Desktop HST Tester Design Requirements and Descriptions

Forrest Xu
Date: July 21, 2016
Status: Draft
V0.41

Change History

Date	Revision	Comments
March 31, 2016	V0.1	The first draft by Forrest Xu
April 6, 2016	V0.2	Add in Chapter 4 and onwards by Forrest Xu
April 15, 2016	V0.3	1. Modification for IO assignment. 2. Modification for HST_dt_set_output. 3. Add in “SM_jog_cw” , “SM_jog_ccw”, “SM_jog_stop”.
June 16, 2016	V0.4	1. Add in two commands: SM_get_status and SM_rel_move. 2. Change to the command SM_get_io_input 3. Change to the position index parameter for command HST_dt_servo_move_to_taught_position to be consistent with command HST_dt_teach_servo_position. 4. Add in Flattener 2 sensor inputs to HST_get_all_inputs command. 5. Add in Flattener 2 output to HST_set_output command. 6. Removed command “HST_dt_precisor_vac_on”. 7. Changes to GUI in Figure 6.2 and Figure 6.3 to add in some new command buttons, such as Flattener 2 and etc.
July 21, 2016	V0.41	Add in more detailed description for Desktop Start Multiple Mesaurements. (page 35~36).

Reference Documents

[1] “HST Electronics Design and Description” by Forrest Xu, Rev. 0.9. March 13, 2016.

Table of Contents

1. Background and Objectives

Figure 1.1 HST Static Test Machine

As shown in the Figure 1.1 above, the HST static tester (referred as HST machine) is an automatic test machine used in FOLA production line. Within the machine, it consists of three major sub-stations: Input Station, Test Station and Output Station.

Input Station: Its major task is to pick up 10 HGAs from a carrier on input conveyor and load them onto a precisor for probing and testing.

Test Station: It contains test probe and measurement electronics. When the precisor carries 10 HGAs to the test position. The test probe moves down to engage the Pogo pins with test pads on HGAs. Then the electronics starts the measurement. After completion of measurements, the precior moves the 10 HGAs to Output Station.

Output Station: Its major task is to pick up the 10 HGAs from precisor and load them onto the carrier. After that, the carrier is transferred out from the machine through conveyor.

The test results are recorded in the factory information system (FIS). Pass/Fail for each HGAs is written into the RFID attached on each boat for next processes.

The HST machine is meant for production purposes and it can run up to 3000+ UPH. Due to cycle time constraints, the measurement shows only the basic parameters such as resistance, capacitance and short/open. With these parameters, it is good enough to reject those failed HGAs.

From production point of view, more detailed Failure Analysis on those rejected HGAs is required to identify what and where goes wrong; and therefore to take necessary corrective actions to improve the processes. Can this be done with the HST machine? Yes, it can be done but not wise to do it. The HST machine is high-cost and meant for high UPH production. It is not wise to sacrifice the cycle time for those rejected HGAs. Another reason is that Failure Analysis needs some domain expertise and good product knowledge.

Major Objectives of HST Desktop Tester

1. It is low-cost and serves as Failure Analysis Tool for production test engineers to identify the root causes. All detailed measurement parameters such as resistances, capacitances, pad voltages, bias currents, leakage currents and etc, shall be presented to user. With all these data, the test engineers can locate where and what exactly goes wrong in the failed HGAs. Therefore, corrective actions can be defined to improve production process quality.

2. It is used as Research Tool for future new products. Before investing on actual high-cost HST machines for future new products, this tool can be used for feasibility study and experiments to collect necessary data for analysis. It helps to minimize technical risks and improve the time-to-market.

2. Top Level Design Requirements

2.1 Mechanical Requirements

1. Overall Dimensions and Weight

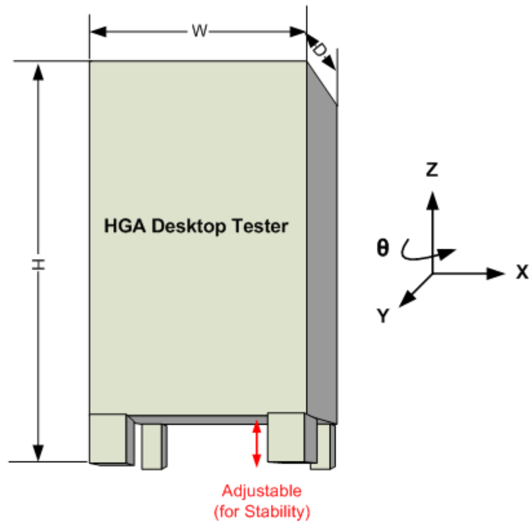

Figure 2.1 Indicative Dimension for HGA Desktop Tester

Dimension Requirements: H (height) < 800mm, W (width) < 600mm, D (depth) < 400mm.

Total Weight (excluding LCR meter): < 30Kg.

The machine is supported with 4 legs, one of them shall be adjustable for machine stability (table top flatness).

2. HGA Capacity and Loading/Unloading

The machine shall be able to test with maximum of 20 HGAs, 10 for Up Tab and 10 for Down Tab. It shall be able to handle tests with fewer HGAs. HGAs are manually loaded onto or unloaded from precisor.

3. Test Probing and Servo Actuations

The machine has to provide X, Y, Z and θ servo actuations for precise test probing with both Up or Down tab HGAs. Positioning accuracy requirements are listed below:

Servo Axis	Accuracy Requirements	Comments
X	<±20µm	It is about 10% of minimum test pad size on HGA flying lead.
Y	<±20µm	
Z	<±30µm	It is about 10% of test probe full stroke (300µm).
θ	±0.023°	It is equivalent to ±20µm position error at R=50mm.

Total accumulated probe positioning error shall be <±60µm when the pad is at its nominal position.

4. HGA Flattener and Vacuum on Precisor

Four independent vacuum channels are required to secure proper position for HGAs on precisor. No vacuum sensor is required. The vacuum can be generated from compressed air using ejector or directly from facility vacuum supply. For the first prototype, it can be designed with facility vacuum. But space has to be reserved for future using vacuum ejectors.

After loading HGAs onto precisor, a Flattener is required to put HGAs into proper position before the vacuum is turned on.

5. Precisor Parking Position for HGA Loading/Unloading

A parking position shall be provided for precisor HGA manual loading/unloading without any obstacles.

6. Probing Contact Force Control

The probing contact force shall be manually adjustable. The maximum contact force per Pogo pin shall not exceed 15g. The total contact force is dependent to the total number of Pogo pins. For example, the total number of Pogo pins for Rosewood and V11 is 100, therefore, the adjustable contact force range shall be within 0~1500g. In any situation during normal operation, no damages are allowed to the test Pogo pins.

7. Facility Outlets

Three outlets are required on machine panels.

Outlet for electrical power input:	220~240VAC, 10A, Single Phase.
Outlet for compressed air input:	Tube size to be defined.
Outlet for facility vacuum input:	Tube size to be defined.

8. Compatibility and Product Conversion

The following sub-modules in HST Static Tester (for production) shall be able to be reused:

- Measurement Electronics PCBAs and Flex Cables
- Pogo Pin Assembly
- Precisor Assembly

The Pogo Pin and Precisor assembles shall be detachable and replaceable due to product conversion. The conversion time shall be < 1 hour.

2.2 Electrical and Pneumatic Requirements

1. A power switch for 100~240VAC/20A input is required. Power-on light indication is recommended.

2. E-Stop is required for emergence machine stop. When activated, it shall cut off the power supplies to all high power servo actuators. However, the +24VDC to measurement electronics PCBA shall remain.

3. An air press regulator is required for incoming compressed air.

4. For safety reason, all machine frames shall be connected to the Earth. No exposure is allowed for any voltage above 48V (both DC and AC).

5. The LCR meter used for HGA capacitance measurement is externally connected to the machine via RS232 and SMA cables.

2.3 Software Requirements

1. Backward compatible with existing HST BENCH TEST TOOL V2.9 or onwards.

2. To provide GUI interface for servo calibration and teaching points.

3. To provide detailed measurement parameters (log-file) for failure analysis and root cause identification.

4. Except for HGA manual loading/unloading, the rest of measurement processes shall be automated.

5. To provide GUI for different product configuration and setup.
6. To provide identification about current machine configurations (product type and measurement parameters).
7. To provide protection against product mismatching, abnormal servo positions and etc.
8. To provide a help file for machine basic operations.

3. System Design and Key Component Selection

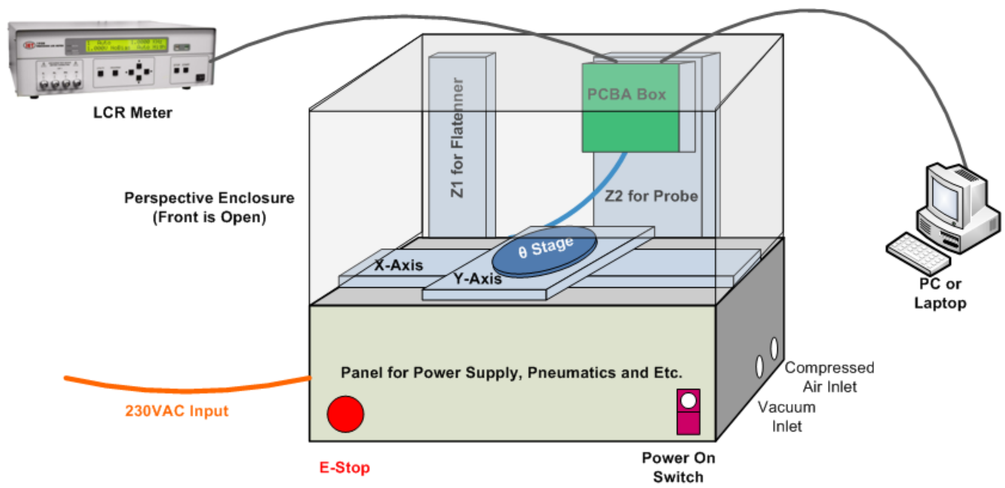

Figure 3.1 Proposed Overview Design for Desktop HST Tester

Figure 3.1 is a proposal for the overview design of Desktop HST Tester. At bottom is the base frame and plate to support the whole tester. Space inside can be used as a panel for power supply, pneumatics regulators and valves, switches, electrical cables and etc.

For ease of access, all facility outlets shall be on the two sides of the tester rather than on the back side.

3.1 Servo Axis and Actuators

There are 5 servo-axis: X-Y-θ for precisor positioning; Z1 for HGA flattener and Z2 for HGA probing. The two cantilevered end-effectors flattener and probing are not shown in the Figure 3.1.

X-Y Actuators for Precisor

These 3 actuators are stacked up together, i.e. X is at bottom, Y is on top of X and θ is on top of Y as shown in the Figure 3.1.

With considerations for positioning accuracy, SKR series LM Guide Actuators are recommended for X and Y axis. The stroke shall be greater than twice of the corresponding dimensions of existing precisor design for Rosewood and V11. The recommended servo motors and actuators for X-Y are listed in Table 3.1 below.

Table 3.1 Recommended Key Components for X and Y

Precisor Axis	Linear Actuator			Servo Motor
	Supplier & Series	Lead(Pitch)	Stroke	
X	THK SKR33-B	6mm	395mm	SM34165D or SM34165DT
Y	THK SKR33-B	6mm	295mm	SM23165DT or SM34165D

θ Actuator for Precisor

The θ-axis is mainly to compensate the angular difference between Up and Down Tab. Its relationship with linear accuracy is illustrated in the Figure 3.2 below.

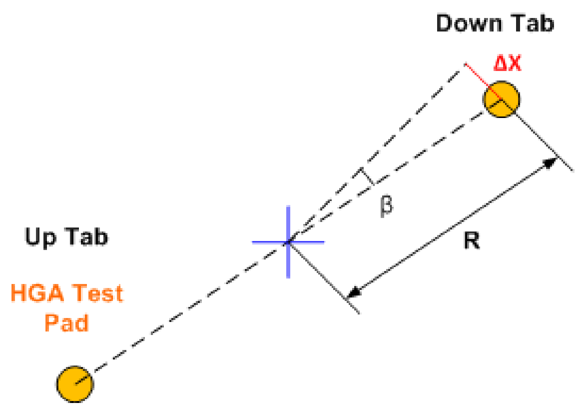

Figure 3.2 Angular Position Error vs. Linear Position Error

Assuming the θ-axis has angular positioning error of β°, the resulted linear positioning error Δx for Pogo pins can be estimated as:

$$\Delta x \approx \frac{2\pi \cdot \beta}{360} \times R$$

Assuming the rotational center is the same as procisor geometry center, based on the Precisor design for Rosewood and V11, the R is approximately 50mm. Then the linear positioning error above can be estimated as:

$\Delta x \approx 872.67 \cdot \beta$ in unit of µm.

To achieve 20µm linear accuracy, the angular positioning error must be < 0.023°. To achieve such a high angular accuracy, one option is to use high precision rotary stage with encoder counts of 65536/revolution. Typically this is very expensive. Another option is to have a special design as illustrated in the Figure 3.3.

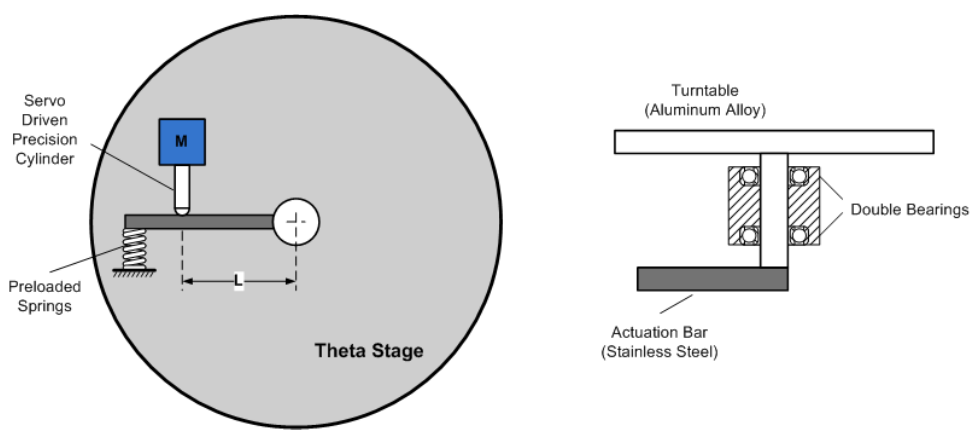

Figure 3.3 Conceptual Design for Low-Cost Theta Stage

As shown in the Figure 3.3, a linear actuator is used to convert a linear motion to angular motion. For expected angular stroke of ±10°, it is good enough. With this design, the angular positioning accuracy can be approximated as:

$\beta \approx \frac{\Delta y}{L} \times \frac{360}{2\pi}$; Δy is the linear actuator accuracy.

Assuming Δy=10µm and L>=50mm, the achievable angular accuracy β < 0.011°.

Design Requirements for θ Stage

1. Double Bearings with press-fit for turntable shaft. The intention is to minimize the wobbling and clearance.
2. To minimize the wear and tear, the actuation bar and the linear actuated stopper shall be stainless steel.
3. The spring preloaded torque T shall be within 0.8 ~ 1.0Nm.
4. L >= 50mm.
5. The lead (pitch) of the linear actuator shall be as small as possible to minimize the preloaded bias torque onto the servo motor. The linear actuator accuracy < 20µm. Minimum stroke > 10mm.

Table 3.2 Recommended Key Components for Theta

Precisor Axis	Linear Actuator			Servo Motor
	Supplier & Series	Lead(Pitch)	Stroke	
θ	THK SKR20-C	1mm	30mm	SM23165D

Z1-Axis for HGA Fattener

When HGAs are manually loaded onto precisor, the precisor first moves to the station underneath Z1 to flatten the HGAs and then turn on the vacuums to secure the HGAs positions. The flattener up/down actuations can be achieved using a pneumatic cylinder.

The stopper position shall be adjustable for future different precisor design and also to prevent hard contact with HGA flex.

An air regulator is required for cylinder speed adjustment.

Z2-Axis for HGA Probing

Z2-axis is meant for test probe actuation. Because it is used for vertical up and down, its servo motor selection is very much dependent to following three factors of mechanical design:

1. The total assembly mass attached on the actuator sliding block.
2. The pitch and shaft diameter if the sliding actuator.
3. Intended acceleration when moving up.

As a general guideline, the continuous torque of SmartMotor shall be 1.5~2.0 times of the total static torque of the Z2 assmbly. Below is recommendation for Z2 actuator and SmartMotor selection. But it is to be confirmed with final mechanical design.

Table 3.3 Recommended Key Components for Z2

Probe Axis	Linear Actuator			Servo Motor
	Supplier & Series	Lead(Pitch)	Stroke	
Z2	THK SKR33-B	2mm	~80~100mm	SM23165DT

In order to minimize the height of center gravity, it is recommended to have the SmartMotor at bottom rather than on the top.

Homing and Limit Sensor Position

For each SmartMotor axis (X-Y-Theta), there are three sensors for Home, Positive Limit and Negative Limit. To simplify the design, the Home and Negative Limit sensors are combined into one, i.e. the Negative Limit sensor is used also for homing.

The homing sensor position is very critical for the design. Typically, the homing position is determined as the first index mark position at the moment when home sensor is triggered. Therefore, as general guideline, the home sensor position shall be about 1~1.5 of linear actuator pitch size from the negative limit.

3.2 Communication with LCR Meter IET 1920

As shown in the Figure 3.1, the HST measurement PCBA provides a set of connectors for control and communication with the LCR meter IET 1920. It includes: RS232 DB9 (Male), 4 SMA shielded cables connectors.

The power supply for the IET 1920 is single-phase 230VAC from test bench.

3.3 Communication with Host PC or Laptop

The Desktop HST Tester is connected to a host PC or laptop via standard RS232 DB9 connector (Female). The Host PC is installed with application software "HST Desktop Tester" and it acts as a master to the tester.

It provides all necessary GUI command interfaces and features in order to operate the machine.

3.4 Servo Controls and Communications

The 3 servo motors are from Animatic SmartMotor Class 5 Series. Each servo motor has its own built-in encoder, servo controller and drive. It provides standard RS485 interface for multi-drop network connection. As shown in the Figure 3.1, these 3 SmartMotors are network connected to the HST Measurement PCBA via RS485. Figure 3.4 below shows the hierarchy for servo controls.

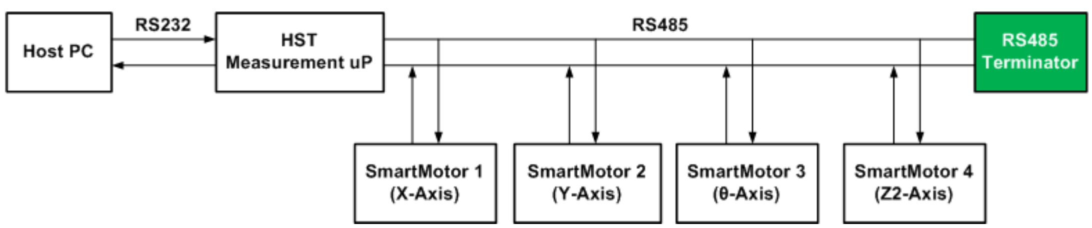

Figure 3.4 Network Connection for Servo Motors

The HST Measurement uP is a slave to the Host PC and it acts also as a master to the 4 SmartMotors. The RS485 shall be terminated with 120Ω resistors at the last node of the network (SmartMotor 3 as shown in the Figure 3.4).

To avoid potential bus contention between HST measurement uP and the 3 SmartMotors, the communication topology over RS485 is typical Master-Slave. HST Measurement uP is the only master and SmartMotors are slave. Only the Master is allowed to initiate a command over the RS485 network to SmartMotors. At the end of command execution, the SmartMotor has to wait for the RS485 to be Idle before acknowledging to the Master. Details are to be discussed later in Software Design.

3.5 List of Key Components and Estimation of BOM Cost

As shown in the list below, the total estimated cost is about USD$24,900. It will be some variation when the final BOM is confirmed.

S/N	Item & Description	Model and Vendor	Unit Price (USD$)	QTY	Extended Cost
	Mechanical				
1	Base frame & enclosure	TBD	1200	1	1200
2	X-Axis Servo & Linear Actuator Assembly	Precision Automation (SM34165D + THK SKR33-B, Stroke 395mm)	3500	1	3800
3	Y-Axis Servo & Linear Actuator Assembly	Precision Automation (SM23165DT + THK SKR33-B, Stroke 295mm)	3000	1	3300
4	Z-Axis Servo & Linear Actuator Assembly	Precision Automation (SM23165DT + THK SKR33-B, Stroke 195mm)	2500	1	3000
5	Theta-Axis Servo & Linear Actuator Assembly	Precision Automation (SM23165D + THK SKR20-C, Stroke 30mm)	2200	1	2800
6	Turntable Assembly	TBD	700	1	700
7	Z2 Cylinder & Acutator Assembly	TBD	700	1	700
8	Vacuum Vales	TBD	250	1	250
9	Air Pressure Regulator	TBD	250	1	250
10	Cylinder Speed Regulator	TBD	250	1	250
11	Precisor Assembly	TBD	2000	1	1800
12	Pogo Pin Probe Assembly	Equaty	3000	1	3000
	Electrical				
13	24VDC Power Supply	PLUS CS10	250	3	750
14	Power Switches, E-Stop Button, Relay & Contactors	TBD	300	1	300
15	Terminal Blocks and Wiring	TBD	800	1	800
	Electronics & Measurements				
16	Measurement PCBA (from Spares)	Glendale	2500	1	0
17	RS485 Transceiver & PCB Terminal Block for SmartMotors (New Design Setup Cost) (from Spares)	Glendale	2500	1	0
18	Miscellaneous & Others	TBD	2000	1	2000
Total					**24900**

4. Electrical Design and Wiring Requirements

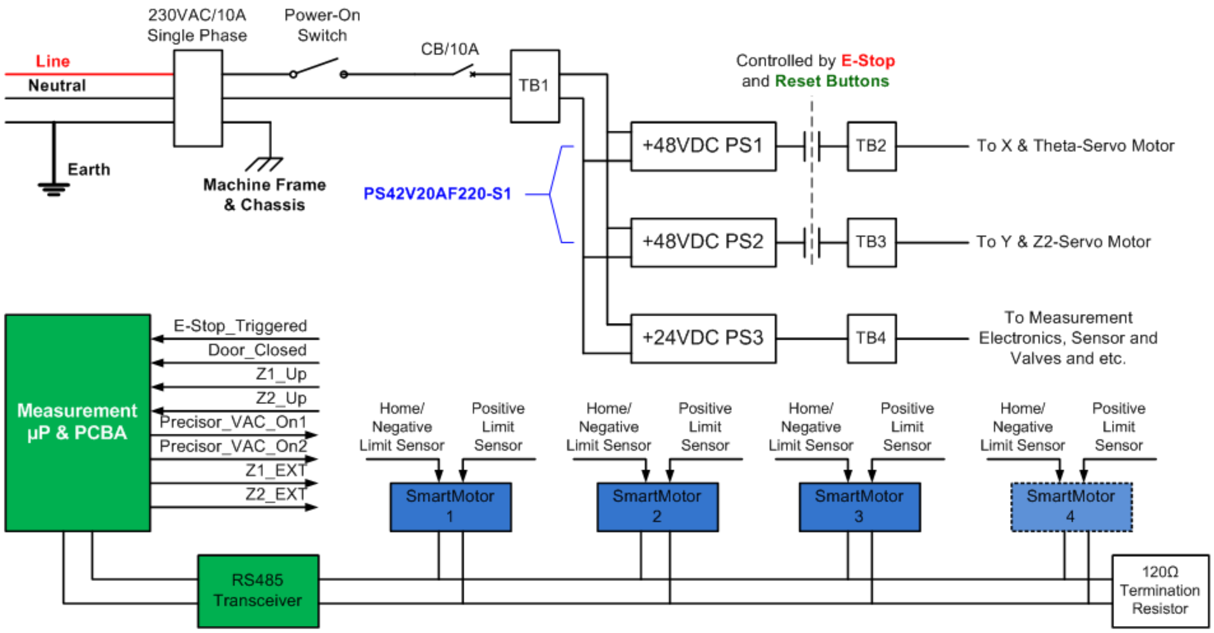

Figure 4.1 Electrical Design, Wiring and IOs

Figure 4.1 gives overview for electrical design, wiring and IO assignments.

Main Power Input

230VAC/10A, Single Phase.
Three wires: Line, Neutral and Earth. The Earth must be connected to the tester frame and chassis.

Power-On Switch

A power-on switch is applied to the main power input. It recommended to have light-on indication when power is switched ON.

For most of users, they are used to directly switch OFF the power after use. Because the four servo drives may be active and consume certain currents (a few amperes), electrical arcing happens at the moment when the power switch is suddenly turned OFF. Therefore, it is recommended to use a power switch with snub-circuit to extend the switch life cycle.

Circuit Breaker

A 10A rating circuit breaker shall be in place for protections against potential short-circuit in the tester.

+48VDC Power Supplies for Servo Motors

Due to space constraints and application needs, there are two separate +48VDC power supplies, each of them is shared between two servo motors. Their current ratings are all 10A.

+48VDC PS1: shared between X and Theta
+48VDC PS2: shared between Y and Z2

Another option is to use PS42V20AF220-S1 to replace the two +48VDC power supplies.

+24VDC Power Supplies for Measurement PCBA, Sensors and Valves

This is +24VDC power supply for Measurement PCBA, Sensors and Valves. The current rating is 5A.

SmartMotors and Their Limit Sensors

There are 4 SmartMotors in total, i.e.

SmartMotor 1 for X axis
SmartMotor 2 for Y axis
SmartMotor 3 for Theta axis
SmartMotor 4 for Z2 probing (to replace the Z2 cylinder, to be confirmed with mechanical design)

Each motor has two sensors mounted on the linear actuator, i.e.

Negative Limit Sensor: it is also used as Home sensor, connected to the negative limit input of the motor.
Positive Limit Sensor: it is connected to the positive limit input of the motor.

RS485 Transceiver and Network Connections

As shown in the Figure 4.1, all four SmartMotors are connected to the Measurement uP PCBA through RS485 transceiver. We need to make a conversion PCBA which converts UART TTL signals (SF_TXD and SF_RXD on HST Measurement PCBAs) to standard RS232 signals. This PCBA will be manually soldered in the Lab. Figure 4.2 below shows the schematics design and signal connections.

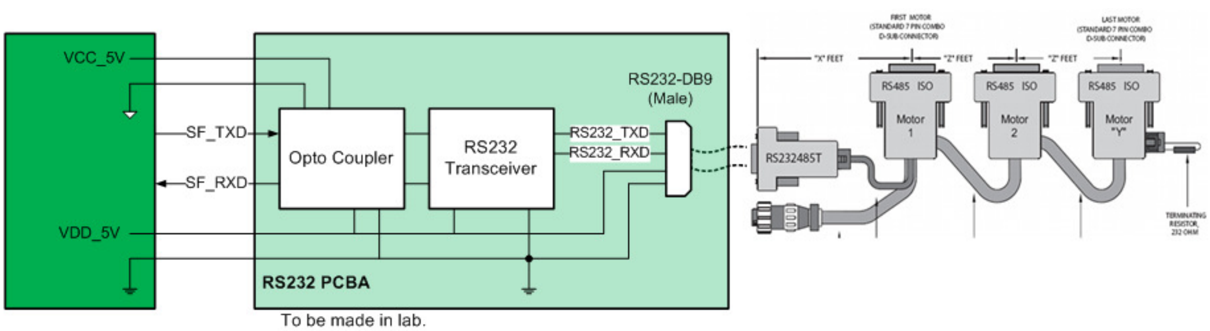

Figure 4.2 Conversion PCB Schematcis

The connection cable for the 4 SmartMotors is shown as in the Figure 4.3 below. The cable length “X”, “Y”, “Z”, … , can be customized and ordered from Animatics (vendor for SmartMotor). This is to be finalized when the mechanical top-level design is ready.

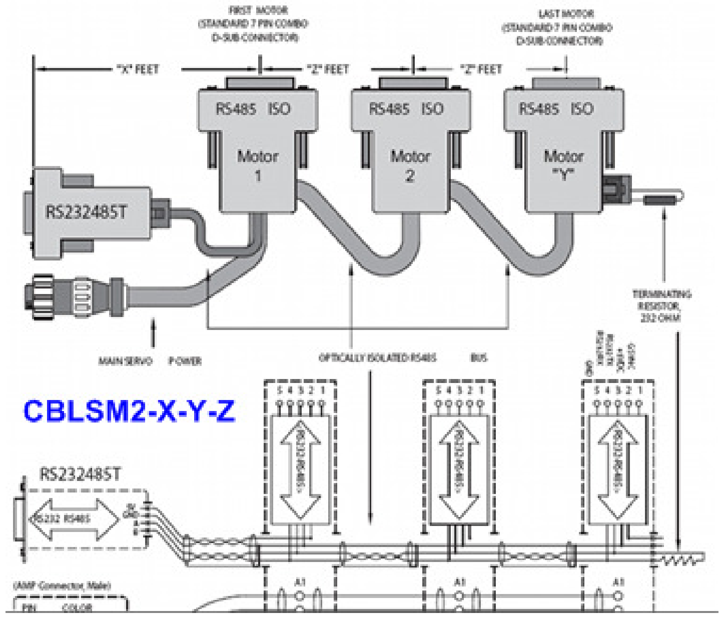

Figure 4.3 RS485 Cable Connections with Multi SmartMotors

5. Recommendations for Precisor Design

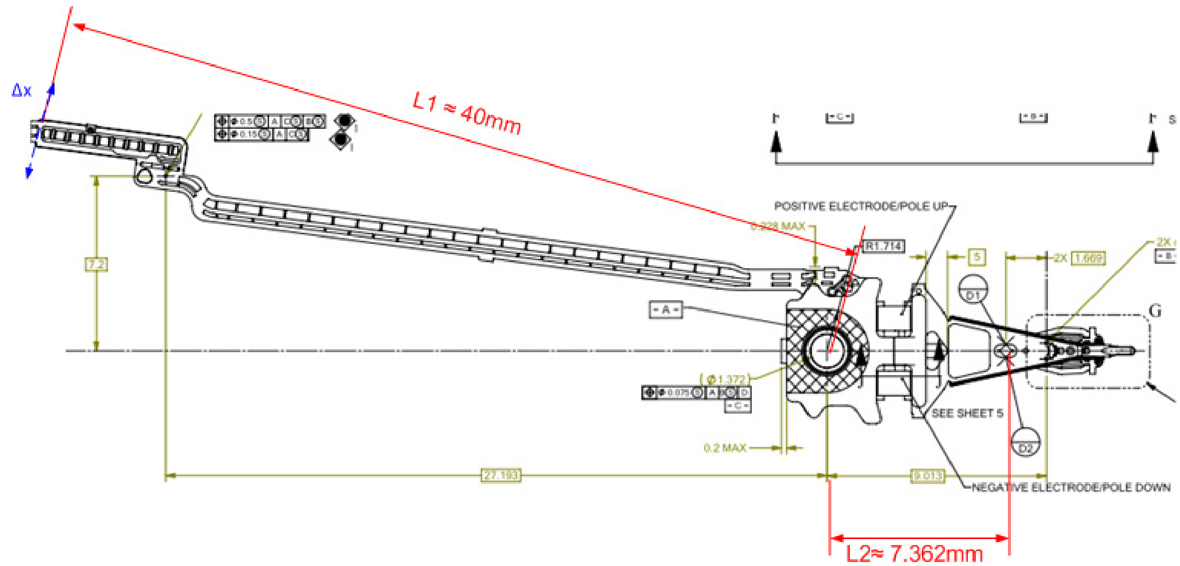

Figure 5.1 Outline Dimensions of HGA GrenadaBP2

There are three types of HGAs in factory production, i.e. Short Tail, Long Tail and Extra Long Tail. As shown in the Figure 5.1 above is the HGA GrenadaBP2 (Long Tail). Assuming the clearance of the two dowel pin holes is δ. The last pad (typically is the GND pad) position variation Δx from its nominal position can be estimated as:

$$\Delta x \approx 2 \times \frac{L1}{L2} \times \delta$$

For a typical value of δ = 30µm, the last pad position variation can be Δx ≈ 323µm. As a consequence of such a large position variation, the Pogo Pin cannot be probed with the test pad reliably. This has been observed during the buy-off tests for Iris V11 HGA.

To improve the probing reliability, we have to minimize the pad position variation to be within ±25µm. This requires some design improvements for the existing precisor used in HST machine.

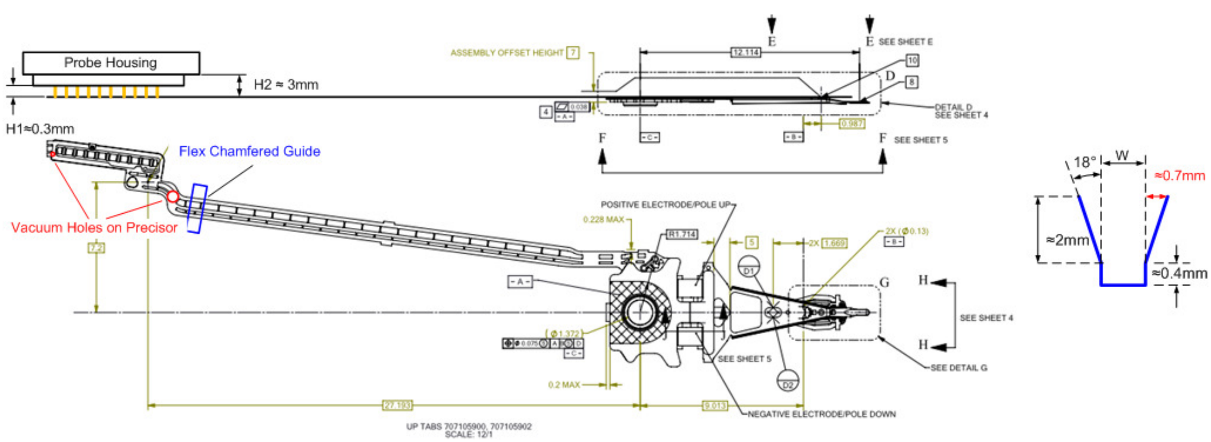

Figure 5.2 Proposed Chamfered Guide for HGA Flex

As shown in the Figure 5.2, along the HGA flex, a V-shape chamfered guide (in **Blue**) can be placed near to the flying lead position. The total height shall be about 2.4mm which is less than the clearance height H2

(≈3mm) of probe housing. The chamfer angle is recommended to be <= 18°. In this way, the HGA flex can be still guided to précised position even if it is out of position by ±0.7mm.

At bottom of the V-shape chamfer, it is pocket nest for flying lead to sit in. The width **W** shall be slightly bigger than the flex width by 20~30µm.

Near the V-shape chamfer, a vacuum hole is required in the precisor to secure the flex inside the chamfer. If feasible, it is also nice to have a vacuum hole on the top of flying lead as shown in **Red**.

The flying leads shall sit on a non-conductive surface. The original Semitron design can remain the same.

6. Host API Commands & GUI Requirements

As shown in Figure 3.4, the Host PC controls the test operation via RS232 communication with HST Measurement µP. To be backward compatible with HST Static Tester, the communication protocol remains the same as what defined in Section 4.3 of Reference Document [1].

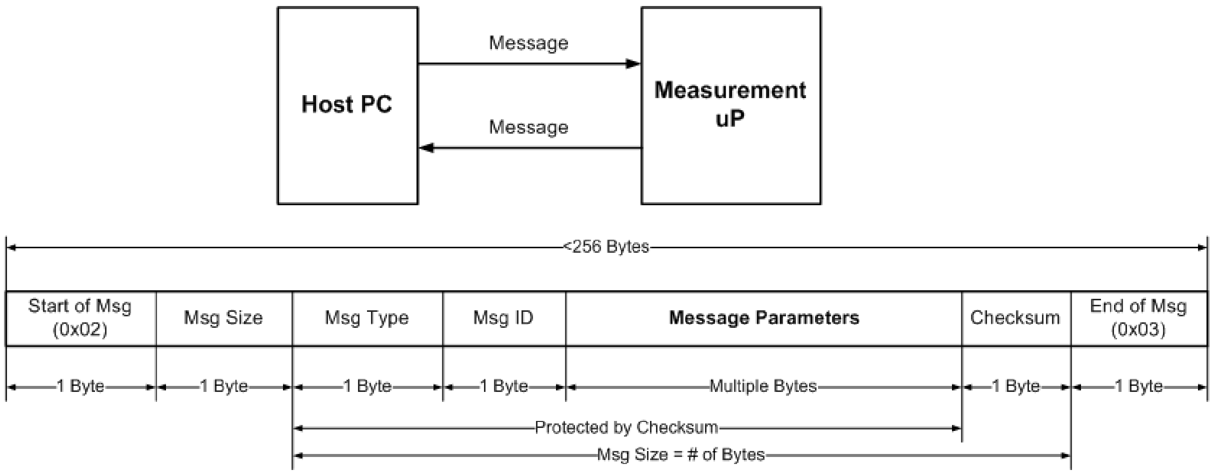

Figure 6.1 Communication Protocol between Host PC and Measurement uP

Table 6.1 Definitions of Communication Message Protocol

Name	Size (byte)	Definition
Start of Message	1	Hex value of **0x02**. It indicates the Start of Message and can be used for message framing in software. Every new message always starts with 0x02. But it does not mean all bytes with content of 0x02 are the Start of Message.
Message Size	1	It defines the total number of bytes including the Message Type, Message ID, Message Parameters and Checksum.
Message Type	1	This byte defines the current message type. 1: Command Message, indicating that the Server has to execute a command upon receiving it from Client. An acknowledge message is required sending back to the Client upon receiving this command message. Typically Host sends the command message. 2: Acknowledge Message, indicating that this is an acknowledgement message for the received command message. Typically the uP sends this acknowledge message. 3: Unsolicited Message, a message from Client to Server which does not require acknowledge message to be sent back. Typically this message type is used for measurement uP to report its measurement results to the Host PC after completion of tests. Other: Reserved for future use.
Message ID	1	This is message ID indicating what command/task is required for the Server to complete. Its meaning is also dependent to the Message Type. If Message Type =1 or 3, All commands are as defined in the Section 4.4. If Message Type =2, It is the copy of the previous received Message/Command ID. Others: Reserved.
Message Parameters	<250	It contains all necessary command parameters related to the Message ID. Refer to each command details.
Checksum	1	It is the checksum for Message Type, Message ID and Message Parameters. It is calculated as: Message Type + Message ID + Message Size.
End of Message	1	Hex value of **0x03**. All messages end with 0x03. But it does NOT mean that all bytes with 0x03 are the end of message.

6.1 API Commands and Descriptions

The set of API commands defined in Ref. Document [1] are all re-used in this HST Desktop Tester. They are mainly used for electronics measurements or calibrations. Besides, a set of servo control and calibration commands are to be added. These commands are unique to this Desktop Tester and their command IDs start with value of 80 or above.

6.1.1 HST_dt_connect

For ease of software sustaining, it is intended to use one GUI software and µP firmware for both lab and factory applications. It means that the existing "HST Bench Tool" will be replaced with "HST Desktop Tester" software. To avoid potential conflicts between two different hardware configurations, this command is meant to identify the hardware attached with the "HST Desktop Tester".

When this command is initiated, the Host PC sends an enquiring password 0xA5A5 to the HST uP. Then the measurement uP sends servo ID request commands to all the 4 SmartMotors. After receiving correct acknowledgements from the 4 servo controllers, the uP then replies the Host PC with 0x5A5A. Once this identification process is successfully completed, the connection between "HST Desktop Tester" software and the test machine is established. Otherwise, all commands which are unique to the Desktop Tester are rejected by the measurement uP.

Prior to waiting for the response from the measurement uP, the Host PC shall set up a Timeout value of 2s. If the identification process fails within 2s, no connection is established.

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 80

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Password (LSB)	Password for identification, 0xA5A5. Unsigned 16-bit.
2	Password (MSB)	

Acknowledge Message from Measurement uP

Message Size = 7 (if READY) or 5 (if ERROR)
Message Type = 2
Message ID = 80

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.
3	Password (LSB)	Password for identification, 0x5A5A. Unsigned 16-bit.
4	Password (MSB)	

6.1.2 HST_dt_homing_axis

This command is for the Desktop Tester to perform homing for a specified servo axis or all axes. To perform homing for all servo axes, the measurement uP will issue an individual servo homing command to each axis in a specific sequence to prevent potential mechanical crash.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 81

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Servo Axis	Unsigned Byte, 8-bit. 1: X axis; 2: Y axis; 3: Theta axis; 4: Z2 axis (Test Probe). 5: Precisor (combined X, Y and Theta). 255: All axis. Typically used for whole machine homing. Others: Reserved.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 81

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.3 HST_dt_servo_off

This command is to deactivate servo PID control loop for all axes. It is typically used before turn OFF the machine to avoid electrical arcing at the main power switch.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 82
Command Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 82

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.4 HST_dt_servo_abs_move_axis

This command is to move a servo axis to a specific absolute target position.

Command Message from Host

Message Size = 20
Message Type = 1
Message ID = 83

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Servo Axis	Unsigned Byte, 8-bit. 1: X-axis only; 2: Y-axis only; 3: θ-axis only; 4: Z2-axis (test probe) only. 5: X-Y-θ Combined. Others: reserved.
2	X Abs. Pos (LSB Low)	X-axis Absolute Target Position in unit of µm. Signed 32-bit Value.
3	X Abs. Pos (LSB Mid)	
4	X Abs. Pos (MSB Low)	
5	X Abs. Pos (MSB High)	
6	Y Abs. Pos (LSB Low)	Y-axis Absolute Target Position in unit of µm. Signed 32-bit Value.
7	Y Abs. Pos (LSB Mid)	
8	Y Abs. Pos (MSB Low)	
9	Y Abs. Pos (MSB High)	
10	θ Abs. Pos (LSB Low)	θ-axis Absolute Target Position in unit of µm. Signed 32-bit Value.
11	θ Abs. Pos (LSB Mid)	
12	θ Abs. Pos (MSB Low)	
13	θ Abs. Pos (MSB High)	
14	Z2 Abs. Pos (LSB Low)	Z2-axis Absolute Target Position in unit of µm. Signed 32-bit Value.
15	Z2 Abs. Pos (LSB Mid)	
16	Z2 Abs. Pos (MSB Low)	
17	Z2 Abs. Pos (MSB High)	

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 83

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.5 HST_dt_servo_rel_move_axis

This command is to move a servo axis to relative target position with respect to its current position.

Command Message from Host

Message Size = 8
Message Type = 1
Message ID = 84

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Servo Axis	Unsigned Byte, 8-bit. 1: X axis; 2: Y axis; 3: Theta axis; 4: Z2 axis (test probe). Others: reserved.
2	Rel. Pos (LSB Low)	Relative Target Position in unit of µm. Signed 32-bit Value.
3	Rel. Pos (LSB Mid)	
4	Rel. Pos (MSB Low)	
5	Rel. Pos (MSB High)	

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 84

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.6 HST_dt_get_servo_positions

This command is to read out current servo absolute positions in unit of encoder counts.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 85
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = 21
Message Type = 2
Message ID = 85

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.
3	X Pos (LSB Low)	X-axis Absolute Position in unit of µm. Signed 32-bit Value.
4	X Pos (LSB Mid)	

5	X Pos (MSB Low)	
6	X Pos (MSB High)	
7	Y Pos (LSB Low)	Y-axis Absolute Position in unit of µm. Signed 32-bit Value.
8	Y Pos (LSB Mid)	
9	Y Pos (MSB Low)	
10	Y Pos (MSB High)	
11	θ Pos (LSB Low)	θ-axis Absolute Position in unit of µm. Signed 32-bit Value.
12	θ Pos (LSB Mid)	
13	θ Pos (MSB Low)	
14	θ Pos (MSB High)	
15	Z2 Pos (LSB Low)	Z2-axis Absolute Position in unit of µm. Signed 32-bit Value.
16	Z2 Pos (LSB Mid)	
17	Z2 Pos (MSB Low)	
18	Z2 Pos (MSB High)	

6.1.7 HST_dt_teach_servo_position

This command is to inform the measurement uP to record current servo position as a specific taught position for normal operations. Table below lists the positions to be taught manually by user. Unless change of mechanical parts or reassembling, these positions need to be taught once only during commission.

		Standby Position	Load/Unloading Station	Flattener Station		Test Station	
				Up Tab	Down Tab	Up Tab	Down Tab
Precisor	X Y θ	√ (1)	√ (3)	√ (4)	√ (5)	√ (6)	√ (7)
Test Probe	Z2	√ (2)				√ (8)	√ (9)

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 86

Definition of Command Message Parameters

Byte Index	**Byte Definition**	**Description**
1	Taught Position Index #	Unsigned Byte, 8-bit. Referring to the Table above: 1: Precisor Standby Position; 2: Test Probe Standby Position; 3: Precisor Loading/Unloading Station; 4: Precisor at Flattener Station, Up Tab; 5: Precisor at Flattener Station, Down Tab; 6: Precisor Test Station, Up Tab; 7: Precisor Test Station, Down Tab; 8: Test Probe at Test Station, Up Tab; 9: Test Probe at Test Station, Down Tab; Others: reserved.

Acknowledge Message from Measurement uP

Message Size = 17 for Precisor; or 9 for Test Probe.
Message Type = 2
Message ID = 86

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.
3	X or Z2 Pos (LSB Low)	**X or Z2-axis** Absolute Position in unit of µm. Signed 32-bit Value.
4	X or Z2 Pos (LSB Mid)	
5	X or Z2 Pos (MSB Low)	
6	X or Z2 Pos (MSB High)	
7	Y Pos (LSB Low)	Y-axis Absolute Position in unit of µm. Signed 32-bit Value.
8	Y Pos (LSB Mid)	
9	Y Pos (MSB Low)	
10	Y Pos (MSB High)	
11	θ Pos (LSB Low)	θ-axis Absolute Position in unit of µm. Signed 32-bit Value.
12	θ Pos (LSB Mid)	
13	θ Pos (MSB Low)	
14	θ Pos (MSB High)	

6.1.8 HST_dt_servo_move_to_taught_position

After servo position teaching, this command is used to verify the taught positions by user.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 87

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Taught Position Index #	Unsigned Byte, 8-bit. Referring to the Table above: 1: Precisor Standby Position; 2: Test Probe Standby Position; 3: Precisor Loading/Unloading Station; 4: Precisor at Flattener Station, Up Tab; 5: Precisor at Flattener Station, Down Tab; 6: Precisor Test Station, Up Tab; 7: Precisor Test Station, Down Tab; 8: Test Probe at Test Station, Up Tab; 9: Test Probe at Test Station, Down Tab; 10: Precisor, test probe and Flattener to their standby position; Others: reserved.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 87

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.9 HST_dt_save_taught_position

After teaching and verification, this command is to save the servo taught positions into uP's EEPROM for normal operations. Each time after power-on, these taught positions are automatically uploaded from EEPROM into uP's RAM.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 88
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 88

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.10 HST_dt_set_output

This command is to set individual output IO to be ON or OFF. These IOs are typically used for vacuum or cylinder valve's control.

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 89

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Output Index #	Unsigned Byte, 8-bit. Referring to the Table above: 1: Precisor Up Tab Vacuum On/Off; 2: Precisor Down Tab Vacuum On/Off; 3: Flattener 1 Extend/Retract; 4: Flattener 2 Extend/Retract; Others: Reserved.
2	ON/OFF	0: Output OFF or Flattener Retract; 1: Output ON or Flattener Extend. Others: Reserved.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 89

Definition of Message Parameters

Byte Index	Byte Definition	Description

1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.11 HST_dt_get_all_inputs

This command is to read out all sensor input status.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 90
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = 13
Message Type = 2
Message ID = 90

Definition of Message Parameters

Byte Index	**Byte Definition**	**Description**
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.
3	E-Stop Input Status	E-Stop Trigger Input Status.
4	Flattener Retract Sensor Input 1	Sensor Input Status.
5	Flattener Extend Sensor Input 1	Sensor Input Status.
6	Flattener Retract Sensor Input 2	Sensor Input Status.
7	Flattener Extend Sensor Input 2	Sensor Input Status.
8	Reserved	
9	Reserved	
10	Reserved	

6.1.12 HST_dt_load_hga

This is a combo command which consists of several servo actions. After loading HGAs onto precisor, the Desktop Tester will perform following actions in sequence once this command is received:

1. Retract the Test Probe and Flattener.
2. Move the precisor from Loading/Unloading Station to Flattener Station.
3. Extend the flattener end-effect.
4. Turn on the two precisor vacuums in sequence with time interval of 0.1s.
5. Retract the flattener.
6. Move the precisor to Test Station.
7. Extend the Test Probe to be engaged with HGA test pads.

If any error happens during Step 1~7, the command execution is terminated with error message acknowledgement to the Host PC.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 91

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Up/Down Tab Index #	Unsigned Byte, 8-bit. 1 – Up Tab; 2 – Down Tab. Others: Reserved.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 91

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.13 HST_dt_eject_hga

This is a combo command and its ser. After loading HGAs onto action sequence is just opposite to the HST_dt_load_hga command. After completion of measurements, the Desktop Tester will perform following actions in sequence once this command is received:

1. Retract the Test Probe and Flattener (if necessary).
2. Move the precisor from Test Station to Loading/Unloading Station.
3. Turn OFF the two precisor vacuums in sequence with time interval of 0.1s.

If any error happens during Step 1~3, the command execution is terminated with error message acknowledgement to the Host PC.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 92
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 92

Definition of Message Parameters

Byte Index	Byte Definition	Description

1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.14 HST_dt_disconnect

Opposite to the "HST_dt_connect", this command is to break the connection estabilished between the Host PC and Desktop Tester. Once disconnected, all servo commands are considered as illegal commands and rejected by the Desktop Tester.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 93
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 93

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.15 HST_dt_jog

This command is used during machine commissioning. It is intended for user to manually jog one of the servo axes toward specified direction at predefined slow speed.

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 94

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Servo Axis Index #	Unsigned Byte, 8-bit. Referring to below: 1: Precisor X-axis; 2: Precisor Y-axis; 3: Precisor θ-axis; 4: Test Probe Z2-axis; Others: Reserved.
2	Jog Direction	1: Left for X; or Inside for Y; or CW for θ; or Up for Z2. 2: Right for X; or Outside for Y; or CCW for θ; or Down for Z2. Others: Reserved

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 94

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.1.16 HST_dt_jog_stop

This command is to immediately stop servo jogging in progress.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 95

Definition of Command Message Parameters

Byte Index	Byte Definition	Description
1	Servo Axis Index #	Unsigned Byte, 8-bit. Referring below: 1: Precisor X-axis; 2: Precisor Y-axis; 3: Precisor θ-axis; 4: Test Probe Z2-axis; Others: Reserved.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 95

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

6.2 Host PC GUI Requirements

The purpose of GUI is to provide user-friendly graphic interfaces for machine commissioning and operating. Therefore, there are two GUI Tabs, one for commission and the other one for operation. These two can be inserted into the existing "HST BENCH TEST TOOL". The GUI for machine commissioning shall be password protected to prevent from unintentional changes to machine calibration data.

No command queuing is supported. Any new command shall be ignored if previous command is in progress.

6.2.1 GUI Design and Requirements for Machine Commissioning

Figure 6.2 below is a proposed GUI design for machine commissioning.

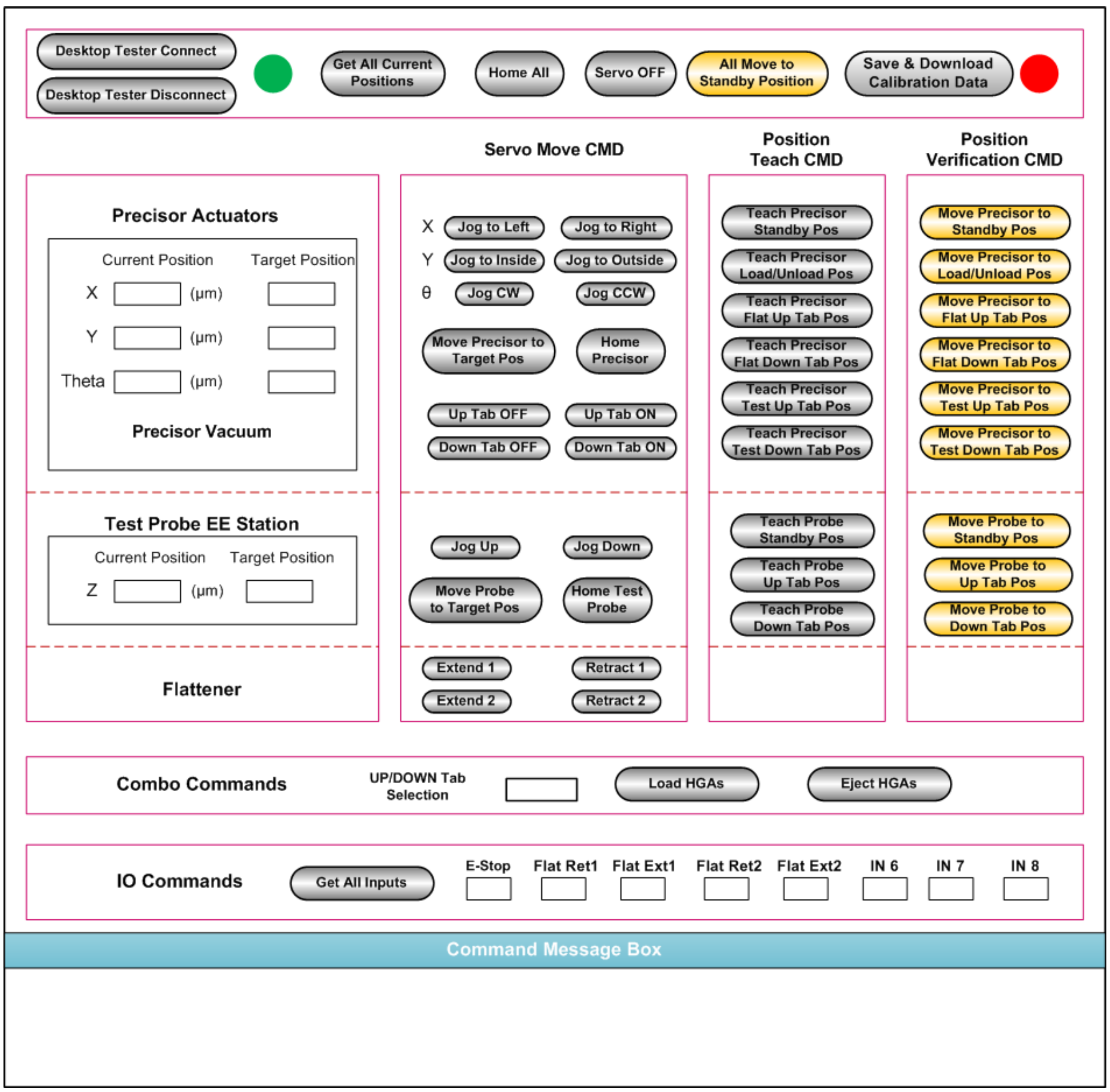

Figure 6.2 Proposed GUI Design for Machine Commissioning

Desktop Tester Connect/Disconnect

This command button is to enable or disable logic connection with the Desktop Tester. Associated API commands are "HST_dt_connect" and "HST_dt_disconnect". When the connection is successful, the green LED next to the command button shall turn ON. Otherwise, the LED turns OFF.

After launching this GUI application, the connection shall be OFF by default.

If the connection is OFF, all command buttons on this tab shall be deactivated. No command can be sent to the Desktop Tester except for the "Desktop Tester Connect" command.

Get All Current Positions

This command button is to read back current servo positions from all 4-axis. The readback values are directly updated into message boxes for X, Y, Theta and Z accordingly. Message Boxes for Current Position are read-only. User is not allowed to change any displayed contents.

Associated API command is "HST_dt_get_servo_positions".

Home All, Home Precisor, Home Test Probe

These three homing command buttons are associated with the same API command "HST_dt_homing_axis" but different command parameters. Refer to Section 6.1.2 for details.

This command may take up to 20s.

Servo OFF

This is a special command button for user before switching off the machine. To minimize electrical arcing at power-off switch, user is recommended to turn OFF the 4 servos and reduce the current consumption before turning off the power switches.

The associated API command is "HST_dt_servo_off".

All Move to Standby Position

This command is to move Precisor, Test Probe and Cylinders to their standby position. To avoid potential mechanical collision, the cylinders and Test Probe move to their standby positions first, then the precisor moves later. This sequence is automatically controlled by the measurement uP.

Associated API command is "HST_dt_servo_move_to_taught_position" with command parameter:

Taught Position Index = 10

This command may take up to 20s.

Save & Download Calibration Data

After verification on calibrated servo positions by user, this command is to inform the measurement uP to save all data into its EEPROM for use in future.

Message Boxes for Precisor and Test Probe Actuators

There are two sets of message boxes, one set for Current Position which is read-only and the other set is Target Position which is meant for user to key in desired target position during calibration. These two boxes are 32-bit intgers with decimal format. Units are all in µm.

Servo Jog Commands: X - Jog to Left/Jog to Right; Y- Jog to Inside/Jog to Outside; θ – Jog CW/Jog CCW
Z – Jog Up/Jog Down

There are two Jog Commands for each servo axis. These command buttons are associated with the same API commands: "HST_dt_jog" and "HST_dt_jog_stop", but with different command parameters. Refer to Section 6.1.15 and 6.1.16 for details.

If the command button is pressed and held, the corresponding axis continues to jog in a specified direction. As soon as the command button is released, jogging shall immediately stop.

These command buttons must be combination of both **Static State and Event-driven**, i.e.

1. At the moment of JOG command button is pressed, Host PC sends API command "HST_dt_jog" to the uP, then wait for acknowledgement.
2. As soon as the command button is released, the Host PC sends API command "HST_dt_jog_stop" to uP.
3. Because of communication latency, it is possible that the event of releasing JOG command happens before the uP acknowledgement to the "HST_dt_jog" command at Step 1. If this event is not captured by the Host PC, the Jog will continue going. This is a dangerous situation for machine safety. This must be considered in the Host PC software design.

Home Precisor to Target Position and **Move Probe to Target Position**

This command is to move the Precisor or Test Probe to a target position which is specified in the Target Position message boxes.

Associated command is "HST_dt_servo_abs_move_axis". The servo axis selection shall be 4 (Z2 only) or 5 (X-Y-θ combined).

Precisor Vacuum ON/OFF

This command is to turn on/off the two vacuums on the precisor, one for Up Tab and one for Down Tab. Associated API command is "HST_dt_set_output" with command parameter Output Index = 1 or 2.

Flattener Extend/Retract

In the mechanical design, there are two Flatteners, i.e. Flattener 1 and Flattener 2. Similar to Precisor Vacuum ON/OFF command, this is to extend or retract the Flattener end-effecter. Associated API command is "HST_dt_set_output" with parameter Output Index = 3 (for Flattener 1) or 4 (for Flattener 2) and On_Off = 0 for Retract or 1 for Extend.

Position Teach Commands

There are 9 Servo Position Teach Command buttons in total. These commands are associated with API command "HST_dt_teach_servo_position" but different command parameters. The returned taught position values shall be displayed in the Current Position Text Box accordingly for user's verification. When teach precisor position, X-Y-θ are updated. When teach test probe, only θ is updated.

Position Verification Commands

This set of command buttons are used for user verification of taught servo positions. They are associated with API command “HST_dt_servo_move_to_taught_position” but different command parameters.

Load HGAs and **Eject HGAs**

These two commands are combo-commands which consist of several servo/cylinder/values actuations. They are associated with corresponding API commands “HST_dt_load_hga” and “HST_dt_eject_hga”.

Get All Inputs

This is a IO to read out all input states. There are 8 inputs in total. The returned values for each input shall be updated in their corresponding message box as shown in the Figure 6.2.

Command Message Box

This message box is for user or developer to trace the communication/message flow between Host PC and measurement uP.

6.2.2 GUI Design and Requirements for Desktop Tester Operation

In general, the GUI design for Desktop Tester Operation shall be very similar to the existing “Bench Test Tool” design except for the commands highlighted in Green below.

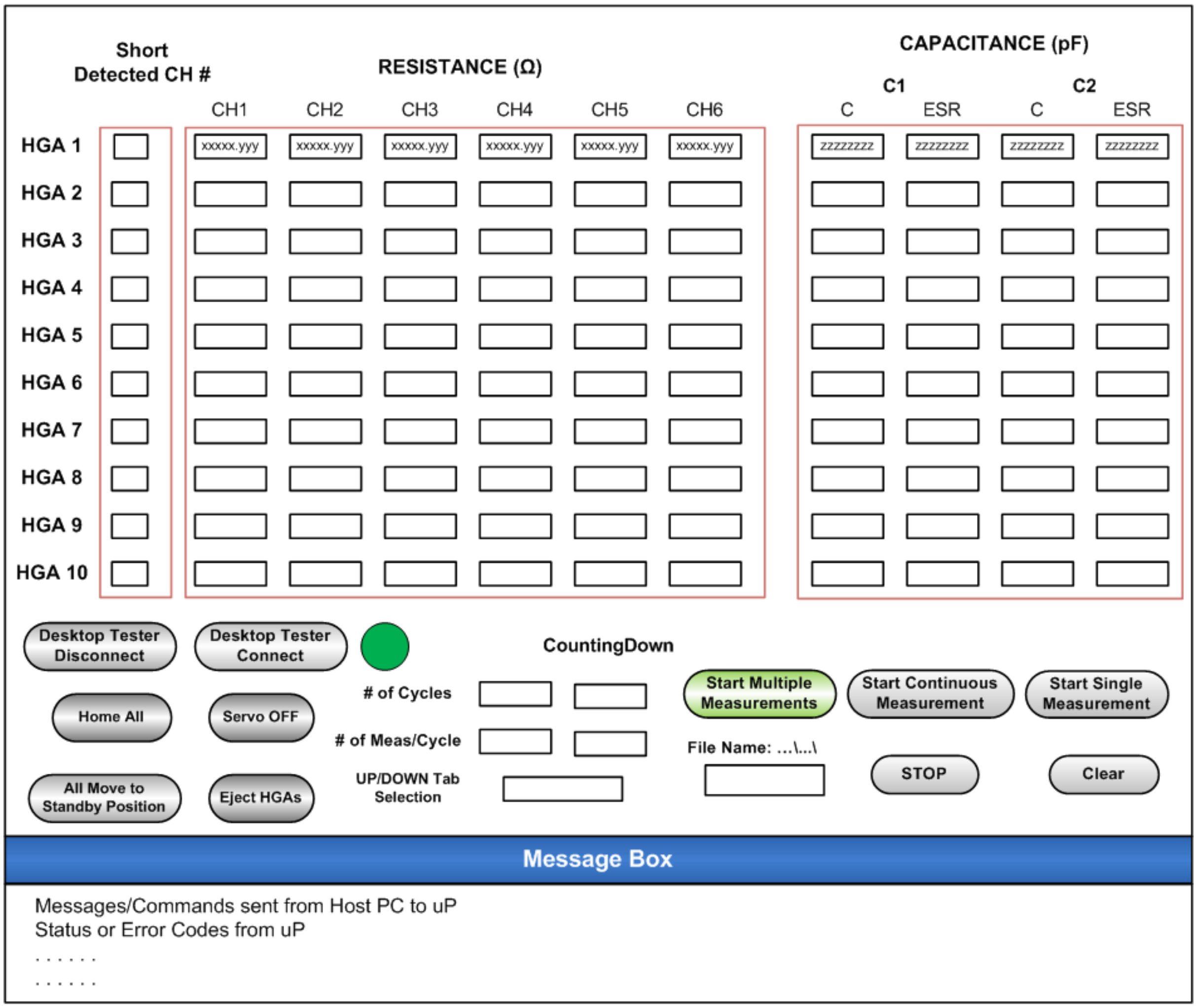

Figure 6.3 GUI Design for Desktop Normal Operations

Desktop Tester Connect/Disconnect

This command is to setup or disable a logic connection between Host PC and Desktop Tester. All command buttons are activated only when this connection is built up. There is Green LED associated with this connection. If the connection is active, the LED turns ON; otherwise, it turns OFF.

Associated API commands are “HST_dt_connect” and “HST_dt_disconnect”.

Home All, Servo OFF, All Move to Standby Position, Eject HGAs

Referring to the description of Machine Calibration and Commissioning Tab page, these four command buttons are also duplicated in this Tab Page, because they are often used during normal operation. The definitions and requirements are the same as what defined in previous section of this document.

Start Multiple Measurements

There are two parameters/message boxes associated with this command button, i.e. # of Cycles and # of Measurements/Cycle. Figure 6.4 below illustrates the definitions for Cycle and Measurements/Cycle.

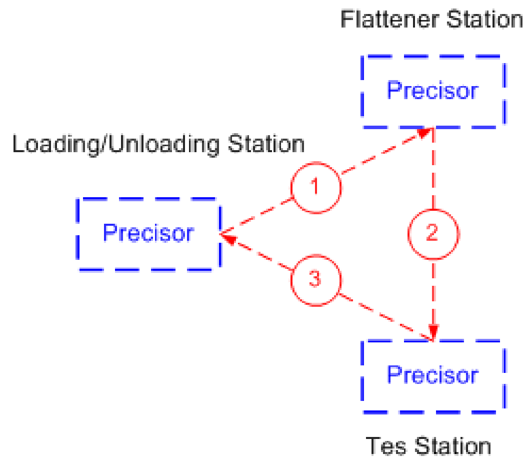

Figure 6.4 Definition of Measurement Cycle

Definition of Measurement Cycle and Measurements/Cycle

A complete measurement cycle consists of three major steps.

Step 1: The precisor loaded with HGAs moves from Loading/Unloading Station to Flattener Station for securing the HGAs positions on precisor.

Step 2: After Step 1, the precisor continues moving to the Test Station for resistance/capacitance measurements. If the parameter, # of Meas/Cycle, is greater than 1, multiple measurements are repeated and results are updated in corresponding message boxes at end of each measurement.

Step 3: After completion of measurements, the precisor moves to the Loading/Unloading Station.

If multiple cycles are chosen by user, the above 3 steps are repeated with time interval of 0.5s between two consecutive cycles. There are two Counting-Down message boxes to count down the number of cycles and measurements/cycle. At the end of each cycle, the Counting-Down message box shall be reloaded with the value from “# of Meas/Cycle” message box.

In the Logfile, measurement results shall be segregated with cycle index and measurement index.

What is the Objective for Multi Cycle Tests? It is meant for Probing Reliability evaluation on precisor design.

If user selects # of Cycles = 0 and # of Meas/Cycle != 0, in this situation, it starts measurements straight away without precisor movements, i.e. without calling API commands “HST_dt_load_hga” and “HST_dt_eject_hga”.

Associated API commands are “HST_dt_load_hga”, “HST_start_meas”, “HST_get_short_detection”, “HST_get_res_results”, “HST_get_cap_results”, “HST_get_cap_secondary_results”, “HST_dt_eject_hga”.

Below is HST Command Sequence flow for an example of: # of Cycle = 2 and # Meas/Cycle = 5.

(Copy the # of Cycles into Counting Down Text Box)
(Copy the # of Meas/Cycle into Counting Down Text Box)
(if # of Cycles = 0 and # of Meas/Cycle = 0, do nothing)

(if # of Cycles = 0 but # of Meas/Cycle != 0, just do measurements as Bench Test Tool did till counting becomes 0.)
(if # of Cycles != 0 but # of Meas/Cycle != 0, follow the sequence below)

1. Send “HST_dt_load_HGAs”
2. Send “HST_start_meas”
3. Send “HST_get_short_detection”, “HST_get_res_results”, “HST_get_cap_results”, “HST_get_cap_secondary_results”, and then log the results.
4. Decrementing Counting Down for the # of Meas/Cylce by 1.
5. Check if # of Meas/Cylce = 0. If NOT,Repeat the Step 3 and 4 till Counting Down for the # of Meas/Cylce to be 0. If 0, Send “HST_dt_eject_HGAs” command. After successful acknowledgment, Go to next Step 6.
6. Decrementing the Counting for # of Cycles by 1.
7. Check if # of Cycles = 0.
   If NOT, copy the # of Meas/Cylce to its Counting Down Test Box. Repeat the Step 1~7.
   If 0, go to next Step 8.
8. Close the Logfile. End.

Other Command Buttons: **Start Continuous Measurement, Start Single Measurement, Stop, Clear**

They are remained as the same as what defined in existing “HST BENCH TEST TOOL”.

7. Communication Protocol and API Commands with SmartMotor

7.1 Node ID and Communication Protocol

Node ID Assignments

As shown in the Figure 4.1, the uP communicates with the 4 SmartMotor through RS485 multi-drop network. Messages over the RS485 are visible to all 4 SmartMoror. In order to have a specific peer-to-peer communication, each SmartMotor needs to have a unique Node ID. Table 7.1 below shows the Node ID for each device on the RS485 network.

Table 7.1 Node ID Assignments

Device	uP	X-axis	Y-axis	θ-axis	Test Probe
Node ID	0	1	2	3	4

The communication topology is Master-Slave, i.e. the uP is the Master and others are Slaves. Only the Master is allowed to initiate a communication with Slaves. Each slave has to acknowledge to the Master immediately for every command. For servo commands which may take some time for execution, the Slave has to acknowledge to the Master immediately after receiving the command. During servo command execution, the Master will poll the Slave for the execution status.

Communication Protocol

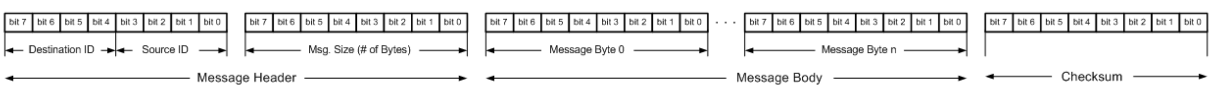

Figure 7.1 Message Frame Structure

As shown in the Figure 7.1 above, a message always consists of Message Header, Message Body and Checksum. Within message header, two bytes are used to define message destination address, source address and message size. Message body includes actual messages that are to be transferred from the source to destination. The actual messages can be commands, status info, error codes, configuration data or other parameters. Checksum is a value which is computed based on n+1 message bytes in the message body. It is intended for validity check of message bytes in a noisy environment. The message header is not included in the calculation of checksum. As long as a node detects the destination ID, if it is for other nodes and not relevant to itself, it can just discard the message after receiving the whole frame.

Destination ID: 4 bits (bit 7:4)

0b 0000	uP.
0b 0001	X-axis.
0b 0010	Y-axis.
0b 0011	θ-axis.
0b 0100	Z2-axis (Test Probe).
Others	Reserved for future.

Source ID: 4 bits (bit 3:0) Same as Destination ID.

As an example, if the uP sends a command to a X-axis SmartMotor, the Destination and Source IDs shall be set as below:

Destination ID = 1 and Source ID = 0;

However, when the X-axis (SmartMotor 1) acknowledges to the uP, the Destination and Source IDs shall be set as below:

Destination ID = o and Source ID = 1;

Message Size: 8 bits (bit 7:0)

It is the number of bytes in Message Body (excluding Checksum). The maximum number of message size is 255 bytes.

Message Body:

The total number of message bytes is determined by the Message Size. The contents are specific to each command and details are defined in following sections.

For a command from Master to Slave, it typically includes Command ID and associated CMD Parameters as shown in the Figure 7.2 below. For an acknowledgement from Slave to Master, it typically consists of Status, Error Code and Ack Parameters.

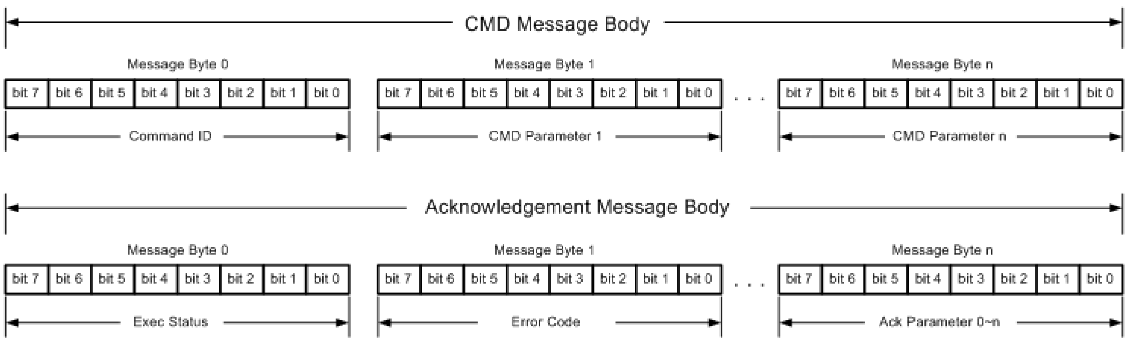

Figure 7.2 Typical Message Body Structure for Command and Acknowledgement Messages

Checksum Calculation

The checksum calculation is defined as:

$$Checksum = \sum_{i=0}^{n} MessageByte_i$$

Carries in the summation are ignored and individual message byte is treated as unsigned value.

Baudrate Settings

At Physical Layer, the uP communicates with SmartMotors through RS485. At Logic or Data Link Layer, UART is actually used. Communication settings are recommended as below:

Baudrate: 19200
Data Bit: 8
Stop Bit: 1
Parity: None

7.2 SmartMotor API Commands and Descriptions

This Section defines all API commands between the measurement uP and SmartMotors. To make the command description concise, the message headers and checksums are ignored. Only the command/message body is specified.

7.2.1 SM_homing

This command is to request SmartMotor to turn on the servo PID loop and perform homing. After completion of execution, the SmartMotor stays in its home position and waits for next command.

Command Message from uP

Message Size = 1
Command ID = 1
Command Parameters = None

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.2 SM_abs_move

This command is to move the SmartMotor to a user specified absolute target position using pre-defined servo motion parameters, such acceleration, velocity and etc.

Command Message from uP

Message Size = 5
Command ID = 2

Definition of Command Parameters

Byte Index	Byte Definition	Description
1	Target Enc. Pos (LSB Low)	Absolute Target Position in unit of encoder counts. Signed 32-bit Value.
2	Target Enc. Pos (LSB Mid)	
3	Target Enc. Pos (MSB Low)	
4	Target Enc. Pos (MSB High)	

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.3 SM_rel_move

Using pre-defined servo motion parameters, such acceleration, velocity and etc, this command is to move the SmartMotor with incremental encoder counts to its current position.

Command Message from uP

Message Size = 5
Command ID = 3

Definition of Command Parameters

Byte Index	Byte Definition	Description
1	Relative Enc. Counts (LSB Low)	Move the SmartMotor by relative encoder counts to its current position. In unit of encoder counts. Signed 32-bit Value.
2	Relative Enc. Counts (LSB Mid)	
3	Relative Enc. Counts (MSB Low)	
4	Relative Enc. Counts (MSB High)	

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.4 SM_get_enc_pos

This command is for uP to read back current SmartMotor position encoder counts.

Command Message from uP

Message Size = 1
Command ID = 4
Command Parameters = None

Acknowledge Message from SmartMotor

Message Size = 6

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.
3	Current Enc. Pos (LSB Low)	Current Servo Position in unit of encoder counts. Signed 32-bit Value.
4	Current Enc. Pos (LSB Mid)	
5	Current Enc. Pos (MSB Low)	
6	Current Enc. Pos (MSB High)	

7.2.5 SM_set_io_output

In SmartMotor, there are 7 IOs (I/O 0~9) can be used for generic purposes. This command is set the SmartMotor I/O output to ON of OFF.

Command Message from uP

Message Size = 3
Command ID = 5

Definition of Command Parameters

Byte Index	Byte Definition	Description
1	I/O Index #	0~9: Stands for SmartMotor Expansion I/O 0~9. Other Values: Reserved.
2	On/Off	0: OFF; 1: ON. Other Values: Reserved.

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.6 SM_get_io_inputs

This command is to read back current states of SmartMotor Expansion IO 0~9.

Command Message from uP

Message Size = 2
Command ID = 6

Definition of Command Parameters

Byte Index	Byte Definition	Description
1	IO Index #	0~9: Indicating SmartMotor Expansion IO.

Acknowledge Message from SmartMotor

Message Size = 3

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.
3	IO Input Status	0: OFF; 1: ON.

7.2.7 SM_servo_off

This command is to turn OFF the SmartMotor servo PID control loop. But the SmartMotor remains powered on until it is powered off by user.

Command Message from uP

Message Size = 1
Command ID = 7
Command Parameters = None

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.8 SM_jog_cw

This command is to jog the SmartMotor in Clockwise direction at a predefined low speed.

Command Message from uP

Message Size = 1
Command ID = 8
Command Parameters = None

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description

1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.9 SM_jog_ccw

This command is to jog the SmartMotor in Counter Clockwise direction at a predefined low speed.

Command Message from uP

Message Size = 1
Command ID = 9
Command Parameters = None

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.10 SM_jog_stop

This command is to stop immediately the SmartMotor jogging in progress.

Command Message from uP

Message Size = 1
Command ID = 10
Command Parameters = None

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.

7.2.11 SM_get_status

This command is to get execution status from a specific SmartMotor attached on the RS485 bus.

Command Message from uP

Message Size = 1
Command ID = 11
Command Parameters = None

Acknowledge Message from SmartMotor

Message Size = 2

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. When Status = READY or BUSY, this byte is 0.