL), or the minus direction (MEL), configured as follows.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

CN2 Pin	Signal	Description	Axis #
37	PEL0	End limit signal (+)	0
38	MEL0	End limit signal (-)	0
43	PEL1	End limit signal (+)	1
44	MEL1	End limit signal (-)	1
87	PEL2	End limit signal (+)	2
88	MEL2	End limit signal (-)	2
93	PEL3	End limit signal (+)	3
94	MEL3	End limit signal (-)	3
137	PEL4	End limit signal (+)	4
138	MEL4	End limit signal (-)	4
143	PEL5	End limit signal (+)	5
144	MEL5	End limit signal (-)	5
187	PEL6	End limit signal (+)	6
188	MEL6	End limit signal (-)	6
193	PEL7	End limit signal (+)	7
194	MEL7	End limit signal (-)	7

Table 2-7: End-Limit Signal Pin Assignment

As shown, the external limit switch should have a minimum contact capacity of +24V @ 8mA. Either 'A-type' (normal open) or 'B-type' (normal closed) contact switches can be used.

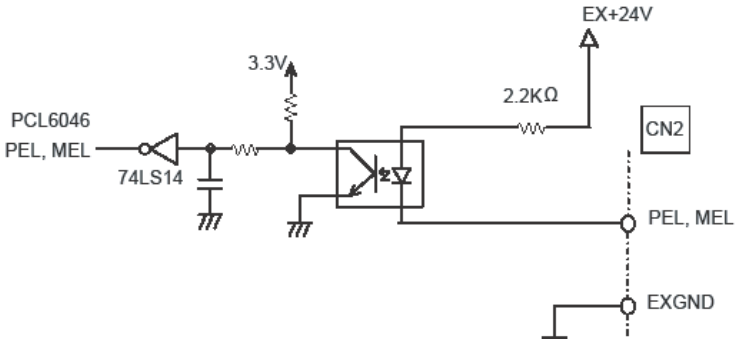


Figure 2-8: End-Limit Signals Circuit

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.6.5 In-Position Signal INP

The in-position signal INP from a servo motor driver indicates its deviation error. If there is no deviation error then the servo's position indicates zero.

CN2 Pin	Signal	Description	Axis #
10	INP0	In-position signal	0
28	INP1	In-position signal	1
60	INP2	In-position signal	2
78	INP3	In-position signal	3
110	INP4	In-position signal	4
128	INP5	In-position signal	5
160	INP6	In-position signal	6
178	INP7	In-position signal	7

Table 2-8: INP Signal Connection

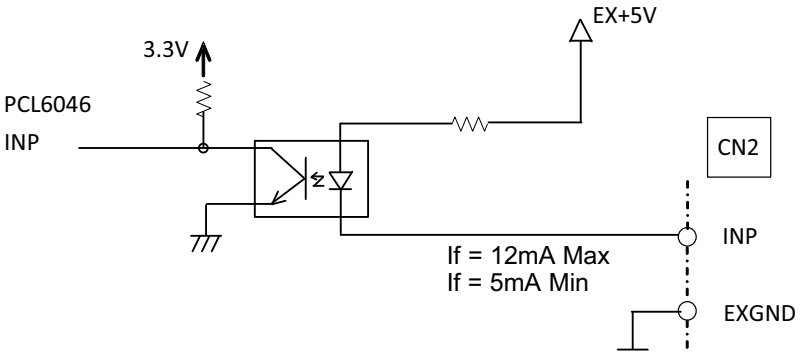


Figure 2-9: INP Signal Circuit

The in-position signal is usually generated by the servomotor driver and is ordinarily an open collector output signal. An external circuit must provide at least 8mA current sink capabilities to drive the INP signal.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.6.6 Alarm Signal ALM

The alarm signal ALM is used to indicate the alarm status from the servo driver.

CN2 Pin	Signal	Description	Axis #
9	ALM0	Alarm signal	0
27	ALM1	Alarm signal	1
59	ALM2	Alarm signal	2
77	ALM3	Alarm signal	3
109	ALM4	Alarm signal	4
127	ALM5	Alarm signal	5
159	ALM6	Alarm signal	6
177	ALM7	Alarm signal	7

Table 2-9: Alarm Signal Connection

The ALM signal is normally an open collector output signal generated by the servomotor driver. An external circuit must provide at least 8mA current sink capabilities to drive the ALM signal.

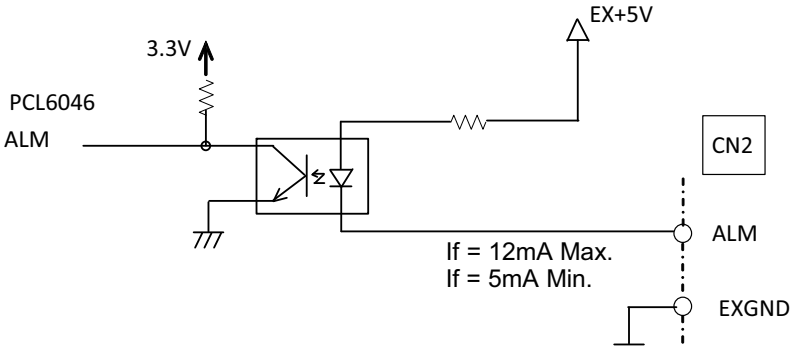


Figure 2-10: Input Alarm Circuit

2.6.7 Deviation Counter Clear Signal ERC

The deviation counter clear signal (ERC) is active when:

- Home return is complete
- End-limit switch is active
- An alarm signal stops OUT and DIR signals
- An emergency stop command is issued by software (operator)

CN2 Pin	Signal	Description	Axis #
8	ERC0	Dev. ctr, clr. Signal	0
26	ERC1	Dev. ctr, clr. Signal	1
58	ERC2	Dev. ctr, clr. Signal	2
76	ERC3	Dev. ctr, clr. Signal	3
108	ERC4	Dev. ctr, clr. Signal	4
126	ERC5	Dev. ctr, clr. Signal	5
158	ERC6	Dev. ctr, clr. Signal	6
176	ERC7	Dev. ctr, clr. Signal	7

Table 2-10: ERC Connection

The ERC signal is used to clear the deviation counter of the servomotor driver. The ERC output circuit is an open collector with a maximum of 35V at 50mA driving capacity.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

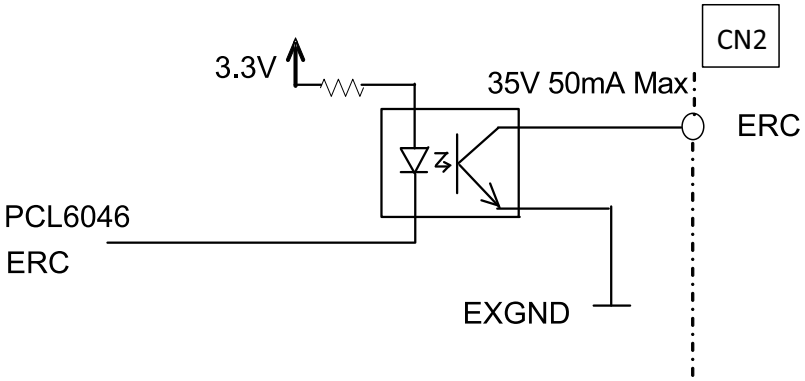


Figure 2-11: ERC Circuit

2.6.8 General Purpose Signal SVON

The SVON signal can be used as a servomotor-on control or general purpose output signal.

CN2 Pin	Signal	Description	Axis #
7	SVON0	Servo On/Off	0
25	SVON1	Servo On/Off	1
57	SVON2	Servo On/Off	2
75	SVON3	Servo On/Off	3
107	SVON4	Servo On/Off	4
125	SVON5	Servo On/Off	5
157	SVON6	Servo On/Off	6
175	SVON7	Servo On/Off	7

Table 2-11: SVON Connection

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

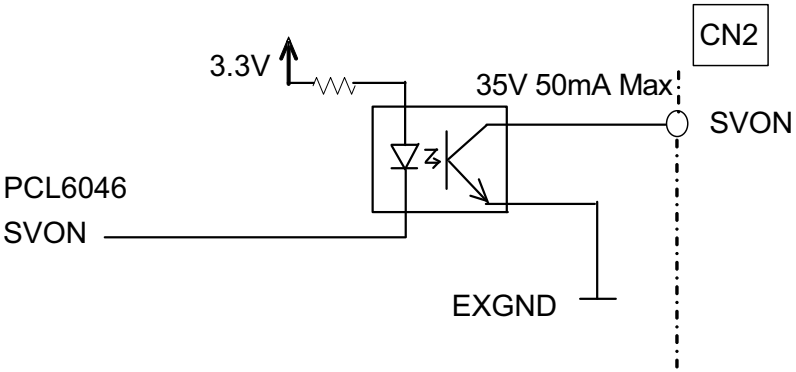


Figure 2-12: SVON Circuit

2.6.9 General-purpose Signal RDY

The RDY signals can be used as motor driver ready input or general purpose input signals.

CN2 Pin	Signal	Description	Axis #
11	RDY0	Multi purpose Input	0
29	RDY1	Multi purpose Input	1
61	RDY2	Multi purpose Input	2
79	RDY3	Multi purpose Input	3
111	RDY4	Multi purpose Input	4
129	RDY5	Multi purpose Input	5
161	RDY6	Multi purpose Input	6
179	RDY7	Multi purpose Input	7

Table 2-12: RDY Signal Connection

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

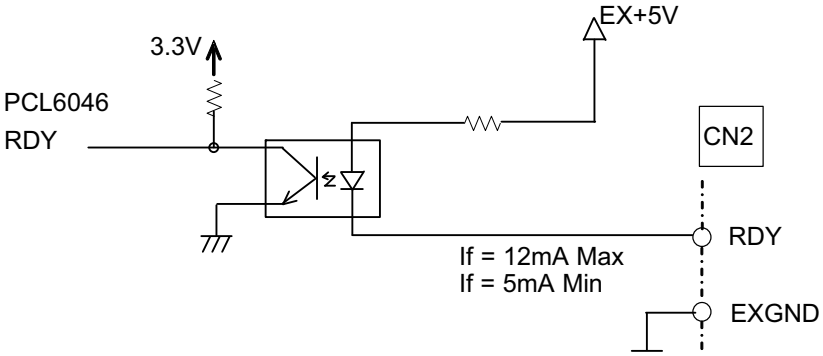


Figure 2-13: RDY Circuit

2.6.10 Multi-Functional Output Pin: DO/CMP

The PCIe-8158 provides multi-functional output channels DO0/CMP0 to DO7/CMP7, corresponding to 8 axes. Each output pin can be individually configured as Digit Output (DO) or as Comparison Output (CMP). When configured as a Comparison Output pin, the pin generates a pulse signal when the encoder counter matches a preset value.

CN2 Pin	Signal	Description	Axis #
40	DO0	General Output / CMP	0
46	DO1	General Output / CMP	1
90	DO2	General Output / CMP	2
96	DO3	General Output / CMP	3
140	DO4	General Output / CMP	4
146	DO5	General Output / CMP	5
190	DO6	General Output / CMP	6
196	DO7	General Output / CMP	7

Table 2-13: DO/CMP Connection

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

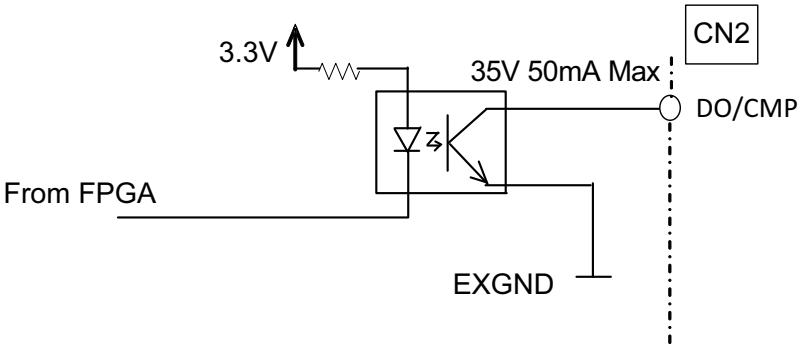


Figure 2-14: DO/CMP Circuit

2.6.11 Multi-Functional Input Pin: DI/LTC/SD/PCS/CLR/EMG

Each of the 4 multi-functional input pins on CN2 can be configured as DI (Digit Input), LTC (Latch), SD (Slow down), PCS (Target position override), CLR (Counter clear), or EMG (Emergency).

CN2 Pin	Signal	Description	Axis #
39	GDI0	DI/LTC/PCS/SD/CLR	0
45	GDI1	DI/LTC/PCS/SD/CLR	1
89	GDI2	DI/LTC/PCS/SD/CLR	2
95	GDI3	DI/LTC/PCS/SD/CLR	3
139	GDI4	DI/LTC/PCS/SD/CLR	4
145	GDI5	DI/LTC/PCS/SD/CLR	5
189	GDI6	DI/LTC/PCS/SD/CLR	6
195	GDI7	DI/LTC/PCS/SD/CLR	7

Table 2-14: DI/LTC/SD/PCS/CLR/EMG Connection

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

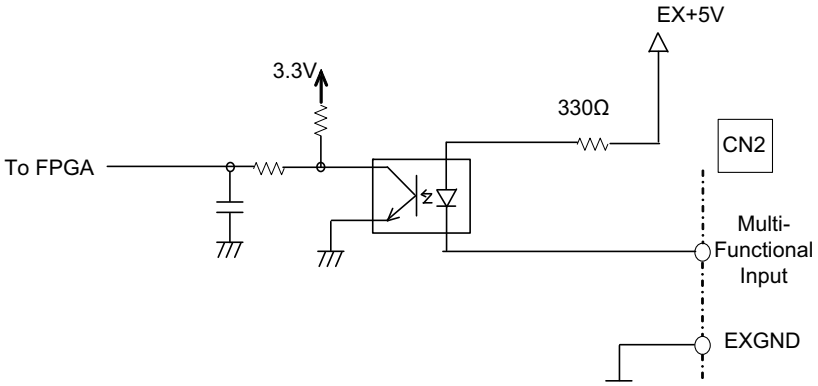


Figure 2-15: DI/LTC/SD/PCS/CLR/EMG Circuit

2.6.12 Manual Pulse Generator Input Signals PA and PB

The PCIe-8158 can accept differential manual pulse generator input signals through the pins of P1 listed below. The manual pulse generator acts as an encoder, with A-B phase signals generating positioning information to guide the motor.

P1 Pin	Signal Name	Axis #	P1 Pin	Signal Name	Axis #
2	PA+	0-7	3	PA-	0-7
4	PB+	0-7	5	PB-	0-7

Table 2-15: Manual Pulse Generator Input Signal Connection

The manual pulse generator signals are used for axes 0 to 7, where each axis' manual pulse generator can be disabled with the _8158_disable_pulser_input function.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

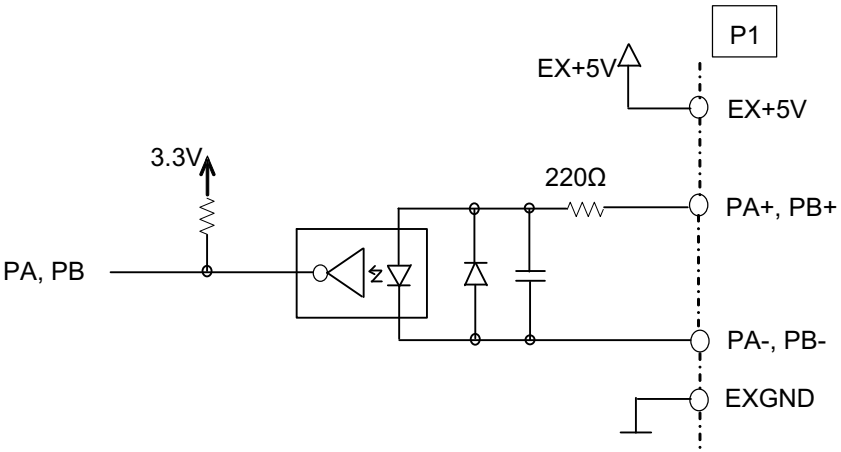


Figure 2-16: Manual Pulse Generator Input Signal Circuit

2.6.13 Simultaneous Start/Stop Signals STA and STP

The PCIe-8158 provides STA and STP signals, which enable simultaneous starting and stopping of motion on multiple axes. The STA and STP signals are on K1/K2.

The STP and STA signals are both input and output signals. To start and stop simultaneously, both software control and external control are needed. With software control, the signals can be generated from any connected PCIe-8158. Alternatively, an external open collector or switch can drive the STA/STP signals for simultaneous start and stop.

If there are two or more PCIe-8158 cards, connect the K2 connector on the previous card to K1 connector on the next card. K1 and K2 connectors on the same PCIe-8158 are connected internally.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

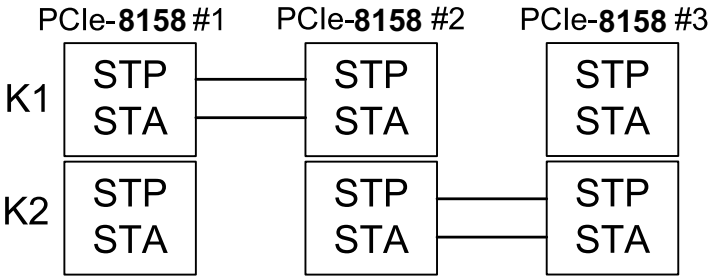


Figure 2-17: STA & STP Connection

External start and stop signals can initiate simultaneous cross-card motor operations, when connected to STA and STP pins on the K1 connector.

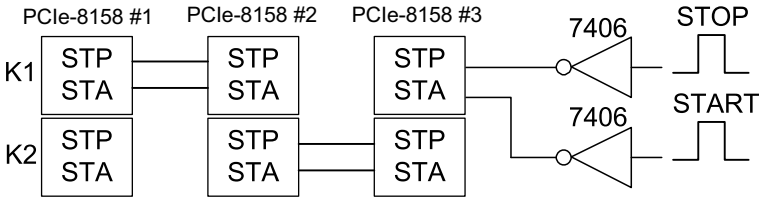


Figure 2-18: STA & STP Connection With External Start/Stop

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.6.14 General Purpose TTL I/O EDI And EDO

32 general purpose TTL digital input/ outputs are provided on CN5.

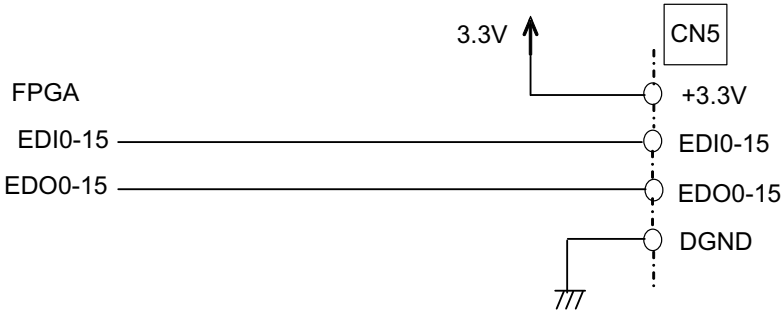


Figure 2-19: EDI And EDO Circuit

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Appendix A MotionCreatorPro

After installing the hardware, it is necessary to correctly configure all cards and check the system. MotionCreatorPro provides simple yet powerful setup, configuration, testing, and debugging for motion control systems using 8158 cards.

MotionCreatorPro functions under Windows 7/8.1. Recommended screen resolution is 1024x768.

A.1 About MotionCreatorPro

Before running MotionCreatorPro, please note the following.

1. MotionCreatorPro is written in VB.NET 2003.
2. MotionCreatorPro allows settings and configurations for 8158 cards to be saved. Saved configurations will be automatically loaded the next time MotionCreatorPro is executed. Two files, **8158.ini** and **8158MC.ini**, in the **windows root directory** are used to save all settings and configurations.
3. To duplicate configurations from one system to another, copy 8158.ini and 8158MC.ini into the Windows root directory.
4. If multiple 8158 cards use the same MotionCreatorPro saved configuration files, the DLL function call **_8158_config_from_file**() can be invoked within a user developed program.

A.2 Initiating MotionCreatorPro

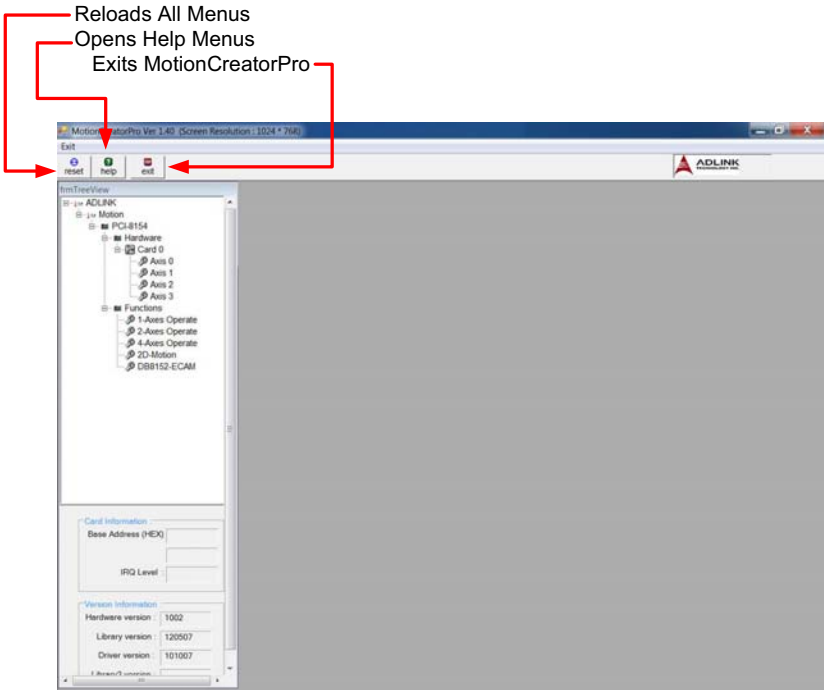
After installing the software drivers for the 8158 in Windows 7/8.1, the MotionCreatorPro program can be located at <chosen path>\ADLINK\8158\\MotionCreatorPro. To execute the program, double click the executable file or use Start>Program Files>ADLINK>8158>MotionCreatorPro.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

A.3 MotionCreatorPro Introduction

Main Menu

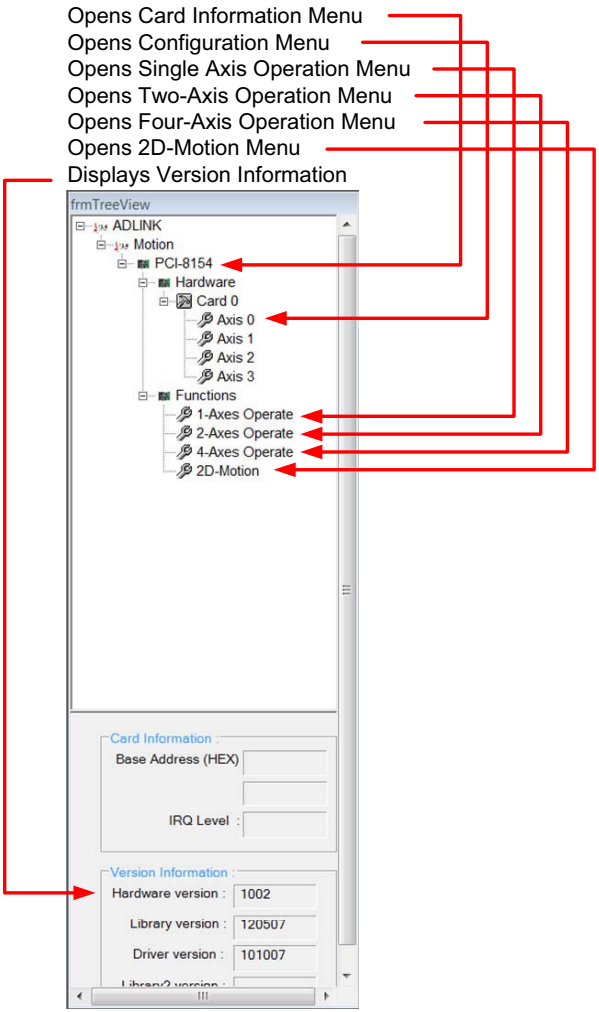
The main menu opens after starting MotionCreatorPro.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

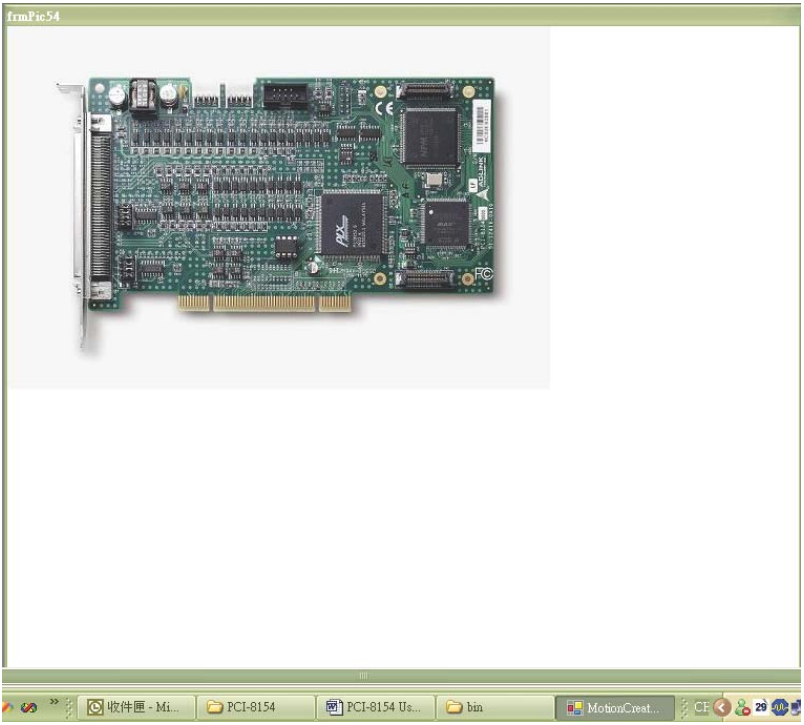
Select Menu

The Select menu appears after starting MotionCreatorPro and enables selection of operating card and axis.



Card Information Menu

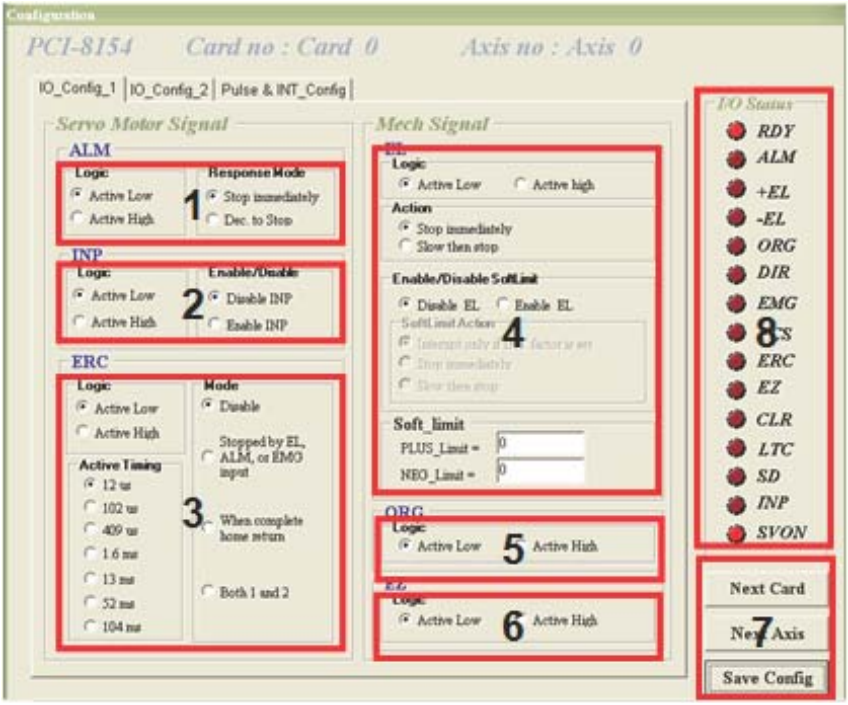
Provides Information about the card:



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Configuration Menu

In the **IO_Config_1** menu, users can configure ALM, INP, ERC, EL, ORG, and EZ, as follows.

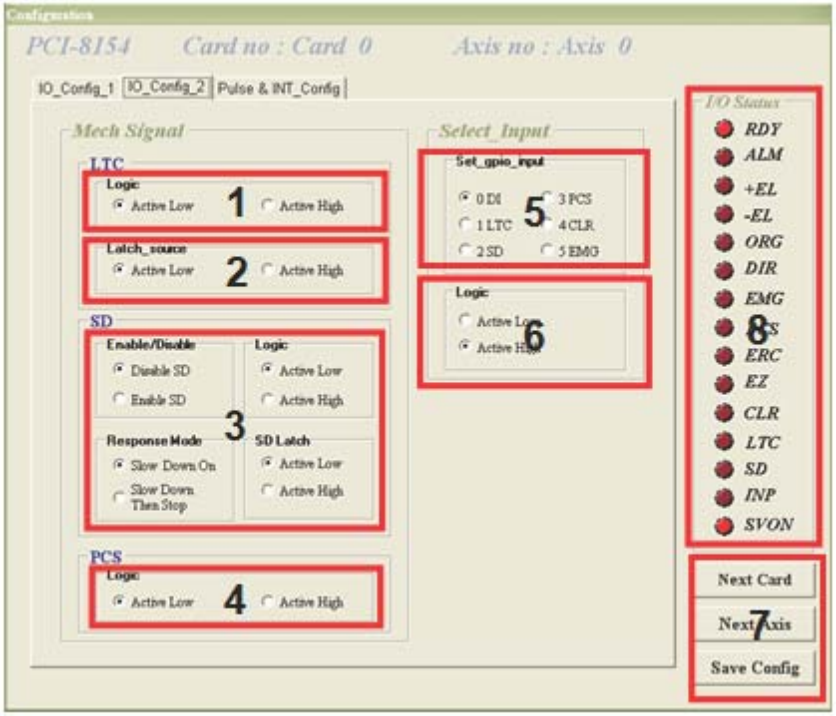


1. **ALM Logic and Response mode**: Select logic and response modes of ALM signal. The related function call is *_8158_set_alm*().
2. **INP Logic and Enable/Disable selection**: Select logic, and Enable/ Disable the INP signal. The related function call is *_8158_set_inp*()
3. **ERC Logic, Active timing and ERC mode**: Select the Logic, Active timing and mode of the ERC signal. The related function call is *_8158_set_erc*().
4. **EL Response mode**: Select the response mode of the EL signal. The related function call is *_8158_set_limit_logic*().

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5. **ORG Logic**: Select the logic of the ORG signal. The related function call is *_8158_set_home_config*().
6. **EZ Logic**: Select the logic of the EZ signal. The related function call is *_8158_set_home_config*().
7. **Buttons**:
   - ▷ **Next Card**: Change operating card.
   - ▷ **Next Axis**: Change operating axis.
   - ▷ **Save Config**: Save current configuration to 8158.ini and 8158MC.ini.
8. **I/O Status**: The status of motion I/O. Light-On means Active, while Light-Off indicates inactive. The related function is *_8158_get_io_status*

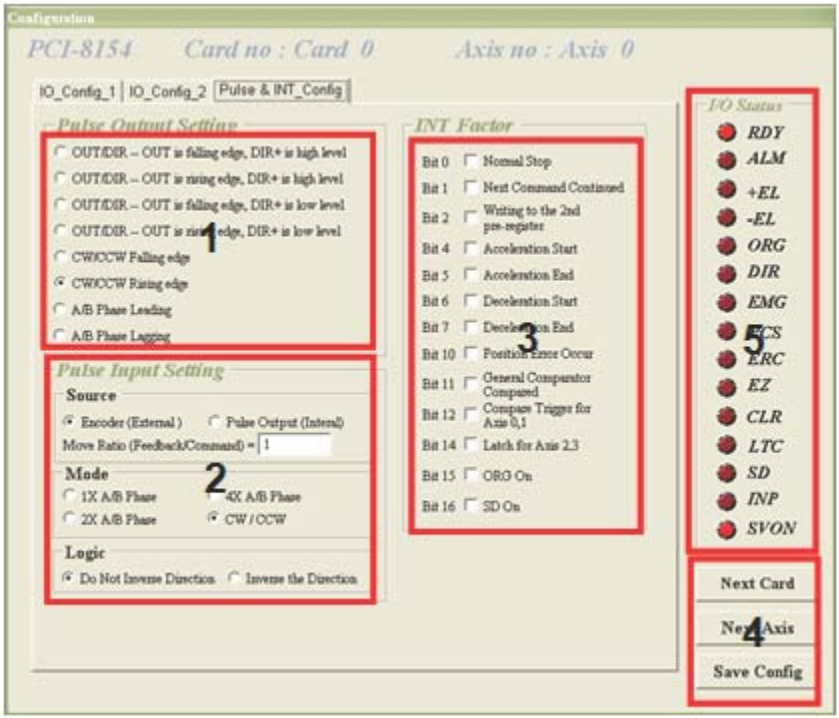
In the **IO_Config_2** menu, users can configure LTC, SD, PCS, and Select_Input.



1. **LTC Logic**: Select the logic of the LTC signal. The related function call is *_8158_set_ltc_logic*().
2. **LTC latch_source**: Select the logic of the latch_source signal. The related function call is *_8158_set_latch_source*().
3. **SD Configuration**: Configure the SD signal. The related function call is *_8158_set_sd*().
4. **PCS Logic**: Select the logic of the SelectNo signal. The related function call is *_8158_set_pcs_logic*().
5. **Set gpio input**: Select the configurations of the gpio input. The related function call is *_8158_set_gpio_input_function*.
6. **Gpio Logic**: Select the logic of the gpio. The related function call is *_8158_set_gpio_input_function*.
7. **Buttons**:
   - ▷ **Next Card**: Change operating card.
   - ▷ **Next Axis**: Change operating axis.
   - ▷ **Save Config**: Save current configuration to 8158.ini And 8158MC.ini.
8. **I/O Status**: Displays I/O status, with lit indicating Active unlit inactive. The related function is *_8158_get_io_status*

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

In the **Pulse & INT_Config** menu, users can configure pulse input/output and move ratio and INT factor.



1. **Pulse Output Mode**: Select the output mode of the pulse signal (OUT/ DIR). The related function call is *_8158_set_pls_outmode*().
2. **Pulse Input**: Sets the configurations of the Pulse input signal(EA/EB). The related function calls are *_8158_set_pls_iptmode*(), *_8158_set_feedback_src*().
3. **INT Factor**: Select factors to initiate the event int. The related function call is *_8158_set_int_factor*().
4. **Buttons**:
   - ▷ **Next Card**: Change operating card.
   - ▷ **Next Axis**: Change operating axis.

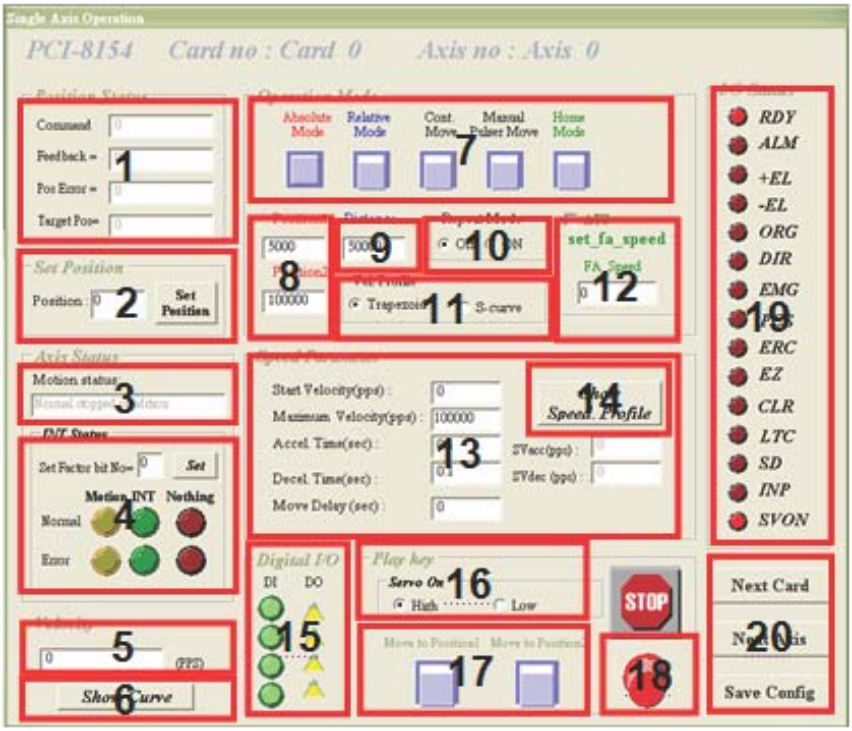
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

▷ **Save Config**: Save current configuration to 8158.ini And 8158MC.ini.

5. **I/O Status**: The status of motion I/O. Light-On means Active, while Light-Off indicates inactive. The related function is *_8158_get_io_status.*

Single Axis Operation Menu

In this menu, users can change the settings for a specific axis, such as velocity mode motion, preset relative/absolute motion, manual pulse move, and home return.



1. **Position**:

▷ **Command**: displays the value of the command counter. The related function is *_8158_get_command*().

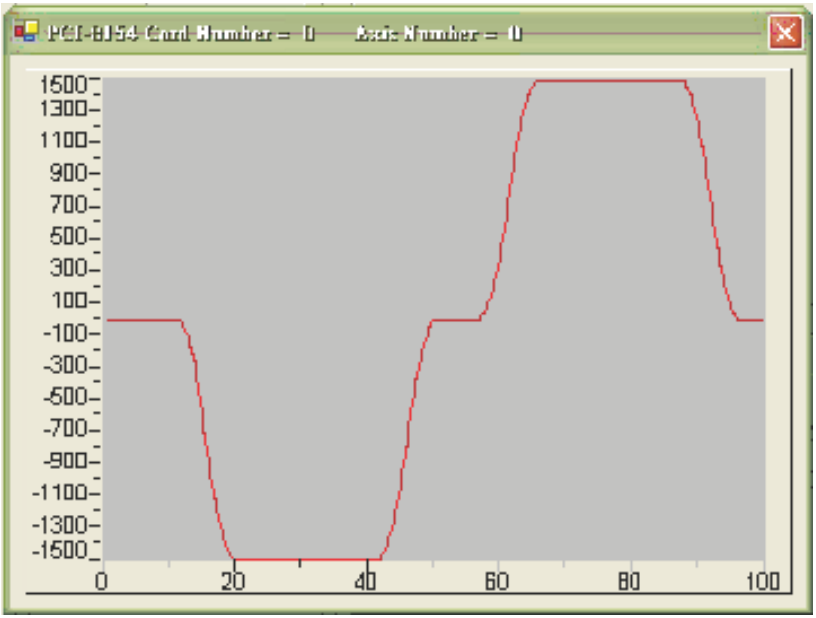
▷ **Feedback**: displays the value of the feedback position counter. The related function is *_8158_get_position*()

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- ▷ **Pos Error**: displays the value of the position error counter. The related function is *_8158_get_error_counter*().
- ▷ **Target Pos**: displays the value of the target position recorder. The related function is *_8158_get_target_pos*().

2. **Position Reset**: Resets all positioning counters to a specified value. The related functions are:
   - ▷ *_8158_set_position*(),
   - ▷ *_8158_set_command*(),
   - ▷ *_8158_reset_error_counter*()
   - ▷ *_8158_reset_target_pos*()
3. **Motion Status**: Displays the returned value of the *_8158_motion_done* function. The related function is *_8158_motion_done*().
4. **INT Status**:
   - ▷ **int_factor bit no**: Set int_factor bit.
   - ▷ **Normal INT:** display of Normal INT status. The related function is *_8158_wait_motion_interrupt* ().
   - ▷ **Error INT**: display of Error INT status. The related function is *_8158_wait_error_interrupt* ().
   - ▷ **GPIO INT**: display of GPIO INT status. The related function is *_8158_wait_gpio_interrupt* ().
5. **Velocity**: The absolute value of velocity in units of PPS. The related function is *_8158_get_current_speed*().
6. **Show Velocity Curve Button**: Clicking this button will open a window showing a velocity vs. time curve. In this curve, every 100ms, a new velocity data point will be added. To close, click again, and to clear data, click the curve.

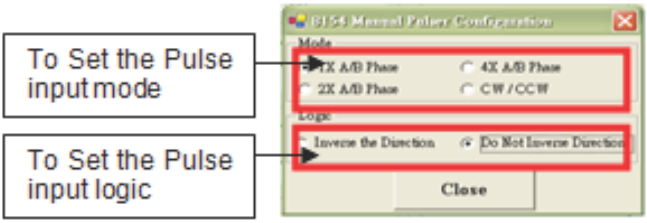
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000



7. **Operating Mode**: Select operating mode.
   - ▷ **Absolute Mode**: “Position1” and “position2” is used as absolution target positions for motion. The related functions are *_8158_start_ta_move*(), *_8158_start_sa_move*().
   - ▷ **Relative Mode**: “Distance” is used as relative displacement for motion. The related function is *_8158_start_tr_move*(), *_8158_start_sr_move*().
   - ▷ **Cont. Move**: Velocity motion mode. The related function is *_8158_tv_move*(), *_8158_start_sv_move*().

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

▷ **Manual manual pulse generator Move**: Manual Pulse motion. Clicking this button will invoke the manual pulse configuration window.



▷ **Home Mode**: Home return motion. Clicking this button will invoke the home move configuration window. The related function is *_8158_set_home_config*(). *If the check box "ATU" is checked, it will execute auto homing when motion starts.*

ERC Output: Select if the ERC signal will be sent when home move completes.

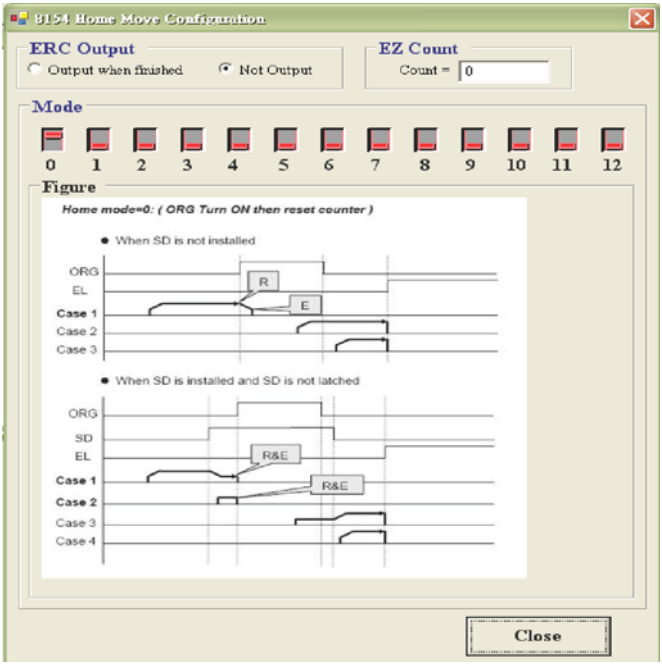
EZ Count: Set the EZ count number, which is effective on certain home return modes.

Mode: Select the home return mode. There are 13 modes available.

Home Mode figure: The figure shown explains the actions of the individual home modes.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Close: Click this button close this window.



8. **Position**: Set the absolute position for “Absolute Mode.” It is only effective when “Absolute Mode” is selected.
9. **Distance**: Set the relative distance for “Relative Mode.” It is only effective when “Relative Mode” is selected.
10. **Repeat Mode**: When “On” is selected, the motion will become repeat mode (**forward<-->backward** or **position1<-->position2**). It is only effective when “Relative Mode” or “Absolute Mode” is selected.
11. **Vel. Profile**: Select the velocity profile. Both Trapezoidal and S-Curve are available for “Absolute Mode,” “Relative Mode,” and “Cont. Move.”
12. **FA Speed/ATU**: Sets the configurations of the FA Speed. The related function calls are

_8158_set_fa_speed(). If the check box “ATU” is checked, it will execute auto homing when motion starts.

13. **Motion Parameters**: Set the parameters for single axis motion. This parameter is meaningless if “Manual manual pulse generator Move” is selected, since the velocity and moving distance is decided by pulse input.

    ▷ **Start Velocity**: Set the start velocity of motion in units of PPS. In “Absolute Mode” or “Relative Mode,” only the value is effective. For example, -100.0 is the same as 100.0. In “Cont. Move,” both the value and sign are effective. –100.0 means 100.0 in the minus direction.

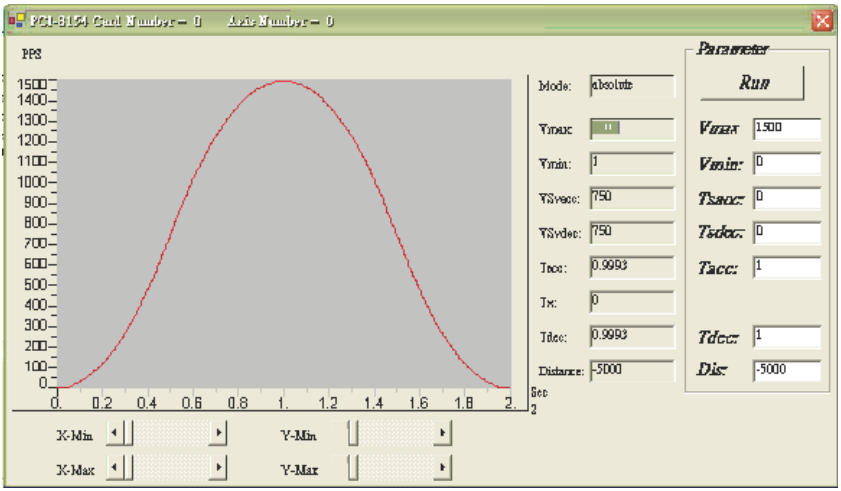
    ▷ **Maximum Velocity**: Set the maximum velocity of motion in units of PPS. In “Absolute Mode” or “Relative Mode,” only the value is effective. For example, -5000.0 is the same as 5000.0. In “Cont. Move,” both the value and

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

sing is effective. –5000.0 means 5000.0 in the minus direction.

- ▷ **Accel. Time**: Set the acceleration time in units of second.
- ▷ **Decel. Time**: Set the deceleration time in units of second.
- ▷ **SVacc**: Set the S-curve range during acceleration in units of PPS.
- ▷ **SVdec**: Set the S-curve range during deceleration in units of PPS.
- ▷ **Move Delay**: This setting is effective only when repeat mode is set “On.” It will cause the 8158 to delay for a specified time before it continues to the next motion.

14. **Speed_Profile**: Clicking this button will show the Speed Profile.



15. **Digital I/O**: Display and set Digital I/O. The related functions are: *_8158_get_gpio_output*(), *_8158_get_gpio_input*(), *_8158_set_gpio_output*().
16. **Servo On**: Set the SVON signal output status. The related function is _8158_set_servo().
17. **Play Key**:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Left play button: Causes the 8158 start to outlet pulses according to previous setting.

- ▷ In “Absolute Mode,” it causes the axis to move to position1.
- ▷ In “Relative Mode,” it causes the axis to move forward.
- ▷ In “Cont. Move,” it causes the axis to start to move according to the velocity setting.
- ▷ In “Manual manual pulse generator Move,” it causes the axis to go into pulse move. The speed limit is the value set by “Maximum Velocity.”

Right play button: Causes the 8158 start to outlet pulses according to previous setting.

- ▷ In “Absolute Mode,” it causes the axis to move to position.
- ▷ In “Relative Mode,” it causes the axis to move backwards.
- ▷ In “Cont. Move,” it causes the axis to start to move according to the velocity setting, but in the opposite direction.
- ▷ In “Manual manual pulse generator Move,” it causes the axis to go into pulse move. The speed limit is the value set by “Maximum Velocity.”

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

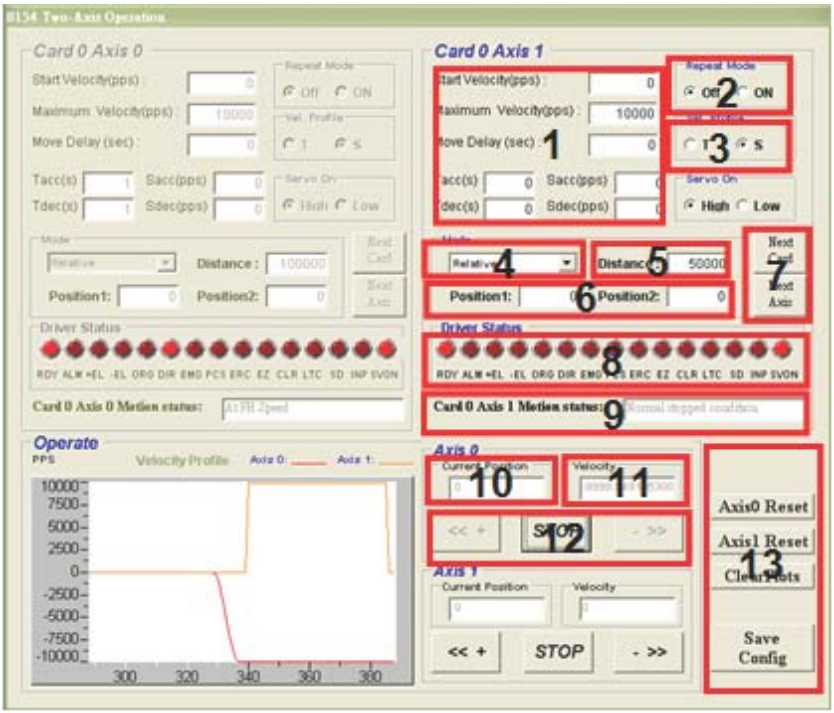
18. **Stop Button**: Causes the 8158 to decelerate and stop. The deceleration time is defined in “Decel. Time.” The related function is _8158_sd_stop().
19. **I/O Status**: The status of motion I/O. Light-On means Active, while Light-Off indicates inactive. The related function is _8158_get_io_status().
20. **Buttons**:
    - ▷ **Next Card**: Change operating card.
    - ▷ **Next Axis**: Change operating axis.
    - ▷ **Save Config**: Save current configuration to 8158.ini And 8158MC.ini.
    - ▷ **Close**: Close the menu.

Two-Axis and Four-Axis Operation Menu

In two-axis and four-axis menu, users can change the settings of two or four selected axis, including velocity mode motion, preset relative/absolute motion. User can discover two-axis and four-

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

axis operation menu are similarly, that's because we just introduce two-axis menu.



1. **Motion Parameters**: Set the parameters for single axis motion. This parameter is meaningless if “Manual manual pulse generator Move” is selected, since the velocity and moving distance is decided by pulse input.
   - ▷ **Start Velocity**: Set the start velocity of motion in units of PPS. In “Absolute Mode” or “Relative Mode,” only the value is effective. For example, -100.0 is the same as 100.0.
   - ▷ **Maximum Velocity**: Set the maximum velocity of motion in units of PPS. In “Absolute Mode” or “Relative Mode,” only the value is effective. For example, -5000.0 is the same as 5000.0.
   - ▷ **Tacc**: Set the acceleration time in units of second.
   - ▷ **Tdec**: Set the deceleration time in units of second.
   - ▷ **Sacc**: Set the S-curve range during acceleration in units of PPS.
   - ▷ **Sdec**: Set the S-curve range during deceleration in units of PPS.
2. **Repeat Mode**: When “On” is selected, the motion will become repeat mode (**forward<-->backward** or **position1<-->position2**). It is only effective when “Relative Mode” or “Absolute Mode” is selected.
3. **Vel. Profile**: Select the velocity profile. Both Trapezoidal and S-Curve are available for “Absolute Mode,” “Relative Mode,” and “Cont. Move.”
4. **Operating Mode**: Select operating mode.
   - ▷ Absolute Mode: “Position1” and “position2” is used as absolution target positions for motion. The related functions are *_8158_start_ta_move*(), *_8158_start_sa_move*().
   - ▷ **Relative Mode**: “Distance” is used as relative displacement for motion. The related function is *_8158_start_tr_move*(), *_8158_start_sr_move*().
5. **Distance**: Set the relative distance for “Relative Mode.” It is only effective when “Relative Mode” is selected.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6. **Position**: Set the absolute position for “Absolute Mode.” It is only effective when “Absolute Mode” is selected.
7. **Buttons**:
   - ▷ **Next Card**: Change operating card.
   - ▷ **Next Axis**: Change operating axis.
8. **I/O Status**: The status of motion I/O. Light-On means Active, while Light-Off indicates inactive. The related function is *_8158_get_io_status*().
9. **Motion status**: Displays the returned value of the *_8158_motion_done* function. The related function is *_8158_motion_done*().
10. **Current Position**:
    - ▷ **Command**: displays the value of the command counter. The related function is *_8158_get_position*().
11. **Velocity**: The absolute value of velocity in units of PPS. The related function is *_8158_get_current_speed*().
12. **Play Key**:

Left play button: Causes the 8158 start to outlet pulses according to previous setting.

- ▷ In “Absolute Mode,” it causes the axis to move to position1.
- ▷ In “Relative Mode,” it causes the axis to move forward.
- ▷ Right play button: Causes the 8158 start to outlet pulses according to previous setting.
- ▷ In “Absolute Mode,” it causes the axis to move to position2.
- ▷ In “Relative Mode,” it causes the axis to move backwards.

Stop Button: Causes the 8158 to decelerate and stop. The deceleration time is defined in “Decel. Time.” The related function is *_8158_sd_stop*().

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

13. **Buttons**:
    - ▷ **Axis0 Reset**: clicking this button will set all positioning counters of selected axis to zero. The related functions are:

      _8158_set_position()

      _8158_set_command()

      _8158_reset_error_counter()

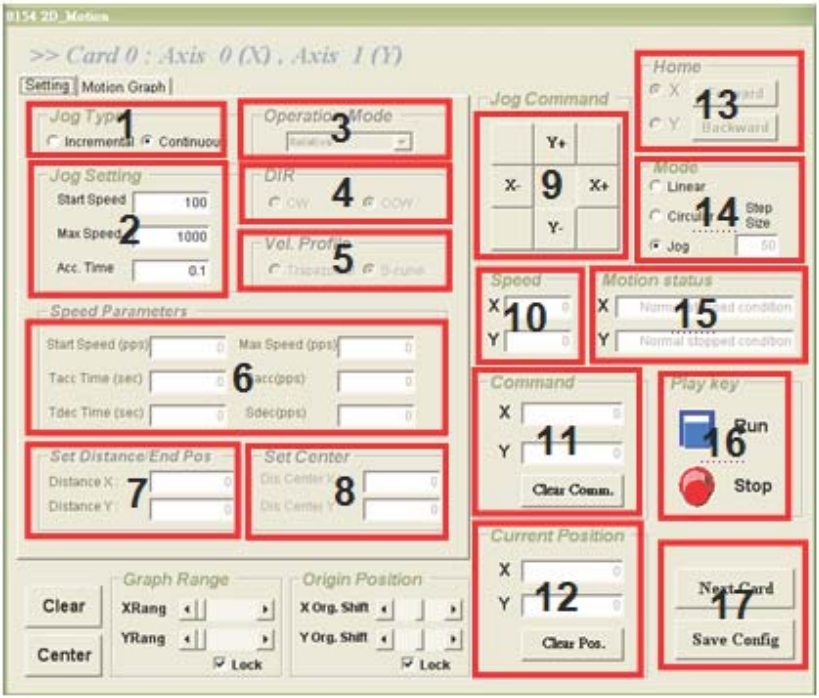
      _8158_reset_target_pos()
    - ▷ **Axis1 Reset**: clicking this button will set all positioning counters of selected axis to zero.
    - ▷ **ClearPlots**: Clear the Motion Graph.
    - ▷ **Save Config**: Save current configuration to 8158.ini and 8158MC.ini.
    - ▷ **Close**: Close the menu.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

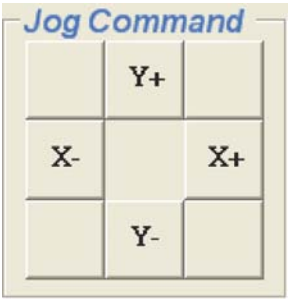
2D_Motion Menu

Press 2-D button in operating window will enter this window. This is for 2-D motion test. It includes the following topics:

- ▷ Linear Interpolation
- ▷ Circular Interpolation
- ▷ Incremental Jog
- ▷ Continuous Jog
- ▷ Other Control Objects



1. **Jog Type**:
   - ▷ **Continuous Jog**: Pressing a directional button directs the axis to continuously accelerate and releasing the button stops the axis immediately.



   - ▷ **Incremental Jog**: When one directional button is pressed, the axis steps the distance entered.



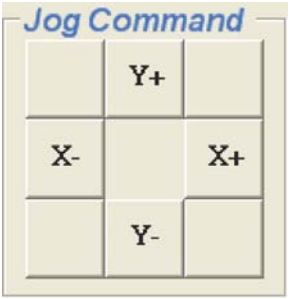
2. **Jog Setting**: Sets parameters for single axis motion. Inactive if "Jog Mode" is selected, since velocity and moving distance are decided by pulse input.
   - ▷ **Start Velocity**: Sets start velocity of motion in PPS.
   - ▷ **Maximum Velocity**: Sets the maximum velocity of motion in PPS.
   - ▷ **Tacc**: Sets the acceleration time in seconds.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3. **Operating Mode**: Selects operating mode.
   - ▷ **Absolute Mode**: “Position” is used as absolution target position for motion when “Linear Interpolation Mode” is selected. “ABS EndPos” and “ABS Center” are used as absolution target positions for motion when “Circular Interpolation Mode” is selected. The related functions are _8158_start_ta_move(), _8158_start_sa_move().
   - ▷ **Relative Mode**: “Distance” is used as absolution target position for motion when “Linear Interpolation Mode” is selected. “Dis EndPos” and “Dis Center” are used as absolution target positions for motion when “Circular Interpolation Mode” is selected. The related function is _8158_start_tr_move(), _8158_start_sr_move().
4. **DIR**: Specified direction of arc, CW/CCW, only effective when “Circular Interpolation Mode” is selected.
5. **Vel. Profile**: Selects velocity profile. Both Trapezoidal and S-Curve are available for “Linear Interpolation Mode” and “Circular Interpolation Mode”.
6. **Speed Parameters**: Sets the parameters for single axis motion. This parameter is meaningless if “Linear Interpolation Mode” or “Circular Interpolation Mode” is selected, since the velocity and moving distance is decided by pulse input.
   - ▷ **Start Velocity**: Sets the start velocity of motion in units of PPS.
   - ▷ **Maximum Velocity**: Sets the maximum velocity of motion in units of PPS.
   - ▷ **Accel. Time**: Sets acceleration time in seconds.
   - ▷ **Decel. Time**: Sets deceleration time in seconds.
   - ▷ **SVacc**: Sets S-curve range during acceleration in PPS.
   - ▷ **SVdec**: Sets the S-curve range during deceleration in PPS.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

7. Set Distance/End Pos: Sets the absolution target positions or relative distance for “Linear Interpolation Mode” . Sets the position end of arc for “Circular Interpolation Mode”. Available for “Linear Interpolation Mode” and “Circular Interpolation Mode”.
8. **Set Center**: Sets the position of center for “Circular Interpolation Mode”. Only available when “Circular Interpolation Mode” is selected.
9. **Jog Command**: Pressing one directional button generates a move.



10. **Velocity**: The absolute value of velocity in PPS. The related function is *_8158_get_current_speed*().
11. **Interpolation Command**:
    - ▷ **Command**: displays the value of the command counter. The related function is *_8158_get_command*().
12. **Current Position**:
    - ▷ **Feedback**: displays the value of the feedback position counter. The related function is *_8158_get_position*().
13. **Home Mode**: Home return motion. Clicking this button invokes the home move configuration window. The related function is *_8158_set_home_config*(). Two home return buttons in the lower left corner of the window return to the origin.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

14. **Mode**:

- ▷ **Linear Interpolation**: After motion parameters have been set in “Motion Parameters Setting Frame”, the destination can be entered in this frame. Run launches linear interpolation motion.
- ▷ **Circular Interpolation**: Circular interpolation mode has arc degree, division axis, and optimize option parameters in “Motion Parameters Setting Frame”.

After setting these parameters, arc center and degree can be entered in “Play Key Frame”. Run launches circular interpolation motion.

15. **Motion status**: Displays the returned value of the *_8158_motion_done* function. The related function is *_8158_motion_done*().

16. **Play Key**:

Play button: Causes the 8158 to outlet pulses according to previous settings.

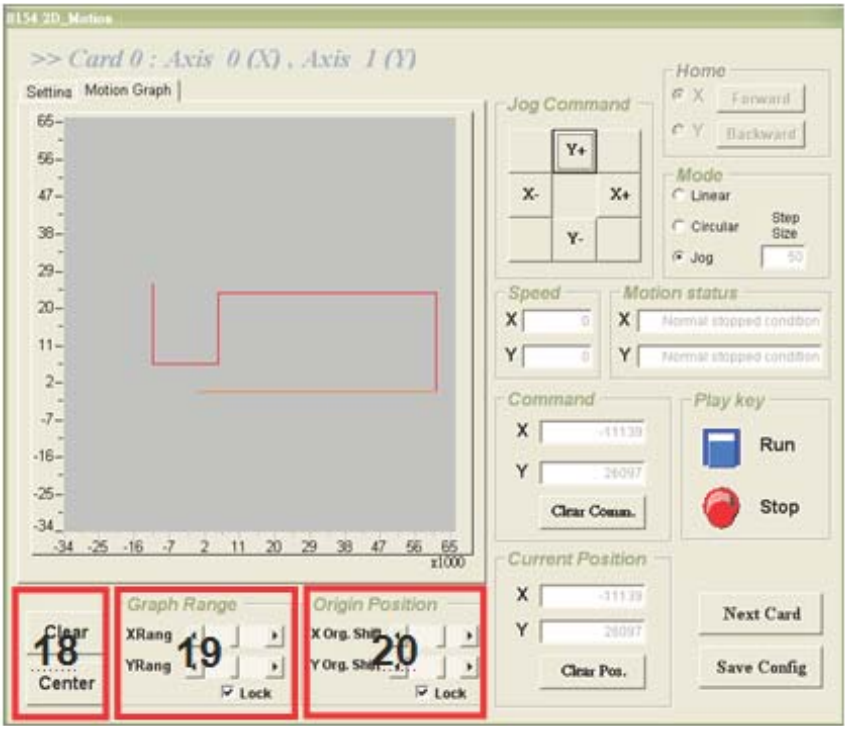
- ▷ In “Linear Mode,” the axis moves to Distance. The related function is *_8158_start_tr_move_xy*, *_8158_start_sr_move_xy*.
- ▷ In “Circular Mode,” the axis moves to Distance(By Pos/Dist(pulse)).The related function is *_8158_start_tr_arc_xy*, *_8158_start_sr_arc_xy*.

Stop Button: Causes the 8158 to decelerate and stop, with deceleration time is defined in “Decel. Time.” The related function is _8158_sd_stop().

17. **Buttons**:

- ▷ **Next Card**: Changes operating card.
- ▷ **Save Config**: Saves current configuration to 8158.ini And 8158MC.ini.
- ▷ **Close**: Closes the menu.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000



18. **Graph Range Frame**:
    - ▷ **Clear**: Clears the Motion Graph.
    - ▷ **Center**: Displays the Motion Graph in center position.
19. **Graph Range**: controls X or Y axis display range.
20. **Origin Position**: allows panning of display location.

Help Menu

Right clicking shows Help Information.



This page intentionally left blank.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Appendix B Function Library Reference

The functions listed support development of programs in C, C++, and Visual Basic. If Delphi is used as the programming environment, it is necessary to transform the header files, pci_8158.h manually.

B.1 Data Types

The library uses these data types in **pci_8158.h**. We suggest you use these data types in your application programs. Data type names and ranges in C/C++ are as follows

Type	Description	Range
U8	8-bit ASCII character	0 to 255
I16	16-bit signed integer	-32768 to 32767
U16	16-bit unsigned integer	0 to 65535
I32	32-bit signed long integer	-2147483648 to 2147483647
U32	32-bit unsigned long integer	0 to 4294967295
F32	32-bit single-precision floating-point	-3.402823E38 to 3.402823E38
F64	64-bit double-precision floating-point	-1.797683134862315E308 to 1.797683134862315E309
Boolean	Boolean logic value	TRUE, FALSE

Function Naming

The PCIe-8158 software drivers use full names to identify the function. Naming conventions are as follows.

In a ‘C’ programming environment:

_{hardware_model}_{action_name}. e.g. **`_8158_initial()`**.

To differentiate between a C library and a VB library, a capital “B” is placed at the beginning of each function name e.g. **`B_8158_initial()`**.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

B.2 List of Functions

Category	Function
System & Initialization	_8158_initial
	_8158_close
	_8158_get_version
	_8158_set_security_key
	_8158_check_security_key
	_8158_reset_security_key
	_8158_config_from_file
Pulse Input/Output Configuration	_8158_set_pls_outmode
	_8158_set_pls_iptmode
	_8158_set_feedback_src
Velocity Mode Motion	_8158_set_pls_outmode
	_8158_set_pls_iptmode
	_8158_set_feedback_src
	_8158_set_pls_outmode
	_8158_set_pls_iptmode
	_8158_set_feedback_src
	_8158_set_pls_outmode
Single Axis Position Mode	_8158_start_tr_move
	_8158_start_ta_move
	_8158_start_sr_move
	_8158_start_sa_move
	_8158_set_move_ratio
	_8158_position_override

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Category	Function
Linear Interpolated Motion	_8158_start_tr_move_xy
	_8158_start_ta_move_xy
	_8158_start_sr_move_xy
	_8158_start_sa_move_xy
	_8158_start_tr_move_zu
	_8158_start_ta_move_zu
	_8158_start_sr_move_zu
	_8158_start_tr_line2
	_8158_start_ta_line2
	_8158_start_sr_line2
	_8158_start_sa_line2
	_8158_start_tr_line3
	_8158_start_ta_line3
	_8158_start_sr_line3
	_8158_start_sa_line3
	_8158_start_tr_line4
	_8158_start_ta_line4
	_8158_start_sr_line4
	_8158_start_sa_line4
Circular Interpolated Motion	_8158_start_tr_arc_xy
	_8158_start_ta_arc_xy
	_8158_start_sr_arc_xy
	_8158_start_sa_arc_xy
	_8158_start_tr_arc_zu
	_8158_start_ta_arc_zu
	_8158_start_sr_arc_zu
	_8158_start_sa_arc_zu
	_8158_start_tr_arc2
	_8158_start_ta_arc2
	_8158_start_sr_arc2
	_8158_start_sa_arc2

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Category	Function
Helical Interpolation Motion	_8158_start_tr_helical
	_8158_start_ta_helical
	_8158_start_sr_helical
	_8158_start_sa_helical
Home Return Mode	_8158_set_home_config
	_8158_home_move
	_8158_home_search
Manual Pulse Generator Mode	_8158_set_pulser_iptmode
	_8158_disable_pulser_input
	_8158_pulser_vmove
	_8158_pulser_pmove
	_8158_set_pulser_ratio
Motion Status, Section	_8158_motion_done
Motion Interface I/O	_8158_set_servo
	_8158_set_pcs_logic
	_8158_set_pcs
	_8158_set_clr_mode
	_8158_set_inp
	_8158_set_alm
	_8158_set_erc
	_8158_set_erc_out
	_8158_clr_erc
	_8158_set_sd
	_8158_enable_sd
	_8158_set_limit_logic
	_8158_set_limit_mode
	_8158_get_io_status
Interrupt Control	_8158_int_control
	_8158_wait_error_interrupt
	_8158_wait_motion_interrupt
	_8158_set_motion_int_factor

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Category	Function
Position Control and Counters	_8158_get_position
	_8158_set_position
	_8158_get_command
	_8158_set_command
	_8158_get_error_counter
	_8158_reset_error_counter
	_8158_get_general_counter
	_8158_set_general_counter
	_8158_get_target_pos
	_8158_reset_target_pos
	_8158_get_res_distance
	_8158_set_res_distance
	_8158_get_ring_counter
	_8158_set_ring_counter
	_8158_escape_home
Position Compare and Latch	_8158_set_trigger_logic
	_8158_set_error_comparator
	_8158_set_general_comparator
	_8158_set_trigger_comparator
	_8158_set_latch_source
	_8158_set_ltc_logic
	_8158_get_latch_data
Continuous Motion	_8158_set_continuous_move
	_8158_check_continuous_buffer
	_8158_dwell_move
Multi-Axis Simultaneous Operation	_8158_set_tr_move_all
	_8158_set_ta_move_all
	_8158_set_sr_move_all
	_8158_set_sa_move_all
	_8158_start_move_all
	_8158_stop_move_all

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Category	Function
General Purpose I/O	_8158_set_gpio_output
	_8158_get_gpio_output
	_8158_get_gpio_input
	_8158_set_gpio_input_function
Soft Limit	_8158_disable_soft_limit
	_8158_enable_soft_limit
	_8158_set_soft_limit
Backlash Compensation/ Vibration Suppression	_8158_backlash_comp
	_8158_suppress_vibration
	_8158_set_fa_speed
Speed Profile Calculation	_8158_get_tr_move_profile
	_8158_get_ta_move_profile
	_8158_get_sr_move_profile
	_8158_get_sa_move_profile
Extended General Purpose TTL Input/Output	_8158_set_gpio_output_ex
	_8158_get_gpio_output_ex
	_8158_get_gpio_input_ex
	_8158_set_gpio_output_ex_CH
	_8158_get_gpio_output_ex_CH
	_8158_get_gpio_input_ex_CH

B.3 System and Initialization

@ Name

_8158_initial – Card initialization

_8158_close – Card close

_8158_get_version – Check hardware and software version information

_8158_set_security_key – Set the security password

_8158_check_security_key – Check the security password

_8158_reset_security_key – Rest the security password to default

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_config_from_file – Config PCIe-8158 settings from file

@ Description

_8158_initial:

This function is used to initialize an 8158 card without assigning the hardware resources. All 8158 cards must be initialized by this function before calling other functions in your applications. By setting the parameter "**Manual_ID**", user can choose the type that the card's ID is assigned manually or automatically.

_8158_close:

This function is used to close 8158 card and release its resources, which should be called at the end of your applications.

_8158_get_version:

Lets users read back the firmware's, driver's and DLL's version information.

_8158_set_security_key:

This function is used to set a security code to the PCIe card.

See also:

_8158_check_security_key
_8158_reset_security_key

_8158_check_security_key:

This function is used to verify the security code which the user set by the function "**_8158_set_security_key**".

See also:

_8158_set_security_key
_8158_reset_security_key

_8158_reset_security_key:

By this function, Users can reset the security code on the PCIe card to default value. The default security code is0.

See also:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
_8158_check_security_key
_8158_set_security_key
```

_8158_config_from_file:

This function is used to load the configuration of the PCIe-8158 according to specified file. By using MotionCreatorPro, user could test and configure the 8158 correctly. After saving the configuration, the file would be existed in user's system directory as 8158.ini.

When this function is executed, all 8158 cards in the system will be configured as the following functions were called according to parameters recorded in 8158.ini.

```
_8158_set_limit_logic
_8158_set_pcs_logic
_8158_set_ltc_logic
_8158_set_inp
_8158_set_erc
_8158_set_alm
_8158_set_pls_iptmode
_8158_set_pls_outmode
_8158_set_move_ratio
_8158_set_latch_source
_8158_set_feedback_src
_8158_set_home_config
_8158_set_soft_limit
_8158_set_fa_speed
_8158_set_sd
```

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_initial(U16 *CardID_InBit, I16
     Manual_ID);
I16 _8158_close(void);
I16 _8158_get_version(I16 card_id, I16
     *firmware_ver, I32 *driver_ver, I32
     *dll_ver);
I16 _8158_set_security_key(I16 card_id, I16
     old_secu_code, I16 new_secu_code);
I16 _8158_check_security_key(I16 card_id, I16
     secu_code);
I16 _8158_reset_security_key(I16 card_id);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_config_from_file();
```

Visual Basic 6 (Windows 7/8.1)

```
B_8158_initial(CardID_InBit As Integer, ByVal
     Manual_ID As Integer) As Integer
B_8158_close() As Integer
B_8158_get_version(ByVal card_id As Integer,
     firmware_ver As Integer, driver_ver As Long,
     dll_ver As Long) As Integer
B_8158_set_security_key(ByVal card_id As Integer,
     ByVal old_secu_code As Integer, ByVal
     new_secu_code As Integer) As Integer
B_8158_check_security_key(ByVal card_id As
     Integer, ByVal secu_code As Integer)As
     Integer
B_8158_reset_security_key(ByVal card_id As
     Integer);
B_8158_config_from_file() As Integer
```

@ Argument

`CardID_InBit`:

`Manual_ID`: Enable the On board dip switch (SW1) to decide the Card ID

The CardID could be decided by :

0: the sequence of PCIe slot.

1: on board DIP switch (SW1).

`card_id`: Specify the PCIe-8158 card index. The card_id could be decided by DIP switch (SW1) or depend on slot sequence. Please refer to **`_8158_initial`**().

`firmware_ver`: The current firmware version.

`driver_ver`: The current device driver version.

`dll_ver`: The current DLL library version.

`old_secu_code`: Old security code.

`new_secu_code`: New security code.

`secu_code`: security code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

B.4 Pulse Input/Output Configuration

@ Name

_8158_set_pls_iptmode – Set the configuration for feedback pulse input.

_8158_set_pls_outmode – Set the configuration for pulse command output.

_8158_set_feedback_src – Enable/Disable the external feedback pulse input

@ Description

_8158_set_pls_iptmode:

Configure the input modes of external feedback pulses. There are 4 types for feedback pulse input. Note that this function makes sense only when the Src parameter in **_8158_set_feedback_src**() function is enabled.

_8158_set_pls_outmode:

Configure the output modes of command pulses. There are 6 modes for command pulse output.

_8158_set_feedback_src:

If external encoder feedback is available in the system, set the Src parameter in this function to an **Enabled** state. Then, the internal 28-bit up/down counter will count according to the configuration of the **_8158_set_pls_iptmode**() function. Else, the counter will count the command pulse output.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_pls_iptmode(I16 AxisNo, I16
     pls_iptmode, I16 pls_logic);
I16 _8158_set_pls_outmode(I16 AxisNo, I16
     pls_outmode);
I16 _8158_set_feedback_src(I16 AxisNo, I16 Src);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic6 (Windows 7/8.1)

```
B_8158_set_pls_iptmode(ByVal AxisNo As Integer,
     ByVal pls_iptmode As Integer, ByVal
     pls_logic As Integer) As Integer
B_8158_set_pls_outmode(ByVal AxisNo As Integer,
     ByVal pls_outmode As Integer) As Integer
B_8158_set_feedback_src(ByVal AxisNo As Integer,
     ByVal Src As Integer) As Integer
```

@ Argument

AxisNo: Axis number designated to configure the pulse input/output.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

pls_iptmode: Encoder feedback pulse input mode setting (EA/EB signals).

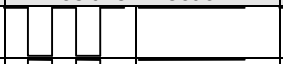
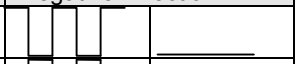
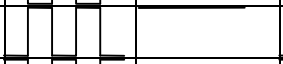
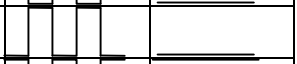
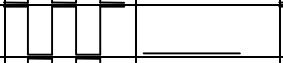
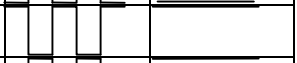
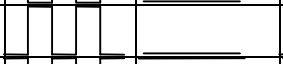
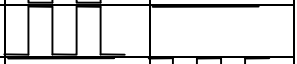
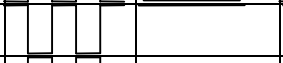
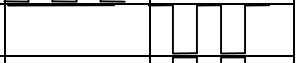
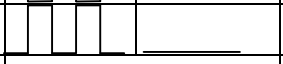
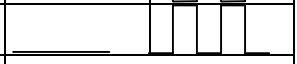
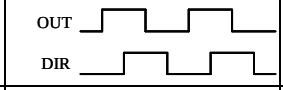
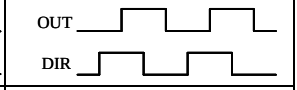
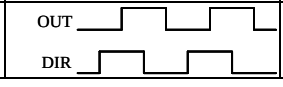
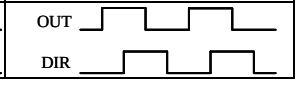
Value	Meaning
0	1X A/B
1	2X A/B
2	4X A/B
3	CW/CCW

pls_logic: Logic of encoder feedback pulse.

Value	Meaning
0	Not inverse direction
1	inverse direction

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

pls_outmode: Setting of command pulse output mode.

Value	Type	Positive Direction	Negative Direction
0	OUT/DIR		
1	OUT/DIR		
2	OUT/DIR		
3	OUT/DIR		
4	CW / CCW		
5	CW / CCW		
6	AB		
7	AB		

Src: Counter source

Value	Meaning
0	External signal feedback
1	Command pulse

B.5 Velocity mode motion

@ Name

_8158_tv_move – Accelerate an axis to a constant velocity with trapezoidal profile

_8158_sv_move – Accelerate an axis to a constant velocity with S-curve profile

_8158_emg_stop – Immediately stop

_8158_sd_stop – Decelerate to stop

_8158_get_current_speed – Get current speed

_8158_speed_override – Change speed on the fly

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Description

_8158_tv_move:

This function is to accelerate an axis to the specified constant velocity with a trapezoidal profile. The axis will continue to travel at a constant velocity until the velocity is changed or the axis is commanded to stop. The direction is determined by the sign of the velocity parameter.

_8158_sv_move:

This function is to accelerate an axis to the specified constant velocity with a S-curve profile. The axis will continue to travel at a constant velocity until the velocity is changed or the axis is commanded to stop. The direction is determined by the sign of velocity parameter.

_8158_emg_stop:

This function is used to immediately stop an axis. This function is also useful when a preset move (both trapezoidal and S-curve motion), manual move, or home return function is performed.

_8158_sd_stop:

This function is used to decelerate an axis to stop with a trapezoidal or S-curve profile. This function is also useful when a preset move (both trapezoidal and S-curve motion), manual move, or home return function is performed. Note: The velocity profile is decided by original motion profile.

_8158_get_current_speed:

This function is used to read the current pulse output rate (pulse/sec) of a specified axis. It is applicable in any time in any operating mode.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_tv_move(I16 AxisNo, F64 StrVel, F64
     MaxVel, F64 Tacc);
I16 _8158_sv_move(I16 AxisNo, F64 StrVel, F64
     MaxVel, F64 Tacc, F64 SVacc);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_emg_stop(I16 AxisNo);
I16 _8158_sd_stop(I16 AxisNo, F64 Tdec);
I16 _8158_get_current_speed(I16 AxisNo, F64
    *speed)
```

Visual Basic6 (Windows 7/8.1)

```
B_8158_tv_move(ByVal AxisNo As Integer, ByVal
    StrVel As Double, ByVal MaxVel As Double,
    ByVal Tacc As Double) As Integer
B_8158_sv_move(ByVal AxisNo As Integer, ByVal
    StrVel As Double, ByVal MaxVel As Double,
    ByVal Tacc As Double, ByVal SVacc As Double)
    As Integer
B_8158_emg_stop(ByVal AxisNo As Integer) As
    Integer
B_8158_sd_stop(ByVal AxisNo As Integer, ByVal
    Tdec As Double) As Integer
B_8158_get_current_speed(ByVal AxisNo As Integer,
    ByRef Speed As Double) As Integer
```

@ Argument

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

StrVel: Starting velocity in units of pulse per second

MaxVel: Maximum velocity in units of pulse per second

Tacc: Specified acceleration time in units of second

SVacc: Specified velocity interval in which S-curve acceleration is performed.

Note: SVacc = 0, for pure S-Curve

Tdec: specified deceleration time in units of second

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

***Speed**: Variable to get current speed (pulse/sec).

B.6 Single Axis Position Mode

@ Name

_8158_start_tr_move – Begin a relative trapezoidal profile move

_8158_start_ta_move – Begins absolute trapezoidal profile move

_8158_start_sr_move – Begin a relative S-curve profile move

_8158_start_sa_move – Begins absolute S-curve profile move

_8158_set_move_ratio – Set the ration of command pulse and feedback pulse

_8158_position_override – Change position on the fly

_8158_set_max_override_speed – Set the maximum over-ride speed

@ Description

General:

The moving direction is determined by the sign of the Pos or Dist parameter. If the moving distance is too short to reach the specified velocity, the controller will automatically lower the **MaxVel**, and the **Tacc**, **Tdec**, **VSacc**, and **VSdec** will also become shorter while dV/dt(acceleration / deceleration) and d(dV/dt)/dt (jerk) are keep unchanged.

_8158_start_tr_move:

This function causes the axis to accelerate form a starting velocity (StrVel), rotate at constant velocity (MaxVel), and decelerate to stop at the **relative distance** with **trapezoidal** profile. The acceleration (Tacc) and deceleration (Tdec) time is specified independently–it does not let the program wait for motion completion but immediately returns control to the program.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_start_ta_move:

This function causes the axis to accelerate from a starting velocity (StrVel), rotate at constant velocity (MaxVel), and decelerates to stop at the specified **absolute position** with **trapezoidal** profile. The acceleration (Tacc) and deceleration (Tdec) time is specified independently. This command does not let the program wait for motion completion, but immediately returns control to the program.

_8158_start_sr_move:

This function causes the axis to accelerate from a starting velocity (StrVel), rotate at constant velocity (MaxVel), and decelerates to stop at the **relative distance** with **S-curve** profile. The acceleration (Tacc) and deceleration (Tdec) time is specified independently. This command does not let the program wait for motion completion, but immediately returns control to the program.

_8158_start_sa_move:

This function causes the axis to accelerate from a starting velocity (StrVel), rotate at constant velocity, and decelerates to stop at the specified **absolute position** with **S-curve** profile. The acceleration and deceleration time is specified independently. This command does not let the program wait for motion completion but immediately returns control to the program.

_8158_set_move_ratio:

This function configures scale factors for the specified axis. Usually, the axes only need scale factors if their mechanical resolutions are different. For example, if the resolution of feedback sensors is two times resolution of command pulse, then the parameter "**move_ratio**" could be set as 2.

_8158_position_override:

This function is used to change target position on the fly. There are some limitations on this function. Please refer to section 4.2.15 before use it.

_8158_set_max_override_speed:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_start_tr_move(I16 AxisNo, F64 Dist, F64
      StrVel, F64 MaxVel, F64 Tacc, F64 Tdec);
I16 _8158_start_ta_move(I16 AxisNo, F64 Pos, F64
      StrVel, F64 MaxVel, F64 Tacc, F64 Tdec);
I16 _8158_start_sr_move(I16 AxisNo, F64 Dist, F64
      StrVel, F64 MaxVel, F64 Tacc, F64 Tdec, F64
      SVacc, F64 SVdec);
I16 _8158_start_sa_move(I16 AxisNo, F64 Pos, F64
      StrVel, F64 MaxVel, F64 Tacc, F64 Tdec, F64
      SVacc, F64 SVdec);
I16 _8158_set_move_ratio(I16 AxisNo, F64
      move_ratio);
I16 _8158_position_override(I16 AxisNo, F64
      NewPos);
I16 _8158_set_max_override_speed(I16 AxisNo, F64
      OvrdSpeed, I16 Enable);
```

Visual Basic6 (Windows 7/8.1)

```
B_8158_start_tr_move(ByVal AxisNo As Integer,
      ByVal Dist As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double) As Integer
B_8158_start_ta_move(ByVal AxisNo As Integer,
      ByVal Pos As Double, ByVal StrVel As Double,
      ByVal MaxVel As Double, ByVal Tacc As
      Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_move(ByVal AxisNo As Integer,
      ByVal Dist As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double, ByVal SVacc
      As Double, ByVal SVdec As Double) As Integer
B_8158_start_sa_move(ByVal AxisNo As Integer,
      ByVal Pos As Double, ByVal StrVel As Double,
      ByVal MaxVel As Double, ByVal Tacc As
      Double, ByVal Tdec As Double, ByVal SVacc As
      Double, ByVal SVdec As Double) As Integer
B_8158_set_move_ratio(ByVal AxisNo As Integer,
      ByVal move_ratio As Double) As Integer
B_8158_position_override(ByVal AxisNo As Integer,
      ByVal NewPos As Double) As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
B_8158_set_max_override_speed(ByVal AxisNo As
     Integer, ByVal OvrdSpeed As Double, ByVal
     Enable As Integer) As Integer
```

@ Argument

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

Dist: Specified relative distance to move (unit: pulse)

Pos: Specified absolute position to move (unit: pulse)

StrVel: Starting velocity of a velocity profile in units of pulse per second

MaxVel: Maximum velocity in units of pulse per second

Tacc: Specified acceleration time in units of seconds

Tdec: Specified deceleration time in units of seconds

SVacc: Specified velocity interval in which S-curve acceleration is performed.

Note: SVacc = 0, for pure S-Curve. For more details, see section 2.4.4

SVdec: specified velocity interval in which S-curve deceleration is performed.

Note: SVdec = 0, for pure S-Curve. For more details, see section 4.2.4

Move_ratio: ratio of (feedback resolution)/(command resolution), should not be 0

NewPos: specified new absolute position to move

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

B.7 Linear Interpolated Motion

@ Name

_8158_start_tr_move_xy – Begin a relative 2-axis linear interpolation for X & Y axis with trapezoidal profile

_8158_start_ta_move_xy – Begins absolute 2-axis linear interpolation for X & Y axis with trapezoidal profile

_8158_start_sr_move_xy –Begin a relative 2-axis linear interpolation for X & Y axis with S-curve profile

_8158_start_sa_move_xy –Begins absolute 2-axis linear interpolation for X & Y axis with S-curve profile

_8158_start_tr_move_zu – Begin a relative 2-axis linear interpolation for Z & U axis with trapezoidal profile

_8158_start_ta_move_zu – Begins absolute 2-axis linear interpolation for Z & U axis with trapezoidal profile

_8158_start_sr_move_zu –Begin a relative 2-axis linear interpolation for Z & U axis with S-curve profile

_8158_start_sa_move_zu –Begins absolute 2-axis linear interpolation for Z & U axis with S-curve profile

_8158_start_ta_line2 – Begins absolute 2-axis linear interpolation for any 2 of 4 axes, with trapezoidal profile

_8158_start_sr_line2 – Begin a relative 2-axis linear interpolation for any 2 of 4 axes, with S-curve profile

_8158_start_sa_line2 – Begins absolute 2-axis linear interpolation for any 2 of 4 axes, with S-curve profile

_8158_start_tr_line3 – Begin a relative 3-axis linear interpolation for any 3 of 4 axes, with trapezoidal profile

_8158_start_ta_line3 – Begin a absolute 3-axis linear interpolation for any 3 of 4 axes, with trapezoidal profile

_8158_start_sr_line3 – Begin a relative 3-axis linear interpolation for any 3 of 4 axes, with S-curve profile

_8158_start_sa_line3 – Begin a absolute 3-axis linear interpolation for any 3 of 4 axes, with S-curve profile

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_start_tr_line4 – Begin a relative 8-axis linear interpolation for any 4 of 4 axes, with trapezoidal profile

_8158_start_ta_line4 – Begin a absolute 8-axis linear interpolation for any 4 of 4 axes, with trapezoidal profile

_8158_start_sr_line4 – Begin a relative 8-axis linear interpolation for any 4 of 4 axes, with S-curve profile

_8158_start_sa_line4 – Begin a absolute 8-axis linear interpolation for any 4 of 4 axes, with S-curve profile

@ Description

These functions perform linear interpolation motion with different profile. Detail Comparisons of those functions are described by follow table.

Function	Total axes	Velocity Profile	Relative Absolute	Target Axes
_8158_start_tr_move_xy	2	T	R	Axes 0 & 1
_8158_start_ta_move_xy	2	T	A	Axes 0 & 1
_8158_start_sr_move_xy	2	S	R	Axes 0 & 1
_8158_start_sa_move_xy	2	S	A	Axes 0 & 1
_8158_start_tr_move_zu	2	T	R	Axes 2 & 3
_8158_start_ta_move_zu	2	T	A	Axes 2 & 3
_8158_start_sr_move_zu	2	S	R	Axes 2 & 3
_8158_start_sa_move_zu	2	S	A	Axes 2 & 3

Function	Total axes	Velocity Profile	Relative Absolute	Target Axes
_8158_start_tr_line2	2	T	R	Any 2 of 4 axes
_8158_start_ta_line2	2	T	A	Any 2 of 4 axes
_8158_start_sr_line2	2	S	R	Any 2 of 4 axes
_8158_start_sa_line2	2	S	A	Any 2 of 4 axes

Note: The target two axes of linear interpolation are the 2 of 4 axes on a card.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Function	Total axes	Velocity Profile	Relative Absolute	Target Axes
_8158_start_tr_line3	3	T	R	Any 3 of 4 axes
_8158_start_ta_line3	3	T	A	Any 3 of 4 axes
_8158_start_sr_line3	3	S	R	Any 3 of 4 axes
_8158_start_sa_line3	3	S	A	Any 3 of 4 axes

Note: The target 3 axes of linear interpolation are the 3 of 4 axes on a card.

Function	Total axes	Velocity Profile	Relative Absolute	Target Axes
_8158_start_tr_line4	4	T	R	Any 4 of 4 axes
_8158_start_ta_line4	4	T	A	Any 4of 4 axes
_8158_start_sr_line4	4	S	R	Any 4 of 4 axes
_8158_start_sa_line4	4	S	A	Any 4 of 4 axes

Note: The target 4 axes of linear interpolation are the 4 of 4 axes on a card.

Velocity profile:

T: trapezoidal profile

S: S-curve profile

Relative / Absolute:

R: Relative distance

A: Absoulte position

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_start_tr_move_xy(I16 Card_id, F64
     DistX, F64 DistY, F64 StrVel, F64 MaxVel,
     F64 Tacc, F64 Tdec);
I16 _8158_start_ta_move_xy(I16 Card_id, F64 PosX,
     F64 PosY, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_start_sr_move_xy(I16 Card_id, F64
     DistX, F64 DistY, F64 StrVel, F64 MaxVel,
     F64 Tacc, F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_sa_move_xy(I16 Card_id, F64 PosX,
     F64 PosY, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_tr_move_zu(I16 Card_id, F64
     DistX, F64 DistY, F64 StrVel, F64 MaxVel,
     F64 Tacc, F64 Tdec);
I16 _8158_start_ta_move_zu(I16 Card_id, F64 PosX,
     F64 PosY, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec);
I16 _8158_start_sr_move_zu(I16 Card_id, F64
     DistX, F64 DistY, F64 StrVel, F64 MaxVel,
     F64 Tacc, F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_sa_move_zu(I16 Card_id, F64 PosX,
     F64 PosY, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_tr_line2(I16 *AxisArray, F64
     *DistArray, F64 StrVel, F64 MaxVel, F64
     Tacc, F64 Tdec);
I16 _8158_start_ta_line2(I16 *AxisArray, F64
     *PosArray, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec);
I16 _8158_start_sr_line2(I16 *AxisArray, F64
     *DistArray, F64 StrVel, F64 MaxVel, F64
     Tacc, F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_sa_line2(I16 *AxisArray, F64
     *PosArray, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_tr_line3(I16 *AxisArray, F64
     *DistArray, F64 StrVel, F64 MaxVel, F64
     Tacc, F64 Tdec);
I16 _8158_start_ta_line3(I16 *AxisArray, F64
     *PosArray, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec);
I16 _8158_start_sr_line3(I16 *AxisArray, F64
     *DistArray, F64 StrVel, F64 MaxVel, F64
     Tacc, F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_sa_line3(I16 *AxisArray, F64
     *PosArray, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec, F64 SVacc, F64 SVdec);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_start_tr_line4(I16 *AxisArray, F64
     *DistArray, F64 StrVel, F64 MaxVel, F64
     Tacc, F64 Tdec);
I16 _8158_start_ta_line4(I16 *AxisArray, F64
     *PosArray, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec);
I16 _8158_start_sr_line4(I16 *AxisArray, F64
     *DistArray, F64 StrVel, F64 MaxVel, F64
     Tacc, F64 Tdec, F64 SVacc, F64 SVdec);
I16 _8158_start_sa_line4(I16 *AxisArray, F64
     *PosArray, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec, F64 SVacc, F64 SVdec);
```

Visual Basic6 (Windows 7/8.1)

```
B_8158_start_tr_move_xy(ByVal Card_id As Integer,
     ByVal DistX As Double, ByVal DistY As
     Double, ByVal StrVel As Double, ByVal MaxVel
     As Double, ByVal Tacc As Double, ByVal Tdec
     As Double) As Integer
B_8158_start_ta_move_xy(ByVal Card_id As Integer,
     ByVal PosX As Double, ByVal PosY As Double,
     ByVal StrVel As Double, ByVal MaxVel As
     Double, ByVal Tacc As Double, ByVal Tdec As
     Double) As Integer
B_8158_start_sr_move_xy(ByVal Card_id As Integer,
     ByVal DistX As Double, ByVal DistY As
     Double, ByVal StrVel As Double, ByVal MaxVel
     As Double, ByVal Tacc As Double, ByVal Tdec
     As Double, ByVal SVacc As Double, ByVal
     SVdec As Double) As Integer
B_8158_start_sa_move_xy(ByVal Card_id As Integer,
     ByVal PosX As Double, ByVal PosY As Double,
     ByVal StrVel As Double, ByVal MaxVel As
     Double, ByVal Tacc As Double, ByVal Tdec As
     Double, ByVal SVacc As Double, ByVal SVdec
     As Double) As Integer
B_8158_start_tr_move_zu(ByVal Card_id As Integer,
     ByVal DistX As Double, ByVal DistY As
     Double, ByVal StrVel As Double, ByVal MaxVel
     As Double, ByVal Tacc As Double, ByVal Tdec
     As Double);
B_8158_start_ta_move_zu(ByVal Card_id As Integer,
     ByVal PosX As Double, ByVal PosY As Double,
     ByVal StrVel As Double, ByVal MaxVel As
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
      Double, ByVal Tacc As Double, ByVal Tdec As
      Double) As Integer
B_8158_start_sr_move_zu(ByVal Card_id As Integer,
      ByVal DistX As Double, ByVal DistY As
      Double, ByVal StrVel As Double, ByVal MaxVel
      As Double, ByVal Tacc As Double, ByVal Tdec
      As Double, ByVal SVacc As Double, ByVal
      SVdec As Double) As Integer
B_8158_start_sa_move_zu(ByVal Card_id As Integer,
      ByVal PosX As Double, ByVal PosY As Double,
      ByVal StrVel As Double, ByVal MaxVel As
      Double, ByVal Tacc As Double, ByVal Tdec As
      Double, ByVal SVacc As Double, ByVal SVdec
      As Double) As Integer
B_8158_start_tr_line2(AxisArray() As Integer,
      DistArray() As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double) As Integer
B_8158_start_ta_line2(AxisArray() As Integer,
      PosArray() As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_line2((AxisArray() As Integer,
      DistArray() As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double, ByVal Svacc
      As Double, ByVal Svdec As Double) As Integer
B_8158_start_sa_line2(AxisArray() As Integer,
      PosArray() As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double, ByVal Svacc
      As Double, ByVal Svdec As Double) As Integer
B_8158_start_tr_line3(AxisArray() As Integer,
      DistArray() As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double) As Integer
B_8158_start_ta_line3(AxisArray() As Integer,
      PosArray() As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_line3((AxisArray() As Integer,
      DistArray() As Double, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        As Double, ByVal Tdec As Double, ByVal Svacc
        As Double, ByVal Svdec As Double) As Integer
B_8158_start_sa_line3(AxisArray() As Integer,
        PosArray() As Double, ByVal StrVel As
        Double, ByVal MaxVel As Double, ByVal Tacc
        As Double, ByVal Tdec As Double, ByVal Svacc
        As Double, ByVal Svdec As Double) As Integer
B_8158_start_tr_line4(AxisArray() As Integer,
        DistArray() As Double, ByVal StrVel As
        Double, ByVal MaxVel As Double, ByVal Tacc
        As Double, ByVal Tdec As Double) As Integer
B_8158_start_ta_line4(AxisArray() As Integer,
        PosArray() As Double, ByVal StrVel As
        Double, ByVal MaxVel As Double, ByVal Tacc
        As Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_line4((AxisArray() As Integer,
        DistArray() As Double, ByVal StrVel As
        Double, ByVal MaxVel As Double, ByVal Tacc
        As Double, ByVal Tdec As Double, ByVal Svacc
        As Double, ByVal Svdec As Double) As Integer
B_8158_start_sa_line4(AxisArray() As Integer,
        PosArray() As Double, ByVal StrVel As
        Double, ByVal MaxVel As Double, ByVal Tacc
        As Double, ByVal Tdec As Double, ByVal Svacc
        As Double, ByVal Svdec As Double) As Integer
```

@ Argument

`card_id`: Specify the PCIe-8158 card index. The card_id could be decided by DIP switch (SW1) or depend on slot sequence. Please refer to **`_8158_initial`**().

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

DistX: specified relative distance of **axis 0** to move (unit: pulse).

DistY: specified relative distance of **axis 1** to move (unit: pulse).

PosX: specified absolute position of **axis 0** to move (unit: pulse).

PosY: specified absolute position of **axis 1** to move (unit: pulse).

StrVel: Starting velocity of a velocity profile in units of pulse per second.

MaxVel: Maximum velocity in units of pulse per second.

Tacc: Specified acceleration time in units of seconds.

Tdec: Specified deceleration time in units of seconds.

SVacc: Specified velocity interval in which S-curve acceleration is performed.

Note: SVacc = 0, for pure S-Curve. For more details, see section 4.2.4

SVdec: specified velocity interval in which S-curve deceleration is performed.

Note: SVdec = 0, for pure S-Curve. For more details, see section 4.2.4

***AxisArray**: Array of axis number to perform interpolation.

Example:

```
I16 AxisArray[2] = {0, 3}; //axis 0, & axis 3
     (correct)
I16 AxisArray[3] = {0,2, 3}; //axis 0, 2 & 3
     (correct)
I16 AxisArray[2] = {1, 6}; //axis 1, & axis 6
     (incorrect)
```

***DistArray**: Array of relative distance for linear interpolation.

Example:

```
I16 AxisArray[2] = {0, 3}; //axis 0, & axis 3
F64 DistArray[2] = {1000.0, 2000.0} //for axis 0
     & 3
```

***PosArray**: Array of absolute position for linear interpolation.

Example:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 AxisArray[3] = {0,2, 3}; //axis 0, 2 & 3
F64 PosArray[3] = {200.0, 300.0, 400.0} //
     absolute position for axis 0, 2 & 3
```

B.8 Circular Interpolation Motion

@ Name

_8158_start_tr_arc_xy – Begin a T-curve relative circular interpolation for X & Y axis

_8158_start_ta_arc_xy – Begin a T-curve absolute circular interpolation for X & Y axis

_8158_start_sr_arc_xy – Begin a S-curve relative circular interpolation for X & Y axis

_8158_start_sa_arc_xy –Begin a S-curve absolute circular interpolation for X & Y axis

_8158_start_tr_arc_zu – Begin a T-curve relative circular interpolation for Z & U axis

_8158_start_ta_arc_zu – Begin a T-curve absolute circular interpolation for Z & U axis

_8158_start_sr_arc_zu – Begin a S-curve relative circular interpolation for Z & U axis

_8158_start_sa_arc_zu –Begin a S-curve absolute circular interpolation for Z & U axis

_8158_start_tr_arc2 – Begin a T-curve relative circular interpolation for any 2 of 4 axes

_8158_start_ta_arc2 – Begin a T-curve absolute circular interpolation for any 2 of 4 axes

_8158_start_sr_arc2 – Begin a S-curve relative circular interpolation for any 2 of 4 axes

_8158_start_sa_arc2 – Begin a S-curve absolute circular interpolation for any 2 of 4 axes

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Description

Those functions perform Circular interpolation motion with different profile. Detail Comparisons of those functions are described by follow table.

Function	Total axes	Velocity Profile	Relative Absolute	Target Axes
_8158_start_tr_arc_xy	2	trapezoidal	R	Axes 0 & 1
_8158_start_ta_arc_xy	2	trapezoidal	A	Axes 0 & 1
_8158_start_sr_arc_xy	2	S-curve	R	Axes 0 & 1
_8158_start_sa_arc_xy	2	S-curve	A	Axes 0 & 1
_8158_start_tr_arc_zu	2	trapezoidal	R	Axes 2 & 3
_8158_start_ta_arc_zu	2	trapezoidal	A	Axes 2 & 3
_8158_start_sr_arc_zu	2	S-curve	R	Axes 2 & 3
_8158_start_sa_arc_zu	2	S-curve	A	Axes 2 & 3
_8158_start_tr_arc2	2	trapezoidal	R	Any 2 of 4 Axis
_8158_start_ta_arc2	2	trapezoidal	A	Any 2 of 4 Axis
_8158_start_sr_arc2	2	S-curve	R	Any 2 of 4 Axis
_8158_start_sa_arc2	2	S-curve	A	Any 2 of 4 Axis

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_start_tr_arc_xy(I16 card_id, F64
     OffsetCx, F64 OffsetCy, F64 OffsetEx, F64
     OffsetEy, I16 CW_CCW, F64 StrVel,F64
     MaxVel,F64 Tacc,F64 Tdec);
I16 _8158_start_ta_arc_xy(I16 card_id, F64 Cx,
     F64 Cy, F64 Ex, F64 Ey, I16 CW_CCW, F64
     StrVel,F64 MaxVel,F64 Tacc,F64 Tdec);
I16 _8158_start_sr_arc_xy(I16 card_id, F64
     OffsetCx, F64 OffsetCy, F64 OffsetEx, F64
     OffsetEy, I16 CW_CCW, F64 StrVel,F64
     MaxVel,F64 Tacc,F64 Tdec,F64 SVacc,F64
     SVdec);
I16 _8158_start_sa_arc_xy(I16 card_id, F64 Cx,
     F64 Cy, F64 Ex, F64 Ey, I16 CW_CCW, F64
     StrVel,F64 MaxVel,F64 Tacc,F64 Tdec,F64
     SVacc,F64 SVdec);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_start_tr_arc_zu(I16 card_id, F64
     OffsetCx, F64 OffsetCy, F64 OffsetEx, F64
     OffsetEy, I16 CW_CCW, F64 StrVel,F64
     MaxVel,F64 Tacc,F64 Tdec);
I16 _8158_start_ta_arc_zu(I16 card_id, F64 Cx,
     F64 Cy, F64 Ex, F64 Ey, I16 CW_CCW, F64
     StrVel,F64 MaxVel,F64 Tacc,F64 Tdec);
I16 _8158_start_sr_arc_zu(I16 card_id, F64
     OffsetCx, F64 OffsetCy, F64 OffsetEx, F64
     OffsetEy, I16 CW_CCW, F64 StrVel,F64
     MaxVel,F64 Tacc,F64 Tdec,F64 SVacc,F64
     SVdec);
I16 _8158_start_sa_arc_zu(I16 card_id, F64 Cx,
     F64 Cy, F64 Ex, F64 Ey, I16 CW_CCW, F64
     StrVel,F64 MaxVel,F64 Tacc,F64 Tdec,F64
     SVacc,F64 SVdec);
I16 _8158_start_tr_arc2(I16 *AxisArray, F64
     *OffsetCenter, F64 *OffsetEnd, I16 CW_CCW,
     F64 StrVel,F64 MaxVel,F64 Tacc,F64 Tdec);
I16 _8158_start_ta_arc2(I16 *AxisArray, F64
     *CenterPos, F64 *EndPos, I16 CW_CCW, F64
     StrVel,F64 MaxVel,F64 Tacc,F64 Tdec);
I16 _8158_start_sr_arc2(I16 *AxisArray, F64
     *OffsetCenter, F64 *OffsetEnd, I16 CW_CCW,
     F64 StrVel,F64 MaxVel,F64 Tacc,F64 Tdec, F64
     SVacc,F64 SVdec);
I16 _8158_start_sa_arc2(I16 *AxisArray, F64
     *CenterPos, F64 *EndPos, I16 CW_CCW, F64
     StrVel,F64 MaxVel, F64 Tacc, F64 Tdec, F64
     SVacc, F64 SVdec);
```

Visual Basic6 (Windows 7/8.1)

```
B_8158_start_tr_arc_xy( ByVal card_id As Integer,
     ByVal OffsetCx As Double, ByVal OffsetCy As
     Double, ByVal OffsetEx As Double, ByVal
     OffsetEy As Double, ByVal CW_CCW As Integer,
     ByVal StrVel As Double, ByVal MaxVel As
     Double, ByVal Tacc As Double, ByVal Tdec As
     Double);
B_8158_start_ta_arc_xy(ByVal card_id As Integer,
     ByVal Cx As Double, ByVal Cy As Double,
     ByVal Ex As Double, ByVal Ey As Double,
     ByVal CW_CCW As Integer, ByVal StrVel As
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_arc_xy(ByVal card_id As Integer,
      ByVal OffsetCx As Double, ByVal OffsetCy As
      Double, ByVal OffsetEx As Double, ByVal
      OffsetEy As Double, ByVal CW_CCW As Integer,
      ByVal StrVel As Double, ByVal MaxVel As
      Double, ByVal Tacc As Double, ByVal Tdec As
      Double, ByVal Svacc As Double, ByVal Svdec
      As Double) As Integer
B_8158_start_sa_arc_xy(ByVal card_id As Integer,
      ByVal Cx As Double, ByVal Cy As Double,
      ByVal Ex As Double, ByVal Ey As Double,
      ByVal CW_CCW As Integer, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double, ByVal Svacc
      As Double, ByVal Svdec As Double) As Integer
B_8158_start_tr_arc_zu( ByVal card_id As Integer,
      ByVal OffsetCx As Double, ByVal OffsetCy As
      Double, ByVal OffsetEx As Double, ByVal
      OffsetEy As Double, ByVal CW_CCW As Integer,
      ByVal StrVel As Double, ByVal MaxVel As
      Double, ByVal Tacc As Double, ByVal Tdec As
      Double);
B_8158_start_ta_arc_zu(ByVal card_id As Integer,
      ByVal Cx As Double, ByVal Cy As Double,
      ByVal Ex As Double, ByVal Ey As Double,
      ByVal CW_CCW As Integer, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
      As Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_arc_zu(ByVal card_id As Integer,
      ByVal OffsetCx As Double, ByVal OffsetCy As
      Double, ByVal OffsetEx As Double, ByVal
      OffsetEy As Double, ByVal CW_CCW As Integer,
      ByVal StrVel As Double, ByVal MaxVel As
      Double, ByVal Tacc As Double, ByVal Tdec As
      Double, ByVal Svacc As Double, ByVal Svdec
      As Double) As Integer
B_8158_start_sa_arc_zu(ByVal card_id As Integer,
      ByVal Cx As Double, ByVal Cy As Double,
      ByVal Ex As Double, ByVal Ey As Double,
      ByVal CW_CCW As Integer, ByVal StrVel As
      Double, ByVal MaxVel As Double, ByVal Tacc
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        As Double, ByVal Tdec As Double, ByVal Svacc
        As Double, ByVal Svdec As Double) As Integer
B_8158_start_tr_arc2(AxisArray() As Integer,
        OffsetCenter() As Double, OffsetEnd() As
        Double, Byval CW_CCW As Integer, ByVal
        StrVel As Double , ByVal MaxVel As Double,
        ByVal Tacc As Double, ByVal Tdec As Double)
        As Integer
B_8158_start_ta_arc2(AxisArray() As Integer,
        CenterPos() As Double, EndPos() As Double,
        Byval CW_CCW As Integer, ByVal StrVel As
        Double , ByVal MaxVel As Double, ByVal Tacc
        As Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_arc2(AxisArray() As Integer,
        OffsetCenter() As Double, OffsetEnd() As
        Double, Byval CW_CCW As Integer, ByVal
        StrVel As Double , ByVal MaxVel As Double,
        ByVal Tacc As Double, ByVal Tdec As Double,
        ByVal Svacc As Double, ByVal Svdec As
        Double) As Integer
B_8158_start_sa_arc2(AxisArray() As Integer,
        CenterPos() As Double, EndPos() As Double,
        Byval CW_CCW As Integer, ByVal StrVel As
        Double , ByVal MaxVel As Double, ByVal Tacc
        As Double, ByVal Tdec As Double, ByVal Svacc
        As Double, ByVal Svdec As Double) As Integer
```

@ Argument

`card_id`: Specify the PCIe-8158 card index. The card_id could be decided by DIP switch (SW1) or depend on slot sequence. Please refer to **`_8158_initial`**().

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

card_id	Physical axis	AxisNo
1	0	4
	1	5
	…	…

OffsetCx: X-axis (first axis of target axes) offset to center

OffsetCy: Y-axis (second axis of target axes) offset to center

OffsetEx: X-axis (first axis of target axes) offset to end of arc

OffsetEy: Y-axis offset to end of arc

Cx: X-axis (first axis of target axes) absolute position of center of arc

Cy: Y-axis (second axis of target axes) absolute position of center of arc

Ex: X-axis (first axis of target axes) absolute position of end of arc

Ey: Y-axis (second axis of target axes) absolute position of end of arc

CW_CCW: Specified direction of arc

Value	Meaning
0	Clockwise(cw)
1	Counterclockwise(ccw)

StrVel: Starting velocity of a velocity profile in units of pulse per second.

MaxVel: Maximum velocity in units of pulse per second.

Tacc: Specified acceleration time in units of seconds.

Tdec: Specified deceleration time in units of seconds.

SVacc: Specified velocity interval in which S-curve acceleration is performed.

Note: SVacc = 0, for pure S-Curve. For more details, see section 4.2.4

SVdec: specified velocity interval in which S-curve deceleration is performed.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Note: SVdec = 0, for pure S-Curve. For more details, see section 4.2.4

***AxisArray**: Array of axis number to perform interpolation.

Example:

```
I16 AxisArray[2] = {0, 3}; //axis 0, & axis 3
     (correct)
I16 AxisArray[2] = {1, 6}; //axis 1, & axis 6
     (incorrect)
```

***OffsetCenter**: Array of the offset to center (relative to the start position)

Example:

```
F64 OffsetCenter[2] = {2000.0, 0.0}; //offset
     from start position(initial point) for 1st &
     2nd axes
```

***OffsetEnd**: Array of the offset to end of arc (relative to the start position)

Example: F

```
64 OffsetEnd[2] = {4000.0, 0.0}; //offset from
     start position(initial point for 1st & 2nd
     axes
```

*CenterPos: Array of the center of arc absolute position

Example:

```
F64 CenterPos[2] = {2000.0, 0.0}; //absolute
     center position for 1st & 2nd axes
```

*EndPos: Array of the end point of arc absolute position

Example:

```
F64 EndPos[2] = {4000.0, 0.0}; //absolute end
     position for 1st & 2nd axes
```

B.9 Helical Interpolation Motion

@ Name

_8158_start_tr_helical – Begin a T-curve relative helical interpolation for X, Y and Z axis

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_start_ta_helical – Begin a T-curve absolute helical interpolation for X, Y and Z axis

_8158_start_sr_helical – Begins S-curve relative helical interpolation for X, Y and Z axis

_8158_start_sa_helical –Begins S-curve absolute helical interpolation for X, Y and Z axis

@ Description

These functions perform helical interpolation motion with different profiles. Detail comparsions of these functions are described in follow table. These function can be used for circular interpolation between the axes X and Y and to adjust the angle of a jig toward an arc tangent point with the Z axis. Also, in this operation, the U axis operation will be a “dummy motion” and it cannot be used for any other purpose.

Function	Total axes	Velocity Profile	Relative Absolute	Target Axes
_8158_start_tr_helical	4	trapezoidal	R	Axes 0, 1 and 2
_8158_start_ta_helical	4	trapezoidal	A	Axes 0, 1 and 2
_8158_start_sr_helical	4	S-curve	R	Axes 0, 1 and 2
_8158_start_sa_helical	4	S-curve	A	Axes 0, 1 and 2

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_start_tr_helical(I16 card_id, F64
     OffsetCx, F64 OffsetCy, F64 OffsetEx, F64
     OffsetEy, F64 PitchDist, I16 CW_CCW, F64
     StrVel, F64 MaxVel, F64 Tacc, F64 Tdec);
I16 _8158_start_ta_helical(I16 card_id, F64 Cx,
     F64 Cy, F64 Ex, F64 Ey, F64 PitchPos, I16
     CW_CCW, F64 StrVel, F64 MaxVel, F64 Tacc,
     F64 Tdec);
I16 _8158_start_sr_helical(I16 card_id, F64
     OffsetCx, F64 OffsetCy, F64 OffsetEx, F64
     OffsetEy, F64 PitchDist, I16 CW_CCW, F64
     StrVel, F64 MaxVel, F64 Tacc, F64 Tdec, F64
     SVacc, F64 SVdec);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_start_sa_helical(I16 card_id, F64 Cx,
      F64 Cy, F64 Ex, F64 Ey, F64 PitchPos, I16
      CW_CCW, F64 StrVel, F64 MaxVel, F64 Tacc,
      F64 Tdec, F64 SVacc, F64 SVdec);
```

Visual Basic6 (Windows 7/8.1)

```
B_8158_start_tr_helical Lib "8158.dll" Alias
      "_8158_start_tr_helical" (ByVal card_id As
      Integer, ByVal OffsetCx As Double, ByVal
      OffsetCy As Double, ByVal OffsetEx As
      Double, ByVal OffsetEy As Double, ByVal
      PitchDist As Double, ByVal CW_CCW As
      Integer, ByVal StrVel As Double, ByVal
      MaxVel As Double, ByVal Tacc As Double,
      ByVal Tdec As Double) As Integer
B_8158_start_ta_helical Lib "8158.dll" Alias
      "_8158_start_ta_helical" (ByVal card_id As
      Integer, ByVal Cx As Double, ByVal Cy As
      Double, ByVal Ex As Double, ByVal Ey As
      Double, ByVal PitchPos As Double, ByVal
      CW_CCW As Integer, ByVal StrVel As Double,
      ByVal MaxVel As Double, ByVal Tacc As
      Double, ByVal Tdec As Double) As Integer
B_8158_start_sr_helical Lib "8158.dll" Alias
      "_8158_start_sr_helical" (ByVal card_id As
      Integer, ByVal OffsetCx As Double, ByVal
      OffsetCy As Double, ByVal OffsetEx As
      Double, ByVal OffsetEy As Double, ByVal
      PitchDist As Double, ByVal CW_CCW As
      Integer, ByVal StrVel As Double, ByVal
      MaxVel As Double, ByVal Tacc As Double,
      ByVal Tdec As Double, ByVal SVacc As Double,
      ByVal SVdec As Double) As Integer
B_8158_start_sa_helical Lib "8158.dll" Alias
      "_8158_start_sa_helical" (ByVal card_id As
      Integer, ByVal Cx As Double, ByVal Cy As
      Double, ByVal Ex As Double, ByVal Ey As
      Double, ByVal PitchPos As Double, ByVal
      CW_CCW As Integer, ByVal StrVel As Double,
      ByVal MaxVel As Double, ByVal Tacc As
      Double, ByVal Tdec As Double, ByVal SVacc As
      Double, ByVal SVdec As Double) As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Argument

card_id: Specify the PCIe-8158 card index. The card_id could be decided by DIP switch (SW1) or depend on slot sequence.Please refer to _8158_initial().

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

OffsetCx: X-axis (first axis of target axes) offset to center

OffsetCy: Y-axis (second axis of target axes) offset to center

OffsetEx: X-axis (first axis of target axes) offset to end of arc

OffsetEy: Y-axis offset to end of arc

PitchDist: Z-axis specified relative distance to move

Cx: X-axis (first axis of target axes) absoult position of center of arc

Cy: Y-axis (second axis of target axes) absoult position of center of arc

Ex: X-axis (first axis of target axes) absoult position of end of arc

Ey: Y-axis (second axis of target axes) absoult position of end of arc

PitchPos: Z-axis specified absolute position to move

CW_CCW: Specified direction of arc

Value	Meaning
0	Clockwise(cw)
1	Counterclockwise(ccw)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

StrVel: Starting velocity of a velocity profile in units of pulse per second.

MaxVel: Maximum velocity in units of pulse per second.

Tacc: Specified acceleration time in units of seconds.

Tdec: Specified deceleration time in units of seconds.

SVacc: Specified velocity interval in which S-curve acceleration is performed.

Note: SVacc = 0, for pure S-Curve. For more details, see section 4.2.4

SVdec: specified velocity interval in which S-curve deceleration is performed.

Note: SVdec = 0, for pure S-Curve. For more details, see section 4.2.4

B.10 Home Return Mode

@ Name

_8158_set_home_config – Set the configuration for home return move motion

_8158_home_move – Perform a home return move.

_8158_home_search – Perform an auto search home

@ Description

_8158_set_home_config

Configures the home return mode, origin(ORG) and index signal(EZ) logic, EZ count, and ERC output options for the home_move() function. Refer to section 4.2.10 for the setting of home_mode control.

_8158_home_move

This function will cause the axis to perform a home return move according to the **_8164_set_home_config**() function settings. The direction of movement is determined by the sign of

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

velocity parameter (MaxVel). Since the stopping condition of this function is determined by the home_mode setting, users should take care in selecting the initial moving direction. Users should also take care to handle conditions when the limit switch is touched or other conditions that are possible causing the axis to stop. For more detail description, see section 4.2.10

_8158_home_search

This function will cause the axis to perform a home-search move according to the **_8164_set_home_config**() function settings. The direction of movement is determined by the sign of velocity parameter (MaxVel). Since the stopping condition of this function is determined by the home_mode setting, users should take care in selecting the initial moving direction. Users should also take care to handle conditions when the limit switch is touched or other conditions that are possible causing the axis to stop. For more detail description, see section 4.2.11

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_home_config(I16 AxisNo, I16
     home_mode, I16 org_logic, I16 ez_logic, I16
     ez_count, I16 erc_out);
I16 _8158_home_move(I16 AxisNo, F64 StrVel, F64
     MaxVel, F64 Tacc);
I16 _8158_home_search(I16 AxisNo, F64 StrVel, F64
     MaxVel, F64 Tacc, F64 ORGOffset);
```

Visual Basic (Windows 7/8.1)

```
B_8158_set_home_config(ByVal AxisNo As Integer,
     ByVal home_mode As Integer, ByVal org_logic
     As Integer, ByVal ez_logic As Integer, ByVal
     ez_count As Integer, ByVal erc_out As
     Integer) As Integer
B_8158_home_move(ByVal AxisNo As Integer, ByVal
     StrVel As Double, ByVal MaxVel As Double,
     ByVal Tacc As Double) As Integer
B_8158_home_search(ByVal AxisNo As Integer, ByVal
     StrVel As Double, ByVal MaxVel As Double,
     ByVal Tacc As Double, ByVal ORGOffset As
     Double) As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Argument

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

home_mode: Stopping modes for home return, This value is between 0 to 12. Please refer to the operation theory section 4.2.10

org_logic: Action logic configuration for ORG

Value	Meaning
0	Active low
1	Active high

ez_logic: Action logic configuration for EZ

Value	Meaning
0	Active low
1	Active high

ez_count: 0-15 (Please refer to section 4.2.10)

erc_out: Set ERC output options.

Value	Meaning
0	no ERC out
1	ERC signal out when home-move finishing

StrVel: Starting velocity of a velocity profile. (unit: pulse/sec)

MaxVel: Maximum velocity. (unit: pulse/sec)

Tacc: Specified acceleration time (Unit: sec)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ORGOffset: The escape pulse amounts when home search touches the ORG singal (Unit: pulse)

B.11 Manual Pulse Generator Motion

@ Name

_8158_disable_manual pulse generator_input – Disable the manual pulse generator input

_8158_manual pulse generator_pmove – Manual manual pulse generator p_move

_8158_manual pulse generator_vmove – Manual manual pulse generator v_move

_8158_set_manual pulse generator_ratio – Set manual manual pulse generator ratio for actual output pulse rate

_8158_set_manual pulse generator_iptmode – Set the input signal modes of manual pulse generator

@ Description

_8158_disable_manual pulse generator_input

This function is used to set the manual pulse generator input disable or enable.

_8158_manual pulse generator_pmove

With this command, the axis begins to move according to the manual pulse input. The axis will output one pulse when it receives one manual pulse, until the **_8158_disable_manual pulse generator_input** function disables the manual pulse generator or the output pulse number reaches the distance.

_8158_manual pulse generator_vmove

With this command, the axis begins to move according to the manual pulse input. The axis will output one pulse when it receives one manual pulse, until the **_8158_disable_manual pulse generator_input** function disables the manual pulse generator.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_set_manual pulse generator_ratio

Set manual pulse ratio for actual output pulse rate. The formula for manual pulse output rate is:

Output Pulse Count = Input manual pulse generator Count x (MultiF + 1) x DivF / 2048

The DivF = 1~2047 Divide Factor

The MultiF= 0~31 Multiplication Factor

_8158_set_manual pulse generator_iptmode

This function is used to configure the input mode of manual manual pulse generator.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_disable_manual pulse
     generator_input(I16 AxisNo, U16 Disable );
I16 _8158_manual pulse generator_pmove(I16
     AxisNo, F64 Dist, F64 SpeedLimit);
I16 _8158_manual pulse generator_vmove(I16
     AxisNo, F64 SpeedLimit);
I16 _8158_set_manual pulse generator_ratio(I16
     AxisNo, I16 DivF, I16 MultiF);
I16 _8158_set_manual pulse generator_iptmode(I16
     AxisNo, I16 InputMode, I16 Inverse);
```

Visual Basic (Windows 7/8.1)

```
B_8158_disable_manual pulse generator_input(ByVal
     AxisNo As Integer, ByVal Disable As Integer)
     As Integer
B_8158_manual pulse generator_pmove(ByVal AxisNo
     As Integer, ByVal Dist As Double, ByVal
     SpeedLimit As Double) As Integer
B_8158_manual pulse generator_vmove(ByVal AxisNo
     As Integer, ByVal SpeedLimit As Double) As
     Integer
B_8158_set_manual pulse generator_ratio(ByVal
     AxisNo As Integer, ByVal DivF As Integer,
     ByVal MultiF As Integer) As Integer
B_8158_set_manual pulse generator_iptmode(ByVal
     AxisNo As Integer, ByVal InputMode As
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
Integer, ByVal Inverse As Integer) As
Integer
```

@ Argument

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

Disable: Disable manual pulse generator input.

Disable = 1, disable manual pulse generator

Disable = 0, enable manual pulse generator

Dist: Specified relative distance to move (unit: pulse)

For example, if SpeedLimit is set to be 100pps, then the axis can move at fastest 100pps, even the input manual pulse generator signal rate is more then 100pps.

DivF: Divide factor (1-2047)

Note: When 0 or 2048 is entered, the division circuit will be OFF.

MultiF: Multiplication factor (0-31)

InputMode: Setting of manual pulse input mode from the PA and PB pins

Value	Meaning
0	1X AB phase type pulse input
1	2X AB phase type pulse input
2	4X AB phase type pulse input
3	CW/CCW type pulse input

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Inverse: Reverse the moving direction from pulse direction

Value	Meaning
0	no inverse
1	Reverse moving direction

B.12 Motion Status

@ Name

_8158_motion_done – Return the motion status

@ Description

_8158_motion_done:

Return the motion status of the 8158. The return code show as below:

0	Normal stopped condition
1	Waiting for DR
2	Waiting for CSTA input
3	Waiting for an internal synchronous signal
4	Waiting for another axis to stop
5	Waiting for a completion of ERC timer
6	Waiting for a completion of direction change timer
7	Correcting backlash
8	Wait PA/PB
9	At FA speed
10	At FL Speed
11	Accelerating
12	At FH Speed
13	Decelerating
14	Wait INP
15	Others(Controlling Start)
16	SALM
17	SPEL

Table 1:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

18	SMEL
19	SEMG
20	SSTP
21	SERC

Table 1:

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_motion_done(I16 AxisNo)
```

Visual Basic (Windows 7/8.1)

```
B_8158_motion_done(ByVal AxisNo As Integer) As
     Integer
```

@ Argument

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

B.13 Motion Interface I/O

@ Name

`_8158_set_servo` – Set the ON-OFF state of the SVON signal

`_8158_set_pcs_logic` – Set the logic of PCS signal

`_8158_set_pcs` – Enable the PCS for position override

`_8158_set_clr_mode` – Set the mode of CLR signal

`_8158_set_inp` – Set the logic of INP signal and operating mode

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_set_alm – Set the logic of ALM signal and operating mode

_8158_set_erc – Set the logic of ERC signal and operating mode

_8158_set_erc_out – Output an ERC signal

_8158_clr_erc – Clear the ERC signal

_8158_set_sd – Set the logic SD signal and operating mode

_8158_enable_sd – Enable SD signal

_8158_set_limit_logic – Set the logic of PEL/MEL signal

_8158_set_limit_mode – Set PEL/MEL operating mode

_8158_get_io_status –Get all the motion I/O statuses of each 8158

@ Description

_8158_set_servo:

You can set the ON-OFF state of the SVON signal with this function. The default value is 1(OFF), which means the SVON is open to GND.

_8158_set_pcs_logic:

Set the active logic of the PCS signal input

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_set_pcs:

Enable the position override when input signal PCS is turn ON. The PCS terminal status can be monitored by the "**_8158_get_io_status**" function.

_8158_set_clr_mode

CLR inputted signal can reset specified counters(command, position, error and general purpose counter). The reset action could be set by this function. The reset action mode has 4 types. For details refer to arguments description.

_8158_set_inp:

Set the active logic of the In-Position signal input from the servo driver. Users can select whether they want to enable this function. It is disabled by default.

_8158_set_alm:

Set the active logic of the ALARM signal input from the servo driver. Two reacting modes are available when the ALARM signal is active.

_8158_set_erc:

Users can set the logic and on time of the ERC with this function. It also can set the manual pulse generator width of ERC signal.

_8158_set_erc_out:

This function is used to output the ERC singal manually.

_8158_clr_erc:

This function is used to reset the output when the ERC signal output is specified to a level type output.

_8158_set_sd:

Set the active logic, latch control, and operating mode of the SD signal input from a mechanical system. Users can select whether they want to enable this function by **_8158_enable_sd**. It is disabled by default

_8158_enable_sd:

Enable the SD signal input. Default setting is default.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_set_limit_logic:

Set the EL logic, normal open or normal closed.

_8158_set_limit_mode:

Set the reacting modes of the EL signal.

_8158_get_io_status:

Get all the I/O statuses for each axis. The definition for each bit is as follows:

Bit	Name	Description
0	RDY	RDY pin input
1	ALM	Alarm Signal
2	+EL	Positive Limit Switch
3	-EL	Negative Limit Switch
4	ORG	Origin Switch
5	DIR	DIR output
6	EMG	EMG status
7	PCS	PCS signal input
8	ERC	ERC pin output
9	EZ	Index signal
10	CLR	Clear signal
11	LTC	Latch signal input
12	SD	Slow Down signal input
13	INP	In-Position signal input
14	SVON	Servo-ON output status

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_servo(I16 AxisNo, I16 on_off);
I16 _8158_set_pcs_logic(I16 AxisNo, I16
     pcs_logic);
I16 _8158_set_pcs(I16 AxisNo, I16 enable);
I16 _8158_set_clr_mode(I16 AxisNo, I16 clr_mode,
     I16 targetCounterInBit);
I16 _8158_set_inp(I16 AxisNo, I16 inp_enable, I16
     inp_logic);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_set_alm(I16 AxisNo, I16 alm_logic, I16
     alm_mode);
I16 _8158_set_erc(I16 AxisNo, I16 erc_logic, I16
     erc_pulse_width, I16 erc_mode);
I16 _8158_set_erc_out(I16 AxisNo);
I16 _8158_clr_erc(I16 AxisNo);
I16 _8158_set_sd(I16 AxisNo, I16 sd_logic, I16
     sd_latch, I16 sd_mode);
I16 _8158_enable_sd(I16 AxisNo, I16 enable);
I16 _8158_set_limit_logic(I16 AxisNo, U16 Logic
     );
I16 _8158_set_limit_mode(I16 AxisNo, I16
     limit_mode);
I16 _8158_get_io_status(I16 AxisNo, U16 *io_sts);
```

Visual Basic (Windows 7/8.1)

```
B_8158_set_servo(ByVal AxisNo As Integer, ByVal
     on_off As Integer) As Integer
B_8158_set_pcs_logic(ByVal AxisNo As Integer,
     ByVal pcs_logic As Integer) As Integer
B_8158_set_pcs(ByVal AxisNo As Integer, ByVal
     enable As Integer)As Integer
B_8158_set_clr_mode(ByVal AxisNo As Integer,
     ByVal clr_mode As Integer, ByBal
     targetCounterInBit as Integer) As Integer
B_8158_set_inp(ByVal AxisNo As Integer, ByVal
     inp_enable As Integer, ByVal inp_logic As
     Integer) As Integer
B_8158_set_alm(ByVal AxisNo As Integer, ByVal
     alm_logic As Integer, ByVal alm_mode As
     Integer) As Integer
B_8158_set_erc(ByVal AxisNo As Integer, ByVal
     erc_logic As Integer, ByVal erc_pulse_width
     As Integer, ByVal erc_mode As Integer) As
     Integer
B_8158_set_erc_out(ByVal AxisNo As Integer) As
     Integer
B_8158_clr_erc(ByVal AxisNo As Integer) As
     Integer
B_8158_set_sd(ByVal AxisNo As Integer, ByVal
     sd_logic As Integer, ByVal sd_latch As
     Integer, ByVal sd_mode As Integer) As
     Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
B_8158_enable_sd(ByVal AxisNo As Integer, ByVal
     Enable As Integer) As Integer
B_8158_set_limit_logic(ByVal AxisNo As Integer,
     ByVal Logic As Integer) As Integer
B_8158_set_limit_mode(ByVal AxisNo As Integer,
     ByVal limit_mode As Integer) As Integer
I16 _8158_get_io_status(ByVal AxisNo As Integer,
     io_sts As Integer) As Integer
```

@ Argument

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

`on_off`: ON-OFF state of SVON signal

Value	Meaning
0	ON
1	OFF

`pcs_logic`: PCS signal input logic

Value	Meaning
0	Negative logic
1	Positive logic

`enable`: enable or disable

Value	Meaning
0	Disable
1	Enable

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

clr_mode: Specify a CLR input clear mode

clr_mode = 0 , Clear on the falling edge (default)

clr_mode = 1 , Clear on the rising edge

clr_mode = 2 , Clear on a LOW level

clr_mode = 3 , Clear on a HIGH level

targetCounterInBit: Enable/Disable clear target counter in bit

Value	Meaning
Bit	Description
0	Reset command counter when CLR input turns ON
1	Reset position counter when CLR input turns ON
2	Reset error counter when CLR input turns ON
3	Reset general purpose counter when CLR input turns ON

inp_enable: INP function enabled/disabled

inp_enable = 0, Disabled (default)

inp_enable = 1, Enabled

inp_logic: Set the active logic for the INP signal

Value	Meaning
0	Negative logic
1	Positive logic

alm_logic: Setting of active logic for ALARM signals

Value	Meaning
0	Negative logic
1	Positive logic

alm_mode: Reacting modes when receiving an ALARM signal.

Value	Meaning
0	motor immediately stops (default)
1	motor decelerates then stops

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

erc_logic: Set the active logic for the ERC signal

Value	Mmeaning
0	Negative logic
1	Positive logic

erc_pulse_width: Set the pulse width of the ERC signal

Value	Meaning
0	12 µs
1	102 µs
2	409 µs
3	1.6 ms
4	13 ms
5	52 ms
6	104 ms
7	Level output

erc_mode:

Value	Meaning
0	Disable
1	Output ERC when stopped by EL, ALM, or EMG input
2	Output ERC when complete home return
3	Both 1 and 2

sd_logic:

Value	Meaning
0	Negative logic
1	Positive logic

sd_latch: Set the latch control for the SD signal

Value	Meaning
0	Do not latch
1	latch

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

sd_mode: Set the reacting mode of the SD signal

Value	Meaning
0	slow down only
1	slow down then stop

enable: Set the ramping-down point for high speed feed.

Value	Meaning
0	Automatic setting
1	Manual setting (default)

Logic: Set the PEL/MEL logic.

Value	Meaning
0	Normal low(normal open)
1	Normal high(normal close)

limit_mode:

Value	Meaning
0	Stop immediately
1	Slow down then stop

***io_sts**: I/O status. Please refer to 6.12 function description.

B.14 Interrupt Control

@ Name

_8158_int_control – Enable/Disable INT service

_8158_set_motion_int_factor – Set the factors of motion related interrupts

_8158_wait_error_interrupt – Wait error related interrupts

_8158_wait_motion_interrupt – Wait motion related interrupts

@ Description

_8158_int_control:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This function is used to enable the Windows interrupt event to host PC.

_8158_set_motion_int_factor:

This function allows users to select motion related factors to initiate the event int. The error can never be masked once the interrupt service is turned on by **_8158_int_control**(). Once the Interrupt function is enabled, you can use **_8158_wait_motion_interrupt**() to wait event.

_8158_wait_error_interrupt:

When user enabled the Interrupt function by **_8158_int_control**(). He could use this function to wait the error interrupts. Please refer to the operation theory section 4.8

_8158_wait_motion_interrupt:

When user enabled the Interrupt function by **_8158_int_control**() and set the interrupt factors by **_8158_set_motion_int_factor**(). User could use this function to wait the specific interrupt. When this function was running, the process would never stop until evens were triggered or the function was time out.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_int_control(I16 card_id, I16
     intFlag);
I16 _8158_set_motion_int_factor(I16 AxisNo,
     U32 int_factor );
I16 _8158_wait_error_interrupt(I16 AxisNo,
     I32 TimeOut_ms );
I16 _8158_wait_motion_interrupt(I16 AxisNo, I16
     IntFactorBitNo, I32 TimeOut_ms );
```

Visual Basic (Windows 7/8.1)

```
B_8158_int_control(ByVal card_id As Integer,
     ByVal intFlag As Integer) As Integer
B_8158_wait_error_interrupt(ByVal AxisNo As
     Integer, ByVal TimeOut_ms As Long) As
     Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
B_8158_wait_motion_interrupt(ByVal AxisNo As
     Integer, ByVal IntFactorBitNo As Integer,
     ByVal TimeOut_ms As Long) As Integer
B_8158_set_motion_int_factor(ByVal AxisNo As
     Integer, ByVal int_factor As Long) As
     Integer
```

@ Argument

`card_id`: Specify the index of target PCIe-8158 card. The card_id could be decided by DIP switch (SW1) or depend on slot sequence. Please refer to **`_8158_initial`**().

`intFlag`: Enable/Disable the Interrupt function

Value	Meaning
0	Disable
1	Enable

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

`int_factor`: interrupt factor

motion INT factors (Value, 0: Disable, 1: Enable)

Bit	Description
0	Normal stop
1	Next command in buffer starts
2	Command pre-register 2 is empty and allow new command to write
3	(Reserved) (Always set to 0)
4	Acceleration Start
5	Acceleration End

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Bit	Description
6	Deceleration Start
7	Deceleration End
8	+Soft limit or comparator 1 is ON
9	-Soft limit or comparator 2 is ON
10	Error comparator or comparator 3 is ON
11	General comparator or comparator 4 is ON
12	Trigger comparator or comparator 5 is ON
13	Counter is reset by CLR input
14	Counter is latched by LTC input
15	Counter is latched by ORG Input
16	SD input turns on
17	(Reserved) (Always set to 0)
18	CSTA input or _8158_start_move_all() turns on
19-31	Not define (Always set to 0)

`TimeOut_ms`: Specifies the time-out interval, in milliseconds. If TimeOut_ms is zero, the function tests the states of the specified objects and returns immediately. If TimeOut_ms is -1, the function's time-out interval never elapses (infinite).

`IntFactorBitNo`: Specifies the bit number of the INT factor.

e.g. IntFactorBitNo = 4, It means waiting the factor of “Acceleration Start” interrupt.

B.15 Position Control and Counters

@ Name

`_8158_get_position` – Get the value of feedback position counter

`_8158_set_position` – Set the feedback position counter

`_8158_get_command` – Get the value of command position counter

`_8158_set_command` – Set the command position counter

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_get_error_counter – Get the value of position error counter

_8158_reset_error_counter – Reset the position error counter

_8158_get_general_counter – get the value of general counter

_8158_set_general_counter – Set the general counter

_8158_get_target_pos – Get the value of target position recorder

_8158_reset_target_pos – Reset target position recorder

_8158_get_res_distance – Get remaining pulses accumulated from motion

_8158_set_res_distance – Set remaining pulses record

_8158_get_ring_counter – Acquires limitation value of ring counter

_8158_set_ring_counter – Sets ring counter limitation and enables ring counter function

_8158_escape_home – Leaves from the home position, with positivity of negative move set by the MaxVel

@ Description

_8158_get_position:

This function is used to read the feedback position counter value. Note that this value has already been processed by the move ratio setting by **_8158_set_move_ratio**(). If the move ratio is 0.5, than the value of position will be twice. The source of the feedback counter is selectable by the function **_8158_set_feedback_src**() to be external EA/EB or internal pulse output of 8158 .

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_set_position:

This function is used to change the feedback position counter to the specified value. Note that the value to be set will be processed by the move ratio. If move ratio is 0.5, then the set value will be twice as given value.

_8158_get_command:

This function is used to read the value of the command position counter. The source of the command position counter is the pulse output of the 8158.

_8158_set_command:

This function is used to change the value of the command position counter.

_8158_get_error_counter:

This function is used to read the value of the position error counter.

_8158_reset_error_counter:

This function is used to clear the position error counter.

_8158_get_general_counter:

This function is used to read the value of the general counter.

_8158_set_general_counter:

This function is used to set the counting source of and change the value of general counter (By default, the source is manual pulse generator input).

_8158_get_target_pos:

This function is used to read the value of the target position recorder. The target position recorder is maintained by the 8158 software driver. It records the position to settle down for current running motion.

_8158_reset_target_pos:

This function is used to set new value for the target position recorder. It is necessary to call this function when home return completes, or when a new feedback counter value is set by function **_8158_set_position**().

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_get_res_distance:

This function is used to read the value of the residue distance recorder. The target position recorder is maintained by the 8158 software driver. It records the position to settle down for current running motion.

_8158_set_res_distance:

This function is used to change the value of the residue distance counter

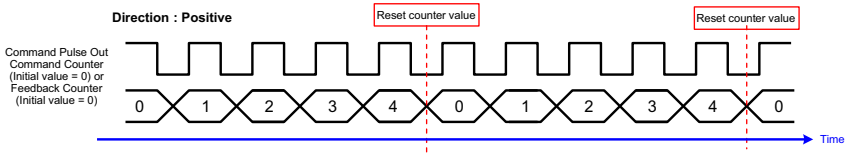
_8158_get_ring_counter

Acquires limitation value of ring counter.

_8158_set_ring_counter

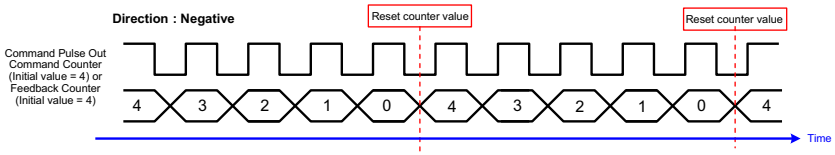
Sets ring counter limitation and enables ring counter function, wherein command and feedback counters operate as a ring counter. When ring counter limitation is set to zero, ring counter function is disabled.

As shown, when ring counter limitation (I32 RingVal) is set to four, command and feedback counters count up until counter value equals four, at which point command and feedback counters reset to zero and the operation is repeated.



Accordingly, as is further shown, when ring counter limitation (I32 RingVal) is set to four, command and feedback counters count down until counter value equals zero, command and feedback counters are reset to four, and the operation is repeated.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000



_8158_escape_home

Leaves from the home position, with positivity of negative move set by the MaxVel

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_get_position(I16 AxisNo, F64 *Pos);
I16 _8158_set_position(I16 AxisNo, F64 Pos);
I16 _8158_get_command(I16 AxisNo, I32 *Command);
I16 _8158_set_command(I16 AxisNo, I32 Command);
I16 _8158_get_error_counter(I16 AxisNo, I16
     *error);
I16 _8158_reset_error_counter(I16 AxisNo);
I16 _8158_get_general_counter(I16 AxisNo, F64
     *CntValue);
I16 _8158_set_general_counter(I16 AxisNo, I16
     CntSrc, F64 CntValue);
I16 _8158_get_target_pos(I16 AxisNo, F64 *T_pos);
I16 _8158_reset_target_pos(I16 AxisNo, F64
     T_pos);
I16 _8158_get_res_distance(I16 AxisNo, F64
     *Res_Distance);
I16 _8158_set_res_distance(I16 AxisNo, F64
     Res_Distance);
I16 _8158_set_ring_counter( I16 AxisNo, I32
     RingVal );
I16 _8158_get_ring_counter( I16 AxisNo, I32
     *RingVal );
I16 _8158_escape_home (I16 AxisNo, F64 StrVel,
     F64 MaxVel, F64 Tacc);
```

Visual Basic (Windows 7/8.1)

```
B_8158_get_position(ByVal AxisNo As Integer, Pos
     As Double) As Integer
B_8158_set_position(ByVal AxisNo As Integer,
     ByVal Pos As Double) As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
B_8158_get_command(ByVal AxisNo As Integer, Cmd
     As Long) As Integer
B_8158_set_command(ByVal AxisNo As Integer, ByVal
     Cmd As Long) As Integer
B_8158_get_error_counter(ByVal AxisNo As Integer,
     ByRef error As Integer) As Integer
B_8158_reset_error_counter(ByVal AxisNo As
     Integer) As Integer
B_8158_set_general_counter(ByVal AxisNo As
     Integer, ByVal CntSrc As Integer, ByVal
     CntValue As Double) As Integer
B_8158_get_general_counter(ByVal AxisNo As
     Integer, ByRef Pos As Double) As Integer
B_8158_reset_target_pos(ByVal AxisNo As Integer,
     ByVal Pos As Double) As Integer
B_8158_get_target_pos(ByVal AxisNo As Integer,
     ByRef Pos As Double) As Integer
B_8158_set_res_distance(ByVal AxisNo As Integer,
     ByVal Res_Distance As Double) As Integer
B_8158_get_res_distance(ByVal AxisNo As Integer,
     ByRef Res_Distance As Double) As Integer
B_8158_set_ring_counter (ByVal AxisNo As Integer,
     ByVal RingVal As Long) As Integer
B_8158_get_ring_counter (ByVal AxisNo As Integer,
     RingVal As Long) As Integer
B_8158_escape_home (ByVal AxisNo As Integer,
     ByVal StrVel As Double, ByVal
MaxVel As Double, ByVal Tacc As Double) As
     Integer
```

@ Argument

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Pos, ***Pos**: Feedback position counter value, (_8158_get/set_position)

range: -134217728 to 134217727

Cmd, ***Cmd**: Command position counter value,

range: -134217728 to 134217727

***error**: Position error counter value,

range: -32768 to 32767

CntSrc: general counter source

Value	Meaning
0	Command pulse
1	EA/EB
2	manual pulse generator input
3	System clock÷2

CntValue, ***CntValue**: the counter value

TargetPos, ***TargetPos**: Target position recorder value,

range: -134217728 to 134217727

ResDistance, ***ResDistance**: residue distance

RingVal, *RingVal: Limitation value of ring counter.(0 < RingVal < 134217727), wherein if RingVal equals zero, disable ring counter function

StrVel: Starting velocity of a velocity profile in pulses per second

MaxVel: Maximum velocity in pulses per second

Tacc: Specified acceleration time in seconds

B.16 Position Compare and Latch

@ Name

_8158_set_trigger_logic – Set the CMP signal's logic

_8158_set_trigger_comparator – Set the trigger comparator

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_set_error_comparator – Set the error comparator

_8158_set_general_comparator – Set the general comparator

_8158_set_latch_source – Set the latch timing for a counter

_8158_set_ltc_logic – Set the logic of LTC signal

_8158_get_latch_data – Get the latch data from counter

@ Description

_8158_set_trigger_logic:

This function is used to set the logic of CMP single.

_8158_set_error_comparator:

This function is used to set the comparing method and value for the error comparator. When the position error counter's value reaches the comparing value, the 8158 will generate an interrupt to the host PC. Also see Section B.14 on page 120.

_8158_set_general_comparator:

This function is used to set the comparing source counter, comparing method and value for the general comparator. When the comparison conditions are met, there is one of the 4 reactions will be done. The detail setting, see the argument description.

_8158_set_trigger_comparator:

This function is used to set the comparing source counter, comparing method and value for the trigger comparator. When the comparison source counter's value reaches the comparing value, the 8158 will generate a pulse output via CMP and an interrupt (event_int_status, bit 12) will also be sent to host PC.

_8158_set_latch_source:

There are 4 latch triggering source. By using this function, user can choose the event source to latch counters' data.

_8158_set_ltc_logic:

This function is used to set the logic of the latch input.

_8158_get_latch_data:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

After the latch signal arrived, the function is used to read the latched value of counters.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_trigger_logic(I16 AxisNo, I16
     Logic);
I16 _8158_set_error_comparator(I16 AxisNo, I16
     CmpMethod, I16 CmpAction, I32 Data);
I16 _8158_set_general_comparator(I16 AxisNo, I16
     CmpSrc, I16 CmpMethod,
I16 CmpAction, I32 Data);
I16 _8158_set_trigger_comparator(I16 AxisNo, I16
     CmpSrc, I16 CmpMethod,
I32 Data);
I16 _8158_set_latch_source(I16 AxisNo, I16
     LtcSrc);
16 _8158_set_ltc_logic(I16 AxisNo, I16 LtcLogic);
I16 _8158_get_latch_data(I16 AxisNo, I16
     CounterNo, F64 *Pos);
```

Visual Basic (Windows 7/8.1)

```
B_8158_set_trigger_logic(ByVal AxisNo As Integer,
     ByVal Logic As Integer) As Integer
B_8158_set_error_comparator(ByVal AxisNo As
     Integer, ByVal CmpMethod As Integer, ByVal
     CmpAction As Integer, ByVal Data As Long) As
     Integer
B_8158_set_general_comparator(ByVal AxisNo As
     Integer, ByVal CmpSrc As Integer, ByVal
     CmpMethod As Integer, ByVal CmpAction As
     Integer, ByVal Data As Long) As Integer
B_8158_set_trigger_comparator(ByVal AxisNo As
     Integer, ByVal CmpSrc As Integer, ByVal
     CmpMethod As Integer, ByVal Data As Long) As
     Integer
B_8158_set_latch_source(ByVal AxisNo As Integer,
     ByVal LtcSrc As Integer) As Integer
B_8158_set_ltc_logic(ByVal AxisNo As Integer,
     ByVal StcLogic As Integer) As Integer
B_8158_get_latch_data(ByVal AxisNo As Integer,
     ByVal CounterNo As Integer, Pos As Double)
     As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Argument

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

Logic: logic of comparing trigger

Value	Meaning
0	Negative logic
1	Positive logic

CmpSrc: The comparing source counters

Value	Meaning
0	Command counter
1	Feedback counter
2	Error counter
3	General counter

CmpMethod: The comparing methods

Value	Meaning
0	No Compare(Disable)
1	Data = Source counter (direction independent)
2	Data = Source counter (Count up only)
3	Data = Source counter (Count down only)
4	Data > Source counter
5	Data < Source counter

Data: Comparing value (Position)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

CmpAction:

Value	Meaning
0	No action
1	Stop immediately
2	Slow down then stop

ltc_src:

Value	Meaning
0	LTC pin input
1	ORG pin input
2	general comparator conditions are met
3	trigger comparator conditions are met

ltc_logic: LTC signal operation edge

Value	Meaning
0	Negative logic
1	Positive logic

CounterNo: Specified the counter to latch

Value	Meaning
0	Command counter
1	Feedback counter
2	Error counter
3	General counter

***Pos**: Latch data (Position)

B.17 Continuous motion

@ Name

_8158_set_continuous_move – Enable continuous motion for absolute motion

_8158_check_continuous_buffer – Check if the buffer is empty

_8158_dwell_move – Set a dwell move

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Description

_8158_set_continuous_move:

This function is necessary before and after continuous motion command sequences

_8158_check_continuous_buffer:

This function is used to detect if the command pre-register (buffer) is empty or not. Once the command pre-register (buffer) is empty, users may write the next motion command into it. Otherwise, the new command will overwrite the previous command in the 2nd command pre-register. If the return code is 1 means buffer is full. Otherwise return code is 0, buffer is not full.

_8158_dwell_move:

This function is used to start a dwell move that means the move does not cause real motion for a specific time.

Example:

```
_8158_set_continuous_move( 2, 1 ); // start
     continuous move
_8158_start_tr_move( 2, 20000.0, 10.0, 10000.0,
     0.1, 0.1);
_8158_dwell_move( 2, 2000); //dwell move for 2
     sec.
_8158_start_sr_move( 2, 20000.0, 10.0, 10000.0,
     0.1, 0.1, 0, 0 );
_8158_set_continuous_move( 2, 0 ); //end
     continuous move
```

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_continuous_move(I16 AxisNo, I16
     Enable);
I16 _8158_check_continuous_buffer(I16 AxisNo);
I16 _8158_dwell_move(I16 AxisNo, F64 ms);
```

Visual Basic (Windows 7/8.1)

```
B_8158_set_continuous_move(ByVal AxisNo As
     Integer, ByVal Enable As Integer) As Integer
B_8158_check_continuous_buffer(ByVal AxisNo As
     Integer) As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
B_8158_dwell_move(ByVal AxisNo As Integer, ByVal
     ms As Double) As Integer
```

@ Argument

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

Enable: continuous motion switch logic

Value	Meaning
0	continuous motion sequence is finished (Disable)
1	continuous motion sequence is started (Enable)

millisecond: Time of dwell move. the unit is in milliseconds.

B.18 Multiple Axes Simultaneous Operation

@ Name

_8158_set_tr_move_all – Multi-axis simultaneous operation setup

_8158_set_ta_move_all – Multi-axis simultaneous operation setup

_8158_set_sr_move_all – Multi-axis simultaneous operation setup

_8158_set_sa_move_all – Multi-axis simultaneous operation setup

_8158_start_move_all – Begin a multi-axis trapezoidal profile motion

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

_8158_stop_move_all – Simultaneously stop Multi-axis motion

@ Description

Theses functions are related to simultaneous operations of multi-axes, even in different cards. The simultaneous multi-axis operation means to start or stop moving specified axes at the same time. The axes moved are specified by the parameter “AxisArray,” and the number of axes are defined by parameter “TotalAxes” in **_8158_set_tr_move_all**().

When properly setup with **_8158_set_xx_move_all**(), the function **_8158_start_move_all**() will cause all specified axes to begin a trapezoidal relative movement, and **_8158_stop_move_all**() will stop them. Both functions guarantee that motion Starting/Stopping on all specified axes are at the same time. Note that it is necessary to make connections according to Section 1.7 on page 11 if these two functions are needed.

The following code demos how to utilize these functions. This code moves axis 0 and axis 1 to distance 80000.0 and 120000.0 respectively. If we choose velocities and accelerations that are proportional to the ratio of distances, then the axes will arrive at their endpoints at the same time.

Example:

```
I16 axes[2] = {0, 1};
F64 dist[2] = {80000.0, 120000.0},
F64 str_vel[2] = {0.0, 0.0},
F64 max_vel[2] = {4000.0, 6000.0},
F64 Tacc[2] = {0.1, 0.6},
F64 Tdec[2] = {0.1, 0.6};

_8158_set_tr_move_all(2, axes, dist, str_vel,
     max_vel, Tacc, Tdec);
_8158_start_move_all(axes[0]);
```

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_tr_move_all(I16 TotalAxes, I16
     *AxisArray, F64 *DistA, F64 *StrVelA, F64
     *MaxVelA, F64 *TaccA, F64 *TdecA);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
I16 _8158_set_ta_move_all(I16 TotalAx, I16
     *AxisArray, F64 *PosA, F64 *StrVelA, F64
     *MaxVelA, F64 *TaccA, F64 *TdecA);
I16 _8158_set_sr_move_all(I16 TotalAx, I16
     *AxisArray, F64 *DistA, F64 *StrVelA, F64
     *MaxVelA, F64 *TaccA, F64 *TdecA, F64
     *SVaccA, F64 *SVdecA);
I16 _8158_set_sa_move_all(I16 TotalAx, I16
     *AxisArray, F64 *PosA, F64 *StrVelA, F64
     *MaxVelA, F64 *TaccA, F64 *TdecA, F64
     *SVaccA, F64 *SVdecA);
I16 _8158_start_move_all(I16 FirstAxisNo);
I16 _8158_stop_move_all(I16 FirstAxisNo);
```

Visual Basic (Windows 7/8.1)

```
B_8158_set_tr_move_all(ByVal TotalAxes As
     Integer, ByRef AxisArray As Integer, ByRef
     DistA As Double, ByRef StrVelA As Double,
     ByRef MaxVelA As Double, ByRef TaccA As
     Double, ByRef TdecA As Double) As Integer
B_8158_set_sa_move_all(ByVal TotalAxes As
     Integer, ByRef AxisArray As Integer, ByRef
     PosA As Double, ByRef StrVelA As Double,
     ByRef MaxVelA As Double, ByRef TaccA As
     Double, ByRef TdecA As Double, ByRef SVaccA
     As Double, ByRef SVdecA As Double) As
     Integer
B_8158_set_ta_move_all(ByVal TotalAxes As
     Integer, ByRef AxisArray As Integer, ByRef
     PosA As Double, ByRef StrVelA As Double,
     ByRef MaxVelA As Double, ByRef TaccA As
     Double, ByRef TdecA As Double) As Integer
B_8158_set_sr_move_all(ByVal TotalAxes As
     Integer, ByRef AxisArray As Integer, ByRef
     DistA As Double, ByRef StrVelA As Double,
     ByRef MaxVelA As Double, ByRef TaccA As
     Double, ByRef TdecA As Double, ByRef SVaccA
     As Double, ByRef SVdecA As Double) As
     Integer
B_8158_start_move_all(ByVal FirstAxisNo As
     Integer) As Integer
B_8158_stop_move_all(ByVal FirstAxisNo As
     Integer) As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Argument

TotalAxes: Number of axes for simultaneous motion

***AxisArray**: Specified axes number array designated to move.

***DistA**: Specified distance array in units of pulse

***StrVelA**: Starting velocity array in units of pulse per second

***MaxVelA**: Maximum velocity array in units of pulse per second

***TaccA**: Acceleration time array in units of seconds

***TdecA**: Deceleration time array in units of seconds

***PosA**: Specified position array in units of pulse

***SvaccA**: Specified velocity interval array in which S-curve acceleration is performed.

***SvdecA**: Specified velocity interval array in which S-curve deceleration is performed.

FirstAxisNo: The first element in AxisArray.

B.19 General-Purpose DIO

@ Name

_8158_set_gpio_output – Set digital output

_8158_get_gpio_output – Get digital output

_8158_get_gpio_input – Get digital input

_8158_set_gpio_input_function – Set the signal types for any digital inputs

@ Description

_8158_set_gpio_output:

The PCIe-8158 has 4 digital output channels. By this function, user could control the digital outputs.

_8158_get_gpio_output:

This function is used to get the digital output status.

_8158_get_gpio_input:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

PCIe-8158 has 4 digital input channels. By this function, user can get the digital input status.

_8158_set_gpio_input_function:

PCIe-8158 has 4 digital input channels. By this function, user can set one of several input signals to any specific DI channels. Those signals include LTCn, SDn, PCSn, CLRn, EMG. (The index word n mean axis index)

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_gpio_output(I16 card_id, I16
     DoValue );
I16 _8158_get_gpio_output(I16 card_id, I16 *
     DoValue );
I16 _8158_get_gpio_input(I16 card_id, I16 *
     DiValue );
I16 _8158_set_gpio_input_function(I16 card_id,
     I16 Channel, I16Select, I16 Logic);
```

Visual Basic (Windows 7/8.1)

```
B_8158_set_gpio_output(ByVal card_id As Integer,
     ByVal DoValue As Integer) As Integer
B_8158_get_gpio_output(ByVal card_id As Integer,
     DoValue As Integer) As Integer
B_8158_get_gpio_input(ByVal card_id As Integer,
     DiValue As Integer) As Integer
B_8158_set_gpio_input_function(ByVal card_id As
     Integer, ByVal Channel As Integer, ByVal
     Select As Integer, ByVal Logic As Integer)As
     Integer
```

@ Argument

card_id: Specify the PCIe-8158 card index. The card_id could be decided by DIP switch (SW1) or depend on slot sequence. Please refer to **_8158_initial**().

DoValue, ***DoValue**: Digital output value. Bit 0-3: D_out 0-3.

***DiValue**: Digital input value, Bit 0-3: D_in 0-3

Channel: Digital channel DI0 - DI3

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Select: signal types select

Value	Meaning
0	General DI (default)
1	LTC(active low)
2	SD(active low)
3	PCS(active low)
4	CLR (active low)
5	EMG (active low)

Logic: input signal logic

Value	Meaning
0	Not inverse (default)
1	Inverse

B.20 Soft Limit

@ Name

_8158_disable_soft_limit – Disable soft limit function

_8158_enable_soft_limit – Enable soft limit function

_8158_set_soft_limit – Set soft limit

@ Description

_8158_disable_soft_limit:

This function is used to disable the soft limit function.

_8158_enable_soft_limit:

This function is used to enable the soft limit function. Once enabled, the action of soft limit will be exactly the same as physical limit.

_8158_set_soft_limit:

This function is used to set the soft limit value.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_disable_soft_limit(I16 AxisNo);
I16 _8158_enable_soft_limit(I16 AxisNo, I16
     Action);
I16 _8158_set_soft_limit(I16 AxisNo, I32
     PlusLimit, I32 MinusLimit);
```

Visual Basic (Windows 7/8.1)

```
B_8158_disable_soft_limit(ByVal AxisNo As
     Integer) As Integer
B_8158_enable_soft_limit(ByVal AxisNo As Integer,
     ByVal Action As Integer) As Integer
B_8158_set_soft_limit(ByVal AxisNo As Integer,
     ByVal PlusLimit As Long, ByVal MinusLimit As
     Long) As Integer
```

@ Argument

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

`Action`: The reacting method of soft limit

Value	Meaning
0	INT only
1	Immediately stop
2	slow down then stop

`PlusLimit`: Soft limit value, positive direction

`MinusLimit`: Soft limit value, negative direction

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

B.21 Backlash Compensation / Vibration Suppression

@ Name

_8158_backlash_comp – Set backlash corrective pulse for compensation

_8158_suppress_vibration – Set vibration suppressing timing

_8158_set_fa_speed – Set the FA speed

@ Description

_8158_backlash_comp:

Whenever direction change occurs, the 8158 outputs backlash corrective pulses before sending commands. This function is used to set the compensation pulse numbers.

_8158_suppress_vibration:

This function is used to suppress vibration of mechanical systems by outputting a single pulse for negative direction and the single pulse for positive direction right after completion of command movement.

_8158_set_fa_speed:

This function is used to specify the low speed for backlash correction or slip correction. It also used as a reverse low speed for home return operation.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_backlash_comp(I16 AxisNo, I16
     CompPulse, I16 Mode);
I16 _8158_suppress_vibration(I16 AxisNo, U16
     ReverseTime,
U16 ForwardTime);
I16 _8158_set_fa_speed(I16 AxisNo, F64 FA_Speed);
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Visual Basic (Windows 7/8.1)

```
B_8158_backlash_comps (ByVal AxisNo As Integer,
     ByVal CompPulse As Integer, ByVal Mode As
     Integer) As Integer
B_8158_suppress_vibration(ByVal AxisNo As
     Integer, ByVal ReverseTime As Integer, ByVal
     ForwardTime As Integer) As Integer
B_8158_set_fa_speed(ByVal AxisNo As Integer,
     ByVal FA_Speed As Double) As Integer
```

@ Argument

AxisNo: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

CompPulse: Specified number of corrective pulses, 12 bit

Mode:

Value	Meaning
0	Turns off
1	enable backlash compensation
2	Slip correction

ReverseTime: Specified Reverse Time, 0 - 65535, unit 1.6 us

ForwardTime: Specified Forward Time, 0 - 65535, unit 1.6 us

FA_Speed: fa speed (unit: pulse/sec)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

B.22 Speed Profile Calculation

@ Name

`_8158_get_tr_move_profile` – Get the relative trapezoidal speed profile

`_8158_get_ta_move_profile` – Get the absolute trapezoidal speed profile

`_8158_get_sr_move_profile` – Get the relative S-curve speed profile

`_8158_get_sa_move_profile` – Get the absolute S-curve speed profile

@ Description

`_8158_get_tr_move_profile`:

This function is used to get the relative trapezoidal speed profiles. By this function, user can get the actual speed profile before running.

`_8158_get_ta_move_profile`:

This function is used to get the absolute trapezoidal speed profiles. By this function, user can get the actual speed profile before running.

`_8158_get_sr_move_profile`:

This function is used to get the relative S-curve speed profiles. By this function, user can get the actual speed profile before running.

`_8158_get_sa_move_profile`:

This function is used to get the absolute S-curve speed profiles. By this function user can get the actual speed profile before running.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_get_tr_move_profile(I16 AxisNo, F64
      Dist, F64 StrVel, F64 MaxVel, F64 Tacc, F64
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
      Tdec, F64 *pStrVel, F64 *pMaxVel, F64
      *pTacc, F64 *pTdec, F64 *pTconst );
I16 _8158_get_ta_move_profile(I16 AxisNo, F64
      Pos, F64 StrVel, F64 MaxVel, F64 Tacc, F64
      Tdec, F64 *pStrVel, F64 *pMaxVel, F64
      *pTacc, F64 *pTdec, F64 *pTconst );
I16 _8158_get_sr_move_profile(I16 AxisNo, F64
      Dist, F64 StrVel, F64 MaxVel, F64 Tacc, F64
      Tdec, F64 SVacc, F64 SVdec,F64 *pStrVel, F64
      *pMaxVel, F64 *pTacc, F64 *pTdec, F64
      *pSVacc, F64 *pSVdec, F64 *pTconst);
I16 _8158_get_sa_move_profile(I16 AxisNo, F64
      Pos, F64 StrVel, F64 MaxVel, F64 Tacc, F64
      Tdec, F64 SVacc, F64 SVdec,F64 *pStrVel, F64
      *pMaxVel, F64 *pTacc, F64 *pTdec, F64
      *pSVacc, F64 *pSVdec, F64 *pTconst);
```

Visual Basic (Windows 7/8.1)

```
B_8158_get_tr_move_profile(ByVal AxisNo As
      Integer, ByVal Dist As Double, ByVal StrVel
      As Double, ByVal MaxVel As Double, ByVal
      Tacc As Double, ByVal Tdec As Double, ByRef
      pStrVel As Double, ByRef pMaxVel As Double,
      ByRef pTacc As Double, ByRef pTdec As
      Double, ByRef pTconst As Double) As Integer
B_8158_get_ta_move_profile(ByVal AxisNo As
      Integer, ByVal Pos As Double, ByVal StrVel
      As Double, ByVal MaxVel As Double, ByVal
      Tacc As Double, ByVal Tdec As Double, ByRef
      pStrVel As Double, ByRef pMaxVel As Double,
      ByRef pTacc As Double, ByRef pTdec As
      Double, ByRef pTconst As Double) As Integer
B_8158_get_sr_move_profile(ByVal AxisNo As
      Integer, ByVal Dist As Double, ByVal StrVel
      As Double, ByVal MaxVel As Double, ByVal
      Tacc As Double, ByVal Tdec As Double, ByVal
      SVacc As Double, ByVal SVdec As Double,
      ByRef pStrVel As Double, ByRef pMaxVel As
      Double, ByRef pTacc As Double, ByRef pTdec
      As Double, ByRef pSVacc As Double, ByRef
      pSVdec As Double, ByRef pTconst As Double)
      As Integer
B_8158_get_sa_move_profile(ByVal AxisNo As
      Integer, ByVal Pos As Double, ByVal StrVel
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
As Double, ByVal MaxVel As Double, ByVal
Tacc As Double, ByVal Tdec As Double, ByVal
SVacc As Double, ByVal SVdec As Double,
ByRef pStrVel As Double, ByRef pMaxVel As
Double, ByRef pTacc As Double, ByRef pTdec
As Double, ByRef pSVacc As Double, ByRef
pSVdec As Double, ByRef pTconst As Double)
As Integer
```

@ Argument

`AxisNo`: Axis number designated to move or stop.

card_id	Physical axis	AxisNo
0	0	0
	1	1
	2	2
	3	3
1	0	4
	1	5
	…	…

`Dist`: Specified relative distance (unit: pulse)

`Pos`: Specified absolute position (unit: pulse)

`StrVel`: Starting velocity (unit: pulse/sec)

`MaxVel`: Maximum velocity (unit: pulse/sec)

`Tacc`: time for acceleration (unit: sec)

`Tdec`: time for deceleration (unit: sec)

`SVacc`: S-curve region during acceleration (unit: pulse/sec)

Note: SVacc = 0, for pure S-Curve. For more details, see section 4.2.4

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

SVdec: S-curve region during deceleration (unit: pulse/sec)

Note: SVdec = 0, for pure S-Curve. For more details, see section 4.2.4

***pStrVel**: Starting velocity by calculation

***pMaxVel**: Maximum velocity by calculation

***pTacc**: Acceleration time by calculation

***pTdec**: Deceleration time by calculation

***pSVacc**: S-curve region during acceleration by calculation

***pSVdec**: S-curve region during deceleration by calculation

***pTconst**: constant speed time(maximum speed)

B.23 Extended General Purpose TTL Input/Output

@ Name

_8158_set_gpio_output_ex – Set digital output

_8158_get_gpio_output_ex – Get digital output

_8158_get_gpio_input_ex – Get digital input

_8158_set_gpio_output_ex_CH – Set digital output by channel

_8158_get_gpio_output_ex_CH – Get digital output by channel

_8158_get_gpio_input_ex_CH – Get digital input by channel

@ Description

_8158_set_gpio_output_ex():

Set the on_off status for digital output pin.

_8158_get_gpio_output_ex():

Read on_off status of all digital output pins.

_8158_get_gpio_input_ex():

Read on_off status of all digital input pins.

_8158_set_gpio_output_ex_CH():

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Set the on_off status for digital output pins by channel.

_8158_get_gpio_output_ex_CH():

Read on_off status of all digital output pins by channel.

_8158_get_gpio_input_ex_CH():

Read on_off status of all digital input pins by channel.

@ Syntax

C/C++(Windows 7/8.1)

```
I16 _8158_set_gpio_output_ex(I16 CardNo, U16
     DoValue);
I16 _8158_get_gpio_output_ex(I16 CardNo, U16
     *DoValue );
I16 _8158_get_gpio_input_ex(I16 CardNo, U16
     *DiValue );
I16 _8158_set_gpio_output_ex_CH(I16 CardNo, U16
     Channel, U16 Value );
I16 _8158_get_gpio_output_ex_CH(I16 CardNo, U16
     Channel, U16 *Value );
I16 _8158_get_gpio_input_ex_CH(I16 CardNo, U16
     Channel, U16 *Value );
```

Visual Basic (Windows 7/8.1)

```
B_8158_set_gpio_output_ex(ByVal CardNo As
     Integer, ByVal DoValue As Integer) As
     Integer
B_8158_get_gpio_output_ex(ByVal CardNo As
     Integer, DoValue As Integer) As Integer
B_8158_get_gpio_input_ex(ByVal CardNo As Integer,
     DiValue As Integer) As Integer
B_8158_set_gpio_output_ex_CH(ByVal CardNo As
     Integer, ByVal Channel As Integer, ByVal
     Value As Integer) As Integer
B_8158_get_gpio_output_ex_CH(ByVal CardNo As
     Integer, ByVal Channel As Integer, Value As
     Integer) As Integer
B_8158_get_gpio_input_ex_CH(ByVal CardNo As
     Integer, ByVal Channel As Integer, Value As
     Integer) As Integer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

@ Argument

CardNo: Axis number designated.

Channel: Designated channel number 0 to 15.

DoValue: All output values.

DiValue: All input values.

Value: On-Off value for output or input in specific channel where Value=0 or 1

B.24 Return Code

The return error code is defined in “8158_err.h”. The meaning is described in following table.

Code	Meaning
0	No error
-10000	Error Card number
-10001	Error operation system version
-10002	Error card’s ID conflict
-10300	Error other process exist
-10301	Error card not found
-10302	Error Open driver failed
-10303	Error ID mapping failed
-10304	Error trigger channel
-10305	Error trigger type
-10306	Error event already enabled
-10307	Error event not enable yet
-10308	Error on board FIFO full
-10309	Error unknown command type
-10310	Error unknown chip type
-10311	Error card not initial
-10312	Error position out of range
-10313	Error motion busy
-10314	Error speed error
-10315	Error slow down point
-10316	Error axis range error

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

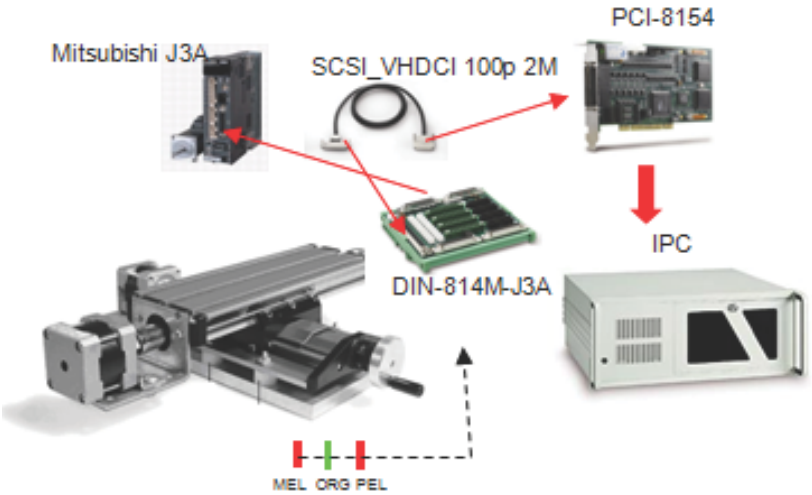
Code	Meaning
-10317	Error compare parameter error
-10318	Error compare method
-10319	Error axis already stop
-10320	Error axis INT wait failed
-10321	Error user code write failed
-10322	Error array size exceed
-10323	Error factor number
-10324	Error enable range
-10325	Error auto accelerate time
-10326	Error dwell time
-10327	Error dwell distance
-10328	Error new position
-10329	Error motion not in running
-10330	Error velocity change time
-10331	Error speed target
-10332	Error velocity percent
-10333	Error position change backward
-10334	Error counter number
-10335	Error gpio input function parameter
-10336	Error channel number
-10337	Error ERC mode
-10338	Error security code

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Appendix C Connection Example

C.1 General Description of Wiring

The following illustrates an exemplary integration of the PCIe-8158 and DIN-814M-J3A.



C.2 Terminal Board User Guide

Please refer to the individual terminal board's User Guide. Supported terminal boards are as follows:

Mitsubishi J2 Super	DIN-814M
Mitsubishi J3A	DIN-814M-J3A
Yaskawa Sigma II	DIN-814Y
Panasonic MINAS A4	DIN-814P-A4

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Important Safety Instructions

For user safety, please read and follow all instructions, Warnings, Cautions, and Notes marked in this manual and on the associated device before handling/operating the device, to avoid injury or damage.

S'il vous plaît prêter attention stricte à tous les avertissements et mises en garde figurant sur l'appareil , pour éviter des blessures ou des dommages.

- Read these safety instructions carefully
- Keep the User's Manual for future reference
- Read the Specifications section of this manual for detailed information on the recommended operating environment
- The device can be operated at an ambient temperature of 50ºC
- When installing/mounting or uninstalling/removing device; or when removal of a chassis cover is required for user servicing (See "Getting Started" on page 15.):
  - Turn off power and unplug any power cords/cables
  - Reinstall all chassis covers before restoring power
- To avoid electrical shock and/or damage to device:
  - Keep device away from water or liquid sources
  - Keep device away from high heat or humidity
  - Keep device properly ventilated (do not block or cover ventilation openings)
  - Always use recommended voltage and power source settings
  - Always install and operate device near an easily accessible electrical outlet
  - Secure the power cord (do not place any object on/over the power cord)
  - Only install/attach and operate device on stable surfaces and/or recommended mountings
- If the device will not be used for long periods of time, turn off and unplug from its power source

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- Never attempt to repair the device, which should only be serviced by qualified technical personnel using suitable tools
- A Lithium-type battery may be provided for uninterrupted backup or emergency power.



Risk of explosion if battery is replaced with one of an incorrect type; please dispose of used batteries appropriately.
Risque d'explosion si la pile est remplacée par une autre de type incorrect. Veuillez jeter les piles usagées de façon appropriée.

- The device must be serviced by authorized technicians when:
  - The power cord or plug is damaged
  - Liquid has entered the device interior
  - The device has been exposed to high humidity and/or moisture
  - The device is not functioning or does not function according to the User's Manual
  - The device has been dropped and/or damaged and/or shows obvious signs of breakage
- Disconnect the power supply cord before loosening the thumbscrews and always fasten the thumbscrews with a screwdriver before starting the system up
- It is recommended that the device be installed only in a server room or computer room where access is:
  - Restricted to qualified service personnel or users familiar with restrictions applied to the location, reasons therefor, and any precautions required
  - Only afforded by the use of a tool or lock and key, or other means of security, and controlled by the authority responsible for the location

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

	**BURN HAZARD** Touching this surface could result in bodily injury. To reduce risk, allow the surface to cool before touching. ***RISQUE DE BRÛLURES*** *Ne touchez pas cette surface, cela pourrait entraîner des blessures.* *Pour éviter tout danger, laissez la surface refroidir avant de la toucher.*

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This page intentionally left blank.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Getting Service

Ask an Expert: http://askanexpert.adlinktech.com

ADLINK Technology, Inc.
Address: 9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
新北市中和區建一路 166 號 9 樓
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.
Address: 5215 Hellyer Avenue, #110
San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.
Address: 上海市浦东新区张江高科技园区芳春路 300 号 (201203)
300 Fang Chun Rd., Zhangjiang Hi-Tech Park
Pudong New Area, Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

ADLINK Technology Beijing
Address: 北京市海淀区上地东路 1 号盈创动力大厦 E 座 801 室(100085)
Rm. 801, Power Creative E, No. 1 Shang Di East Rd.
Beijing, 100085 China
Tel: +86-10-5885-8666
Fax: +86-10-5885-8626
Email: market@adlinktech.com

ADLINK Technology Shenzhen
Address: 深圳市南山区科技园南区高新南七道 数字技术园
A1 栋 2 楼 C 区 (518057)
2F, C Block, Bldg. A1, Cyber-Tech Zone, Gao Xin Ave. Sec. 7
High-Tech Industrial Park S., Shenzhen, 518054 China
Tel: +86-755-2643-4858
Fax: +86-755-2664-6353
Email: market@adlinktech.com

LiPPERT ADLINK Technology GmbH
Address: Hans-Thoma-Strasse 11
D-68163 Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: emea@adlinktech.com

PENTA ADLINK Technology GmbH
Ulrichsbergerstrasse 17
94469 Deggendorf, Germany
Tel: +49 (0) 991 290 94 – 10
Fax: +49 (0) 991 290 94 - 29
Email: emea@adlinktech.com

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ADLINK Technology, Inc. (French Liaison Office)
Address: 6 allée de Londres, Immeuble Ceylan
91940 Les Ulis, France
Tel: +33 (0) 1 60 12 35 66
Fax: +33 (0) 1 60 12 35 66
Email: france@adlinktech.com

ADLINK Technology Japan Corporation
Address: 〒101-0045 東京都千代田区神田鍛冶町 3-7-4
神田 374 ビル 4F
KANDA374 Bldg. 4F, 3-7-4 Kanda Kajicho,
Chiyoda-ku, Tokyo 101-0045, Japan
Tel: +81-3-4455-3722
Fax: +81-3-5209-6013
Email: japan@adlinktech.com

ADLINK Technology, Inc. (Korean Liaison Office)
Address: 경기도 성남시 분당구 수내로 46 번길 4 경동빌딩 2 층
(수내동 4-4 번지) (우) 463-825
2F, Kyungdong B/D, 4 Sunae-ro 46 beon-gil
Bundang-gu, Seongnam-si, Gyeonggi-do, Korea, 463-825
Toll Free +82-80-800-0585
Tel +82-31-786-0585
Fax +82-31-786-0583
Email: korea@adlinktech.com

ADLINK Technology Singapore Pte. Ltd.
Address: 84 Genting Lane #07-02A, Cityneon Design Centre
Singapore 349584
Tel: +65-6844-2261
Fax: +65-6844-2263
Email: singapore@adlinktech.com

ADLINK Technology Singapore Pte. Ltd. (Indian Liaison Office)
Address: #50-56, First Floor, Spearhead Towers
Margosa Main Road (between 16th/17th Cross)
Malleswaram, Bangalore - 560 055, India
Tel: +91-80-65605817, +91-80-42246107
Fax: +91-80-23464606
Email: india@adlinktech.com

ADLINK Technology, Inc. (Israeli Liaison Office)
Address: 27 Maskit St., Corex Building
PO Box 12777
Herzliya 4673300, Israel
Tel: +972-54-632-5251
Fax: +972-77-208-0230
Email: israel@adlinktech.com

ADLINK Technology, Inc. (UK Liaison Office)
Tel: +44 774 010 59 65
Email: UK@adlinktech.com

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000