

MSK5 USB Library Manual

2017 / 4 / 17

Control Commands

Initialization

load_program_file

field correction

load_correction_file

load_correction_file_3d

set_lens_file

get_k_factor

System Control and Parameters

msk5_count_cards

get_msk5_version

get_hex_version

set_control_mode

select_list

set_max_counts

get_counts

get_startstop_info

stop_execution

set_hw_config

get_hw_config

get_uid_info

get_module_info

set_user_info

get_user_info

Laser Control and Parameters

set_laser_mode

set_softstart_mode

set_softstart_level

set_standby

disable_laser

enable_laser

Scanner Control and Parameters

load_varpolydelay

set_delay_mode

get_head_status

goto_xy

goto_xyz

z_out

Coordinate transformations

set_matrix

set_offset

I/O controls

read_io_port

write_io_port

get_io_status

write_8bit_port

write_da_x

write_io_port_ex

get_io_status_ex

set_program_ready

set_marking_ready

set_marking_end

List handling and status

set_start_list

execute_list

stop_list

restart_list

get_status

read_status

get_wait_status

release_wait

auto_change

auto_change_pos

start_loop

quit_loop

get_list_size

set_input_pointer

get_input_pointer

execute_at_pointer

get_time

set_label

get_label

Motion control

set_motion_backspace

get_motor_sensor

get_motion_inpos

get_motor_status
motion_reset_encoder
motion_get_encoder
set_xytable_parameters
set_home_real
motion_moveto_real
get_rotary_pulse_count
motion_jog_start
motion_jog_end
reset_motion_pulse_count
set_home_direction
set_home_mode
set_xytable_real
set_motion_soft_limit

List commands

Vector Command

jump_abs
jump_rel
mark_abs
mark_rel
timed_jump_abs
timed_jump_rel
timed_mark_abs
timed_mark_rel
jump_3d_abs
jump_3d_rel
mark_3d_abs
mark_3d_rel
timed_jump_3d_abs
timed_jump_3d_rel
timed_mark_3d_abs
timed_mark_3d_rel
z_out_list

List Control

set_end_of_list
set_wait
set_list_jump

list_call
list_return
list_call_cond
list_jump_cond
set_cond_timeout
[Scanner Parameters](#)
set_jump_speed
set_mark_speed
set_wobble
[Laser Control and Parameters](#)
set_laser_timing
set_firstpulse_killer_list
set_standby_list
laser_on_list
laser_signal_on_list
laser_signal_off_list

Delays

list_nop
set_laser_delays
set_laser_delays_ex
set_scanner_delays
long_delay

Coordinate transformations

set_matrix_list
set_offset_list

I/O controls

clear_io_cond_list
set_io_cond_list
write_8bit_port_list
write_da_x_list
write_io_port_list
write_io_port_list_ex
set_marking_ready_list

Motion control

list_motion_moveto
list_set_xytable
motion_home_list
xytable_home_list

list_motion_xy

list_set_motion_pos

Mark On Fly

set_fly_x_list

set_fly_y_list

set_ext_start_delay_list

simulate_encoder_list

list_set_encoder_cnt

list_wait_encoder_cnt

Bitmap

set_pixel_line

set_pixel

set_n_pixel

Others

save_and_restart_timer

set_label_list

Data Type

Data Format	Range	C
64-bit IEEE floating point format		double
signed 16-bit value	[-32768; +32767]	short
signed 32-bit value	$[-2^{31}; +2^{31}-1]$	long
pointer to a null-terminated ANSI string, 1 byte per char		char*
unsigned 16-bit value	[0; 65535]	unsigned short

Control Commands

Name	load_program_file
Type	short load_program_file (const char* FileName)
Function	Download firmware file.
Parameter	FileName File name with full path.
Return	ErrorCode: 0 Success 3 Open File Error 11 Card is not found
Comment	<ol style="list-style-type: none"> 1. Must call get_msk5_version to reboot and get card version before calling load_program_file. 2. load_program_file must be called before calling other functions except get_msk5_version, msk5_count_cards, and select_msk5.

Name	load_correction_file
Type	short load_correction_file(const char* FileName)
Function	Download correction file (2D).
Parameter	FileName File name with full path.
Return	ErrorCode: 0 Success 3 Open File Error 10 File Format Error
Comment	<ol style="list-style-type: none"> 1. Support correction file : cor, gcd, and ctb ° 2. Only set compensation values of X and Y. Values of Z are filled by zero.

Name	load_correction_file_3d
Type	short load_correction_file_3d(const char* FileName, short zero_offset_z)
Function	Download correction file (3D).
Parameter	FileName File name with full path. zero_offset_z Zero offset of Z.
Return	ErrorCode: 0 Success 3 Open File Error 10 File Format Error
Comment	<ol style="list-style-type: none"> 3. Support correction file : cor, gcd, and ctb ° 1. Set compensation values of X, Y, and Z. Values of Z are shifted by zero offset.

Ctrl Command	set_lens_file
Type	<code>short set_lens_file(const char* FileName)</code>
Function	Set correction lens file.
Parameter	FileName Full path file name.

Ctrl Command	get_k_factor
Prototype	<code>double get_k_factor(void)</code>
Function	Get K factor.
Comments	K factor: Position scale. (default 1, -32768 ~ +32767)

Name	msk5_count_cards
Type	<code>unsigned short msk5_count_cards(void)</code>
Function	Return count of card.
Return	count of card.
Comment	<ol style="list-style-type: none"> 1. Use to check card exist. 2. Also see select_msk5 and get_serial_number for multcard control.

Name	get_msk5_version
Type	<code>unsigned long get_msk5_version(void)</code>
Function	Get Card version and restart DSP.
Return Value	Card vsrsion (0x00CCAABB): AA.BB.CC
Comment	<ol style="list-style-type: none"> 1. Must call get_msk5_version to reboot and get card version before calling load_program_file. 2. Firmware file map : MSK5_AA_BB_1.srec

Name	get_hex_version
Type	<code>unsigned short get_hex_version(void)</code>
Function	Get HEX version.
Return Value	HEX version

Name	set_control_mode
Type	<code>void set_control_mode(unsigned short control_mode)</code>
Function	Set control mode.
Parameter	control_mode: Bit 0 Enable ext start. Bit 3 Enable ext stop to disable ext start.
Comment	1. Execute counter will be zero after set_control_mode .

	2. Ext start will disable after stop_execution .
--	---

Name	select_list
Type	void select_list(unsigned short list)
Function	Set execute List by ext start.
Parameter	list 0 ~ 11 → List 1 ~ List 12
Comment	List 1 is default.

Name	set_max_counts
Type	void set_max_counts(long max_count)
Function	Set max execution count by ext start.
Parameter	max_count
Comment	1. Set zero to unlimited. (default 0) 2. Auto disable ext start when execution count is arrive max count.

Name	get_counts
Type	long get_counts(void)
Function	Get ext execution count.
Return	Execution count.
Comment	Zero by set_control_mode .

Name	get_startstop_info
Type	unsigned short get_startstop_info(void)
Function	Get Start and Stop status.
Return	StartStop Info.: Bit 0 = 1 internal Start (execute_list, etc.) Bit 1 = 1 external Start (edge trigger) Bit 2 = 1 internal Stop (stop_execution) Bit 3 = 1 external Stop Bit 5 = 1 external Start (external status) Bit 9 = 1 Laser Enable
Comment	StartStop status will be zero after this function be called.

Name	stop_execution
Type	void stop_execution(void)

Function	Stop execution.
Comment	<ol style="list-style-type: none"> 1. Scanner position will keep. 2. LaserON signal will turn off. 3. Motion will stop. 4. Disable ext start.

Name	set_hw_config
Type	<code>void set_hw_config(long IHWCfg)</code>
Function	Set card hardware setting.
Parameter	<p>IHWCfg:</p> <ul style="list-style-type: none"> Bit 0 = 1 Scan XY exchange Bit 1 = 1 LaserON Reverse Bit 2 = 1 START1 Reverse Bit 3 = 1 STOP1 Reverse Bit 4 = 1 STOP2 anode, or STOP2 cathode Bit 5 = 1 STOP1 anode, or STOP1 cathode Bit 6 = 1 START2 anode, or START2 cathode Bit 7 = 1 START1 anode, or START1 cathode Bit 8 = 1 START2 Reverse Bit 9 = 1 STOP2 Reverse Bit 10 = 1 BUSY Reverse Bit 11 = 1 Motion Mode: CW / CCW, or Pulse / Dir Bit 12 = 1 PR Reverse Bit 13 = 1 MR Reverse Bit 14 = 1 ME Reverse Bit 15 = 1 PWM Reverse Bit 16 = 1 FPK Reverse Bit 17 = 1 Scan XY Swap Bit 18 = 1 DA1 Max 5V, or 10V Bit 19 = 1 DA2 Max 5V, or 10V Bit 23 – Bit 20 Card ID Bit 24 = 1 R05, or FPK Bit 27 – Bit 26 Axis (P4) Bit 28 = 1 X mirror Bit 29 = 1 Y mirror Bit 30 = 1 Z mirror

Name	get_hw_config
-------------	----------------------

Type	<code>long get_hw_config(void)</code>
Function	Get card hardware setting.
Return	See <code>set_hw_config</code> .

Name	get_uid_info
Type	<code>int get_uid_info(long *pUID)</code>
Function	Get user ID value.
Parameter	pUID return user ID value
Return	0 = Failed; 1 = Success

Name	get_module_info
Type	<code>int get_module_info(unsigned char *pMod)</code>
Function	Get supported module of card.
Parameter	pMod return module value (30 Bytes): Bit 1 Rotary Bit 2 XY Table Bit 3 Mark on Fly Others Reserved
Return	0 = Failed; 1 = Success
Comment	pMod must be array of 30 items. Ex: unsigned char pMod[30];

Name	set_user_info
Type	<code>int set_user_info(unsigned char *pData)</code>
Function	Set the user data.
Parameter	pData user data
Return	0 = Failed; 1 = Success
Comment	Size for writing must be 30 Bytes.

Name	get_user_info
Type	<code>int get_user_info(unsigned char *pData)</code>
Function	Get the user data.
Parameter	pData return user data
Return	0 = Failed; 1 = Success
Comment	pData must be array of 30 items. Ex:

	unsigned char pData [30];
--	---------------------------

Name	set_laser_mode
Type	void set_laser_mode(unsigned short mode)
Function	Set laser control mode.
Parameter	mode = 0 : CO2 mode = 1 : YAG mode 1 = 2 : YAG mode 2 = 3 : YAG mode 3
Comment	See Appendix A.

Name	set_softstart_mode
Type	void set_softstart_mode (unsigned short mode, unsigned short number, unsigned short restartdelay)
Function	Set soft start control. (for CO2)
Parameter	mode 1 Enable ; 0 Disable number Pulse number · 1 ~ 16 restartdelay reserve
Comment	Only use for CO2 mode.

Name	set_softstart_level
Type	void set_softstart_level (unsigned short index, unsigned short level)
Function	Set soft start Pulse ratio.
Parameter	Index Pulse number index, 0 ~ 15 Level Pulse ratio, 0 ~ 100 (%)
Comment	Only use for CO2 mode.

Name	set_standby
Type	void set_standby(unsigned long half_period, unsigned long pulse_width)
Function	Set standby pulse.
Parameter	half_period Standby standby half period, unit: 1 = 0.02us pulse_width Standby pulse width, unit: 1 = 0.02us
Comment	1. PWM keep Standby when laser OFF. 2. Support for all laser type. (Commonly used in CO2) 3. Disable by set_standby(0,0) .

Name	disable_laser
Type	void disable_laser(void)
Function	Disable laser control signals.
Comment	Laser control signals: LaserON, PWM, and FPK.

Name	enable_laser
Type	void enable_laser(void)
Function	Enable laser control signals.

Name	load_varpolydelay
Type	short load_varpolydelay(const char* STBFileName, unsigned short TableNo)
Function	Set vary polygon delay.
Parameter	STBFileName STB file with full path. TableNo Table number, [VarPolyTable#NO]
return	Error code: 0 Success. 3 File Open Error
Comment	<p>1. Reset after calling load_program_file.</p> <p>2. STB File Format: [VarPolyTable1] ;[VarPolyTable#NO] #NO is TableNo Angle1=20 ; Deg Scale1=0.5 ; Scale rate Angle2=40 Scale2=0.7 Angle3=60 Scale3=1 When Angle > Max Define Angle, the scale is the same as the max scale.</p>

Name	set_delay_mode
Type	void set_delay_mode(unsigned short vapoly, unsigned short directmove3d, unsigned short edgelevel, unsigned short MinJumpDelay, unsigned short

	JumpLengthLimit)
Function	Set delay mode.
Parameter	Vapoly enable / disable vary polygon delay. directmove3d reserve edgelevel reserve MinJumpDelay minmum jump delay JumpLengthLimit limit distance for vary jump delay.
Comment	1. JumpLengthLimit = 0, disable vary jump delay. 2. Jump distance < JumpLengthLimit : Var Jump Delay = (Jump Delay – MinJumpDelay) * Length / JumpLengthLimit + MinJumpDelay (Length < JumpLengthLimit)

Name	get_head_status
Type	unsigned short get_head_status(unsigned short head)
Function	Get XY2-100 Status
Parameter	head reverse
Return	XY2-100 16Bit Status

Name	goto_xy
Type	void goto_xy(short xpos, short ypos)
Function	Move scanner position. (x, y)
Parameter	xpos X position, unit: Pixel, range: -32768 ~ +32767 ypos Y position, unit: Pixel, range: -32768 ~ +32767
Comment	1. Move speed is 983025pixels/s. (for lens 100x100mm, 1500mm/s) 2. This is blocking function, return after move end.

Name	goto_xyz
Type	void goto_xyz(short xpos, short ypos, short zpos)
Function	Move scanner position. (x, y, z)
Parameter	xpos X position, unit: Pixel, range: -32768 ~ +32767 ypos Y position, unit: Pixel, range: -32768 ~ +32767 zpos Z position, unit: Pixel, range: -32768 ~ +32767
Comment	1. Move speed is 983025pixels/s. 2. This is blocking function, return after move end.

Name	z_out
-------------	--------------

Type	<code>void</code> <code>z_out</code> (<code>short</code> value)
Function	Move scanner Z position.
Parameter	value Z position, unit: Pixel, range: -32768 ~ +32767
Comment	3. Move speed is 983025pixels/s. 1. This is blocking function, return after move end.

Name	set_matrix
Type	<code>void</code> <code>set_matrix</code> (<code>double</code> m11, <code>double</code> m12, <code>double</code> m21, <code>double</code> m22)
Function	Set scanner matrix.
Parameter	m11, m12, m21, m22 matrix parameters, default: (1, 0, 0, 1)
Comment	Example: $\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} m11 & m12 \\ m21 & m22 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} Xo \\ Yo \end{bmatrix}$ · X & Y are original · Xo & Yo are offset Rotation α $\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$ Scale $\begin{bmatrix} kx & 0 \\ 0 & ky \end{bmatrix}$ Mixed $\begin{bmatrix} kx \cdot \cos \alpha & ky \cdot (-\sin \alpha) \\ kx \cdot \sin \alpha & ky \cdot \cos \alpha \end{bmatrix}$

Name	set_offset
Type	<code>void</code> <code>set_offset</code> (<code>short</code> x_offset, <code>short</code> y_offset)
Function	Set scanner offset
Parameter	x_offset X Offset, Unit: Pixel, Eange: -32768 ~32767 y_offset Y Offset, Unit: Pixel, Eange: -32768 ~32767

Name	read_io_port
Type	<code>unsigned short</code> <code>read_io_port</code> (<code>void</code>)
Function	Get 16Bit TTL input status.
Return	16Bit TTL Input

Name	write_io_port
Type	<code>void</code> <code>write_io_port</code> (<code>unsigned short</code> value)
Function	Set 16Bit TTL Output
Parameter	value 16Bit TTL Output

Name	get_io_status
Type	unsigned short get_io_status(void)
Function	Get 16Bit TTL output status.
Return	16Bit TTL output status

Name	write_8bit_port
Type	void write_8bit_port(unsigned short Bit,unsigned short bOn)
Function	Set laser control output.
Parameter	Bit 1 ~ 7 → L1 ~ L7 8 → Program Ready 9 → Marking Ready 10 → Marking End bOn 0 / 1
Comment	Bit 0 is not controled. (L0 is LaserON)

Name	write_da_x
Type	void write_da_x (unsigned short x, unsigned short value)
Function	Set analog output.
Parameter	x 1, Analog Out 1 2, Analog Out 2 value 0 ~ 1023 → 0 ~ 10V

Name	write_io_port_ex
Type	void write_io_port_ex(unsigned short value)
Function	Set extension 16Bit output.
Parameter	value Extension 16Bit output

Name	get_io_status_ex
Type	unsigned short get_io_status_ex(void)
Function	Get extension 16Bit output status.
Return	Extension 16Bit output status

Name	set_program_ready
Type	void set_program_ready(unsigned short value)
Function	Set Program Ready.
Parameter	value Program Ready 0 / 1

Comment	Also control by write_8bit_port Bit 8.
---------	---

Name	set_marking_ready
Type	<code>void set_marking_ready(unsigned short value)</code>
Function	Set Marking Ready.
Parameter	value Marking Ready 0 / 1
Comment	Also control by write_8bit_port Bit 9.

Name	set_marking_end
Type	<code>void set_marking_end(unsigned short value)</code>
Function	Set Marking End.
Parameter	value Marking End 0 / 1
Comment	Also control by write_8bit_port Bit 10.

Name	set_start_list
Type	<code>void set_start_list(unsigned short list_no)</code>
Function	Start set List.
Parameter	list_no 1 ~ 12 → List 1 ~ List 12
Comment	Call before list commands.

Name	execute_list
Type	<code>void execute_list(unsigned short list_no)</code>
Function	Execute List.
Parameter	list_no 1 ~ 12 → List 1 ~ List 12

Name	stop_list
Type	<code>void stop_list(void)</code>
Function	Pause List.
Comment	Only pause at laser is turn off.

Name	restart_list
Type	<code>void restart_list(void)</code>
Function	Resume List.
Comment	Use after stop_list .

Name	get_status
Type	<code>BOOL get_status(unsigned short *busy, unsigned long *position)</code>

Function	Get execution status.
Parameter	busy 1 Execute, 0 IDle position Execution position, Position range: 0 ~ 119999
Comment	Real Execution Position: Execute Pos = Current List – 1 * 10000 + List Command Position

Name	read_status
Type	<code>unsigned long read_status(void)</code>
Function	Get List execution status. (bit based)
Return	List Status: (1 Execute, 0 IDle) Bit 0 : List 1 Bit 1 : List 2 ... Bit 11: List 12

Name	get_wait_status
Type	<code>unsigned short get_wait_status(void)</code>
Function	Get wait status.
Return	Wait status
Comment	Wait status is set by set_wait .

Name	release_wait
Type	<code>void release_wait(void)</code>
Function	Release wait.
Comment	Let List continue execute when waiting.

Name	auto_change
Type	<code>void auto_change(short sList)</code>
Function	Auto change list execute.
Parameter	sList 0 ~ 11 → List 1 ~ List 12
Comment	1. If no list execution, it is the same to execute_list . 2. Support auto-change to 12 set.

Name	auto_change_pos
Type	<code>void auto_change_pos(unsigned long start)</code>
Function	Auto change list position execute.
Parameter	start Command Position: 0 ~ 119999

Comment	<ol style="list-style-type: none"> 1. If no list execution, it is the same to execute_at_pointer. 2. Only once time support.
---------	---

Name	start_loop
Type	<code>void start_loop(unsigned short sListNum)</code>
Function	Set List auto loop.
Parameter	sListNum loop List number, 0 ~ 11
Comment	Loop: List 1 to List Num + 1

Name	quit_loop
Type	<code>void quit_loop(void)</code>
Function	End List loop.
Comment	The last Execute List will be done.

Name	get_list_size
Type	<code>long get_list_size(void)</code>
Function	Get all List Command size.
Return	Must be 200000

Name	set_input_pointer
Type	<code>void set_input_pointer(unsigned long pointer)</code>
Function	Start set List position.
Parameter	pointer Command Position: 0 ~ 119999

Name	get_input_pointer
Type	<code>unsigned long get_input_pointer(void)</code>
Function	Get List set position.
Return	Command Position: 0 ~ 119999

Name	execute_at_pointer
Type	<code>void execute_at_pointer(unsigned long pointer)</code>
Function	Execute List position.
Parameter	pointer Command Position: 0 ~ 119999
Comment	<p>Example:</p> <p><code>execute_list(2)</code> → Start Execute Potion = 10000, List 2</p> <p><code>execute_at_pointer (40050)</code> → Start Execute Potion = 40050, List 5</p>

Name	set_label
Type	void set_label(long value)
Function	Set label register.
Parameter	Label value (32Bit Long Integer)

Name	get_label
Type	long get_label(void)
Function	Get label register.
Return	Label value
Comment	used to get execution information. (by set_label_list)

Name	set_motion_backspace
Type	void set_motion_backspace(short axis, long pulse)
Function	Set Axis backspace.
Parameter	axis 0 R 1 X 2 Y 3 Z pulse compensation Pulse number
Comment	Work after homing.

Name	get_motor_sensor
Type	short get_motor_sensor(void)
Function	Get sensor status.
Return	Sensor status: Bit 0 R Axis Home Bit 1 R Axis Limit+ Bit 2 R Axis Limit- Bit 3 X Axis Home Bit 4 X Axis Limit+ Bit 5 X Axis Limit- Bit 6 Y Axis Home Bit 7 Y Axis Limit+ Bit 8 Y Axis Limit- Bit 9 Z Axis Home Bit 10 Z Axis Limit+ Bit 11 Z Axis Limit-

Name	get_motion_inpos
Type	short get_motion_inpos(void)
Function	Get in-position signal.

Return	In-position signal: Bit 0 R Axis Bit 1 X Axis Bit 2 Y Axis Bit 3 Z Axis
--------	---

Name	get_motor_status
Type	<code>short get_motor_status(void)</code>
Function	Get motion status.
Return	Motion status: Bit 0 = 1 R motion Bit 1 = 1 X motion Bit 2 = 1 Y motion Bit 3 = 1 Z motion Bit 4 = 1 R Homing Bit 5 = 1 X Homing Bit 6 = 1 Y Homing Bit 7 = 1 Z Homing

Name	motion_reset_encoder
Type	<code>void motion_reset_encoder(short axis)</code>
Function	Set Encoder zero.
Parameter	Encoder Index: 0 ~ 2

Name	motion_get_encoder																
Type	<code>void motion_get_encoder(long *pulEList)</code>																
Function	Get Encoder values.																
Parameter	<table border="0" style="width: 100%;"> <tr> <td style="width: 30%;">pulEList</td> <td style="width: 20%;">Encoder</td> <td style="width: 20%;">Index 0</td> <td style="width: 30%;">Reserve</td> </tr> <tr> <td></td> <td></td> <td>Index 1</td> <td>Encoder 0</td> </tr> <tr> <td></td> <td></td> <td>Index 2</td> <td>Encoder 1</td> </tr> <tr> <td></td> <td></td> <td>Index 3</td> <td>Encoder 2</td> </tr> </table>	pulEList	Encoder	Index 0	Reserve			Index 1	Encoder 0			Index 2	Encoder 1			Index 3	Encoder 2
pulEList	Encoder	Index 0	Reserve														
		Index 1	Encoder 0														
		Index 2	Encoder 1														
		Index 3	Encoder 2														

Name	set_xytable_parameters				
Type	<code>void set_xytable_parameters(BYTE axis,ROTARY_PARAM* pParam)</code>				
Function	Set axis parameter.				
Parameter	<table border="0" style="width: 100%;"> <tr> <td style="width: 30%;">axis</td> <td style="width: 70%;">0 R 1 X 2 Y 3 Z</td> </tr> <tr> <td>pParam</td> <td>structure pointer</td> </tr> </table>	axis	0 R 1 X 2 Y 3 Z	pParam	structure pointer
axis	0 R 1 X 2 Y 3 Z				
pParam	structure pointer				
Comment	<code>struct{</code>				

	long	m_lAccTime;	Accerated time(ms)
	long	m_lRotarySpeed;	Speed(pulse/s)
	long	m_lPulseCount;	Resolution (Scale 1000) (pulse/mm) for X, Y and Z (pulse/deg) dor Rotary
	BOOL	m_bGoTo0_MarkEnd;	Reserve
	long	m_lMotorCount;	Reserve
	BOOL	m_bAH_InPos;	In-position signal
	BOOL	m_bAH_Home;	Home signal
	long	m_lIniSpeed;	initial speed (pulse/s)
	BOOL	m_bReverse;	Reverse mode
	BOOL	m_bAH_Limit;	limit signal
	long	m_lHomeSpeed;	Homing speed (pulse/s)
	long	m_lHomeBackSpeed;	HomingBack speed (pulse/s)
	BOOL	m_bEnable;	Reserve
	long	m_lINPos_TimeOut;	In-position timeout
	long	m_lINPos_Delay;	In-position delay
	}ROTARY_PARAM		

Name	set_home_real
Type	long set_home_real(short axis)
Function	Set Axis Homing
Parameter	axis 0 R 1 X 2 Y 3 Z
Return	Fixed 1
Comment	1. Set motion parameters before homing. 2. This is nonblocking function. Use get_motor_status to comform motion end.

Name	motion_moveto_real
Type	void motion_moveto_real(BYTE axis, long lPulse)
Function	Set Axis move.
Parameter	axis 0 R 1 X 2 Y 3 Z lPulse absolute position, unit: Pulse Count
Comment	3. Set motion parameters before moving. 1. This is nonblocking function. Use get_motor_status to comform motion end.

Name	get_rotary_pulse_count
Type	long get_rotary_pulse_count(BYTE axis)
Function	Get axis position.

Parameter	axis 0 R 1 X 2 Y 3 Z
Comment	absolute position,, unit: Pulse Count

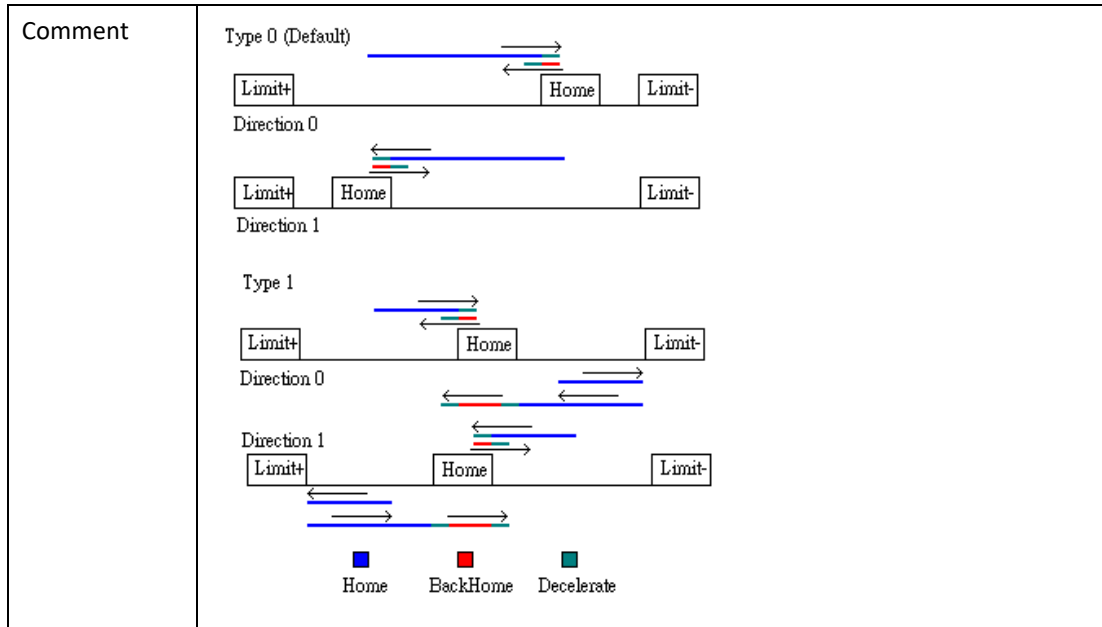
Name	motion_jog_start
Type	<code>void motion_jog_start(int iAxis, long lSpeed, long lAccTime, int iDir)</code>
Function	Start Jog moving.
Parameter	iAxis 0 R 1 X 2 Y 3 Z lSpeed Jog Speed, unit: Pulse/s lAccTime Acc Time, Unit: ms iDir Direction: 1 Positive; -1 negative

Name	motion_jog_end
Type	<code>void motion_jog_end(int iAxis)</code>
Function	Stop Jog moving.
Parameter	iAxis 0 R 1 X 2 Y 3 Z
Comment	Use get_motor_status to conform motion end.

Name	reset_motion_pulse_count
Type	<code>void reset_motion_pulse_count(short axis,long pulse)</code>
Function	Set axis position.
Parameter	axis 0 R 1 X 2 Y 3 Z pulse absolute position, unit: Pulse Count

Name	set_home_direction
Type	<code>void set_home_direction(short axis,short dir)</code>
Function	Set axis Home direction.
Parameter	axis R 1 X 2 Y 3 Z dir 0 home start to negative; 1 home start to positive

Name	set_home_mode
Type	<code>void set_home_mode(short axis,short sMode)</code>
Function	Set Home mode
Parameter	axis 0 R 1 X 2 Y 3 Z sMode Home mode (0 / 1)



Name	set_xytable_real
Type	<code>void set_xytable_real(long IR,long IX , long IY,long IZ)</code>
Function	Set axis moving.
Parameter	IR absolute position, unit: Pulse Count IX absolute position, unit: Pulse Count IY absolute position, unit: Pulse Count IZ absolute position, unit: Pulse Count
Comment	4. Set motion parameters before moving. 1. This is nonblocking function. Use get_motor_status to conform motion end.

Name	set_motion_soft_limit
Type	<code>BOOL set_motion_soft_limit(short sAxis,short sEnable,long lLimitP,long lLimitN)</code>
Function	Set soft axis limit.
Parameter	sAxis 0 R 1 X 2 Y 3 Z sEnable 1 / 0 lLimitP Soft Limit+ Pulse Position lLimitN Soft Limit- Pulse Position
Comment	Auto disable when homing.

List Commands

Name	jump_abs
Type	void jump_abs(short xval, short yval)
Function	Scanner jump move
Parameter	xval X Absolute Position, Range: -32768 ~ 32767 yval Y Absolute Position, Range: -32768 ~ 32767

Name	jump_rel
Type	void jump_rel(short xval, short yval)
Function	Scanner jump move
Parameter	xval X Relative Position, Range: -32768 ~ 32767 yval Y Relative Position, Range: -32768 ~ 32767

Name	mark_abs
Type	void mark_abs(short xval, short yval)
Function	Scanner mark move
Parameter	xval X Absolute Position, Range: -32768 ~ 32767 yval Y Absolute Position, Range: -32768 ~ 32767

Name	mark_rel
Type	void mark_rel (short xval, short yval)
Function	Scanner mark move
Parameter	xval X Relative Position, Range: -32768 ~ 32767 yval Y Relative Position, Range: -32768 ~ 32767

Name	timed_jump_abs
Type	void timed_jump_abs (short xval, short yval, long step)
Function	Scanner jump move in time step.
Parameter	xval X Absolute Position, Range: -32768 ~ 32767 yval Y Absolute Position, Range: -32768 ~ 32767 step time step, 1 = 10us

Name	timed_jump_rel
Type	void timed_jump_rel (short xval, short yval, long step)
Function	Scanner jump move in time step.
Parameter	xval X Relative Position, Range: -32768 ~ 32767

	yval	Y Relative Position, Range: -32768 ~ 32767
	step	time step, 1 = 10us

Name	timed_mark_abs	
Type	void timed_mark_abs (short xval, short yval, long step)	
Function	Scanner mark move in time step.	
Parameter	xval	X Absolute Position, Range: -32768 ~ 32767
	yval	Y Absolute Position, Range: -32768 ~ 32767
	step	time step, 1 = 10us

Name	timed_mark_rel	
Type	void timed_mark_rel(short xval, short yval, long step)	
Function	Scanner mark move in time step.	
Parameter	xval	X Relative Position, Range: -32768 ~ 32767
	yval	Y Relative Position, Range: -32768 ~ 32767
	step	time step, 1 = 10us

Name	jump_3d_abs	
Type	void jump_3d_abs(short xval, short yval, short zval)	
Function	3D Scanner jump move	
Parameter	xval	X Absolute Position, Range: -32768 ~ 32767
	yval	Y Absolute Position, Range: -32768 ~ 32767
	zval	Z Absolute Position, Range: -32768 ~ 32767

Name	jump_3d_rel	
Type	void jump_3d_rel (short xval, short yval, short zval)	
Function	3D Scanner jump move	
Parameter	xval	X Relative Position, Range: -32768 ~ 32767
	yval	Y Relative Position, Range: -32768 ~ 32767
	zval	Z Relative Position, Range: -32768 ~ 32767

Name	mark_3d_abs	
Type	void mark_3d_abs (short xval, short yval, short zval)	
Function	3D Scanner mark move	
Parameter	xval	X Absolute Position, Range: -32768 ~ 32767
	yval	Y Absolute Position, Range: -32768 ~ 32767
	zval	Z Absolute Position, Range: -32768 ~ 32767

Name	mark_3d_rel
Type	void mark_3d_rel (short xval, short yval,short zval)
Function	3D Scanner mark move
Parameter	xval X Relative Position, Range: -32768 ~ 32767 yval Y Relative Position, Range: -32768 ~ 32767 zval Z Relative Position, Range: -32768 ~ 32767

Name	timed_jump_3d_abs
Type	void timed_jump_3d_abs (short xval, short yval,short zval,long step)
Function	3D Scanner jump move in time step.
Parameter	xval X Absolute Position, Range: -32768 ~ 32767 yval Y Absolute Position, Range: -32768 ~ 32767 zval Z Absolute Position, Range: -32768 ~ 32767 step time step, 1 = 10us

Name	timed_jump_3d_rel
Type	void timed_jump_3d_rel (short xval, short yval,short zval,long step)
Function	3D Scanner jump move in time step.
Parameter	xval X Relative Position, Range: -32768 ~ 32767 yval Y Relative Position, Range: -32768 ~ 32767 zval Z Relative Position, Range: -32768 ~ 32767 step time step, 1 = 10us

Name	timed_mark_3d_abs
Type	void timed_mark_3d_abs (short xval, short yval,short zval,long step)
Function	3D Scanner mark move in time step.
Parameter	xval X Absolute Position, Range: -32768 ~ 32767 yval Y Absolute Position, Range: -32768 ~ 32767 zval Z Absolute Position, Range: -32768 ~ 32767 step time step, 1 = 10us

Name	timed_mark_3d_rel
Type	void timed_mark_3d_rel (short xval, short yval,short zval,long step)
Function	3D Scanner mark move in time step.
Parameter	xval X Relative Position, Range: -32768 ~ 32767 yval Y Relative Position, Range: -32768 ~ 32767 zval Z Relative Position, Range: -32768 ~ 32767 step time step, 1 = 10us

Name	z_out_list
Type	void z_out_list (short value)
Function	Z Scanner jump move
Parameter	value Z Absolute Position, Range: -32768 ~ 32767

Name	set_end_of_list
Type	void set_end_of_list(void)
Function	List end
Comment	Must add this command in list end.

Name	set_wait
Type	void set_wait(unsigned short wait_word)
Function	Set wait point.
Parameter	wait_word wait status
Comment	1. Get wait status by get_wait_status . 2. Continue list by release_wait .

Name	set_list_jump
Type	void set_list_jump(unsigned long address)
Function	Jump to setting List Command.
Parameter	address List Command Position: 0 ~ 119999

Name	list_call
Type	void list_call(unsigned long address)
Function	Jump to setting List Command function.
Parameter	address Command Function Position: 0 ~ 119999
Comment	Command Function must end by list_return .

Name	list_return
Type	void list_return (void)
Function	Define command function return.
Comment	Use for list_call .

Name	list_call_cond
Type	void list_call_cond (unsigned short select_io, unsigned short mask_1, unsigned

	<code>short mask_0, unsigned long address)</code>								
Function	Jump to setting List Command function when IN matched.								
Parameter	<table border="0"> <tr> <td><code>select_io</code></td> <td>Reserve</td> </tr> <tr> <td><code>mask_1</code></td> <td>1 check</td> </tr> <tr> <td><code>mask_0</code></td> <td>0 check</td> </tr> <tr> <td><code>address</code></td> <td>Command Function Position: 0 ~ 119999</td> </tr> </table>	<code>select_io</code>	Reserve	<code>mask_1</code>	1 check	<code>mask_0</code>	0 check	<code>address</code>	Command Function Position: 0 ~ 119999
<code>select_io</code>	Reserve								
<code>mask_1</code>	1 check								
<code>mask_0</code>	0 check								
<code>address</code>	Command Function Position: 0 ~ 119999								
Comment	<ol style="list-style-type: none"> 1. Command Function must end by list_return. 2. <code>mask_1</code>: Digital Bit 1 Check (1 enable check bit) 3. <code>mask_0</code>: Digital Bit 0 Check (1 enable check bit) <p>Ex: <code>mask_1 = 0x0001, mask_0 = 0x0002</code> → Input Bit 0 = 1, Bit 1 = 0 → call jump</p>								

Name	list_jump_cond								
Type	<code>void list_jump_cond(unsigned short select_io, unsigned short mask_1, unsigned short mask_0, unsigned long address)</code>								
Function	Jump to setting List Command when IN matched.								
Parameter	<table border="0"> <tr> <td><code>select_io</code></td> <td>Reserve</td> </tr> <tr> <td><code>mask_1</code></td> <td>1 check</td> </tr> <tr> <td><code>mask_0</code></td> <td>0 check</td> </tr> <tr> <td><code>address</code></td> <td>Command Function Position: 0 ~ 119999</td> </tr> </table>	<code>select_io</code>	Reserve	<code>mask_1</code>	1 check	<code>mask_0</code>	0 check	<code>address</code>	Command Function Position: 0 ~ 119999
<code>select_io</code>	Reserve								
<code>mask_1</code>	1 check								
<code>mask_0</code>	0 check								
<code>address</code>	Command Function Position: 0 ~ 119999								
Comment	<ol style="list-style-type: none"> 1. <code>mask_1</code>: Digital Bit 1 Check (1 enable check bit) 2. <code>mask_0</code>: Digital Bit 0 Check (1 enable check bit) <p>Ex: <code>mask_1 = 0x0001, mask_0 = 0x0002</code> → Input Bit 0 = 1, Bit 1 = 0 → list jump</p>								

Name	set_cond_timeout
Type	<code>void set_cond_timeout(long value)</code>
Function	Set check Condition timeout.
Parameter	<code>value</code> Timeout, unit: us
Comment	0 is infinited. (default 0)

Name	set_jump_speed
Type	<code>void set_jump_speed(double jump_speed)</code>
Function	Set scanner jump move speed.
Parameter	<code>jump_speed</code> , unit: Pixel/ms
Comment	Set before jump move.

Name	set_mark_speed
-------------	-----------------------

Type	<code>void set_mark_speed(double mark_speed)</code>
Function	Set scanner mark move speed.
Parameter	mark_speed, unit: Pixel/ms
Comment	Set before mark move.

Name	set_wobble
Type	<code>void set_wobble(unsigned short amplitude, double frequency)</code>
Function	Set wobble parameters.
Parameter	amplitude Unit: Pixel frequency Unit: Hz
Comment	Amplitude = 0 → Disable wobble

Name	set_laser_timing
Type	<code>void set_laser_timing(unsigned long half_period, unsigned long pulse_width1, unsigned long pulse_width2, unsigned short time_base)</code>
Function	Set PWMparameters.
Parameter	half_period PWM half period (Unit refrence to time_base) pulse_width1 PWM pulse width (Unit refrence to time_base) pulse_width2 FPK pulse width (Unit refrence to time_base) time_base 0 Unit: 1us ; 1 Unit: 0.02us
Comment	CO2 Mode: FPK is same as PWM (Phase lag 180 deg.). Pulse width is set by pulse_width2.

Name	set_firstpulse_killer_list
Type	<code>void set_firstpulse_killer_list (unsigned long fpk, unsigned long lead_time)</code>
Function	Set FPK parameters.
Parameter	fpk FPK Pulse width, unit: 0.02us lead_time FPK leading time, unit: 0.02us
Comment	1. fpk = -1 → FPK always High (the same as LaserON) 2. lead_time is used for YAG3.

Name	set_standby_list
Type	<code>void set_standby_list(unsigned long half_period, unsigned long pulse_width)</code>
Function	Set standby parameters.
Parameter	half_period Standby half period, unit: 0.02us pulse_width Standby pulse width, unit: 0.02us
Comment	1. When LaserOFF, PWM keep Standby.

	2. Support for all laser mode. 3. Disable by set_standby(0,0) .
--	---

Name	laser_on_list
Type	void laser_on_list (long delay)
Function	laserON in delay time
Parameter	delay LaserON Time, unit: us

Name	laser_signal_on_list
Type	void laser_signal_on_list (void)
Function	Laser ON

Name	laser_signal_off_list
Type	void laser_signal_off_list (void)
Function	Laser OFF

Name	list_nop
Type	void list_nop(void)
Function	Delay 10us

Name	set_laser_delays
Type	void set_laser_delays (long laser_on_delay, long laser_off_delay)
Function	Set laser delays.
Parameter	laser_on_delay unit: us laser_off_delay unit: us

Name	set_laser_delays_ex
Type	void set_laser_delays_ex (long laser_on_delay, long laser_off_delay)
Function	Set laser delays.
Parameter	laser_on_delay unit: 0.02us laser_off_delay unit: 0.02us

Name	set_scanner_delays
Type	void set_scanner_delays(long jump_delay, long mark_delay, long polygon_delay)
Function	Set scanner delays.
Parameter	jump_delay unit: us mark_delay unit: us

	polygon_delay unit: us
Comment	1. Mark delay >= laser_off delay 2. Scanner delays must be multiple of 10us.

Name	long_delay
Type	void long_delay (unsigned short delay)
Function	Set delay.
Parameter	delay Unit: us
Comment	Delay must be multiple of 10us.

Name	set_matrix_list
Type	void set_matrix_list (unsigned short i, unsigned short j, double m_ij)
Function	Set matrix.
Parameter	i Col Index j Raw Index m_ij matrix parameter
Comment	Matrix Define : m_00m_10 m_01m_11

Name	set_offset_list
Type	void set_offset_list (short x_offset, short y_offset)
Function	Set offset
Parameter	x_offset unit: Pixel x_offset unit: Pixel

Name	clear_io_cond_list
Type	void clear_io_cond_list (unsigned short select_io, unsigned short mask_1, unsigned short mask_0, unsigned short mask_clear)
Function	Clear IO out when IN matched.
Parameter	select_io 0 16Bit Output ; 1 16Bit Extension Output mask_1 1 check mask_0 0 check mask_clear clear out bit mask
Comment	When matched, clear out bit by mask_clear.

Name	set_io_cond_list
-------------	-------------------------

Type	<code>void set_io_cond_list (unsigned short select_io, unsigned short mask_1, unsigned short mask_0, unsigned short mask_set)</code>								
Function	Set IO out when IN matched.								
Parameter	<table> <tr> <td>select_io</td> <td>0 16Bit Output ; 1 16Bit Extension Output</td> </tr> <tr> <td>mask_1</td> <td>1 check</td> </tr> <tr> <td>mask_0</td> <td>0 check</td> </tr> <tr> <td>mask_set</td> <td>set out bit mask</td> </tr> </table>	select_io	0 16Bit Output ; 1 16Bit Extension Output	mask_1	1 check	mask_0	0 check	mask_set	set out bit mask
select_io	0 16Bit Output ; 1 16Bit Extension Output								
mask_1	1 check								
mask_0	0 check								
mask_set	set out bit mask								
Comment	When matched, set out bit by mask_set.								

Name	write_8bit_port_list																				
Type	<code>void write_8bit_port_list (unsigned short Bit, unsigned short bOn)</code>																				
Function	Set laser output.																				
Parameter	<table> <tr> <td>Bit</td> <td>1 ~ 7</td> <td>→</td> <td>L1 ~ L7</td> </tr> <tr> <td></td> <td>8</td> <td>→</td> <td>Program Ready</td> </tr> <tr> <td></td> <td>9</td> <td>→</td> <td>Marking Ready</td> </tr> <tr> <td></td> <td>10</td> <td>→</td> <td>Marking End</td> </tr> <tr> <td>bOn</td> <td>0 / 1</td> <td></td> <td></td> </tr> </table>	Bit	1 ~ 7	→	L1 ~ L7		8	→	Program Ready		9	→	Marking Ready		10	→	Marking End	bOn	0 / 1		
Bit	1 ~ 7	→	L1 ~ L7																		
	8	→	Program Ready																		
	9	→	Marking Ready																		
	10	→	Marking End																		
bOn	0 / 1																				
Comment	L0 is not controled. (L0 is LaserON)																				

Name	write_da_x_list						
Type	<code>void write_da_x_list (unsigned short x, unsigned short value)</code>						
Function	Set analog output						
Parameter	<table> <tr> <td>x</td> <td>1, Analog Out 1</td> </tr> <tr> <td></td> <td>2, Analog Out 2</td> </tr> <tr> <td>value</td> <td>0 ~ 1023 → 0 ~ 10V</td> </tr> </table>	x	1, Analog Out 1		2, Analog Out 2	value	0 ~ 1023 → 0 ~ 10V
x	1, Analog Out 1						
	2, Analog Out 2						
value	0 ~ 1023 → 0 ~ 10V						

Name	write_io_port_list
Type	<code>void write_io_port_list (unsigned short value)</code>
Function	Set 16Bit TTL output
Parameter	value 16Bit output

Name	write_io_port_list_ex
Type	<code>void write_io_port_list_ex (unsigned short value)</code>
Function	Set Extension 16Bit TTL output
Parameter	value Extension 16Bit output

Name	set_marking_ready_list
-------------	-------------------------------

Type	<code>void set_marking_ready_list</code> (unsigned short value)
Function	Set Marking Ready
Parameter	value Marking Ready, 0 / 1
Comment	Can be set by <code>write_8bit_port_list</code> Bit 9.

Name	list_motion_moveto
Type	<code>Void list_motion_moveto</code> (BYTE axis, long IPulse)
Function	Axis move
Parameter	axis 0 R 1 X 2 Y 3 Z IPulse absolute position, unit: Pulse Count

Name	list_set_xytable
Type	<code>void list_set_xytable</code> (long IR, long IX , long IY, long IZ)
Function	Axis move
Parameter	IR R absolute position, unit: Pulse Count IX X absolute position, unit: Pulse Count IY Y absolute position, unit: Pulse Count IZ Z absolute position, unit: Pulse Count

Name	motion_home_list
Type	<code>void motion_home_list</code> (short axis)
Function	Set Homing
Parameter	axis 0 R 1 X 2 Y 3 Z

Name	xytable_home_list
Type	<code>void CALLBACK xytable_home_list</code> (short sel)
Function	Set Homing (bit based)
Parameter	sel Bit 0 = 1 → R 軸 Bit 1 = 1 → X 軸 Bit 2 = 1 → Y 軸 Bit 3 = 1 → Z 軸

Name	list_motion_xy
Type	<code>void list_motion_xy</code> (long xval, long yval)
Function	Axis move
Parameter	xval X absolute position, unit: Pulse Count yval Y absolute position, unit: Pulse Count

Name	simulate_encoder_list
Type	void simulate_encoder_list(unsigned short channel,unsigned short mode)
Function	Set on-fly mode
Parameter	channel 1 X 2 Y mode 1 encoder mode; 2 simulate mode
Comment	Set before enable on fly.

Name	list_set_encoder_cnt
Type	void list_set_encoder_cnt(short encoder,long delay)
Function	Set on fly count delay
Parameter	encoder 0 X 1 Y delay count delay encoder mode: Pulse Count simulate mode: us
Comment	Count start by enable on fly.

Name	list_wait_encoder_cnt
Type	void list_wait_encoder_cnt(void)
Function	Wait on fly count

Name	set_pixel_line
Type	void set_pixel_line (unsigned short pixel_mode, unsigned long pixel_period, double dx, double dy, unsigned short da_ch)
Function	Switches to the pixel output mode, defines various pixel output parameters and marks the beginning of a pixel line.
Parameter	pixel_mode = 0 the laser focus "jumps" from one pixel to the next. = 1 the laser focus moves from one pixel to the next in small steps (microvectors). pixel_period pixel output period, unit: 10us. dx, dy distance in the X and Y directions between adjacent pixels. da_ch Analog output channel (1 or 2).
Comment	<ul style="list-style-type: none"> ● Each image line must start with the command set_pixel_line. This command should be preceded by a jump command to the start point of the image line. ● The stable time For Digital to Analog convertor is about 66us. (Pixel frequency 15KHz)

Name	set_pixel
Type	void set_pixel (unsigned long pulse_width, unsigned long da_value)

Function	Defines the output parameters (laser pulse width and ANALOG OUT value) for one pixel in an image line.
Parameter	pulse_width laser output pulse width, unit: 0.02us. da_value Analog output value (0 - 1023)
Comment	<ul style="list-style-type: none"> ● Each image line must start with the command set_pixel_line. ● Pixels in the image line is defined by set(_n)_pixel command. The set(_n)_pixel commands must follow immediately after the command set_pixel_line. No other commands must be written into the list until the image line is completed.

Name	set_n_pixel
Type	void set_n_pixel (unsigned long pulse_width, unsigned long da_value, unsigned long cnt);
Function	Defines the output parameters (laser pulse width and ANALOG OUT value) for one pixel in an image line.
Parameter	pulse_width laser output pulse width, unit: 0.02us. da_value Analog output value (0 - 1023) cnt number of pixel count.
Comment	<ul style="list-style-type: none"> ● Each image line must start with the command set_pixel_line. ● Pixels in the image line is defined by set(_n)_pixel command. The set(_n)_pixel commands must follow immediately after the command set_pixel_line. No other commands must be written into the list until the image line is completed.

Name	save_and_restart_timer
Type	void save_and_restart_timer (void)
Function	Save and restart timer
Comment	Used for mark time.

Name	set_label_list
Type	void set_label_list (long value)
Function	Set label register
Comment	Get label by get_label.

Appendix A: Laser Timing

CO2

