Seagate

Electronics Circuit Design and Description for HGA Static Tester

Forrest Xu
Status: Draft

Date: March 2, 2016
V0.8

Change History

Date	Revision	Comments
May 14, 2015	V0.1	The first draft by Forrest Xu based on customer procurement specification.
May 25, 2015	V0.2	1. Add in "Counting Down" display for Bench Test GUI Figure 6.1. 2. Modifications on HST_get_short_detection for 10 HGAs (Section 4.4.10). 3. Modification on HST_get_product_id. 4. Modification on HST_get_operation_mode. 5. Modification on Debug GUI requirement. 6. Modification on HST_adc_write, HST_adc_read (add in ADC #) 7. Msg Size correction on HST_get_cap_reading.
July 25, 2015	V0.3	1. Port 10 bit 7 (P10_7) is used to enable offset calibrations for sensing resistance measurement channels. 2. Description on use of HST_meas_channel_enable. 3. Add in new command HST_calibrate_offset (4.4.38). 4. Add in new command HST_get_calibration_offset (4.4.39). 5.Change the Unsolicited Status Command ID from 38 to 255. 6. Modified GUI Calibration Tab Page to add in above two new commands. 7. Add in Section 4.5 for Error Codes and Description. 8. Add in Channel # in HST_get_temperature acknowledge message.
Sept 20, 2015	V0.4	1. Add in a new command "HST_set_offset_switch". 2. Add in a new command "HST_start_short_detection". 3. Add in a new command "HST_flex_cable_calibration". 4. Add in a new command "HST_get_cable_calibration_results". 5. Add in a new command "HST_clear_all_cable_compensation". 6. Add in Chapter 6.6, "GUI Requirement for Cable Calibration".
Oct 15, 2015	V0.5	1. Changes to command "HST_get_temperature". 2. Changes to command "HST_calibrate_offset". 3. Changes to command "HST_get_calibration_offset". 4. Modification to PCB Calibration GUI.
Dec 10, 2015	V0.6	1. Changes to "HST_flex_cable_calibration": add in up/down-tab index as command parameter. Add in cable capacitance calibration. 2. Changes to "HST_start_meas": add in up/down-tab index as command parameter. 3. Changes to "HST_get_cable_calibration_results": add in up/down-tab index as command parameter. Add in cable capacitance calibration. 4. Change the command name "HST_set_cable_compensation_resistance" to "HST_set_cable_compensation": add in up/down-tab index as command parameter. Add in cable capacitance calibration. 5. Add in new commands: "HST_set_short_detection_threshold" and "HST_get_short_detection_threshold". 6. Add in new commands "HST_set_temp1_offset" and "HST_get_temp1_offset". 7. Add in new command "HST_get_all_meas_results". 8. Changes on GUI Requirements: Figure 6.1, Figure 6.4, Figure 6.5. 9. Add in Section 4.6 for Description on Measurement IO Triggering.
Jan 14, 2016	V0.7	1. Changes to "HST_flex_cable_calibration": No calibration results are acknowledged because of limitation of maximum message size has to be below 256. Cable Calibration results have to be transferred with two separate commands. 2. Change the "HST_get_cable_calibration_results" as

		"HST_get_cable_calibration_res_results" because of msg_size limitation: to read out cable Resistance results. 3. Removal of command "HST_get_all_meas_results", because of msg_size constraint. 4. Add in a new command "HST_get_cable_calibration_cap_results". 5. Changes on the Cable Calibration GUI Tab.
Mar 2, 2016	V0.8	Add in Precisor Compensation for capacitance measurements: 1. Add in command "HST_set_precisor_cap_compensation". Section 4.4.52. 2. Add in command "HST_get_precisor_cap_compensation". Section 4.4.53. 3. Add in command "HST_save_precisor_cap_compensation". Section 4.4.54. Add in new GUI Tab for Precisor Compensation. Refer to detail requirements in Section 6.7.

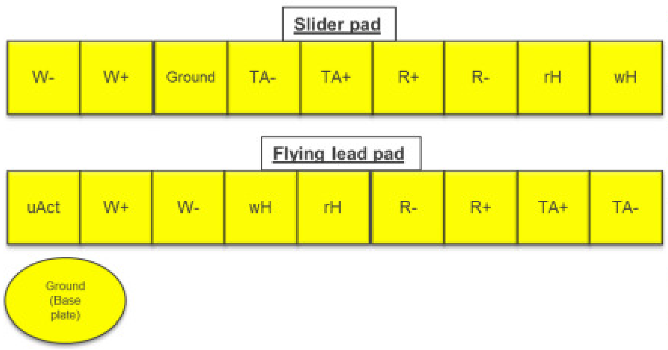
Reference Documents

[1] "HGA Static Tester Station Specification", Rev. x. Jan 2, 2015.
[2] "HGA Static Tester Acceptance Procedure", Rev. 2. Dec 30, 2014.
[3] "Scope of Work of HST", Aug. 5, 2014.
[4] "FOS Tail and Some TGA Design Roadmap", Dec. 12, 2014.
[5] "PZT Capacitance Variation", Rev. 1, 2014
[6] "Iris ST v19.1 & v19.2 Impedance Sweep Analysis", 2014.

Table of Contents

1. Introduction

HGA Static Tester is an automated test machine which measures multiple channels of resistances and capacitances on flying lead of HGA suspensions. As one of examples below, GrenadaBP2 flying lead has 9-pads, i.e. uAct, W+, W-, wH, rH, R-, R+, TA+, TA-.



No	Pad 1	Pad 2	Comment
1	W-	W+	No test
2	W+	Gr	Test
3	Gr	TA-	Test
4	TA-	TA+	No test
5	TA+	R+	Test
6	R+	R-	No test
7	R-	rH	Test
8	rH	wH	Test

Figure 1.1 GrenadaBP2 Flying Lead Pad Layout

For this GrenadaBP2 product, it is required to have 5-pairs resistance measurements, 1-pair capacitance measurement, and short-circuit detection for 5-pairs. Detailed pairs are listed as below:

Resistance Measurements: (W+, W-), (wH, Ground), (rH, Ground), (R+, R-), (TA+, TA-)
Capacitance Measurements: (uAct, Ground)
Short-Circuit Detection: (W+, Ground), (Ground, TA-), (TA+, R+), (R-, rH), (rH, wH)

From the Reference Document [4], there are several of products coming soon in three years and these products have different number of pads and layouts.

Product Name	Number of Pads	# of Resistance Measurements	# of Capacitance Measurements
Iris	9	5	1
Maple	10	5	2
Cadmium	8	?	?
Iris+MSMR2(400um)	11	6	1
Iris+MSMR2(300um)	11	6	1
Maple+MSMR2	12	6	2

To minimize the design efforts and product conversion cost, it is desired to share the same electronics measurement PCBA for all above products.

2. Customer Requirements Analysis and Critical Design Parameters

This section discusses about customer requirements and translates them into critical design parameters and components.

2.1 Customer Requirements

Table 2.1 below summarizes customer requirements for electrical measurements of HGA static tester.

Table 2.1 Summary of Customer Requirements for Electrical Measurements

Item	Parameters	Nominal Range	Recommended Bias Source**		Accuracy & Comments
			V Source	Limit Current	
1	Reader1 Resistance (R1+, R1-)	10~750 Ω	140mV	400µA	0.25Ω or 0.5%.
2	Reader2 Resistance (R2+, R2-)	TBD	TBD	TBD	0.25Ω or 0.5%. Reserved for future product.
3	Writer Resistance (W+, W-)	3~12 Ω	140mV	40000µA	0.25Ω or 0.5%.
4	Heater1 Resistance (H1, Ground)	10~160 Ω	1500mV	-	0.25Ω or 0.5%.
5	Heater2 Resistance (H2, Ground)	10~160 Ω	1500mV	-	0.25Ω or 0.5%.
6	TA Resistance (TA+, TA-)	50~150 Ω	100mV	400µA	0.25Ω or 0.5%.
7	PZT1 Capacitance (Uact1, Ground)	700~1000pF	2.5V	-	10pF or 0.5%.
8	PZT2 Capacitance (Uact2, Ground)	TBD	2.5V	-	10pF or 0.5%. Reserved for future product.
9	**Short Circuit Detection for All Adjacent Pads**	-	-	-	For 9-pad GrenadaBP2 product: (W+, Ground), (Ground, TA-), (TA+, R+), (R-, rH), (rH, wH). For other products, it is To Be Defined.

** The bias voltage in the above table is recommended bias voltage across the two terminal pads on the DUT. They are measured based on customer's ET test machine.

Other than the above specific requirements for the 9-pad GrenadaBP2 product, there are general requirements specified in the SOW as below:

Resistance Measurement Range:	0~10KΩ with accuracy of 0.25Ω or 0.5%
Capacitance Measurement Range:	0~10nF with accuracy of 10pF or 0.5%
Maximum Allowable Bias Voltage on DUT:	<2V
UPH requirement:	>=3000

2.2 Electrical Measurement Requirements and Critical Design Parameters

2.2.1 Requirements for Resistance Measurements

1. Depends on specific suspension failure mode, any of adjacent pads on a flying lead can be potentially shorted together. To prevent any potential damage to the electronics measurement PCBA, it is better to inject a controlled or limited current source to the DUT for resistance measurement. With this limited current source, the cross-talk or impact to other measurement channels can be minimized in case of short circuit.
2. To prevent potential damages to the suspension, a soft start is recommended when applying a current to the DUT. This is meant to minimize any potential high inrush current/voltage onto the suspension.
3. To prevent potential micro-arcing, a soft stop is recommended before the measurement probes (Pogo Pins) disengage with the suspension after completion of measurements.
4. The time constant for current ramping up/down is recommended to be 0.1s.
5. When ejecting a current source for resistance measurement, the power supply (typically using LDO) voltage shall be limited to 2V or below.
6. The measurement accuracy is very much dependent on the following design parameters: ADC resolution, ADC reference voltage for full range, source current, load and noise. This will be discussed later in the detailed design description.
7. Channel-to-channel crosstalk shall be <=-60dB.

Design Considerations for Resistance Measurement Accuracy

Figure 2.1 below is the recommended conceptual circuits for resistance measurements.

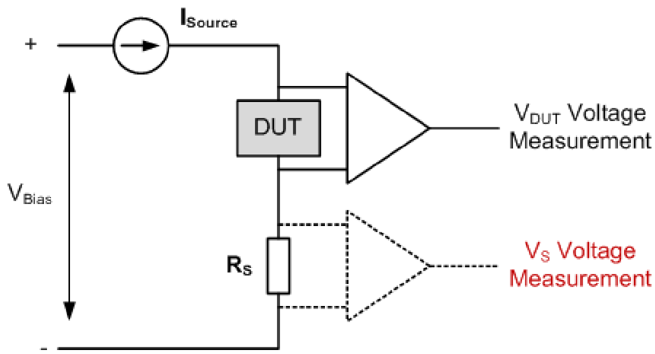


Figure 2.1 Conceptual Circuits for Resistance Measurement

As shown in the Figure 2.1, V_{Bias} is the total voltage applied to the DUT and sampling resistor R_S. I_{Source} is a configurable current source applied to the DUT and sampling resistor. The DUT resistance R_{DUT} can be calculated through the voltage measurement of V_{DUT}. The calculation is as below:

$$R_{DUT} = V_{DUT}/I_{Source} \qquad \text{(EQ.1)}$$

As long as V_{Bias} is greater than I_{Source}*(R_{DUT}+R_S), the measurement result from the Equation EQ.1 is independent of actual V_{Bias}. This gives benefit for power supply design which does not require an accurate voltage.

However, it requires an accurate current source I_{Source} in order to meet the resistance measurement accuracy of 0.5%. To have better understanding about the resistance measurement error, applying the first-order deviation to the Equation EQ.1, then we have:

$$\Delta R_{DUT} = \frac{\Delta V_{DUT}}{I_{Source}} - \frac{V_{DUT}}{I^2_{Source}} \times \Delta I_{Source} \qquad \text{(EQ.2)}$$

Where

ΔR_{DUT}	is the resistance measurement error.
ΔV_{DUT}	is the voltage measurement error, mainly contributed by DC offsets, ADC quantization and noise.
ΔI_{Source}	is the current source error, mainly contributed by electrical offsets, DAC quantization and noise.
VDUT	is the nominal or ideal value of voltage on the DUT.
I_{Source}	is the nominal or ideal current applied to the DUT.

Basically, the resistance measurement error consists of two major items, Item1 and Item2, i.e.

$$Item1 = \frac{\Delta V_{DUT}}{I_{Source}}$$

$$Item2 = -\frac{V_{DUT}}{I^2_{Source}} \times \Delta I_{Source} = -R_{DUT} \times \frac{\Delta I_{Source}}{I_{Source}} \qquad \text{(EQ.3)}$$

Item1 is the error caused by voltage measurement error and Item2 caused by source current error.

The measurement errors in both Equation 2 and 3 are the absolute errors. However, in most of applications, users are more interested in relative accuracy rather than absolute error. For example, a measurement error of 0.25Ω is significant for a 1Ω resistor, but it is negligible for 10KΩ resistor.
The relative accuracy can obtained from EQ.2 and 3 as below:

$$\frac{\Delta R_{DUT}}{R_{DUT}} = \frac{\Delta V_{DUT}}{R_{DUT} \times I_{Source}} - \frac{\Delta I_{Source}}{I_{Source}^2} \qquad \text{(EQ.4)}$$

From the Equation EQ.4, we can conclude the following design guidelines:

1. The higher source current I_{source}, the better measurement accuracy. Within the allowable limit, try to apply high source current to the DUT.
2. The higher nominal resistance, the better measurement accuracy.
3. Higher ADC and DAC resolution, the less quantization error and better measurement accuracy.
4. The DC offsets for ADC and DAC conversion circuits have to be calibrated to achieve better accuracy.

As an internal design specification, we like to achieve total 0.25% relative accuracy. What should be a reasonable resolution for ADC and DAC conversion?

Determination for ADC Resolution

To fully utilize the ADC resolution, the reference voltage for full-range conversion can be selected as 2V. The target measurement error contributed by the ADC is half of 0.5%, i.e. <0.25%.

From the above Table 2.1, we choose the two worst cases for ADC resolution selection.

Case 1: Reader1 Resistance=10Ω, Source Current=50% of Max. Limited Current=200µA

The required Voltage Resolution must be <$5\text{x}10^{-6}$ Volt, which means that the ADC resolution shall be >=19-bit.

Case 2: Writer Resistance=3Ω, Source Current=50% of Max. Limited Current=20000µA

The required Voltage Resolution must be <$1.5\text{x}10^{-4}$ Volt, which means that the ADC resolution shall be >=14-bit.

Practically, when the nominal resistance is below 50Ω, the allowable measurement error defined in SOW is 0.25Ω. The actual ADC resolution can be slightly compromised from the above calculation. However, the minimum resolution has to be 16-bit or above.

In summary, it is recommended to use Analog Device 24-bit SigmaDelta AD7178.

Determination for DAC Resolution

Referring to the above EQ.4, the second item is dominated by the current source offset and DAC resolution. The offset is to be cancelled off through factory calibration process. The DAC resolution is estimated as below.

Assuming the full range DAC current is 100mA and the target measurement error contributed by the DAC current source is half of 0.5%, i.e. <0.25%.

In the Table 2.1, the lowest minimum limit current is 400µA for TA measurement. Take 50% of 400µA, i.e. 200µA, for the calculation. The required current resolution is 0.5µA. It means that the DAC resolution shall be >=18-bit.

Obviously, it is very hard to have 18-bit or above DAC available from market! The most common available DAC is 16-bit. We have to look into alternative solution for the resistance measurements.

Alternative Solution for Resistance Measurements

As shown in the Figure 2.1, a high precision sampling resistor R_S is in series with the DUT. This RS has accuracy of 0.1% (Panasonic ERA3A or ERA6A series) with very low temperature coefficient (<25ppm). Since the same amount of current runs through both DUT and R_S, the resistance values of DUT can be calculated as:

$$R_{DUT} = \frac{V_{DUT}}{V_S} \times R_S$$
$$\frac{\Delta R_{DUT}}{R_{DUT}} = \frac{V_{DUT}}{V_S} \times \frac{\Delta R_S}{R_{DUT}} = \frac{\Delta R_S}{R_S} \qquad \text{(EQ.5)}$$

Ignoring the errors from ADC conversion and other electrical noises, the Equation EQ.5 tells us that the relative measurement accuracy can be independent of current source and dominated by the accuracy of sampling resistor, which provides great benefits to the hardware design.

Conclusions and Decisions

1. ADC resolution has to be 19-bit or above. Recommendation is to use 24-bit ADC from Analog Device. The reference voltage for full range conversion can be 2~2.5V.
2. DAC resolution can be 8-bit or 16-bit. But 16-bit is preferred.
3. DC offset and linearity calibrations shall be performed for ADC conversion circuits but optional for DAC current source. This can be done in factory before shipment.
4. High precision sampling resistors (0.1%) are required in series with DUT as shown in the Figure 2.1. Its value is recommended to be equivalent to the value of R_{DUT}.

2.2.2 Capacitance Measurements and Critical Design Parameters

Customer's intention for capacitance measurement is to verify the soldering joint quality between PZT and ground on the suspension. The measured value is served for two purposes:

1. Pass or Fail for production.
2. If Pass, the measured value is used as input parameter for other process.

To customer, this is a new feature yet to be implemented in production line. Currently it is still at customer's lab evaluation stage. Figure 2.2 shows the conceptual circuit diagram for capacitance measurement used by customer in their lab.

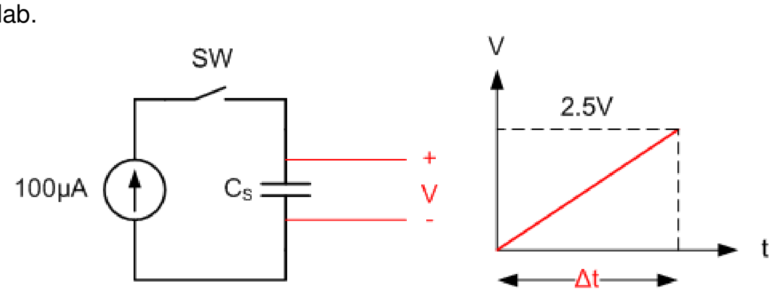


Figure 2.2 Customer's Conceptual Circuits for Capacitance Measurements

As shown in the Figure 2.2, when the switch SW is closed, the 100µA current source starts charging to the PZT (represented by Cs). The voltage on the PZT will increase linearly as the charging time. As soon as the voltage reaches 2.5V, the switch SW is open and charging to the PZT stops. Through the measurement of charging time Δt, the PZT capacitance can be calculated as below:

$$C = \frac{\Delta t \cdot I}{V}; \qquad \text{(EQ.6)}$$

Where V is 2.5V, I is the charging current 0.1A (or 100µA), Δt is the measured time.

As listed in the Table 2.1, the typical capacitance of GrenadaBP2 product ranges from 700~1000pF. Therefore, the measured Δt will be within 17.5~25µs.

This approach of capacitance measurement has some constraints on accuracy, because it is very sensitive to temperature/humidity, DC drift, current leakage, switching delay and noises. Figure 2.2 below shows its sensitivity to humidity. Figure 2.3 shows its sensitivity to temperature.

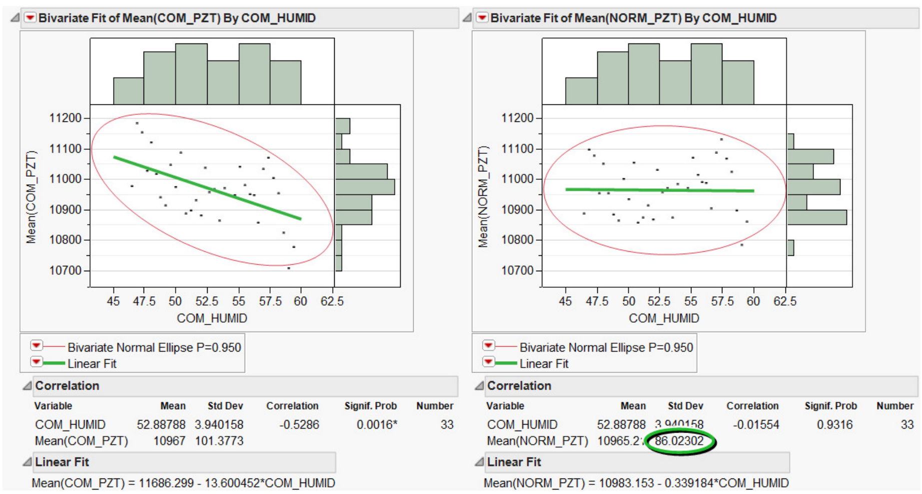


Figure 2.2 PZT Capacitance vs. Humidity

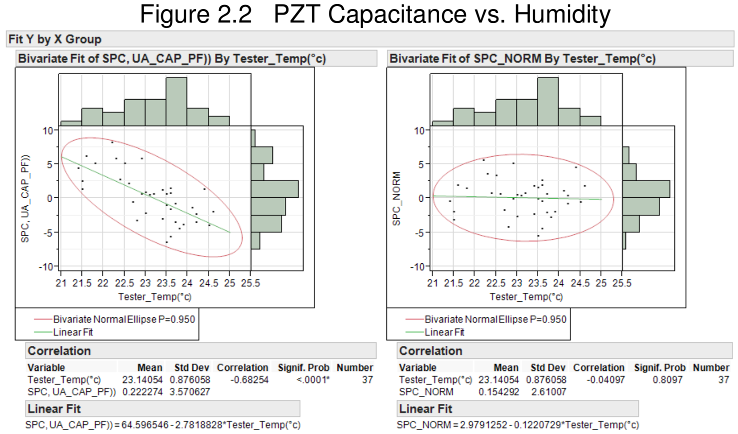


Figure 2.3 PZT Capacitance vs. Temperature

In general, this stepping pulse response approach is not the conventional way for capacitance measurements. For typical impedance measurement equipment (LCR), usually an AC (sine wave) voltage source signal at specification frequency is applied to DUT, the impedance is calculated through measuring the current response. Figure 2.4 shows a typical LCR meter circuit for capacitance measurement.

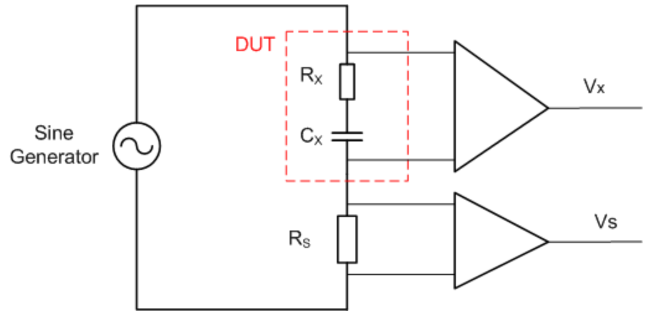
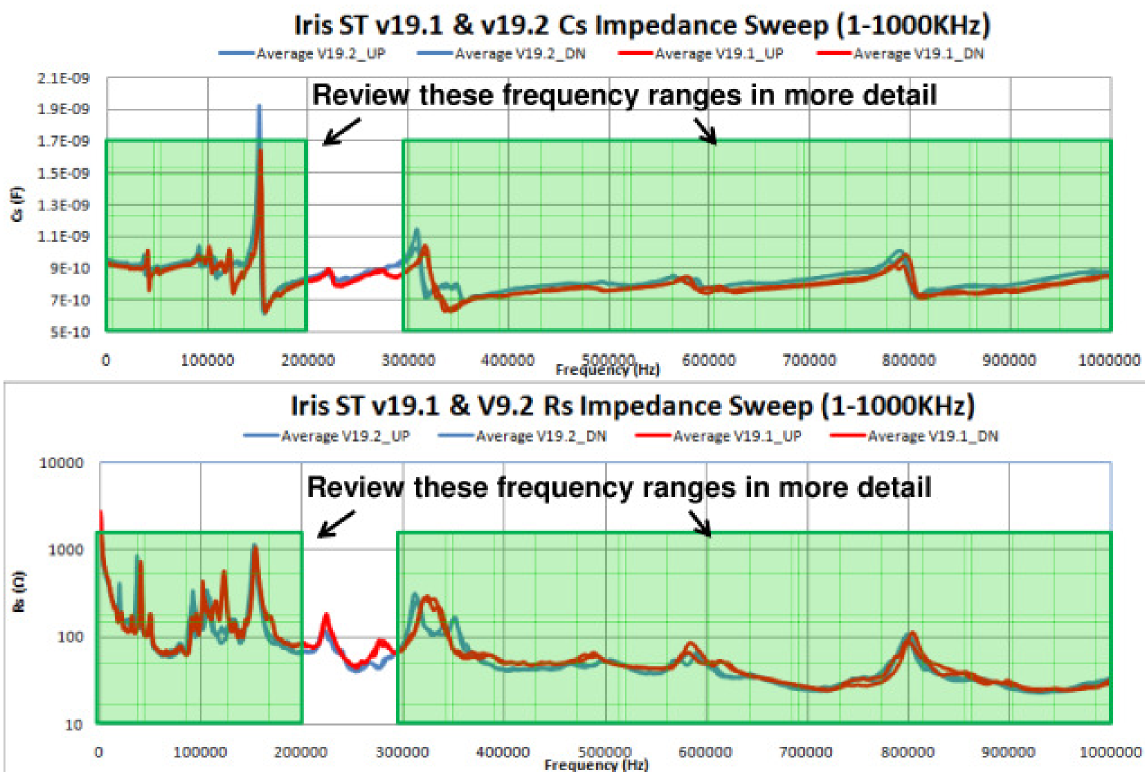


Figure 2.4 Type Circuits for Capacitance Measurements

First, apply a fixed frequency (user configurable) AC sine wave to both the DUT and Rs. Rs is the pure sampling resistor with negligible inductance and capacitance. Then measure the differential voltages V_X and V_S. The impedance of DUT (ZDUT) can be calculated as:

$$Z_{DUT} = \frac{V_X}{V_S} = R_X + \frac{1}{j \cdot 2\pi f \cdot C_X}$$, f is the frequency of sine wave signal. (EQ.7)

The capacitance result is very much related and varies with the measurement frequency. If the frequency is too high, the inductance of the DUT becomes dominant and the capacitance becomes unstable. Typically the measurement frequency for capacitance is below 1MHz. The optimal measurement frequency is dependent to characteristics of a specific DUT. As an example, Figure 2.5 below is a frequency sweep results on a suspension done by Seagate Teparuk team in Thailand.



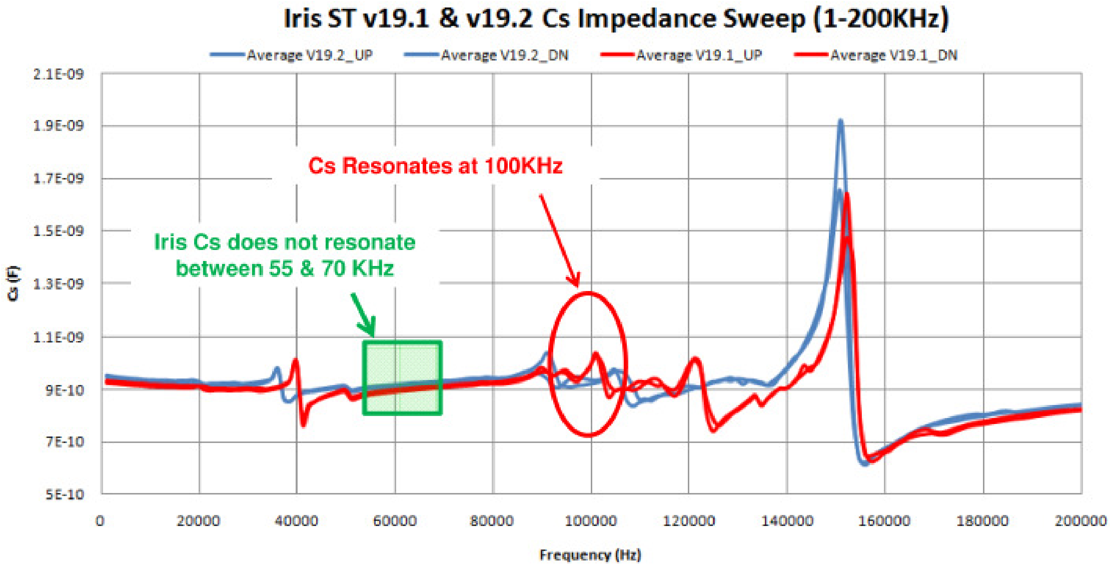


Figure 2.5 Frequency Sweep Response on Iris Suspension for Capacitance Measurements

The Figure 2.5 tells us following:

1. There is no absolutely correct value for capacitor measurement. It is dependent to the measurement frequency and the characteristics of DUT. The objective of measurement is to have a relatively stable result to serve the purpose of Pass or Fail inspection in production.
2. The measurement frequency is critical for the stability of measurement results. The optimal frequency is between 55~70KHz. This may be subjective for changes in the HAG static tester, because the wiring and layout is different from the test setup in the TeparuK lab.
3. PZT does not equal to a capacitor, PZT ≠ Capacitor. Capacitance measurement is an indirect way for checking the PZT soldering quality.

From all the discussions above, we can foresee that the capacitance measurement may cause ambiguity during the customer buy-off tests for HGA Static Tester. Or in another word, it will be an on-going co-development activity with customers after machine delivery. Therefore, some key technical decisions are made here.

1. Rather than developing our own proprietary capacitance measurement circuits, it is decided to use a standard LCR meter which is calibrated and certified by IET authorities.
2. The measurement parameters, such as frequency, bias voltage and etc., are fully configurable by users.
3. The LCR meter costs about USD$10K. For cost consideration, one LCR meter is shared between 10-suspension measurements through a multiplexer. The multiplexer consists of a set solid-state relays. These relays introduce extra measurement error of <2pF, which is acceptable.
4. The capacitance-related circuit design shall be in the form of an independent module, which can be easily upgraded with minimum modification on the existing design in future.
5. For initial internal evaluation and calibrations, we will use a set of standard capacitors which can be purchased from IET lab.

2.2.3 Short Circuit Detection

Referring to Figure 2.6, as one of production failure modes, any two adjacent pads on flying lead can be potentially shorted together. This is required to be tested by the HGA Static Tester.

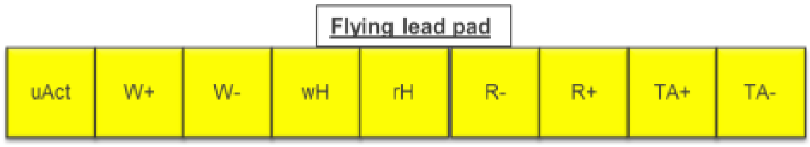


Figure 2.6 Test Pads on Flying Lead for Iris 9-Pad Suspension

Technically, there are many different ways to achieve this by hardware design. However, two major critical points have to be considered in the design:

1. The pad assignments as shown in the Fig. 2.6 are different from product to product. Therefore, the adjacent pairs for short circuit detection are different. For example as shown in Fig. 2.6 above, short detection for (uAct, W+) pair is required and the detection for (W+, TA+) pair is not required. For a different product, the short detection can be vice versa.
2. As one of objectives, we would like to share a common measurement PCBA for different products. It is mainly because that redesign on measurement PCBA takes long time and costly.

With the above two considerations, it is recommended to use a “soft way” of solution for the short detection, which is elaborated below.

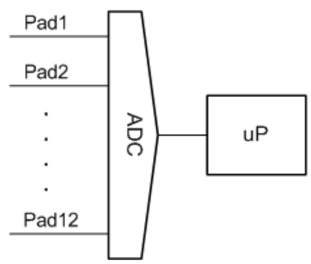


Figure 2.7 “Soft Way” of Solution for Short Detection

As shown in the Fig. 2.7, all pads (assuming 12 pads in total) are electrically connected to an ADC converter directly or indirectly. The basic operation is described as following:

1. Based on product configuration, user defines what pairs are to be tested and these configuration data are downloaded into the tester during initialization. The pairs can be (Pad_x1, Pad_y1), (Pad_x2, Pad_y2), . . ., and so on. X and Y can be any number within range of 1~12.
2. The uP controller ejects a variable source current (typically two different source currents) into related channels and accordingly voltages on (Pad_x1, Pad_y1), (Pad_x2, Pad_y2), ..., are sampled by the ADC.
3. If V_{Pad_X} and V_{Pad_Y} are always the same or differences are too small (threshold value to be defined), then these two pads are considered as shorted together.

To save cycle time, the resistance and capacitance measurement on a suspension will be skipped in case any pair of short circuit on the suspension is detected.

2.3 Design Considerations for Manufacturability and Others

This section discusses about design considerations for UPH, electrical isolation, sustaining, product conversion, factory functional tester and others.

2.3.1 Design Considerations for UPH Requirements

As stated in the feasibility study, the HGA Static Tester has to pick up 10 suspensions each time from testing in order to meet the 3000 UPH requirement. Maximum of 12s are allowed for the 10-suspensions loading, testing and unloading. Among this 12 seconds, 6~8 seconds are allocated for automation and 4~6 seconds are allocated for electrical testing which includes short-circuit detection, resistance measurement and capacitance measurement. There are different operating modes for the measurements shown as below, sequential or concurrent.

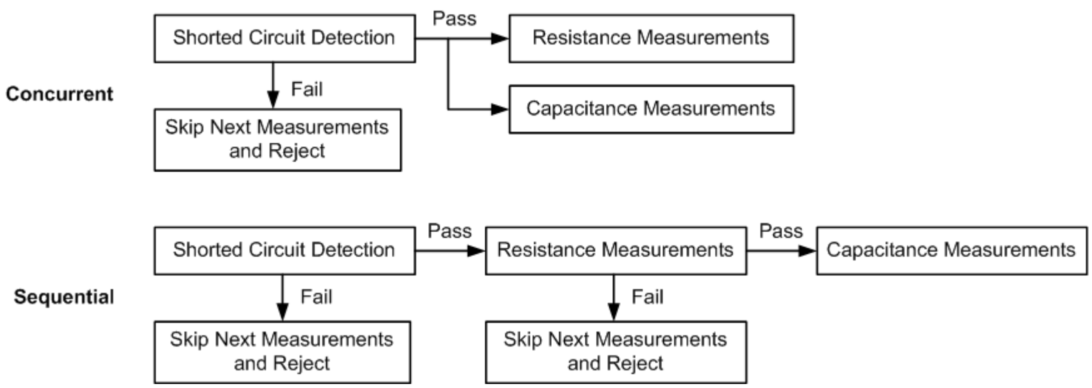


Figure 2.8 Sequential Mode and Concurrent Mode for Testing

Concurrent Mode

After short circuit detection, those passed suspensions are preceded with resistance and capacitance measurements concurrently. To minimize the interference or cross-talk, the resistance and capacitance measurements are synchronized such that they are not performed on the same suspension (as shown in the Figure 2.9 below). The synchronization is controlled by the uP on measurement PCBA.

When in this synchronized concurrent mode, the electrical testing time is expected to be around 4~5 seconds.

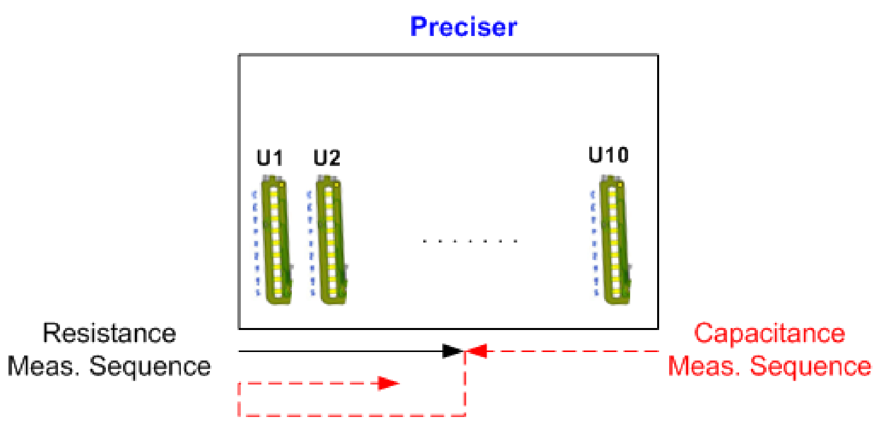


Figure 2.9 Synchronized Concurrent Resistance and Capacitance Measurements

Sequential Mode

There are some design constraints for the success of concurrent process. It requires proper electrical isolation between suspensions. If the isolation is not successful in final implementation, we have to go for sequential process, i.e. capacitance measurement is performed after the completion of resistance measurement. In this case, the testing cycle time will be slightly longer and expected to around 5 or 6 seconds.

Switching between Concurrent and Sequential modes is implemented through software and configurable by user. No additional hardware is required.

2.3.2 Design Considerations for Electrical Isolation between Suspensions

To meet the UPH requirements, 10-pcs of suspensions are loaded onto a precisor for simultaneous measurements as shown in the Figure 2.10 below. The precisor surface and the dowel pins are conductive. On the backside of the suspension, the exposed conductive traces and housing are connected to the Ground (Baseplate) pad on the flying lead. Therefore, when the 10-pcs suspensions are loaded onto the precisor, all the Ground Pads are short-circuit together as shown in the Figure 2.11.

Ideally it is better to have non-conduction surface on the precisor to ensure that 10-pcs of suspensions are completed isolated to each other. However, after consultation with mechanical designers, it is very challenging to achieve this because of accuracy, cost and reliability concerns.

As an alternative plan, the following discussions are focused on electrical design solutions based on assumptions that the suspensions cannot be isolated due to the conductive surface of precisor.

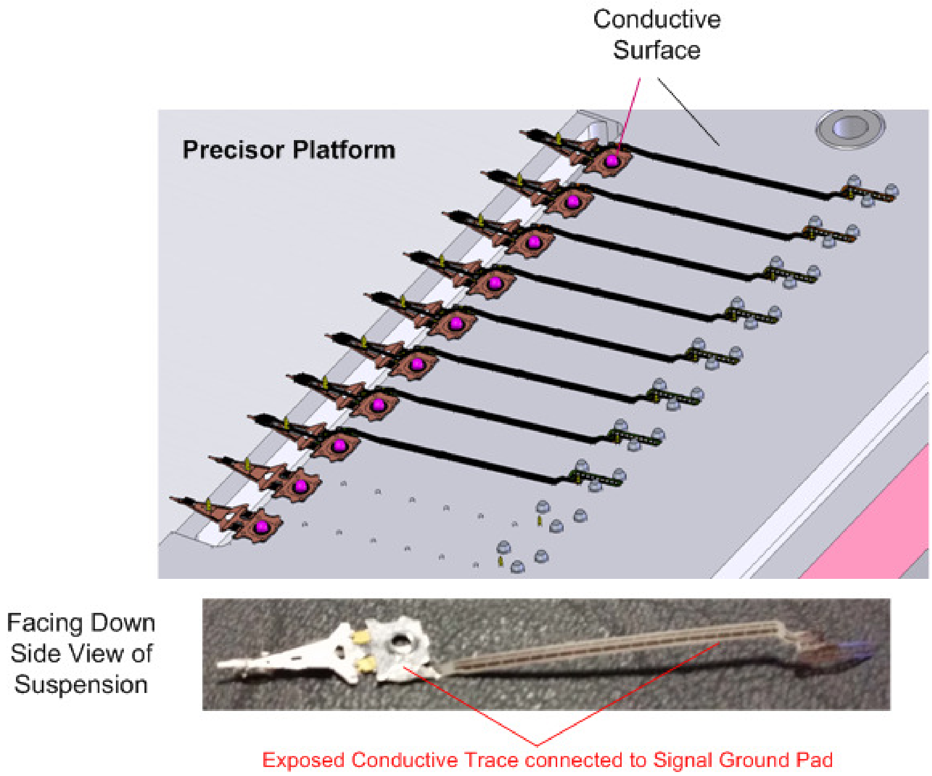


Figure 2.10 Conductive Surface/Trace of Precisor and Suspension

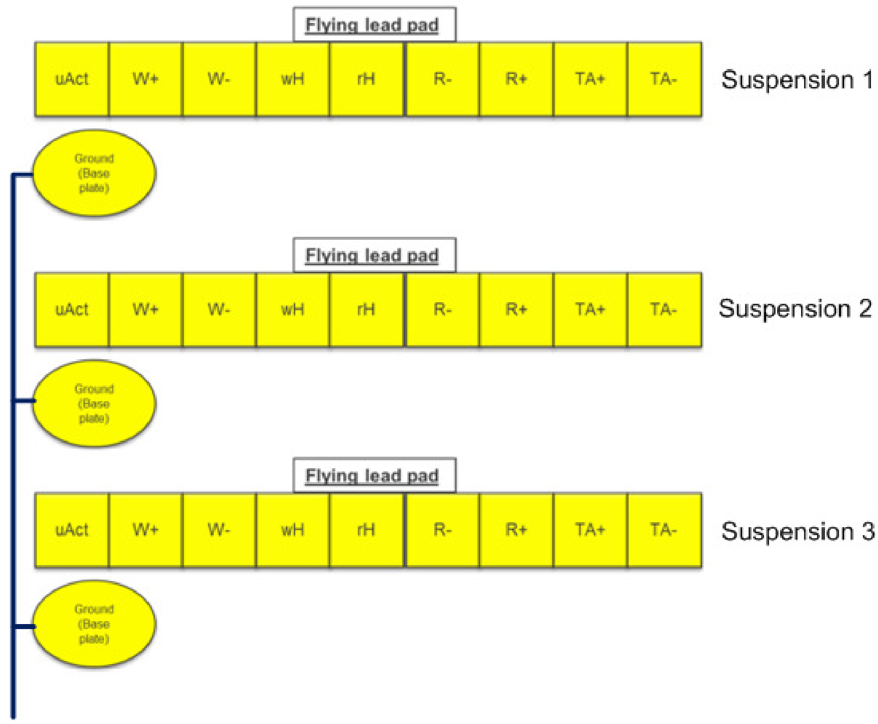


Figure 2.11 All Ground Pads are Short-Circuit due to Precisor Conductive Surface

2.3.2.1 Impact and Analysis on Resistance Measurements in Sequential Mode

As shown in the Figure 2.11, resistance measurements on (W+, W-), (R+, R-) and (TA+, TA-) are not affected when the Ground Pads are shorted together. Only two pairs, (wH, Ground) and (rH, Ground), are potentially affected.

In sequential mode, the LCR meter is physically disconnected (relay open) from the suspensions during resistance measurements. The equivalent measurement circuit for the two pairs (wH, Ground) and (rH, Ground) is shown in the Figure 2.12 below.

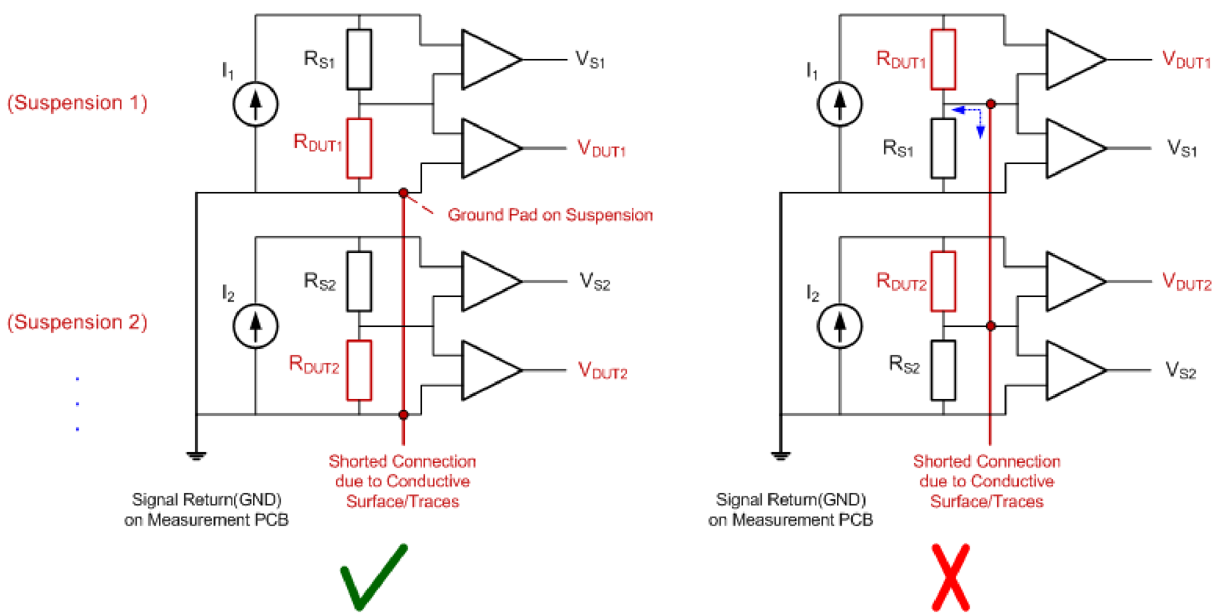


Figure 2.12 Equivalent Measurement Circuits in Sequential Mode

As shown in the left side of Figure 2.12, The short connected ground pads of suspensions are actually also connected with the Signal Return (or called as Signal Ground). The source current $\mathbf{I_X}$ (X=1, 2, ..., 10) runs through the sampling resistor first and then the DUT (Device Under Test). The portion of source current goes to the differential amplifier can be neglected because of its high input impedance. The resistance measurement result from the calculation based on Equation (EQ.5) is still accurate enough.

However, if swap the sampling resistor and DUT as shown in the right side of Figure 2.12, the short connected pads on all suspensions have to maintain the same potential voltage, which causes the current running through the DUT is different from the current through the sampling resistor. Thus the calculated resistance based on Equation EQ.5 becomes inaccurate.

2.3.2.2 Cross-Talk Analysis on Resistance Measurements in Concurrent Mode

As an example shown in the Figure 2.13, the resistance is measured on suspension #1 and the capacitance is measured on PZT of suspension #10 at the same time. The LCR meter ejects a ~60KHz sine wave voltage signal onto the PZT of suspension #10. One terminal of LCR meter, precisor base plate and the signal GND on the measurement PCBA are all shorted connected together in this situation. However, due to the fact that both the base plate and the signal GND are electrically isolated from machine body and external power supply, the 60KHz sine wave signal from the LCR meter does not form a direct close-loop with the suspension #1. Theoretically there is no impact on resistance measurement on the suspension #1. However, there may be small amount of AC noise coupled into the measurement circuit suspension #1 because of stray capacitance. Majority of this noise will be cancelled off by the differential amplifier which has high CMNRR (Common Mode Noise Rejection Ratio). Addition to that, it is recommended to apply a 100Hz Low-Pass Filter before input to the differential amplifier.

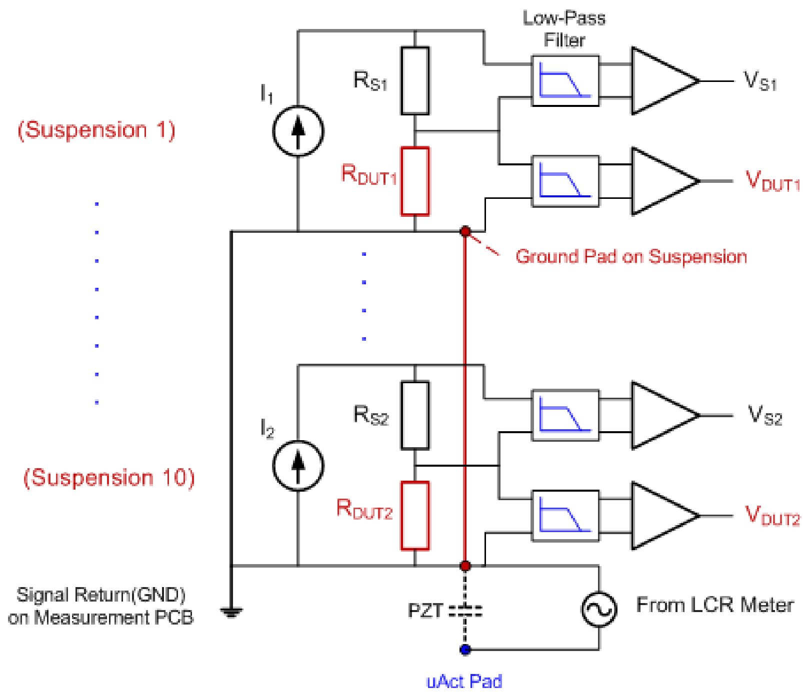


Figure 2.13 Cross-Talk Analysis when Measurements in Concurrent Mode

2.3.2.3 Impact and Analysis on Capacitance Measurements

We are applying a DC source current for the resistance measurements. From cross-talk point of view, the impact from resistance measurement onto the capacitance measurement is negligible.

As shown in the Figure 2.13, the ground pads on suspensions are all shorted together through precisor base plate and connected to a large Signal GND Plane on measurement PCB. This will introduce a certain amount stray capacitance when measuring the PZT. Figure 2.14 shows an equivalent circuit.

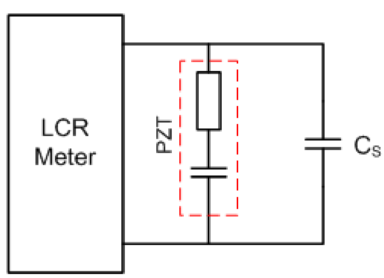


Figure 2.14 Equivalent Circuit for Stray Capacitance

Once the LCR measurement parameters are optimized and finalized, the CS will be calibrated. The PZT capacitance value can be calculated as:

$$C_{PZT} = C_{MEASURED} - C_S \quad \text{(EQ.8)}$$

2.3.2.4 Electronics Isolation for Precisor with Machine Base Frame

As shown in the Figure 2.15 below. The precisor will be electronically isolated from the machine base frame through **ESD-compliant** engineering plastics. The ESD compliance ensures that electrons will be slowly discharged to the Earth Ground and they are not accumulated on the precisor platform. Therefore, the voltage potential can be maintained as the same level as the Earth Ground.

Another important reservation is to have two M2 tap-holes; one is on the side of precisor and the other one is on the base frame. This tap-hole can be used to connect the precisor to either the measurement signal return or Earth Ground in the future if necessary.

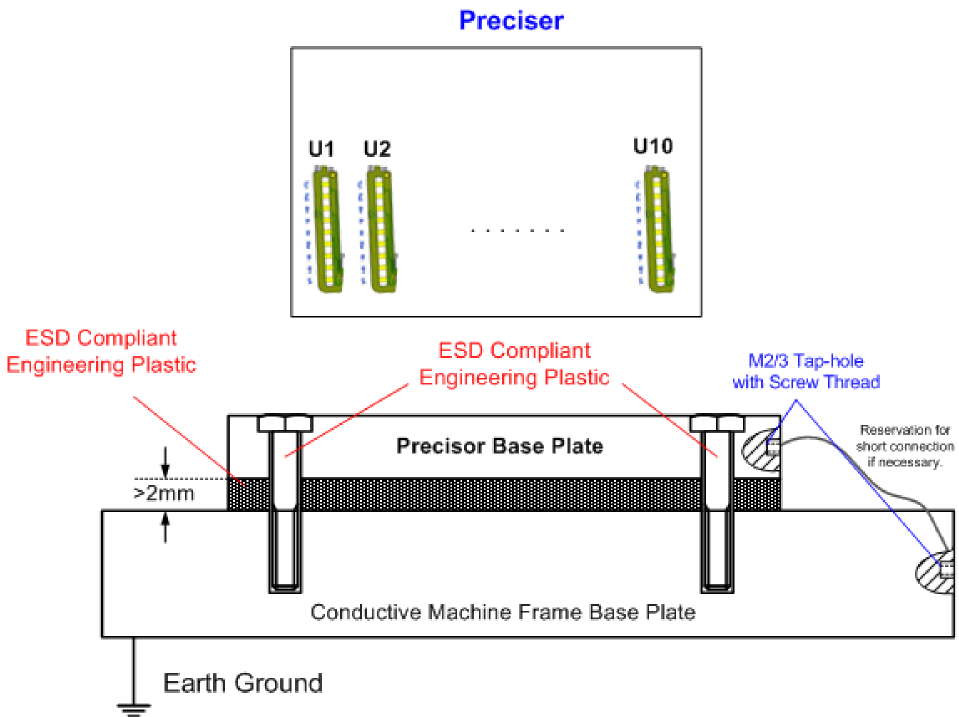
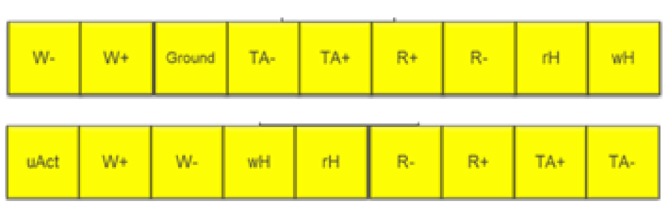


Figure 2.15 Proposed Electronic Isolation between Precisor and Machine Base Frame

2.3.3 Design Consideration for Product Conversions

The pad assignments on flying lead can be different from product to product as shown in the Figure 2.15. To share the same measurement PCBA, a conversion PCB is required to redirect the pads on flying lead to their corresponding test pins on the measurement PCB. When product is changed, the conversion PCB has to be changed if the pad assignment is different. To prevent human error that a wrong conversion PCB is used, each conversion PCB has its unique ID (4 or 5-bits) for verification. This ID is read by the machine at initialization. If the user configured Product Type does not match with conversion PCB ID, the machine will prompt an error message to the user and reject any measurement request. Figure 2.16 shows the conceptual connection between PCBAs.



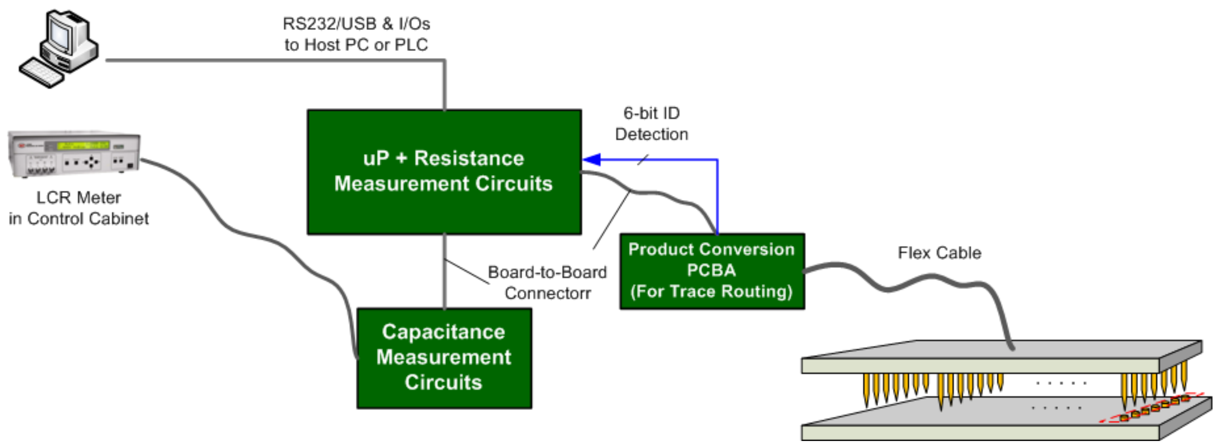


Figure 2.16 Conceptual Connection between PCBAs

3. Circuit Design and Critical Component Selection

This section of document describes the detailed circuit design for critical function blocks.

3.1 Overview and Functional Blocks

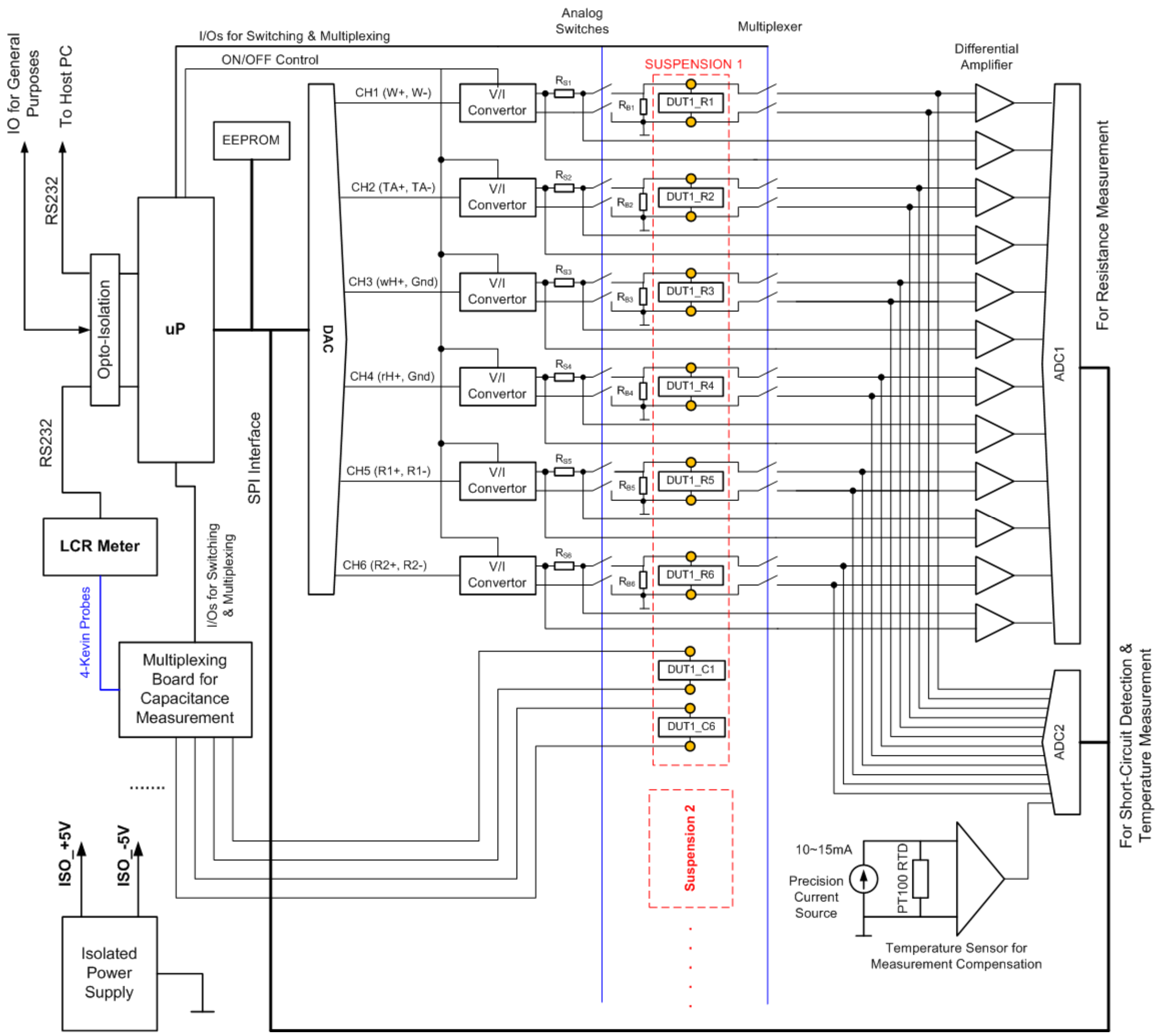


Figure 3.1 Overview and Functional Blocks for HGA Measurement Circuits

From the Figure 3.1 above, the measurement circuits consist of following major functional blocks: power supply, uP controller, DAC and current sources, ADC and differential amplifiers, analog switches/multiplexer, capacitance measurements/LCR meter, temperature measurement.

Isolated Power Supply

The power supply for the measurement PCBA is sourcing from external +24VDC. This +24VDC may be also used for pneumatic values, stepper motors or other high-power servo drives. The power supply for measurements should be isolated from all high power electronics/actuators to avoid potential EMC noises and to ensure measurement accuracy of 0.25% (internal design spec.). This isolation includes all IO signals and RS232 communication, because they all share the common +24VDC return.

Tentatively, this isolated power supply produces two power rails, +5V and -5V. This can be changed (i.e. +12V and -12V) depending on final component selection. In case ±12V supply is selected, other different voltages shall be derived from this isolated supply using LDOs. Switching regulators shall be avoided because of noise consideration. The total power consumption shall be designed for >=10W.

uP Controller and Communication/Control Interfaces

The basic functions for the uP are described in the following.

1. Acting as a slave (server) to the host PC, receive user configuration data for measurements such as bias currents for resistance measurement channels and pairing for short-circuit detection.
2. To read the conversion PCB ID and upload it to the host PC for verification.
3. To receive LCR configuration data from the host and download them into LCR meter.
4. To synchronize the measurements between resistance and capacitance.
5. To perform DAC/ADC measurements and necessary correction based calibration data.
6. To be responsible for signal switching and multiplexing between the 10-suspensions under test.
7. To perform DC offset and linearity calibration for DAC/ADC measurement circuits and save these calibration data into on-board EEPROM.
8. To monitor Input signal from motion controller and start measurements when motion is ready.
9. To upload measurement results after completion of measurements.
10. To output a signal to motion controller after completion of measurements.

DAC and Current Sources

There are 6-channel of DAC used to provide 6-different user specified amount of currents to each resistance measurement channel, the range of currents is between 0~100mA with step size of 10uA. The absolute current accuracy is less critical for measurement and its tolerance can be relaxed to 10%, but the stability is critical and shall be controlled within 0.2%.

As shown in the Figure 3.1, there are two precision resistors, R_{SX} and R_{BX} (X=1, 2, ..., 10). R_{SX} is in series with the R_{DUT} and used for resistance calculation. R_{BX} is parallel to the R_{DUT} which is mainly used for prevention of micro-arcing. Before the test probes engaging/disengaging with the suspension test pads, the current source will be set to 0 current or removed from all suspension. RBX is acting as a bypassing resistor to discharge electrical energy stored in the DUT (if any) before unloading the suspensions from precisor. Details are to be discussed later in current sourcing circuit design.

ADC and Instrument Amplifiers

There are two ADC convertors, ADC1 and ADC2. ADC1 is for high precision resistance measurement and typically it is 24-bit Sigma-Delta AD convertor. ADC2 is meant for short-circuit detection, which needs to be fast with less accuracy. Typically a 16-bit is more than enough.

For each resistance measurement channel, there are two separate differential voltages to be measured, one the voltage across the two terminals of DUT, the other one is the voltage across the sampling resistor. It is recommended to use instrument amplifier TI INA114 or INA101, which has CMNR>100dB.

Analog Switches/Multiplexers

As shown in the Figure 3.1, the DAC and ADC measurement circuits are shared with the 10-suspensions. The analog switches are used to apply the 6-bias currents to different suspensions. The on-resistance of these analog switches shall be < 10Ω and their current offset from inputs shall be <0.25µA.

Similarly, the multiplexers are used for ADC sampling from different suspensions. The input current shall be controlled within 0.1µA.

Capacitance Measurement and LCR Meter

For each suspension, there are two-channel of capacitance are required, one PZT1 and the other one is reserved for PZT2 in future. With the consideration of cost and space, one LCR meter is used for the 10-suspension capacitance measurements. The multiplexing between these 10-suspensions is realized through a set of relay matrix. As a design requirement, the total stray capacitance contributed by the relays in the measurement path shall be <5pF.

Temperature Measurements

The temperature measurement is meant for the compensation of resistance and capacitance measurement results. The temperature range is designed as 0~50℃ with typical measurement point around 25±5℃. As a design target, the accuracy is ±0.2℃ and the commitment to customer is ±0.5℃.

The recommended temperature sensor is PT100, which has resistance of ~105Ω and ~0.4Ω/℃ within temperature range of 10~30℃. The nominal value of the current source is 10mA, which provides ~1V voltage for the ADC conversion (2V for full-range).

The PT100 temperature sensor is externally connected to the PCBA, therefore, two terminals have to be designed in for its external connections.

3.2 Proposed Circuit Design and Key Component Selection

3.2.1 Power Supply Module

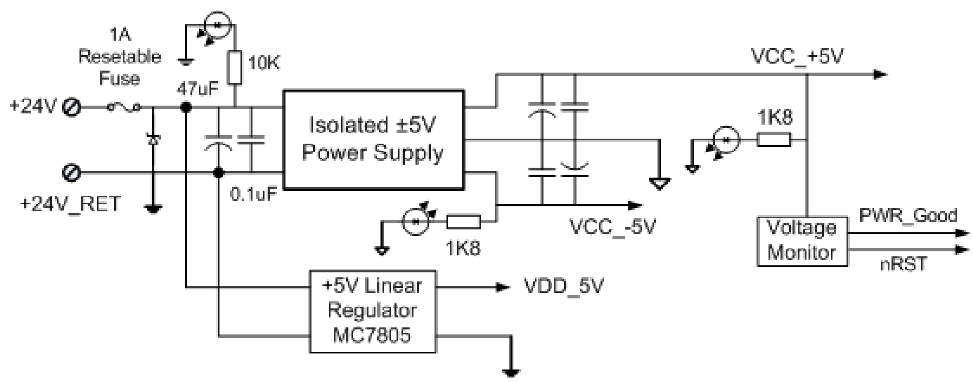


Figure 3.2.1 Power Supply Design & Key Components

There are three power supplies generated from the source of external +24V supply, VDD_5V, VCC_+5V and VCC_-5V.

VDD_5V	is the +5V supply sharing the same return as external +24V. It is used as the supply on the field side of the Opto-Isolating circuits.
VCC_±5V	are generated from an Isolated ±5V Power Supply Module (TDK-Lambda CC10-2405DF-E). They are mainly used as dual supplies for Op-amps and differential/instrument amplifiers in DAC and ADC circuits. The dual supplies are necessary to ensure the accuracy and linearity when the measurement signals are very small or nearly zero. A voltage monitoring IC is associated with the VCC_+5V supply to ensure proper power-on reset or voltage drop during operation.

There are two separate power returns/grounds. They shall be strictly isolated from each other and no short connection is allowed on the measurement PCBA.

1A resettable fuse and reverse protection are required in the circuit design as shown in the Figure 3.2.1.

For the +24V input, it is recommended to use PhonixContact screw terminals rather than connector for the sake of robustness. The recommended supplier part is MPT 0.5/2-2.54.

3.2.2 Circuit Design for DAC and Current Sources

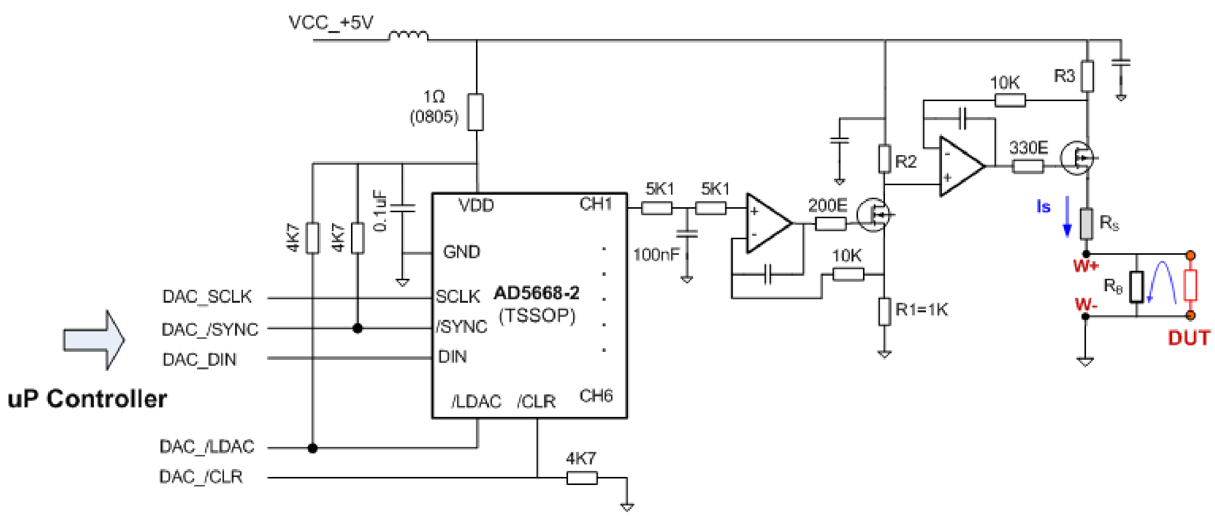


Figure 3.2.2 Proposed Circuit Design for DAC and Current Source

As shown in Figure 3.2.2 above, the six programmable current sources consist of DAC (AD5668-2), low noise and precision operational amplifiers (ADA4661-2), two MOSFETs (1 N-channel and 1 P-channel) and sampling resistor (R_S). These six current sources are used to provide DC currents to individual 6-resistance measurement channels as listed in the Table 2.1. The amount of current is programmed through a uP-controlled DAC. Communication interface between uP and DAC is three-wire SPI (DAC_SCLK, DAC_/SYNC, DAC_IN) and two additional IO signals, i.e. DAC_/LDAC and DAC_/CLR. These two signals are used to enable (/LDAC) or clear (/CLR) the DAC channel outputs (CH1~6, CH7 & 8 are not used). This enable/clear is an important feature for two purposes:

1. To avoid current impulse when switching between the 10-suspensions under testing.
2. To avoid micro-arcing when the test probes engaging/disengaging with the 10-suspension.

When the DAC_/CLR is activated, the DAC channel output is forced to be 0 volt, the MOSFETs shutdown the current loop. To ensure the shutdown without DUT, an optional bypass resistor (dotted-line R_B in Fig. 3.2.2) is proposed to be placed in parallel with DUT. In this way, the test pads from DUT are pulled down to the signal ground when the MOSFETs are shutdown. This is mainly to prevent micro-arcing at the moment when the test probe is engaging with the test pads. To minimize the impact to measurement accuracy, the R_B shall be greater than the R_{DUT}. It has to be a precision resistor as R_S with accuracy of 0.1%. The calculation formula is different from Equation (EQ.5) and shall be following Equation (EQ.9).

$$R_{DUT} = \frac{V_{DUT} \times R_S \times R_B}{R_B \times V_S - V_{DUT} \times R_S} \quad \text{(EQ.9)}$$

Another function of the R_B is meant for ESD protection. Any device, to certain extend, it may have a bit of capacitor's characteristics, which means it is able to store certain small amount of electrical energy. After completion of tests, the DAC current source is switched to another suspension for measurement. With the bypassing resistor R_B, the stored electrical energy (if any) can be discharged via the R_B resistor. Through this way, at the moment when the suspension is unloaded from the precisor, it has the same potential as the Earth Ground and therefore no damage to the suspension.

Furthermore, at each DAC channel output channel, a RC is designed to provide a soft stepping response when a different current setting is applied during short-circuit detection. With R=5.1K and C=100nF, the step response time is expected to be around 0.5ms.

The two operational amplifiers and MOSFETs form a close-loop control to provide a steady/stable current through the sampling resistor and DUT. The relationship between the DAC output voltage V_{DAC} and the current through sensing resistor/DUT is as below:

$$I_S = \frac{V_{DAC} \times R_2}{R_1 \times R_3}$$

Referring to the Table 2.1, the recommended values are listed below.

Table 3.1 Recommended Values for R_L and R_S

Item	Parameters	Nominal Range (Ω)	Recommended Bias Source**		Recommended Values				
			V Source (mV)	Limit Current (uA)	R_S (Ω)	R1 (Ω)	R2 (Ω)	R3 (Ω)	R_B(Ω)
1	Reader1 Resistance (R1+, R1-)	10~750	140	400	100	1K	470	470	100K
2	Reader2 Resistance (R2+, R2-)	TBD	TBD	TBD	100	1K	470	470	100K
3	Writer Resistance (W+, W-)	3~12	140	40000	100	1K	470	47	100K
4	Heater1 Resistance (H1, Ground)	10~160	1500	-	100	1K	470	47	100K
5	Heater2 Resistance (H2, Ground)	10~160	1500	-	100	1K	470	47	100K
6	TA Resistance (TA+, TA-)	50~150	100	400	100	1K	470	470	100K

The DAC (AD5668-2) has its own internal or uses external 2.5V as reference voltage. When internal 2.5V is used, the full-scale output is **5V**. When the external 2.5V is used, the full-scale output is 2.5V. Ignoring the DC offsets in the current source circuit, the nominal DAC setting can be calculated as:

$$DACSetting = \frac{65536 \times R_1 \times R_3}{R_2 \times V_{FullScale}} \times I_S \qquad \text{(EQ.10)}$$

I_S is the target current applied to the DUT, in unit of ampere A;
R_1, R_2, R_3 are the resistor values as defined in the Table 3.1 above, in unit of ohm;
$V_{FullScale}$ is the DAC full-scale output voltage, either 5V (for internal 2.5V reference) or 2.5V (for external 2.5V reference).

3.2.3 Circuit Design for ADC Conversion and Resistance Calculation

Circuit designs for two different ADC ICs (AD7739 and AD7173) are shown in the Figure 3.2.3 and 3.2.4 respectively. The analog-to-digital conversion rate is up to 15K samples/s for AD7739 and 3.1K samples/s for AD7173. Both of them are 24-bit and the effective number of bits is dependent to conversation rate used in the application. AD7173 has more noise filtering and other features than AD7739. If the lab evaluation result shows positive, AD7173 is preferred. The intended conversion rate is 5.2K samples/s with effective resolution of 21-bits (RMS).

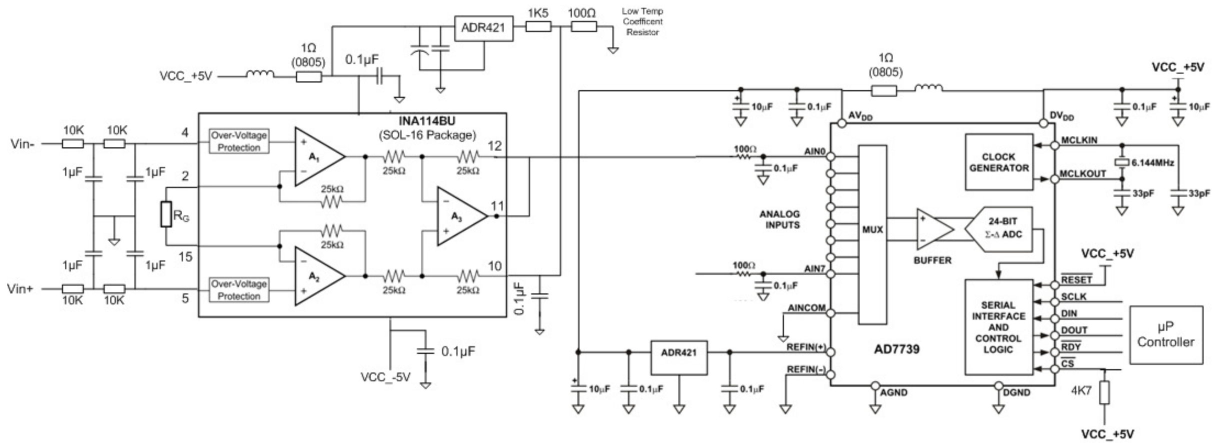


Figure 3.2.3 ADC Conversion Circuits (AD7739) for Resistance Measurements

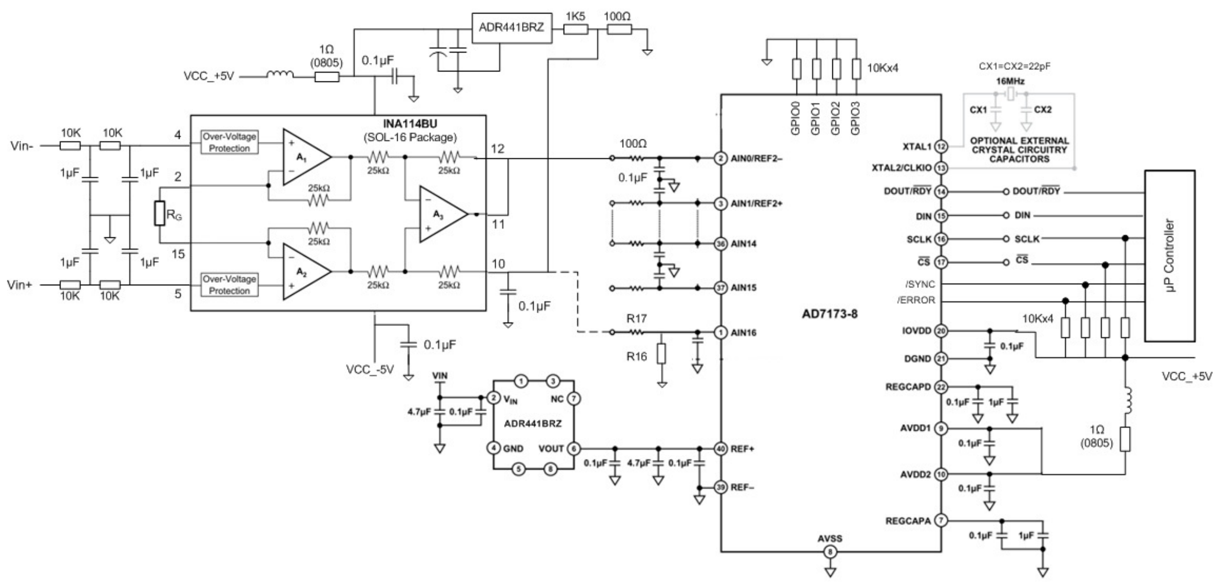


Figure 3.2.4 ADC Conversion Circuits (AD7173) for Resistance Measurements

The circuit design mainly consists of four major blocks, i.e. LPF filter, differential instrument amplifier and ADC conversion and µP controller.

LPF Filter Circuits

Prior entering into the instrument amplifier, the differential signals from suspension test pads are passing to a LPF formed by a π-RC network. Its function is to suppress a potential 60KHz common-mode noise from LCR during concurrent resistance and capacitance measurement. The time-constant is about 10ms. This means that a minimum delay of 10ms is required before starting ADC conversion when switching to another suspension. The target resistance measurement time is <1s for all 10-suspensions. In final product performance evaluation, if result shows less-concern on the 60KHz common mode noises, then the RC time constant can be further reduced to optimize the cycle time.

Differential Instrument Amplifier Circuits

The INA114 from TI is recommended for the amplifier circuit design. It has CMNR of >115dB. The gain setting is determined by an external resistor R_G, which can be calculated as:

$$Gain = 1 + \frac{50K\Omega}{R_G} \qquad \text{(EQ. 11)}$$

Tentatively, the Gain can be set to 2, i.e. R_G≈50KΩ. To ensure temperature stability, the resistor R_G must have very low temperature coefficient, i.e. <20ppm. Keep in mind, the whole ADC conversion and measurement circuits will be calibrated with loop gain and offset. Details will be discussed in the Chapter of “Circuit Calibration Algorithms”.

Typically, an ADC for unipolar input may have certain “dead-zone” or “nonlinear-zone” when the input signal is very small. To prevent this when the measured resistance is very small, the output from the IN114 is level-shift about 150mV. This is achieved by applying a 150mV reference (or bias) voltage to its Pin10 as shown in Figures above. Again to ensure measurement accuracy, the exact reference voltage is to be calibrated.

ADC Conversion Circuits

Two ADC conversion circuits are shown in the above Figures. These two circuits are similar and only difference is the ADC ICs. The outputs from the instrument amplifiers (12 in total) are connected to the ADC IC through a simple RC (R=100Ω, C=0.1µF) filter. For AD7173, it is configured as a 16-channel single-end input AD convertor. The channel AIN16 is connected to ground or optionally to the reference of INA114 as a backup. Referring to datasheet, the AD7173 is mainly designed for 8-channel differential input channels. In this HGA application, AIN16 is used as the common shared “ground” for the other channels AIN0~15 inputs. Inside the AD7173, (AIN0, AIN16), (AIN1, AIN16), ..., (AIN15, AIN16) are still treated as differential-pair inputs to the ADC convertor.

A external 2.5V Voltage Reference is used as ADC conversion. The analog input full-scale range is 0~2.5V unipolar. For AD7173, the conversation result is calculated as below:

$$Data = \left[\frac{0.75 \times V_{IN}}{V_{REF}} \times 2^{23} - (Offset - 0x800000)\right] \times \frac{Gain}{0x400000} \times 2 \qquad \text{(EQ.12)}$$

Where, Offset and Gain is the ADC offset and gain, with default values of Offset=0x800000 and Gain=0x555555.

SPI Interface with µP Controller

The AD7739 and AD7173 support different version of SPI standards, such SPI, QSPI, MICROWIRE and etc. In HGA application, 3-wire SPI protocol is used for the communication between uP and ADC ICs, i.e. SCLK, DI and DO. The chip select /CS is separately controlled by a uP GPIO which provides more flexibility for software control.

For AD7173, an additional /SYNC is used for synchronization during continuous conversation and read. Details are to be discussed in the software interfacing commands.

3.2.4 Circuit Design for Short-Circuit Detection

As shown in the Figure 3.1 overview circuit design, an ADC is used for “soft-way” short-circuit detection. For this purpose, the same AD7173 can be used. The test pads are directly connected to ADC inputs via RC. The circuit design is shown as below:

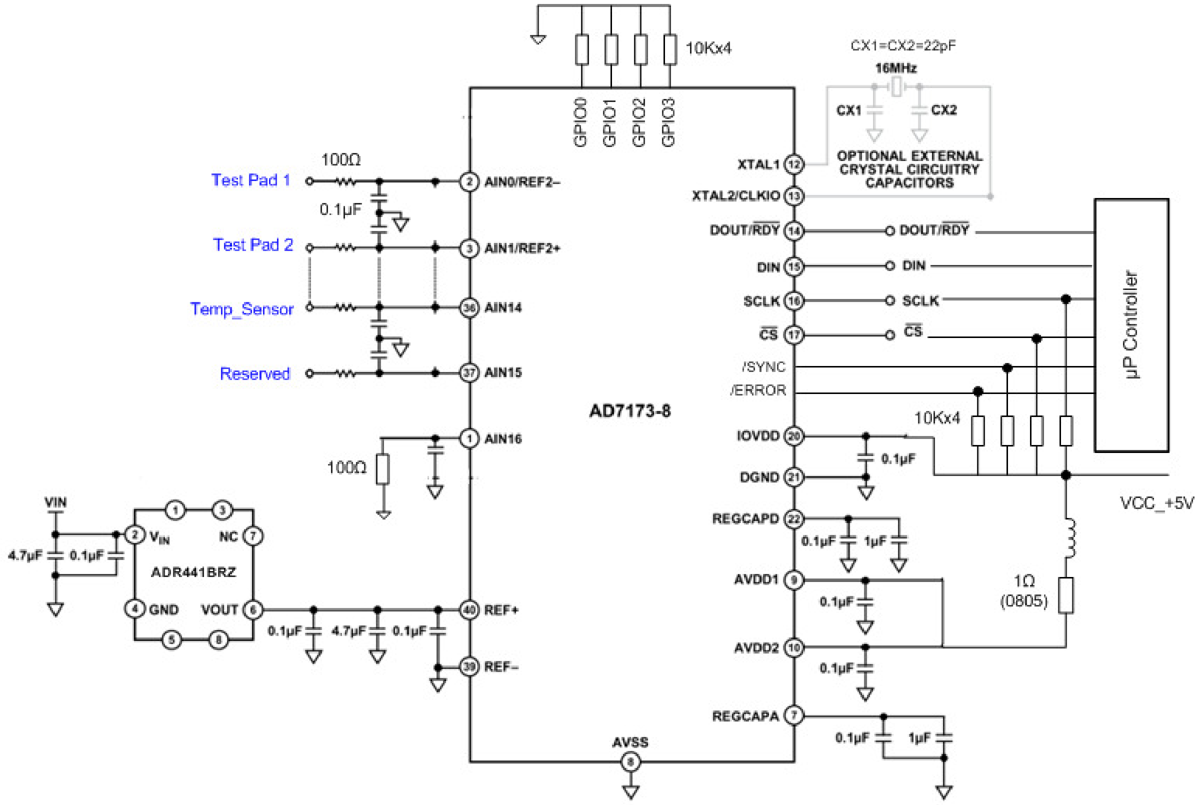


Figure 3.2.5 Circuit Design for Short-Circuit Detection

With the AD7173, the differential pairing can be configured by user in its built-in MUX register. Through applying two different sourcing currents, if the differential voltage on the paired test pads is always 0, these two paired test pads can be considered as short-circuit.

Important Notes:

The circuit design in Figure 3.2.5 is similar to the design in Figure 3.2.4, except for the instrument amplifier circuits. Inside the AD7173, it has also a built-in differential amplifier which performs similar functionality but with Gain≈1 and 80dB common mode noise rejection ratio. During prototype evaluation, if the result shows positive without the INA114 (which has common mode noise rejection ratio >115dB), then we can remove the circuit design as shown in the Figure 3.2.4 to reduce some cost and no PCB re-layout is required. The potential cost saving is about USD$50~80.

3.2.5 Analog Multiplexer/Demultiplexer Circuit Design for Current Source and ADC

As discussed previously, there are six channels of DAC current sources. These 6 DAC current sources are applied to the corresponding 6 channels of one suspension for resistance measurements. After completion of measurements on one suspension, these 6 DAC current sources are switched or demultiplexed to another suspension and repeat the similar measurements till all 10 suspensions are measured. On the other hand, for one suspension, there are 12 outputs from instrument/differential amplifier circuits to the ADC for voltage sampling. One resistance channel on a suspension has two outputs for ADC conversion, one from sampling resistor and the other one from the suspension channel (DUT). There is no multiplexing required for the sampling resistor voltage measurement. The other 6-channels of voltages from the 6 channels on a suspension have to be multiplexed in order to share the same ADC convertor between 10 suspensions. Figure 3.2.6 shows the overview circuit design.

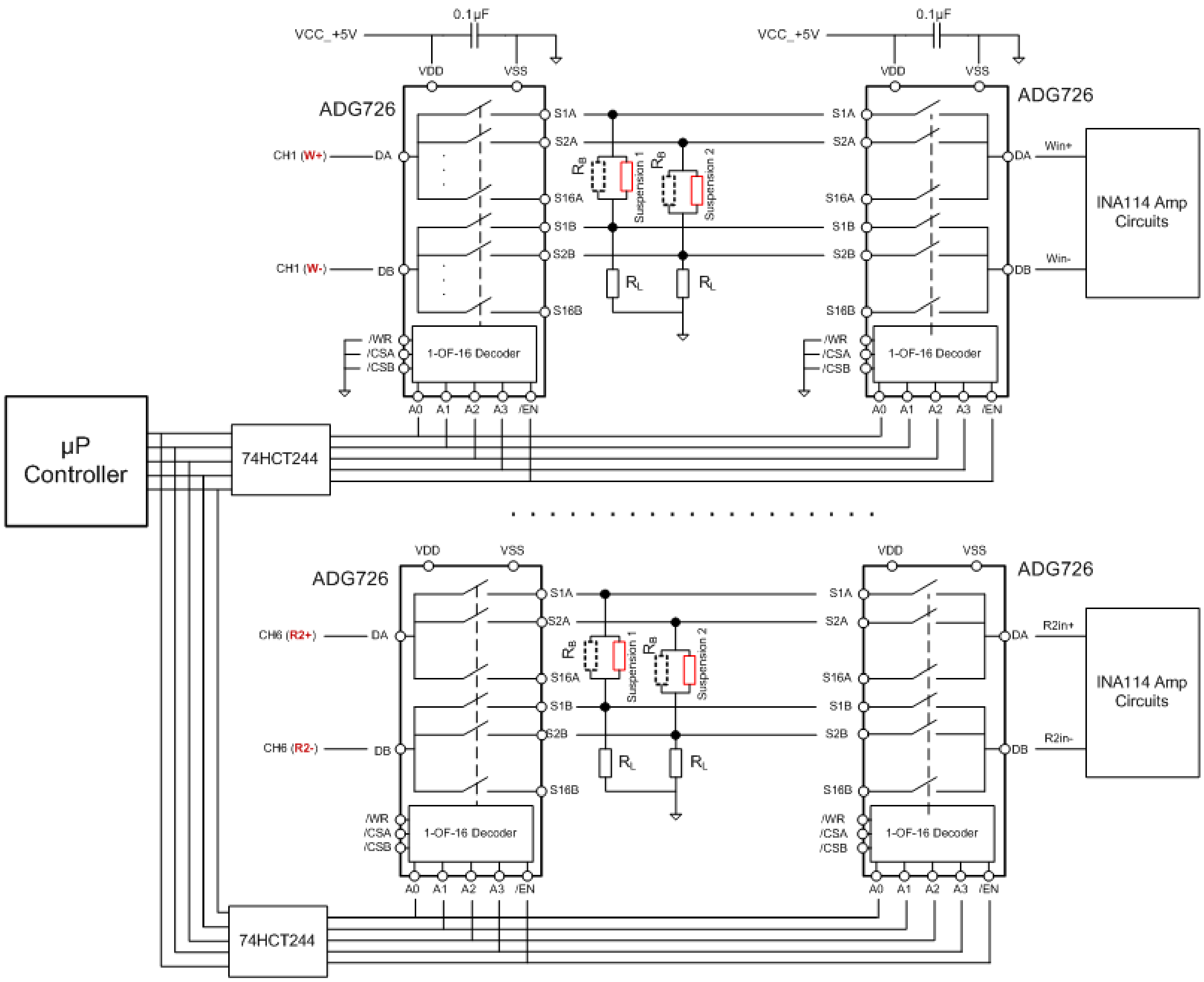


Figure 3.2.6 Multiplexer/Demultiplexer Circuit Design for HGA Resistance Measurements

ADG726 from Analog Device is used for the signal multiplexing and demultiplexing. The nominal On-Resistance is 4Ω at 10mA. In HGA application, its On-Resistance can be greater than 4Ω because of less current. But it is not critical for the HGA application. The switching between suspensions is controlled by the IO signals A0~A3. There are 12 ADG726 ICs in total. With consideration of signal driving strength from the uP controller, these A0~3 and /EN shall be split into two groups and tied together and connected to two bus driver 74HCT244 as shown above. Referring the datasheet and TRUE TABLE, /WR, /CSA and /CSB can be tied together to ground.

ADG726 is a 16-channel differential analog switch. Only 10 channels are used as shown in the Figure 3.2.6. The rest of 6 channels shall be connected to high precision resistors(0.1%) for calibration purposes in future. The recommended resistor values are: 0Ω (or the two terminals are short together), 10Ω, 100Ω, 500Ω, 1KΩ, 10KΩ.

3.2.6 Circuit Design for Accurate Temperature Measurements

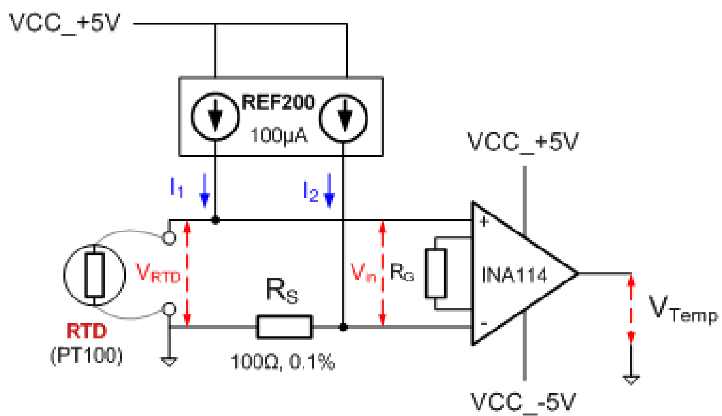


Figure 3.2.7 Temperature Measurement Circuits

As shown in the Figure 3.2.7, REF200 is standard high accurate current source which has two 100µA outputs. One provides 100µA current flowing through the RTD (PT100) and the other one provides 100µA flowing through a high accuracy 100Ω resistor (temperature coefficient < 15ppm). R_G is used to set the amplification gain G for the instrument amplifier INA114. V_{Temp} is the measured voltage for temperature which is to be connected to the ADC input. The nominal relationship between V_{RTD}, V_{IN} and V_{Temp} is as following:

$$V_{Temp} = G \times V_{in} = G \times (I_1 \times R_{RTD} - I_2 \times R_S) = G \times (I_1 \times R_{RTD0} - I_2 \times R_S + I_1 \times \Delta R_T) \qquad \text{(EQ.13)}$$

R_{RTD0} is the nominal resistance of PT100 at 0℃ which equals to 100Ω. ΔR_T is the PT100 resistance variation with respect to 0℃. In HGA application, the expected temperature to be measured is always above 0℃. Therefore, ΔR_T is always > 0. Considering I1=I2 and RS=100Ω, the Equation EQ.13 can be rewrite as below:

$$V_{Temp} = G \times I_1 \times \Delta R_T \qquad \text{(EQ.14)}$$

ΔR_T is about 0.4Ω/℃ and ADC resolution is about $2.5V/2^{20} \approx 2.4µV/bit$. To achieve 0.1℃ measurement accuracy, the proposed gain setting for INA114 can be set to 200, i.e.

$$Gain = 1 + \frac{50K\Omega}{R_G} \quad \text{or} \quad R_G \approx 250\Omega. \qquad \text{(EQ.15)}$$

The R_G shall have low temperature coefficient. Panasonic E96 series resistor is recommended. With this gain setting, the temperature range can be from 0℃ up to highest value which the PT100 can withstand.

To achieve high accuracy temperature measurements, eventually the whole circuit will be calibrated using high accuracy resistance decay in the production of lab. The wiring length for PT100 connection with the PCBA has to be taken into consideration.

3.2.7 Circuit Design for Sink/Source Configurable IOs

Figure 3.2.8 shows opto-isolated sink/source configurable Input circuit design. It is planned to have 4-in/4-out such IOs for connections with external devices. The 4 Inputs must be connected as all sourcing or all sinking.

Figure 3.2.9 shows the circuit design for opto-isolated sink/source configurable outputs. The 4 Outputs must be connected as all sourcing or all sinking.

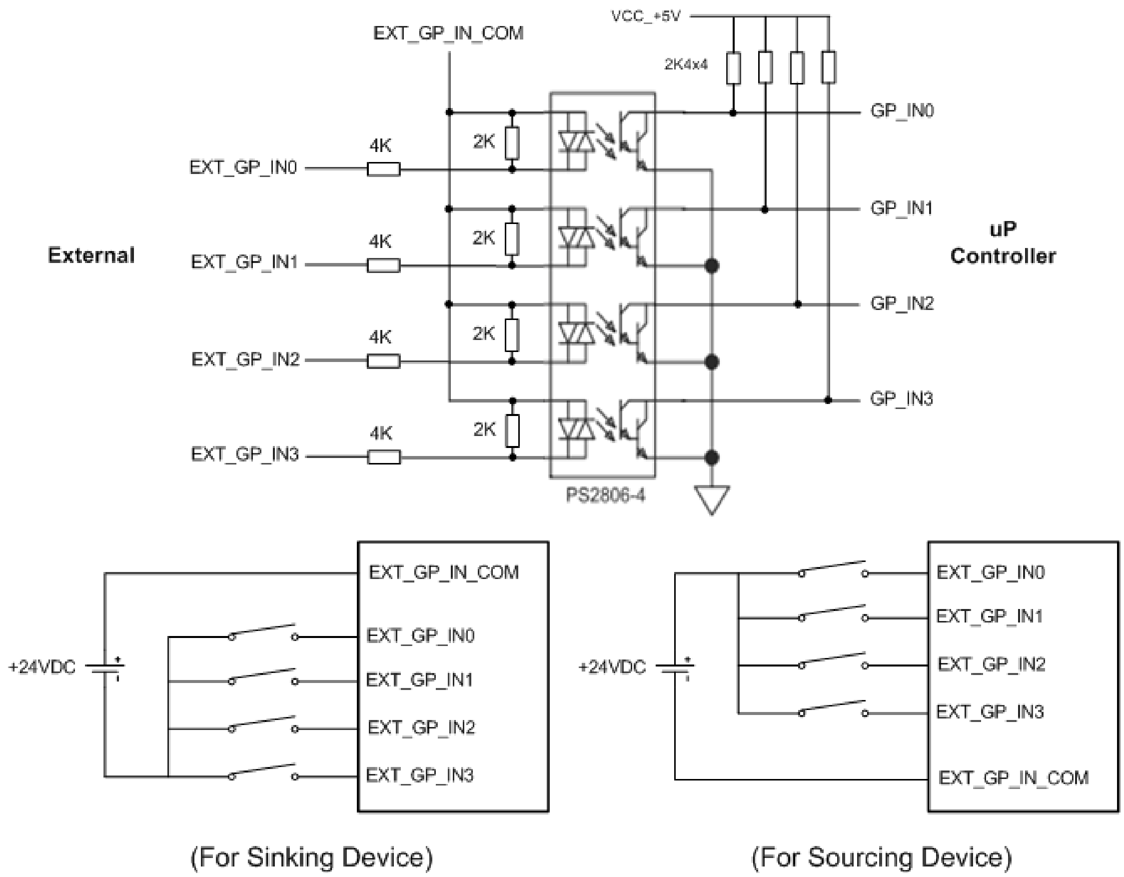
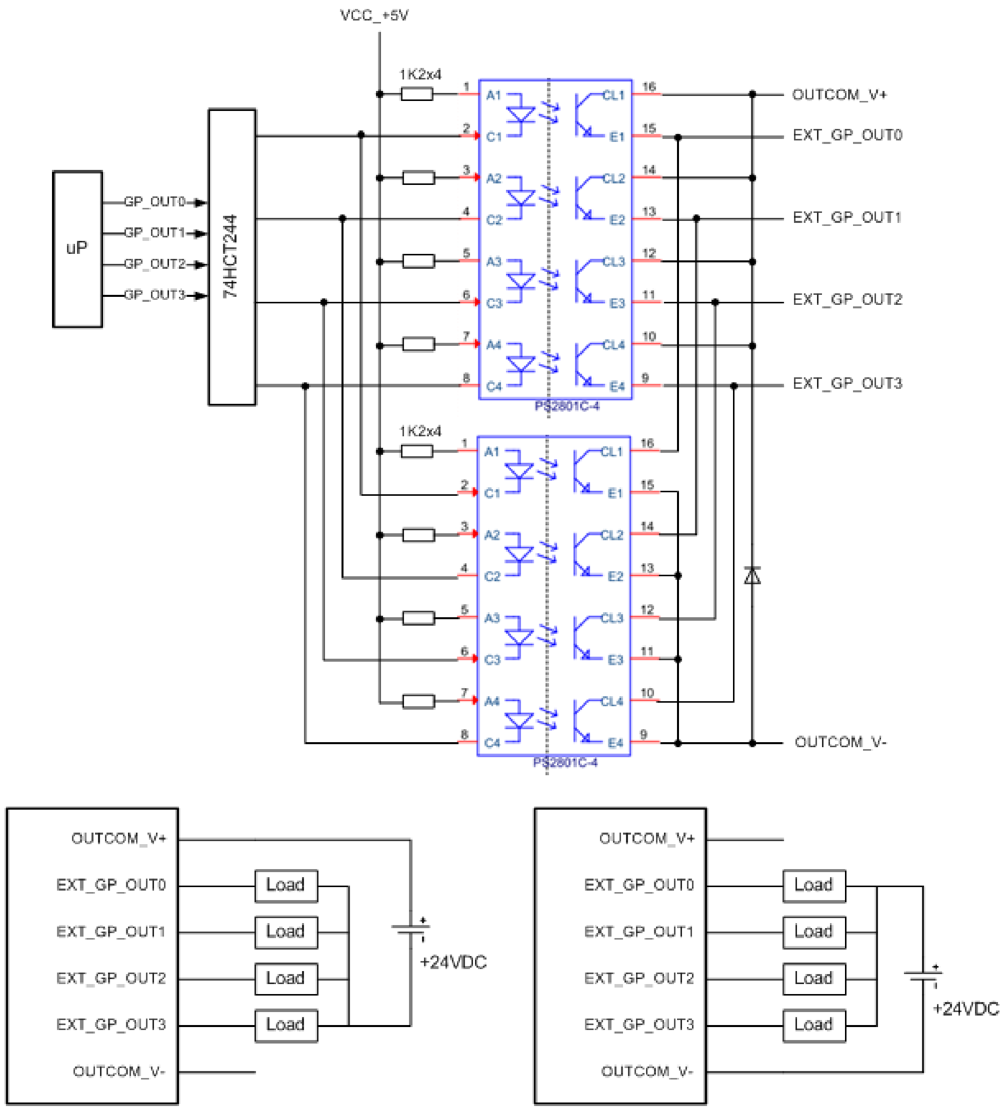


Figure 3.2.8 Circuit Design for Sink/Source Configurable Inputs



(For Current Sinking Load) (For Current Sourcing Load)

Figure 3.2.9 Circuit Design for Sink/Source Configurable Outputs

3.3 Circuit Calibration Algorithms and Tools

3.3.1 Algorithms for DAC Circuits Offset and Gain

Figure 3.3.1 shows the overview loop circuit design for resistance measurements. There are two major functional blocks to be calibrated, i.e. DAC current sourcing circuits and ADC voltage sampling circuits.

The purpose of this calibration is to cancel off the DC offset contributed from DAC, op-amps and related circuits and to build up actual relationship of DAC value vs. actual current.

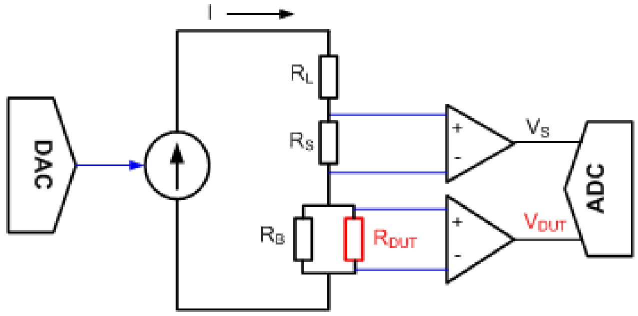


Figure 3.3.1 Loop Circuit for Resistance Measurements

From the theory of operation and analysis discussed early, as long as the current source is stable, the actual amount of sourcing current is less critical. Therefore, this calibration is performed prior to the ADC circuit calibration, i.e. the ADC circuits are assumed with their nominal design parameters.

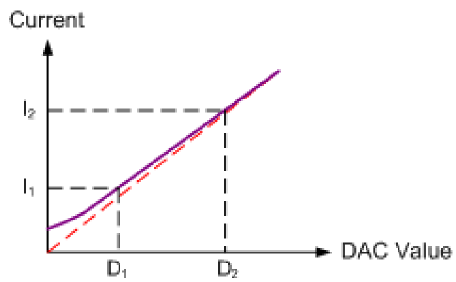


Figure 3.3.2 Method for DAC Current Source Calibration

As shown in the Figure 3.3.2, the basic calibration method and steps are described as below:

1. Referring to Section 3.2.5, switch DAC demultiplexer to channel 11 where the $R_{DUT} = 0\Omega$.
2. Calculate the two nominal DAC values D_1 and D_2 corresponding to two desired currents $I_{1desired}$ and $I_{2desired}$ according Equation 10 below:

$$DACSetting = \frac{65536 \times R_1 \times R_3}{R_2 \times V_{FullScale}} \times I_S$$

The selection of two desired currents must be within the limited current range set by the limiting resistor RL and RS. They are channel-dependent values.

3. Applying D_1 to the DAC and wait for 100ms. Then read back the ADC value on R_S as V_{1ADC}.
4. Applying D_2 to the DAC and wait for 100ms. Then read back the ADC value on R_S as V_{2ADC}.
5. Calculate the actual currents I_1 and I_2 as below:

$$I_1 = \frac{\frac{V_{1ADC}}{2^{24}} \times V_{REF}}{G_V \times R_S} = \frac{V_{1ADC} \times V_{REF}}{2^{24} \times G_V \times R_S} \quad \text{and} \quad I_2 = \frac{V_{2ADC} \times V_{REF}}{2^{24} \times G_V \times R_S}; \qquad \text{(EQ.16)}$$

Where V_{REF}=2.5V, G_V=2, R_S=100Ω.

6. Compare the calculated I_1/I_2 with their corresponding desired values $I_{1desired}/I_{2desired}$. If the deviation is >15%, the calibration is considered failure and reports error message to the Host PC.
7. Store the four values, i.e. D_1, I_1, D_2 and I_2, into EEPROM. I_1 and I_2 shall be stored as 32-bit signed integers in unit of µA.

Repeat the above calibration process for all six current-sourcing channels and we obtain a set of DAC current-source calibration data as (D_{1j}, I_{1j}, D_{2j}, I_{2j}) where j = 1, 2, 3, ..., 6.

After calibration, the user specified bias current for resistance measurements can be converted to DAC setting D_{User} as:

$$D_{User} = D_1 + \frac{D_2 - D_1}{I_2 - I_1} \times I_{User} \quad \text{(EQ.17)}$$

3.3.2 Calibrations for ADC Voltage Sampling Circuits

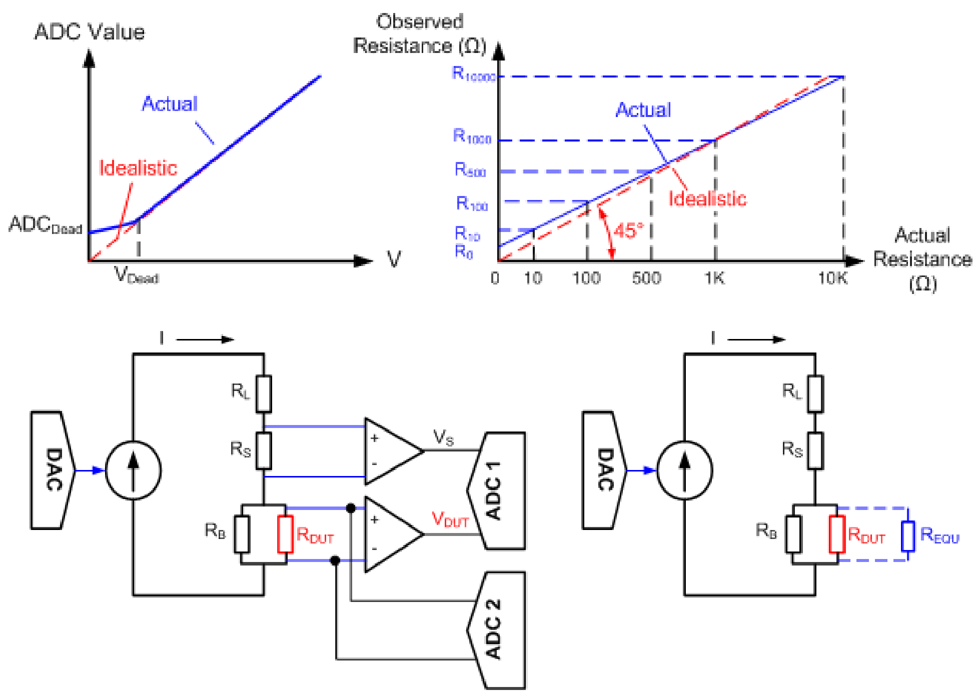


Figure 3.3.3 Calibration Method for ADC Sampling Circuits

Determination of ADC Dead (Nonlinear) Zone

As shown in Figure 3.3.3, the output ADC reading (after A/D conversion) is ideally always linearly proportional to input voltage. However, most of ADCs have typically a dead zone. In this dead zone, the ADC conversion value is no longer linear to the input voltage. This dead zone needs to be evaluated or calibrated to ensure that:

1. A proper biasing current (minimum threshold) is applied to DUT based on its expected nominal resistance value.
2. If necessary, a proper level-shift is applied to the output of instrument amplifier (INA114). As of now, ~150mV is designed for this level-shift.

This dead zone can be characterized by two parameters: ADC_{Dead} and V_{Dead}.

ADC_{Dead} is the ADC conversion value when input voltage is 0.
V_{Dead} is the voltage turning-point where the linearity (voltage vs. ADC value) error >1% (to be concluded).

The ADC_{Dead} can be found just by setting R_{DUT} and DAC current to 0 and read out the ADC conversion value.

The V_{DEAD} can be found with following steps:

1. Switch the multiplexer to channel 12 with built-in R_{DUT}=100Ω (0.1%).
2. Turn ON one of the six current sources and set the current to 400µA.
3. Wait for 100ms. Read out the ADC conversion value.
4. Reduce the current by 40µA and wait for 100ms.
5. Read out the ADC conversion value.
6. Calculate the current linearity L_{CUR} and averaged linearity L_{AVG} from all the previous measurement.
7. If the $|L_{CUR}-L_{AVG}|>0.01*|L_{AVG}|$, stop the calibration process and the current ADC value is set as V_{Dead}. Otherwise, repeat the Step 4~7.

For the 6 resistance measurement channels, we will have 6 sets of calibration results, i.e.

$(ADC_{Dead1}, V_{Dead1}), (ADC_{Dead2}, V_{Dead2}), . . . , (ADC_{Dead6}, V_{Dead6})$

These data are to be stored in EEPROM and they will be used for two purposes in future:

1. To evaluate the hardware design and AD7173-8 performance.
2. According to customer requirement, the DAC current is user-configurable. These data are used for verification if customer selected measurement current is acceptable or not based on nominal expected resistance value. In theory, higher measurement current yields better measurement accuracy.

Calibration for Resistance Measurements

For each measurement channel, there are 6 high precision (0.1%) resistors built-in on board, i.e. 0Ω, 10Ω, 100Ω, 500Ω, 1KΩ, 10KΩ. Switch the multiplexer to these known resistors one by one and set the DAC current source to 300µA for all six channels. Measure the voltages from sensing resistors and the built-in precision resistors. Then calculate observed resistance values according to the Equation (EQ. 9) in previous discussion.

Therefore, as shown in the Figure 3.3.3, we obtain 6 observed resistance values corresponding to the 6 built-in precision resistor values for each channel, i.e.

$(R_{0\_j}, R_{10\_j}, R_{100\_j}, R_{500\_j}, R_{1000\_j}, R_{10000\_j})$,   j=1, 2,..., 6. @ Bias Current = 300µA.

Then change the DAC current to 20mA, switch the multiplexer to built-in resistors 0Ω, 10Ω and 100Ω and perform measurements. We have another set of observed resistance values as below:

$(R_{0\_j}, R_{10\_j}, R_{100\_j})$,   j=1, 2, ..., 6 @ Bias Current = 20mA

The above two sets of calibration data are to be saved into EEPROM.

Ideally, R_{0j}, R_{10j}, ..., R_{10000j} shall be equal to 0, 10Ω, ..., 10KΩ. However, they are not. The measurement errors are mainly due to the resistance contributed by electrical noises, PCB traces, leakage currents flowing to multiplexer, ADC quantization errors, and etc.

.

Correction for the Measurement Results

In normal operation, for a specific resistance channel of suspension under test, its observed value will be linearly interpolated or extrapolated into actual resistance using one of above two calibrated sets of data as shown in the Figure 3.3.3. For channels of Reader 1, Reader 2, TA, Heater 1 and Heater 2, the first set of calibration data is used. For the Writer channel, the second set of data is used.

3.3.3 Calibrations for LCR Meter

A LCR meter IET-1920 is used for capacitance measurements. The 4-Kelvin cables are connected from LCR meter to the individual PZT channels via a Relay Array on the capacitance measurement PCB. Figure 3.3.4 shows the measurement path and concept of calibrations.

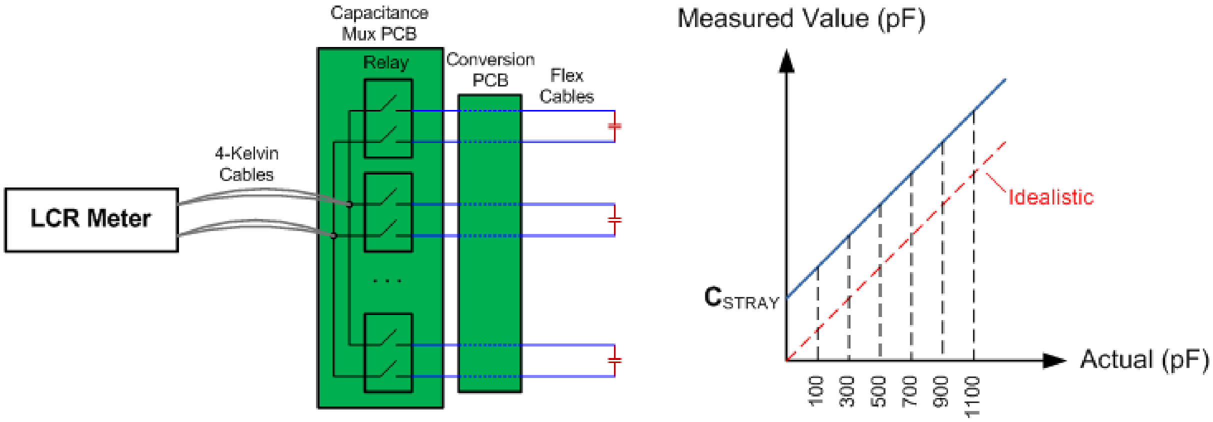


Figure 3.3.4 Capacitance Measurement Path and Concept of Calibrations

There are separate calibrations to be done in the system. One is the calibration of LCR meter itself and the other is the calibration for the measurement path.

Details of the calibration for LCR meter can be found in the “IET 1920 Precision LCR Meter User and Service Manual”. Basically short connect the 4-Kelvin cable terminals and then issue a calibration command to the LCR meter. It will automatically perform all internal calibrations and save the data internally.

Addition to the LCR calibration itself, a calibration is required to cancel off the stray capacitance introduced by the components (mainly relays) and PCB traces/cables along the measurement path as shown in the Figure 3.3.4 above. On the capacitance measurement PCB, there are 6 standard high precision capacitors (accuracy of 1%) are built-in. Their values are 100pF, 300pF, 500pF, 700pF, 900pF and 1100pF. Through software control, these capacitor’s values are measured respectively. Ideally these values shall be the same as their actual values. However, due to stray capacitances introduced along the measurement path, the measured values are expected to be slightly higher than their actual values. The differences are calculated as the stray capacitances which are to be saved in the EEPROM. The following two points need attention in the evaluation:

1. The stray capacitance C_{STRAY} can be measurement frequency-dependent. The higher frequency, the larger C_{STARY} value. Therefore, the measurement frequency for calibration shall be the same as what used for normal operation. The recommended measurement frequency is 60KHz (refer to Figure 2.5 above).
2. The proposed calibration method above does not actually include the stray capacitance contributed by the conversion PCB and flex-cable. For ease of implementation, this will be compensated with a fixed value in software. This fixed value is to be defined during prototype evaluations.

3.3.4 Development Tools and Maintenance Kits for Field Services

Development Tools

1. Renesas E30A Emulator
   This emulator is used for R32C117 µP firmware development and costs about USD$3000. The software IDE (Integrated Development Environment) tool can be free-downloaded from Renesas website.

2. Standard Resistor Decay Box
   This standard resistor decay is used for calibration of resistance measurements and RTD. The recommended part number is HARS-X-6-0.01 from IET lab. It has full range of 11.1111K with resolution of 0.01Ω. Below is the picture of this resistor decay.

3. Standard Precision Decade Capacitor
   Similar to Resistor Decay Box, this precision decade capacitor is used for calibration of capacitance measurement circuits. The recommended part number is IET 1413 Precision Decade Capacitor. It has maximum range of 0~1.11111µF, resolution of 1pF and accuracy of (±0.05%+0.5pF). Below is the picture.

Maintenance Kits

As shown in the Figure 3.3.5 below, a set of Pogo Pins are engaged with suspension test pads during measurements and dis-engaged after completion of measurements. These Pogo pins have typical life cycle about 500K and they are consumables. During the Pogo pin life span, a tool kit is necessary to verify the Pogo pin's performance. One of the key parameter checks is the Pogo pin's contact resistance. If the contact resistance is too much (or out spec, typically <100mΩ), a service shall be called to clean up the Pogo pins or request for replacement.

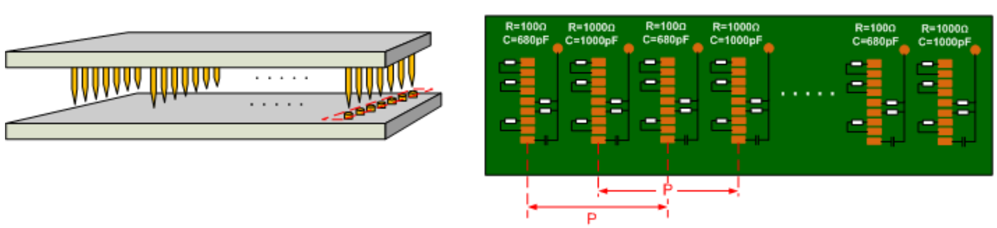


Figure 3.3.5 Verification Kits for Pogo Pins

To achieve the purpose, a calibration PCB is proposed as in the Figure 3.3.5 above. On this PCB, all pads in pair (corresponding to suspension test pads) are connected with either known precision resistors or known precision capacitors. This PCB is situated in a mechanical housing which can be manually clipped onto the Pogo Pins by Field Service Engineers (FSE). Once it is clipped onto the Pogo Pins, FSE can trigger verification process though GUI command interface. Basically, the uP controller measures each resistor and compare with its actual known value. Therefore, the Pogo Pin's contact resistance can be estimated. If it is out of spec, a service request will be generated to users.

3.4 uP Controller Interface Function and Pin Assignments

Renesas 32-bit microprocessor R32C117 is chosen for this HGA Static Tester measurement and control. It has processing capability up to 50 MIPS and rich choice of peripheral functions, such as GPIO, I2C, SPI, Timers, A/D, D/A, UART and etc. The IO pins can be configurable for multiple functional purposes, which provide much flexibility for different applications. It needs only one single +5V supply. Figure 3.4.1 shows its overview functional blocks.

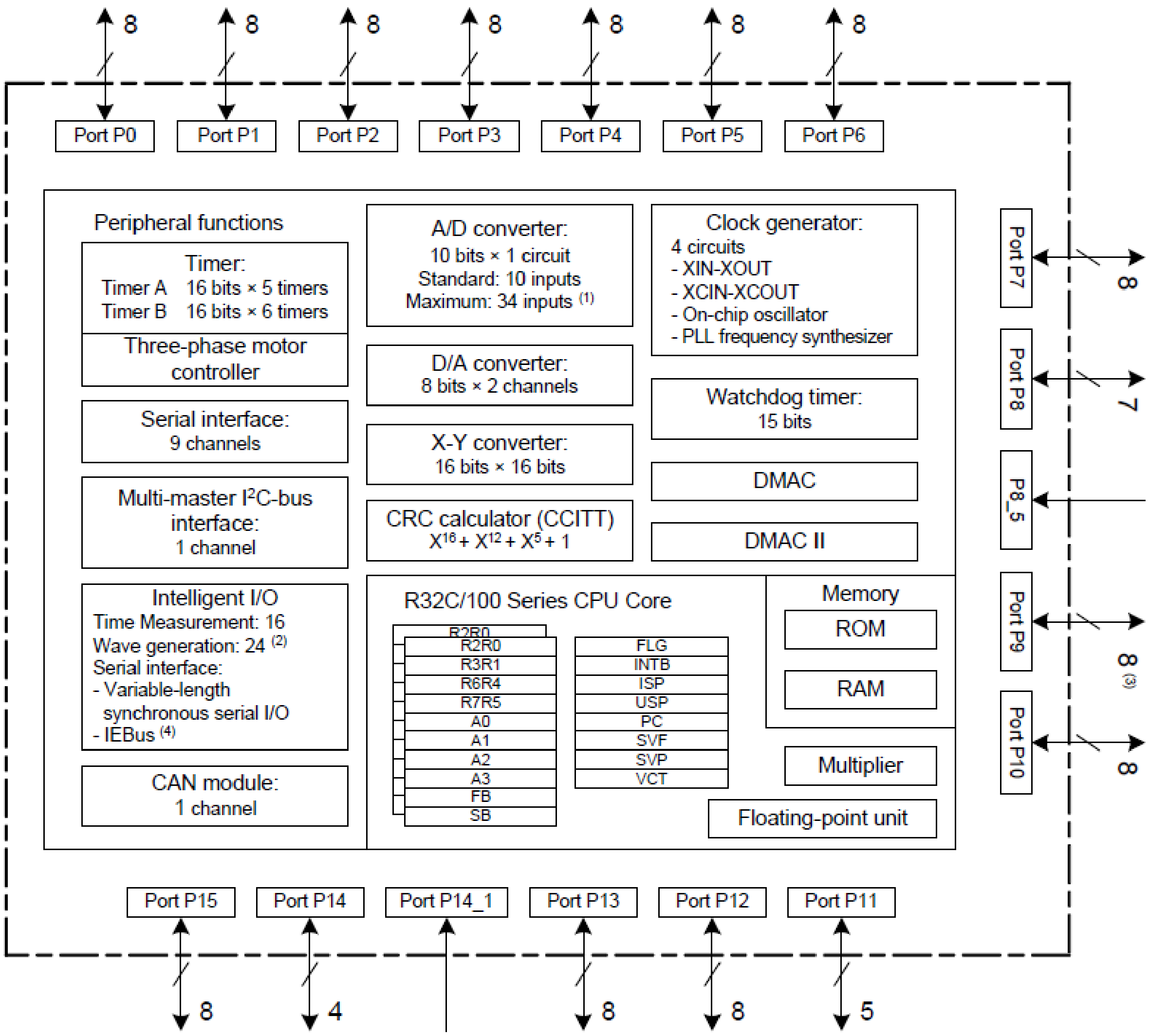


Figure 3.4.1 R32C117 Functional Block Diagram

3.4.1 uP Control and Interface Functional Description in HGA Tester

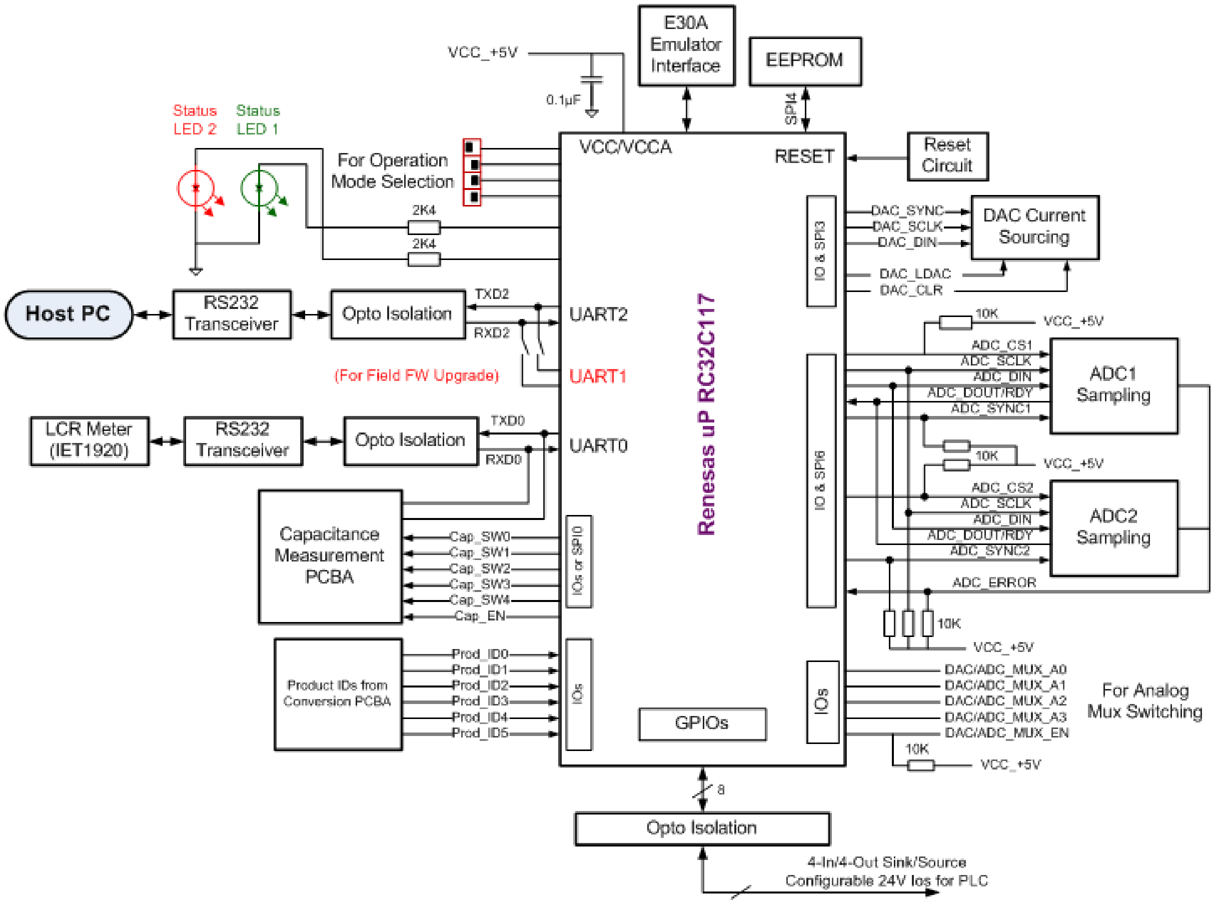


Figure 3.4.2 Overview of Functional Blocks for uP Control Circuits

The uP control circuits include multiple functional blocks as shown in the Figure 3.4.2 above. They are described as below.

Emulator Interface for Firmware Development

For Renesas R32C117 microprocessor, an E30A emulator is used for firmware development and it needs a dedicated serial interface and SMB (Coaxial Cable) connector. The IDE for firmware is called "High-Performance Embedded Workshop", which can be free downloaded from Renesas website.

Operation Mode Selection and Status Indication

There is a 4-way DIP switch or jumpers reserved for user to configure different operation modes such as self-test, normal operation with LCR meter, normal operation with on-board capacitance measurement, and etc. When mode setting is changed, it becomes effective only after power-on or manual reset.

Besides the mode selection, there are two LEDs reserved for command execution status indication. Details are to be defined later.

Host PC Communication Interface and Field Firmware Upgrade

As shown in the Figure 3.4.2 above, UART2 in RC32C117 uP is used for the RS232 communication with Host PC. They are electrically isolated from each via opto-coupler. The RS232 shall be hardware configured as DCE (Data Carrier Equipment) to the Host PC. The connector shall be standard DB-9 female. Communication signals are TXD, RXD, CTS and RTS. CTS and RTS are used for data flow control (to prevent buffer overflow or under-flow. If the Host PC has buffer size >=256 bytes for RS232, the RTS and CTS can be optional. Typically, the data packet size for HGA will be <256 bytes.

External on-board jumpers are available to break the connection between UART2 and RS232 transceiver; and then direct the UART1 to the RS232 transceiver. With the application software provided by Renesas, a Firmware update can be downloaded from Host PC into the uP if necessary. This Firmware Upgrade can be done any time on the field and no additional hardware tools are required.

Interface with LCR Meter and Capacitance Measurement PCBA

For this HGA prototype design, there are scenarios planned for capacitance measurements. Scenario 1 is to use standard LCR meter and Scenario 2 is to use a dedicated self-developed capacitance measurement circuits.

Scenario 1: To use LCR Meter (IEC1920)

In this scenario, the UART0 is reserved for the RS232 communication with the LCR meter. Before starting the HGA machine for normal operation, first the user selects the capacitance measurement configuration parameters from Host PC GUI. Then these configuration data are downloaded to the Renesas uP. After that, the uP downloads these parameters to the LCR meter. Once the normal operation is started, the uP will command the LCR to start/stop capacitance measurements after channel-switching.

The capacitance measurement PCBA consists of mainly an array of solid-state relays. These relays perform the channel switching between suspensions based on the synchronization switching IO signals (Cap_SW0~4, Cap_SW_EN) from the uP. The uP shall control the switching signals such that no resistance and capacitance are measured on the same suspension. This is just to avoid interference or cross-talk noises from LCR meter while performing resistance measurements.

Cap_SW_EN signal is just meant for disconnection between the LCR meter to any of the suspensions. It may be redundant.

Scenario 2: To use self-developed Capacitance Measurement PCBA

Rather than LCR meter, a self-developed capacitance measurement PCBA is used. In this Scenario 2, the UART0 will be reconfigured through firmware as 3-wire SPI interface for data communication between the capacitance measurement PCBA and uP. CAP_SW0~1 functioning as SCLK and CS signals for SPI. Other IOs are to be redefined.

Product Conversion PCB ID Detection

As shown in the Figure 2.16, 6-bit of Inputs are reserved for product conversion PCB ID detection. It has 64 different IDs for different suspension products. These IDs are mainly used to differentiate products with flying lead test pads in following parameters:

1. The number of test pads.
2. The test pad pitch and width.
3. The test pad assignment.
4. The suspension dimensions and layouts.

Rules for ID determination are defined as below:

1. A different ID shall be applied if any one of the above 4 parameters is different.
2. The long tail and short tail in the same suspension family shall use different ID.
3. The UP/DOWN of the same suspension can share the same ID.

Interface with DAC Current Sourcing

SPI3 of R32C117 uP is assigned for the SPI 3-wire communication with the DAC (AD5668-2). Two more outputs DAC_LDAC and DAC_CLR are used for fast load or clear the DAC outputs.

Interface with ADC Voltage Sampling

SPI6 of RC32C117 uP is used for the communication between uP and two ADC convertors. Two separate Chip Select outputs (ADC_CS1 and ADC_CS2) are used in order to share the same SPI6 interface. Additional IOs are used for ADC_CS1, ADC_SYNC1, ADC_CS2, ADC_SYNC2 and ADC_ERROR.

Switching Control Signals for Multiplexer/Demultiplexer

There are two groups IO signals for DAC demultiplexing and ADC multiplexing respectively. DAC_MUX_A0~3 are for DAC and ADC_MUX_A0~3 for ADC. Besides, two enable signals, DAC_MUX_EN for DAC and ADC_MUX_EN for ADC.

In principal, these two groups can be combined as one group. Since the number of IOs from R32C117 is enough, we use two separate group s of IOs to have more flexibility in HGA application.

Sink/Source Configurable IOs

There are 4-in/4-out sink/source configurable +24V IOs available for connection with external PLC controller. A possible scenario for this IO usage is to use an Input signal from PLC for start/stop measurement and one output to PLC to indicate the completion of measurements.

3.4.2 uP Pin Assignments and Descriptions

Pin No.	Control Pin	Port	HGA Application Pin Assignment	Remarks
1		P9_4	EEPROM_CS	SPI Chip Select for EEPROM, Active Low.
2		P9_3	EEPROM_WP	Write Protection for EEPROM, Active Low.
3	VDC0		VDC0	
4		P9_1	PWR_GOOD	Power Good Status from power monitoring IC
5	VDC1		VDC1	
6	NSD		E30_DEBUG	Reserved for E30A emulator
7	CNVSS		E8_CNVSS	Reserved for E8A emulator
8	XCIN	P8_7		
9	XCOUT	P8_6		
10	RESET		nRESET	Reset input, active LOW.
11	XOUT		XOUT	
12	VSS		VSS	
13	XIN		XIN	
14	VCC		VCC	
15		P8_5		
16		P8_4		
17		P8_3	OP_MODE3	Connect to external DIP switch for operating mode selection.
18		P8_2	OP_MODE2	
19		P8_1	OP_MODE1	

20		P8_0	OP_MODE0	
21		P7_7		
22		P7_6		
23		P7_5		
24		P7_4		
25		P7_3	HOST_CTS	CTS for Host Communication
26		P7_2	HOST_RTS	RTS for Host Communication
27		P7_1	HOST_RXD	RS232 communication for Host PC
28		P7_0	HOST_TXD	RS232 communication for Host PC
29		P6_7	E8_TXD	
30		P6_6	E8_RXD	
31		P6_5	E8-SCLK	
32		P6_4	E8_BUSY	
33		P6_3	LCR_TXD	UART TXD0 for LCR meter, or FPGA_DIN (to FPGA) in SPI mode.
34		P6_2	LCR_RXD	UART RXD0 for LCR meter, or FPGA_DOUT (from FPGA) in SPI mode.
35		P6_1	LCR_RTS	RTS for LCR meter, or FPGA_SCLK in SPI mode.
36		P6_0	LCR_CTS	CTS for LCR meter, or FPGA_CS in SPI mode.
37		P5_7		
38		P5_6		
39		P5_5	E8_EPM	
40		P5_4	ADC_SYNC2	Sync output for ADC2
41		P5_3	ADC_SYNC1	Sync output for ADC1
42		P5_2	ADC_ERROR	ADC Error Input
43		P5_1	ADC_CS2	Connected to AD7173 ADC2 as CS pin for serial communication.
44		P5_0	E8_nCE	
45		P4_7	ADC_DIN	Connected to AD7173 DIN input pin for SPI serial communication.
46		P4_6	ADC_DOUT	Connected to AD7173 DOUT output pin for SPI serial communication.
47		P4_5	ADC_SCLK	Connected to AD7173 SCLK input pin for SPI serial communication.
48		P4_4	ADC_CS1	Connected to AD7173 ADC1 as CS pin for serial communication.
49		P4_3	DAC_DIN	Connected to AD5668 DAC as DIN pin for serial communication.
50		P4_2	DAC_DOUT	Reserved as DOUT from DAC data output pin.
51		P4_1	DAC_SCLK	Connected to AD5668 DAC as SCLK pin for serial communication.
52		P4_0	DAC_SYNC	Connected to AD5668 DAC as SYNC pin for serial communication.
53		P3_7	DAC_LDAC	Connected to AD5668 DAC as LDAC signal.
54		P3_6	DAC_CLR	Connected to AD5668 DAC as CLR signal.
55		P3_5	PRD_ID5	Conversion PCB ID input 5
56		P3_4	PRD_ID4	Conversion PCB ID input 4
57		P3_3	PRD_ID3	Conversion PCB ID input 3
58		P3_2	PRD_ID2	Conversion PCB ID input 2

59		P3_1	PRD_ID1	Conversion PCB ID input 1
60	VCC		VCC	
61		P3_0	PRD_ID0	Conversion PCB ID input 0
62	VSS		VSS	
63		P2_7	DEBUG_PAD2	Test Pads for debugging purposes.
64		P2_6	DEBUG_PAD1	
65		P2_5	DEBUG_PAD0	
66		P2_4	DAC_MUX_EN	Analog/DEMUX Switch Enable Signal for DAC
67		P2_3	DAC_MUX_A3	Analog/DEMUX Switch A3 for DAC
68		P2_2	DAC_MUX_A2	Analog/DEMUX Switch A2 for DAC
69		P2_1	DAC_MUX_A1	Analog/DEMUX Switch A1 for DAC
70		P2_0	DAC_MUX_A0	Analog/DEMUX Switch A0 for DAC
71		P1_7	GP_OUT3	Sink/Source configurable outputs
72		P1_6	GP_OUT2	
73		P1_5	GP_OUT1	
74		P1_4	GP_OUT0	
75		P1_3	GP_IN3	Sink/Source configurable inputs
76		P1_2	GP_IN2	
77		P1_1	GP_IN1	
78		P1_0	GP_IN0	
79		P0_7	STATUS_LED2	Connect to external Red LED for operating status indication.
80		P0_6	STATUS_LED1	Connect to external Green LED for operating status indication.
81		P0_5	CAP_SW_EN	CAP_SWx enabled or latch signal to prevent glitches.
82		P0_4	CAP_SW4	Capacitance Relay Switch 4
83		P0_3	CAP_SW3	Capacitance Relay Switch 3
84		P0_2	CAP_SW2	Capacitance Relay Switch 2
85		P0_1	CAP_SW1	Capacitance Relay Switch 1
86		P0_0	CAP_SW0	Capacitance Relay Switch 0
87		P10_7	OFFSET_CAL_EN	ADC offset calibration enable for sensing resistance measurement channels.
88		P10_6		
89		P10_5		
90		P10_4	ADC_MUX_EN	Analog/MUX Switch Enable Signal for ADC
91		P10_3	ADC_MUX_A3	Analog/MUX Switch A3 for ADC
92		P10_2	ADC_MUX_A2	Analog/MUX Switch A2 for ADC
93		P10_1	ADC_MUX_A1	Analog/MUX Switch A1 for ADC
94	AVSS		AVSS	
95		P10_0	ADC_MUX_A0	Analog/MUX Switch A0 for ADC
96	VREF		VREF	
97	AVCC		AVCC	
98		P9_7	EEPROM_DOUT	SPI Data Output from EEPROM
99		P9_6	EEPROM_DIN	SPI Data Input to EEPROM
100		P9_5	EEPROM_SCLK	SPI SCLK for EEPROM

4. Host Communication and API Commands

4.1 Overview of System Communication Interfaces

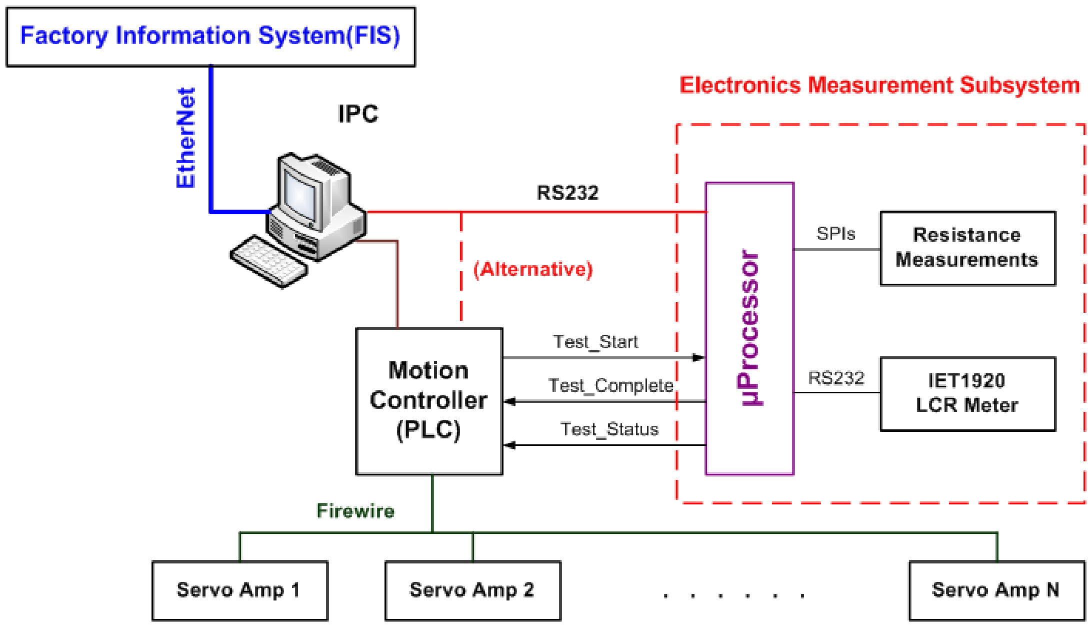


Figure 4.1 Overview of Communication Interfaces in HST Static Tester Design

Figure 4.1 shows an overview of the communication interfaces in HST Static Tester design. The IPC (Industrial PC, also referred as Host PC in this document) acts as a Host/Master for the communication and sequence control of the whole HST machine. On one side, it communicates via RS232 to the electronics measurement subsystem for measurement configuration, results and etc. It also talks to the Motion Controller (tentatively from AeroTech) for motion configuration and sequence control via local Ethernet. On the other side, the IPC communicates with the FIS (Factory Information System) for product configuration and measurement results. Depending on the final system design, the functionalities of IPC can be potentially integrated into the Motion Controller. In this case, the motion controller will communicate with the Electronics Measurement Subsystem via RS232.

In the initial concept and feasibility study, AeroTech A3200 is selected as the motion controller for this HST machine. The main task for A3200 is to control all servo motions via Firewire for multi-axis in the HST, including pneumatic valves, cylinders, IOs and etc. Additionally, it also controls the start/stop for electronics measurements. This is mainly because that it has the latest status information about the Pogo Pins position, i.e. engagement/disconnects with the suspension test pads. Three IO signals are defined for this purpose.

Test_Start: Active High. When the Pogo Pins are engaged with the suspension pads, A3200 shall set this signal to High for at least 10ms to start the electronics measurements. This is a positive-edge sensitive signal.

Test_Complete: Active High. When the electronics measurement subsystem receives the test start command, it immediately set this signal to Low and then starts the measurements. Upon completion of the measurements, this signal will set to High and stays at High until next measurement command is received.

Test_Status: This is status indication signal set by the measurement subsystem upon completion of measurements. When it is High, it indicates that the measurement is successful; Low indicates measurement error. The detailed measurement results can be read through RS232 by the Host PC.

Alternatively, the measurement start/stop can be also controlled via RS232 interface commands from the Host PC. This option is available by proper setting of Operating Mode Switch (SW2) on the uP Measurement PCBA.

4.2 RS232 Communication between Host PC and µP

The RS232 between Host PC and uP on the electrical measurement PCBA meets all industry EIA/TIA-232E and CCITT V.28 specifications. The on-board drivers maintain the ±5V EIA/TIA output signal levels at data rates in excess of 120kbps when loaded in accordance with the EIA/TIA-232E specification. In HST design, four signals are used for full-duplex data communication, TXD, RXD, CTS and RTS.

The CTS and RTS are used mainly used for hand-shaking and data-transfer flow control depending on data buffer size. For uP on measurement PCBA, it has enough bandwidth and response speed to prevent buffer overflow. RTS will be always active High.

For Host PC, typical buffer size is about 256 bytes. If the command data packet size is always below 256 bytes, the CTS and RTS for flow control can be ignored.

The RS232 port parameters shall be set as below:

Baud Rate: 19200; Data Bits: 8; Stop Bit: 1; Parity: None.

With these settings, the data transmission time per byte (8bits) is about 0.52ms.

4.3 Host Communication Protocol

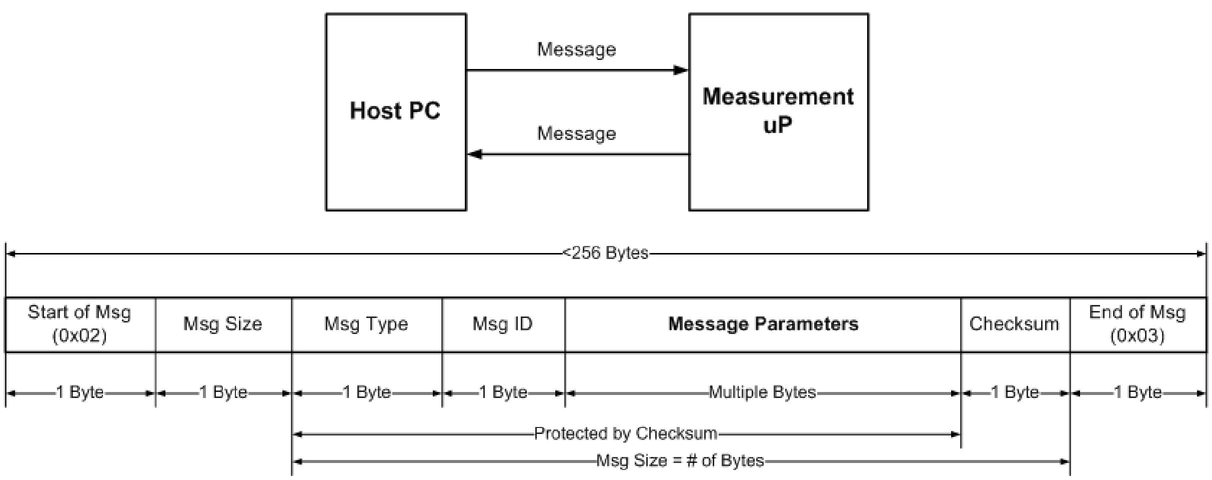


Figure 4.2 Communication Protocol between Host PC and Measurement uP

Communication between Host PC and Measurement uP follows typical Client-Server topology. Any one (Client) of them can initiate a communication request through messaging to the other side (Server). Upon receiving of the message, Server will first verify the data integrity of message through checksum. If it is a valid message (aka “command”), Server will response accordingly based on the Message ID (aka “Command ID”).

In most of cases, the Host initiates communications with the measurement uP. For every message from Host, the uP will send an acknowledgement message back to the Host. The measurement uP initiates a communication only in the situation when requested measurements are completed. The start of measurement request can be either from Host via RS232 command or from Motion Controller via IO signal (as shown in the Figure 4.1).

The rationale for the Client-Server topology is mainly to prevent continuous status polling while the Host is waiting for the measurement completion and results from uP.

No message/command queuing is allowed/supported in this communication protocol. New message or command is allowed to send out only after receiving the acknowledge message for previous command.

A message/command can be acknowledged with Error if the uP is busy with previous command.

The messaging protocol is defined as in Figure 4.3. A message always starts with the first byte equals to 0x02 (Start of Message) and ends with the last byte equals to 0x03 (End of Message). This is meant for the convenience of software message framing. Message bytes between “Start of Message” and “End of Message” are data bytes (not ASCII codes). Therefore, some of their contents can be 0x02 or 0x03. However, they are not the indication of “Start of Message” or “End of Message”. This has to been taken care of in Software Programming.

Table 4.1 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.

All the above messages are transmitted in Binary Data rather than ASCII code. For multiple bytes transmission, the LSB (Least Significant Byte) is always transmitting first and MSB (Most Significant Byte) last. Within one byte, the lsb (least significant bit) is the first and msb (most significant bit) last.

4.4 Host API Commands and Descriptions

This section of document defines all commands initiated by the Host PC to the measurement uP. The command parameters, expected behaviours and acknowledgement messages are explained in details.

4.4.1 HST_get_status

This command is intended for the Host to get current status of electronics measurement subsystem. The acknowledgment from the subsystem can be READY, BUSY or ERROR.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 1
Message Body = None

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 1

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.

4.4.2 HST_config_res_meas

This command is to configure bias currents for resistance measurements.
As discussed in the hardware design, there are 6-channels for resistance measurements: (W+, W-), (TA+, TA-), (wH, Gnd), (rH, Gnd), (R1+, R1-) and (R2+, R2-). Depending on product type, the bias current can be set to a desired value by user. If the bias current is set to 0, the channel is considered as being disabled.

Another parameter to be configured is the number of samples for averaging. When a non-zero value is configured, the uP will continuously read measurement inputs for the specified number of samples and then take the average as final result.

Command Message from Host

Message Size = 16
Message Type = 1
Message ID = 2

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Ch1 Bias Current (LSB)	Bias current for Writer Resistance Measurements.

2	Ch1 Bias Current (MSB)	When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of µA. By default, it is set to 20000µA.
3	Ch2 Bias Current (LSB)	Bias current for TA Resistance Measurements. When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of µA. By default, it is set to 300µA.
4	Ch2 Bias Current (MSB)	
5	Ch3 Bias Current (LSB)	Bias current for Write Heater Resistance Measurements. When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of µA. By default, it is set to 6000µA.
6	Ch3 Bias Current (MSB)	
7	Ch4 Bias Current (LSB)	Bias current for Read Heater Resistance Measurements. When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of µA. By default, it is set to 6000µA.
8	Ch4 Bias Current (MSB)	
9	Ch5 Bias Current (LSB)	Bias current for Reader1 Resistance Measurements. When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of µA. By default, it is set to 300µA.
10	Ch5 Bias Current (MSB)	
11	Ch6 Bias Current (LSB)	Bias current for Reader2 Resistance Measurements. When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of µA. By default, it is set to 300µA.
12	Ch6 Bias Current (MSB)	
13	# for Average	Number of samples for average. 0~64. 8-bit Unsigned Value. Default value is 4.

Acknowledge Message from Measurement uP

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

Definition of Message Parameters

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

4.4.3 HST_config_cap_meas

This command is for user to configure basic LCR parameters for capacitance measurements, i.e. frequency, bias voltage, peak-to-peak voltage and number of samples for averaging.

Command Message from Host

Message Size = 11
Message Type = 1
Message ID = 3

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Frequency (LSB)	It is the LCR meter signal frequency for capacitance measurements. When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of 10Hz. By default, it is set to 6000(60KHz).
2	Frequency (MSB)	
3	Bias Voltage (LSB)	Bias voltage for the capacitance measurement signal. 16-bit Unsigned Value, in unit of mV. By default, it is set to 0.
4	Bias Voltage (MSB)	

5	Peak Voltage (LSB)	Peak-to-Peak voltage for the capacitance measurement signal. When set to 0, the measurement is automatically disabled. 16-bit Unsigned Value, in unit of mV. By default, it is set to 1000mV.
6	Peak Voltage (MSB)	
7	Measurement Mode	For LCR meter, there are two measurement modes for user selection: Serial or Parallel. 0 – Serial Mode; 1 – Parallel Mode. Other values are reserved. Unsigned 8-bit, default value = 0, i.e. Serial Mode.
8	# for Average	Number of samples for average. 0~64. 8-bit Unsigned Value. Default value is 4.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 3

Definition of Message Parameters

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

4.4.4 HST_config_short_detection

This command is for user to configure the pairs which require short detection. As shown in the Figure 4.3 below, it is pairing of short-circuit detection for GrenadaBP2 product.

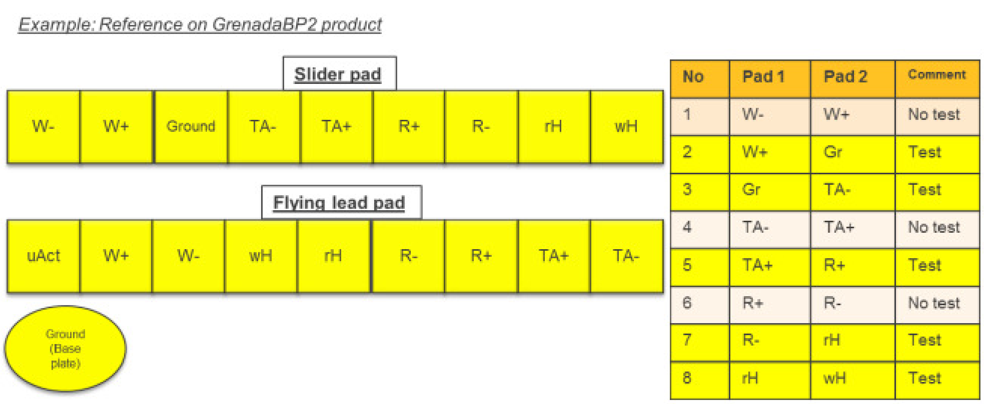


Figure 4.3 Test Pad Pairing for Short-Circuit Detection with GrenadaBP2 Product

On the Fying Lead, the five pairs (W+, W-), (wH, Gnd), (rH, Gnd), (R+, R-) and (TA+, TA-) have been tested with their resistances. If there is any short, open or other failure, they can be detected with the resistance measurement results.

Looking at the Slider Pads, there are potential short-circuit failures between the five pairing of adjacent slider pads, i.e. (W+, Gnd), (Gnd, TA-), (TA+, R+), (R-, rH), (rH, wH). These have to be tested.

Depending on slider layout design, these pairing for short-circuit detection can be different from product to product. In hardware design, all these pads are connected to an ADC and their individual potential voltages can sampled. If any paired pads are short-circuit together, they will have the same ADC sampled values. This is a “soft-way” for short-circuit detection. It is able to handle any paring combination between these connected pads.

With the consideration for future products, there are 12 different pads designed in HST electronics circuits. User has to configure the desired pairing for short-circuit detection. Table below shows the concept that how this pairing is configured by user.

Concept of Pairing Configuration

Msg Param Index #	Test Pads	Test Pads for Pairing (Parameter)												
		0	1 (W+)	2 (W-)	3 (TA+)	4 (TA-)	5 (wH+)	6 (wH-)	7 (rH+)	8 (rH-)	9 (R1+)	10 (R1-)	11 (R2+)	12 (R2-)
1	W+							√						
2	W-	√												
3	TA+										√			
4	TA-							√						
5	wH+								√					
6	wH-	√												
7	rH+						√							
8	rH-	√												
9	R1+				√									
10	R1-								√					
11	R2+	√												
12	R2-	√												

Notes:
1. "√" stands for the default setting for GrenadaBP2 product.
2. "0" stands for No Pairing, which means no short-circuit detection is required.
3. For GrenadaBP2,
   R2+ and R2- do not exist. rH- and wH- are combined as one and connected to the "Ground" pad through product conversion PCBA..
4. One pad can be paired with two adjacent pads, for example, wH- and rH+ in Table above.
5. For ease of result reporting to Host PC (refer to command "HST_get_short_detection"), only one pairing is allowed in a row as shown in the above table.

Command Message from Host

Message Size = 15
Message Type = 1
Message ID = 4

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	W+ Pairing	0 – No Pairing. 1 – Pairing with W+. 2 – Pairing with W-. 3 – Pairing with TA+. 4 – Pairing with TA-. 5 – Pairing with wH+. 6 – Pairing with wH-. 7 – Pairing with rH+. 8 – Pairing with rH-. 9 – Pairing with R1+. 10 – Pairing with R1-. 11 – Pairing with R2+. 12 – Pairing with R2-. Others – Reserved. Default Value: 6 (for GrenadaBP2).
2	W- Pairing	Refer to W+ Pairing. Default Value: 0.
3	TA+ Pairing	Refer to W+ Pairing. Default Value: 9.
4	TA- Pairing	Refer to W+ Pairing. Default Value: 6.

5	wH+ Pairing	Refer to W+ Pairing. Default Value: 7.
6	wH- Pairing	Refer to W+ Pairing. Default Value: 0.
7	rH+ Pairing	Refer to W+ Pairing. Default Value: 5.
8	rH- Pairing	Refer to W+ Pairing. Default Value: 0.
9	R1+ Pairing	Refer to W+ Pairing. Default Value: 3.
10	R1- Pairing	Refer to W+ Pairing. Default Value: 7.
11	R2+ Pairing	Refer to W+ Pairing. Default Value: 0.
12	R2- Pairing	Refer to W+ Pairing. Default Value: 0.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 4

Definition of Message Parameters

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

4.4.5 HST_meas_channel_enable

Referring to HST_config_res_meas command, there are 6-channel of resistance measurements and 2-channel of capacitance measurements for each HGA. For different products, the number of channels to be measured is different. For GrenadaBP2, R2 channel for resistance and C2 channel for capacitance are disabled.

For resistance measurement, if a channel is disabled, its corresponding DAC bias current is set to 0.

This command is for user to enable or disable channel measurements for a specific product.

Command Message from Host

Message Size = 11
Message Type = 1
Message ID = 5

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Resistance CH1, Writer	0 – Disabled. 1 – Enabled. Default Value = 1.
2	Resistance CH2, TA	0 – Disabled. 1 – Enabled. Default Value = 1.
3	Resistance CH3, Write Heater	0 – Disabled. 1 – Enabled. Default Value = 1.
4	Resistance CH4, Read Heater	0 – Disabled. 1 – Enabled. Default Value = 1.
5	Resistance CH5, Read 1	0 – Disabled. 1 – Enabled. Default Value = 1.

6	Resistance CH6, Read 2	0 – Disabled. 1 – Enabled. Default Value = 0.
7	Capacitance CH1, uACT1	0 – Disabled. 1 – Enabled. Default Value = 1.
8	Capacitance CH2, uACT2	0 – Disabled. 1 – Enabled. Default Value = 0.

Acknowledge Message from Measurement uP

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

Definition of Message Parameters

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

4.4.6 HST_hga_enable

This HGA Static Test Machine is designed to test 10 HGAs at one time. This command is for user to enable or disable the tests on specific HGAs when they are not available on the precisor.

Command Message from Host

Message Size = 13
Message Type = 1
Message ID = 6

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	HGA 1	0 – Disabled. 1 – Enabled. Default Value = 1.
2	HGA 2	0 – Disabled. 1 – Enabled. Default Value = 1.
3	HGA 3	0 – Disabled. 1 – Enabled. Default Value = 1.
4	HGA 4	0 – Disabled. 1 – Enabled. Default Value = 1.
5	HGA 5	0 – Disabled. 1 – Enabled. Default Value = 1.
6	HGA 6	0 – Disabled. 1 – Enabled. Default Value = 1.
7	HGA 7	0 – Disabled. 1 – Enabled. Default Value = 1.
8	HGA 8	0 – Disabled. 1 – Enabled. Default Value = 1.
9	HGA 9	0 – Disabled. 1 – Enabled. Default Value = 1.
10	HGA 10	0 – Disabled. 1 – Enabled. Default Value = 1.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2

Message ID = 6

Definition of Message Parameters

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

4.4.7 HST_get_product_id

As discussed in Section 2.3.3, a 6-bit unique ID is assigned for each specific product. 64 different products in total can be recognized by their unique IDs. Through this command, the Host PC is able to read back the existing hardware configuration and compares it with user configured product type through software GUI interface. If they match with each other, it is alright. Otherwise, the Host PC shall prompt an error message to user indicating that a wrong product conversion PCBA is used in the test machine.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 7
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 6
Message Type = 2
Message ID = 7

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. Otherwise, it is 0.
3	Prod ID	If Status = READY, this byte contains Product ID. 1 --- GrenadaBP2. 0xFF --- No Product ID. Others --- Reserved. If Status = ERROR, this byte is undefined.

4.4.8 HST_get_operation_mode

Referring to Figure 3.4.2 and Figure 4.1, there is a DIP switch to set the HST operating mode. Figure 4.4 shows the definition for each bit.

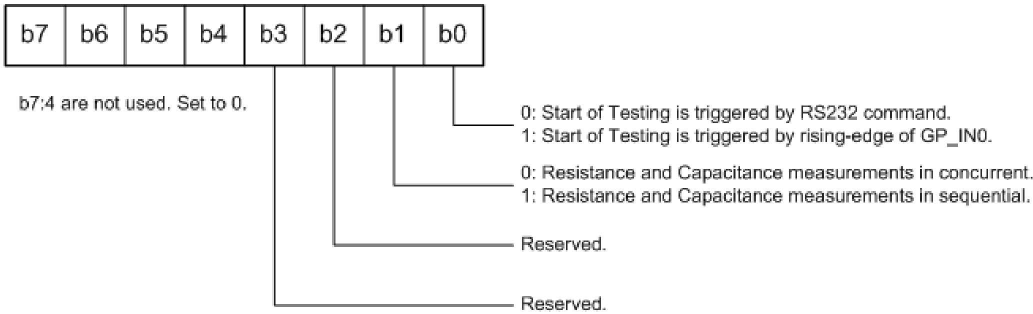


Figure 4.4 Bit Definition for Operating Mode Setting

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 8
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 6
Message Type = 2
Message ID = 8

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 2: ERROR
2	Error Code	When Status = ERROR, this byte indicates the error code. Otherwise, it is 0
3	Operating Mode	If Status = READY, this byte contains operating mode as in Figure 4.4 If Status = ERROR, this byte is undefined.

4.4.9 HST_start_meas

This command is to trigger the uP to start resistance and capacitance measurements on all enabled channels and HGAs. The measurement may take up to 4~8 seconds for completion, depending on the operating mode bit1 setting as shown in the Figure 4.4.

If the operating mode bit0 is set 1, this command is ignored and acknowledged with error and error code.

Upon completion of measurements, the uP will acknowledge back to the Host PC with READY status. Afterwards, the Host PC needs to issue commands to get the measurement results. Refer to commands: HST_get_short_detection, HST_get_res_results, HST_get_cap_results, HST_get_bias_voltages, HST_get_results_by_hga, and HST_get_debug_info.

The Host PC needs to wait for 4~8 seconds before receiving acknowledgment message. It is recommended to set the Timeout value to be >10s.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 9

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Flex Cable Index #	Index # of flex cable. Unsigned Byte. 1 – Up Tab; 2 – Down Tab. Other values are reserved.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 9

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 2: ERROR
2	Error Code	If Status = READY, this byte is 0. If Status = ERROR, this byte contains the error code.

4.4.10 HST_get_short_detection

The command is for user to get all 10 HGA's short-circuit detection results from the last measurement command. After each measurement command, the results are retained inside the uP RAM until next measurement command is executed.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 10
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 125 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 10

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	HGA1 Status of W+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
4	HGA1 Status of W- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
5	HGA1 Status of TA+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
6	HGA1 Status of TA- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
7	HGA1 Status of wH+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
8	HGA1 Status of wH- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
9	HGA1 Status of rH+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.

10	HGA1 Status of rH- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
11	HGA1 Status of R1+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
12	HGA1 Status of R1- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
13	HGA1 Status of R2+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
14	HGA1 Status of R2- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
. . .	. . . . . .	. . . .
111	HGA10 Status of W+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
112	HGA10 Status of W- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
113	HGA10 Status of TA+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
114	HGA10 Status of TA- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
115	HGA10 Status of wH+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
116	HGA10 Status of wH- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
117	HGA10 Status of rH+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
118	HGA10 Status of rH- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
119	HGA10 Status of R1+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
120	HGA10 Status of R1- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
121	HGA10 Status of R1+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
122	HGA10 Status of R1- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.

4.4.11 HST_get_res_results

The command is for user to get all 10 HGA's resistance results from the last measurement command. These results are always corrected with calibration data (refer to command "HST_get_results_by_hga"). After each measurement command, the results are retained inside the uP RAM until next measurement command is executed.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 11
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 245 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 11

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	HGA1, Writer Res. (LSB Low)	HGA1 Writer resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
4	HGA1, Writer Res. (LSB Mid)	
5	HGA1, Writer Res. (MSB Low)	
6	HGA1, Writer Res. (MSB High)	

7	HGA1, TA Res. (LSB Low)	HGA1 TA resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
8	HGA1, TA Res. (LSB Mid)	
9	HGA1, TA Res. (MSB Low)	
10	HGA1, TA Res. (MSB High)	
11	HGA1, wH Res. (LSB Low)	HGA1 Write Heater resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
12	HGA1, wH Res. (LSB Mid)	
13	HGA1, wH Res. (MSB Low)	
14	HGA1, rH Res. (MSB High)	
15	HGA1, rH Res. (LSB Low)	HGA1 Read Heater resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
16	HGA1, rH Res. (LSB Mid)	
17	HGA1, rH Res. (MSB Low)	
18	HGA1, rH Res. (MSB High)	
19	HGA1, Read1 Res. (LSB Low)	HGA1 Reader 1 resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
20	HGA1, Read1 Res. (LSB Mid)	
21	HGA1, Read1 Res. (MSB Low)	
22	HGA1, Read1 Res. (MSB High)	
23	HGA1, Read2 Res. (LSB Low)	HGA1 Reader 2 resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
24	HGA1, Read2 Res. (LSB Mid)	
25	HGA1, Read2 Res. (MSB Low)	
26	HGA1, Read2 Res. (MSB High)	
27~50	HGA2, ...	HGA2, 6 Channel resistance values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
. . .	. . . . .	. . . . . .
219~242	HGA10, ...	HGA10, 6 Channel resistance values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.

4.4.12 HST_get_cap_results

The command is for user to get all the 10 HGA's capacitance results from the last measurement command. These results are always corrected with calibration data (refer to command "HST_get_results_by_hga"). After each measurement command, the results are retained inside the uP RAM until power off or next measurement command is executed.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 12
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 85 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2

Message ID = 12

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	HGA1, Capacitance 1 (LSB Low)	HGA1 Channel 1 capacitance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit capacitance value, in unit of pF.
4	HGA1, Capacitance 1 (LSB Mid)	
5	HGA1, Capacitance 1 (MSB Low)	
6	HGA1, Capacitance 1 (MSB High)	
7	HGA1, Capacitance 2 (LSB Low)	HGA1 Channel 2 capacitance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit capacitance value, in unit of pF.
8	HGA1, Capacitance 2 (LSB Mid)	
9	HGA1, Capacitance 2 (MSB Low)	
10	HGA1, Capacitance 2 (MSB High)	
11~19	HGA2, ...	HGA2, 2 Channel capacitance values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit capacitance value, in unit of pF.
. . .	. . . . .	. . . . . .
75~82	HGA10, ...	HGA2, 2 Channel capacitance values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit capacitance value, in unit of pF.

4.4.13 HST_get_bias_voltages

The command is for user to get all 10 HGA's measured bias voltage on each resistance channel from the last measurement command. After each measurement command, the results are retained inside the uP RAM until next measurement command is executed.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 13
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 245 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 13

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR

3	HGA1, Writer Bias. (LSB Low)	HGA1 Writer bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
4	HGA1, Writer Bias. (LSB Mid)	
5	HGA1, Writer Bias. (MSB Low)	
6	HGA1, Writer Bias. (MSB High)	
7	HGA1, TA Bias. (LSB Low)	HGA1 TA bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
8	HGA1, TA Bias. (LSB Mid)	
9	HGA1, TA Bias. (MSB Low)	
10	HGA1, TA Bias. (MSB High)	
11	HGA1, wH Bias. (LSB Low)	HGA1 Write Heater bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
12	HGA1, wH Bias. (LSB Mid)	
13	HGA1, wH Bias. (MSB Low)	
14	HGA1, rH Bias. (MSB High)	
15	HGA1, rH Bias. (LSB Low)	HGA1 Read Heater bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
16	HGA1, rH Bias. (LSB Mid)	
17	HGA1, rH Bias. (MSB Low)	
18	HGA1, rH Bias. (MSB High)	
19	HGA1, Read1 Bias. (LSB Low)	HGA1 Reader 1 bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
20	HGA1, Read1 Bias. (LSB Mid)	
21	HGA1, Read1 Bias. (MSB Low)	
22	HGA1, Read1 Bias. (MSB High)	
23	HGA1, Read2 Bias. (LSB Low)	HGA1 Reader 2 bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
24	HGA1, Read2 Bias. (LSB Mid)	
25	HGA1, Read2 Bias. (MSB Low)	
26	HGA1, Read2 Bias. (MSB High)	
25~50	HGA2, ...	HGA2, 6 Channel bias voltage values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
. . .	. . . . .	. . . . . .
219~242	HGA10, ...	HGA10, 6 Channel bias voltage values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.

4.4.14 HST_get_results_by_hga

Rather than getting the measurement results of 10 HGAs at one-shot, the command is for user to get the test results (short-circuit status, resistance, capacitance) for a specified single HGA. After each measurement command, the results are retained inside the uP RAM until power off or next measurement command is executed. This command is meant for both evaluation and normal operation.

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 14

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	HGA Index #	Unsigned Byte. HGA Index # between 1~10. Other values are reserved.
2	Correction	Unsigned Byte. 0 --- The resistance and capacitance are raw results. 1 --- The resistance and capacitance are corrected results with calibration data. Others --- Reserved.

Acknowledge Message from Measurement uP

Message Size = 49 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 14

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Status of W+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
4	Status of W- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
5	Status of TA+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
6	Status of TA- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
7	Status of wH+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
8	Status of wH- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
9	Status of rH+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
10	Status of rH- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
11	Status of R1+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
12	Status of R1- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
13	Status of R2+ Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
14	Status of R2- Pad	0 – No Test; 1 – Open; 2 – Shorted to its paired pad. Others – Reserved.
15	Writer Res. (LSB Low)	HGA Writer resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
16	Writer Res. (LSB Mid)	
17	Writer Res. (MSB Low)	
18	Writer Res. (MSB High)	
19	TA Res. (LSB Low)	HGA TA resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
20	TA Res. (LSB Mid)	
21	TA Res. (MSB Low)	
22	TA Res. (MSB High)	
23	wH Res. (LSB Low)	HGA Write Heater resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
24	wH Res. (LSB Mid)	

25	wH Res. (MSB Low)	
26	rH Res. (MSB High)	
27	rH Res. (LSB Low)	HGA Read Heater resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
28	rH Res. (LSB Mid)	
29	rH Res. (MSB Low)	
30	rH Res. (MSB High)	
31	Read1 Res. (LSB Low)	HGA Reader 1 resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
32	Read1 Res. (LSB Mid)	
33	Read1 Res. (MSB Low)	
34	Read1 Res. (MSB High)	
35	Read2 Res. (LSB Low)	HGA Reader 2 resistance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
36	Read2 Res. (LSB Mid)	
37	Read2 Res. (MSB Low)	
38	Read2 Res. (MSB High)	
39	Capacitance 1 (LSB Low)	HGA Channel 1 capacitance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit capacitance value, in unit of pF.
40	Capacitance 1 (LSB Mid)	
41	Capacitance 1 (MSB Low)	
42	Capacitance 1 (MSB High)	
43	Capacitance 2 (LSB Low)	HGA Channel 2 capacitance. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit capacitance value, in unit of pF.
44	Capacitance 2 (LSB Mid)	
45	Capacitance 2 (MSB Low)	
46	Capacitance 2 (MSB High)	

4.4.15 HST_get_bias_by_hga

Rather than getting the measurement results of 10 HGAs at one-shot, the command is for user to get the measured bias voltage on each channel of DUT for a specified single HGA. After each measurement command, the results are retained inside the uP RAM until power off or next measurement command is executed.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 15

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	HGA Index #	Unsigned Byte. HGA Index # between 1~10. Other values are reserved.

Acknowledge Message from Measurement uP

Message Size = 29 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 15

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Writer Bias. (LSB Low)	HGA Writer bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
4	Writer Bias. (LSB Mid)	
5	Writer Bias. (MSB Low)	
6	Writer Bias. (MSB High)	
7	TA Bias. (LSB Low)	HGA TA bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
8	TA Bias. (LSB Mid)	
9	TA Bias. (MSB Low)	
10	TA Bias. (MSB High)	
11	wH Bias. (LSB Low)	HGA Write Heater bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
12	wH Bias. (LSB Mid)	
13	wH Bias. (MSB Low)	
14	rH Bias. (MSB High)	
15	rH Bias. (LSB Low)	HGA Read Heater bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
16	rH Bias. (LSB Mid)	
17	rH Bias. (MSB Low)	
18	rH Bias. (MSB High)	
19	Read1 Bias. (LSB Low)	HGA Reader 1 bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
20	Read1 Bias. (LSB Mid)	
21	Read1 Bias. (MSB Low)	
22	Read1 Bias. (MSB High)	
23	Read2 Bias. (LSB Low)	HGA Reader 2 bias voltage. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
24	Read2 Bias. (LSB Mid)	
25	Read2 Bias. (MSB Low)	
26	Read2 Bias. (MSB High)	

4.4.16 HST_get_sensing_by_hga

This command is meant for evaluation and development purposes. It is used to get the bias voltage on each sensing resistor of a specified HGA. After each measurement command, the results are retained inside the uP RAM until power off or next measurement command is executed.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 16

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	HGA Index #	Unsigned Byte. HGA Index # between 1~10. Other values are reserved.

Acknowledge Message from Measurement uP

Message Size = 29 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 16

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Writer Sensing. (LSB Low)	HGA Writer bias voltage on sensing resistor. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
4	Writer Sensing. (LSB Mid)	
5	Writer Sensing. (MSB Low)	
6	Writer Sensing. (MSB High)	
7	TA Sensing. (LSB Low)	HGA TA bias voltage on sensing resistor. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
8	TA Sensing. (LSB Mid)	
9	TA Sensing. (MSB Low)	
10	TA Sensing. (MSB High)	
11	wH Sensing. (LSB Low)	HGA Write Heater bias voltage on sensing resistor. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
12	wH Sensing. (LSB Mid)	
13	wH Sensing. (MSB Low)	
14	rH Sensing. (MSB High)	
15	rH Sensing. (LSB Low)	HGA Read Heater bias voltage on sensing resistor. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
16	rH Sensing. (LSB Mid)	
17	rH Sensing. (MSB Low)	
18	rH Sensing. (MSB High)	
19	Read1 Sensing. (LSB Low)	HGA Reader 1 bias voltage on sensing resistor. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
20	Read1 Sensing. (LSB Mid)	
21	Read1 Sensing. (MSB Low)	

22	Read1 Sensing. (MSB High)	
23	Read2 Sensing. (LSB Low)	HGA Reader 2 bias voltage on sensing resistor. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit bias voltage value, in unit of µV.
24	Read2 Sensing. (LSB Mid)	
25	Read2 Sensing. (MSB Low)	
26	Read2 Sensing. (MSB High)	

4.4.17 HST_calibration_enable

This command is to set an internal flag in the measurement uP firmware. When this flag is set to TURE, all calibration related commands are allowed and executed. Otherwise, those commands are acknowledged with error messages. The reason for doing so is just to prevent user issues calibration related command by mistakes, which may cause serious measurement error or damages to the calibration data stored inside EEPROM. After power-on or reset, the calibration is disabled by default.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 17

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Enable Flag	0 – Disabled. 1 – Enabled. Default Value = 0.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 17

Definition of Message Parameters

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

4.4.18 HST_start_auto_calibration

This command is allowed only when the calibration is enabled by “HST_calibration_enable” command.

On the HST electronics circuit design, 6 high precision resistors (0Ω, 10Ω, 100Ω, 500Ω, 1KΩ and 10KΩ. 0.1%) and 6 high precision capacitors (100pF, 270pF, 470pF, 680pF, 820pF and 10nF. 1%) are built-in for on-board auto calibration. Once this command is received by the measurement uP, it will sweep through these high precision resistors and capacitors based on configured parameters (refer to “HST_config_res_meas” and “HST_config_cap_meas” commands). Then it samples the feedback signals and performs mathematic calculations. After calculations, it acknowledges this command with calibrated data. The whole calibration process may take up to 10~15 seconds. After verification on these calibration data, user can issue a “SAVE” command to store the data into EEPROM for future use.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 18
Message Parameters = None

Acknowledge Message from Measurement uP

Message Size = 173 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 18

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
	Measured/Observed Resistance Values for 0Ω.	
3	CH1 Writer (LSB Low)	Channel 1 Writer resistance. Unsigned 32-bit resistance value, in unit of mΩ.
4	CH1 Writer (LSB Mid)	
5	CH1 Writer (MSB Low)	
6	CH1 Writer (MSB High)	
7	CH2 TA (LSB Low)	Channel 2 TA resistance. Unsigned 32-bit resistance value, in unit of mΩ.
8	CH2 TA (LSB Mid)	
9	CH2 TA (MSB Low)	
10	CH2 TA (MSB High)	
11	CH3 wH (LSB Low)	Channel 3 Write Heater resistance. Unsigned 32-bit resistance value, in unit of mΩ.
12	CH3 wH (LSB Mid)	
13	CH3 wH (MSB Low)	
14	CH3 wH (MSB High)	
15	CH4 rH (LSB Low)	Channel 4 Read Heater resistance. Unsigned 32-bit resistance value, in unit of mΩ
16	CH4 rH (LSB Mid)	
17	CH4 rH (MSB Low)	
18	CH4 rH (MSB High)	
19	CH5 Read1 (LSB Low)	Channel 5 Reader 1 resistance. Unsigned 32-bit resistance value, in unit of mΩ
20	CH5 Read1 (LSB Mid)	
21	CH5 Read1 (MSB Low)	
22	CH5 Read1 (MSB High)	
23	CH6 Read2 (LSB Low)	Channel 5 Reader 2 resistance. Unsigned 32-bit resistance value, in unit of mΩ
24	CH6 Read2 (LSB Mid)	
25	CH6 Read2 (MSB Low)	

26	CH6 Read2 (MSB High)	
	Measured/Observed Resistance Values (24 Bytes) for 10Ω.	
27~50	CH1~6 Resistances	CH1~6 measured resistance values as above for 10Ω. Unsigned 32-bit resistance values, in unit of mΩ
. . .	. . . . .	. . . . . .
	Measured/Observed Resistance Values (24 Bytes) for 10KΩ.	
123~146	CH1~6 Resistances	CH1~6 measured resistance values as above for 10KΩ. Unsigned 32-bit resistance values, in unit of mΩ
	Measured/Observed Capacitance Values (4 Bytes) for 100pF.	
147	Capacitance (LSB Low)	Measured capacitance value for 100pF Unsigned 32-bit resistance value, in unit of pF.
148	Capacitance (LSB Mid)	
149	Capacitance (MSB Low)	
150	Capacitance (MSB High)	
151~166	. . . . .	Measured capacitance values for 270pF, 470pF, 680pF and 820pF.
	Measured/Observed Capacitance Values (4 Bytes) for 10nF.	
167	Capacitance (LSB Low)	Measured capacitance value for 10nF Unsigned 32-bit resistance value, in unit of pF.
168	Capacitance (LSB Mid)	
169	Capacitance (MSB Low)	
170	Capacitance (MSB High)	

4.4.19 HST_save_calibration_data

This command is allowed only when the calibration is enabled by "HST_calibration_enable" command.

After completion of auto calibration command, user can verify if the calibration data are correct. Then the use can issue this command the data into EEPROM for future use. Within the measurement uP firmware, the uP follows the sequence below:

1. First it erases EEPROM signature bytes.
2. Then write these calibration data into EEPROM.
3. Read back and verify the written data.
4. If verification is OK, write the Signature Bytes to indicate the calibration data are valid.

Each time when the uP uploads the calibration data from EEPROM, it always check first if the Signature Bytes are correct. If the Signature Bytes are wrong, the calibration data are considered as invalid (EEPROM corruption). Error message will be prompt to user.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 19
Message Parameters = None

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 19

Definition of Message Parameters

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

4.4.20 HST_get_calibration_data

For normal operation on site, the calibration data are automatically uploaded from EEPROM to uP RAM after power-on reset. This command is meant for FSE to read back these calibration data from Host GUI interface when necessary.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 20
Message Parameters = None

Acknowledge Message from Measurement uP

Message Size = 173 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 20

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
	Measured/Observed Resistance Values for 0Ω.	
3	CH1 Writer (LSB Low)	Channel 1 Writer resistance. Unsigned 32-bit resistance value, in unit of mΩ.
4	CH1 Writer (LSB Mid)	
5	CH1 Writer (MSB Low)	
6	CH1 Writer (MSB High)	
7	CH2 TA (LSB Low)	Channel 2 TA resistance. Unsigned 32-bit resistance value, in unit of mΩ.
8	CH2 TA (LSB Mid)	
9	CH2 TA (MSB Low)	
10	CH2 TA (MSB High)	
11	CH3 wH (LSB Low)	Channel 3 Write Heater resistance. Unsigned 32-bit resistance value, in unit of mΩ.
12	CH3 wH (LSB Mid)	
13	CH3 wH (MSB Low)	
14	CH3 wH (MSB High)	
15	CH4 rH (LSB Low)	Channel 4 Read Heater resistance.

16	CH4 rH (LSB Mid)	Unsigned 32-bit resistance value, in unit of mΩ
17	CH4 rH (MSB Low)	
18	CH4 rH (MSB High)	
19	CH5 Read1 (LSB Low)	Channel 5 Reader 1 resistance. Unsigned 32-bit resistance value, in unit of mΩ
20	CH5 Read1 (LSB Mid)	
21	CH5 Read1 (MSB Low)	
22	CH5 Read1 (MSB High)	
23	CH6 Read2 (LSB Low)	Channel 5 Reader 2 resistance. Unsigned 32-bit resistance value, in unit of mΩ
24	CH6 Read2 (LSB Mid)	
25	CH6 Read2 (MSB Low)	
26	CH6 Read2 (MSB High)	
	Measured/Observed Resistance Values (24 Bytes) for 10Ω.	
27~50	CH1~6 Resistances	CH1~6 measured resistance values as above for 10Ω. Unsigned 32-bit resistance values, in unit of mΩ
. . .	. . . . .	. . . . . .
	Measured/Observed Resistance Values (24 Bytes) for 10KΩ.	
123~146	CH1~6 Resistances	CH1~6 measured resistance values as above for 10KΩ. Unsigned 32-bit resistance values, in unit of mΩ
	Measured/Observed Capacitance Values (4 Bytes) for 100pF.	
147	Capacitance (LSB Low)	Measured capacitance value for 100pF Unsigned 32-bit resistance value, in unit of pF.
148	Capacitance (LSB Mid)	
149	Capacitance (MSB Low)	
150	Capacitance (MSB High)	
151~166	. . . . .	Measured capacitance values for 270pF, 470pF, 680pF and 820pF.
	Measured/Observed Capacitance Values (4 Bytes) for 10nF.	
167	Capacitance (LSB Low)	Measured capacitance value for 10nF Unsigned 32-bit resistance value, in unit of pF.
168	Capacitance (LSB Mid)	
169	Capacitance (MSB Low)	
170	Capacitance (MSB High)	

4.4.21 HST_manual_set_calibration

Referring to the Section 4.4.18 for HST Auto Calibration, there are two major constraints for auto calibration:

1. The calibration accuracy is highly dependent to the precision resistors (0.1%) and capacitors (1%) on the HST PCBAs. If the incoming PCBAs have wrong resistor/capacitor values or their accuracy is incorrect, all calibration results become inaccurate. In the lab bench test, we have a way to check if resistor/capacitor nominal values are correct, but not for the accuracy.
2. The auto calibration makes use of built-in high precision resistors/capacitors on board. This method itself excludes the additional resistance/capacitance contributed by the flex cable, connectors and Pogo pins; although the majority of them can be cancelled off by fixed values from lab empirical tests.

With considerations above, this command is reserved for manual calibration when necessary during lab evaluations on the HST PCBAs. This command shall be used together with Functional Test PCBA. The basic concept is explained below.

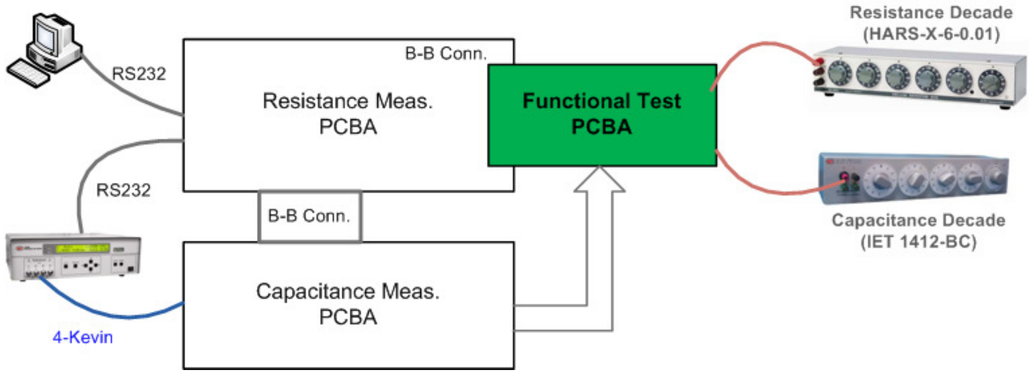


Figure 4.5 Concept of Manual Calibration

1. First select a proper HGA and its channel # through Jumpers on the Functional Test PCBA.
2. Manually set the Resistance or Capacitance Decade to a desired value.
3. Manually trigger the manual calibration command through Host PC GUI.
4. Wait for completion of the command.
5. Repeat the Step 3~4 till all channels are calibrated.
6. Save the calibration data into EEPROM.
7. Wait for the completion of "SAVE" command and then exit.

This command is only allowed when the calibration is enabled. Otherwise, error message is prompt to user.

Command Message from Host

Message Size = 10
Message Type = 1
Message ID = 21

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Calibration Type	0 – Resistance Calibration; 1 – Capacitance Calibration. Others – Reserved.
2	HGA Index #	1 ~ 10: Indicating which HGA is used. Default value is 1. Others: Reserved.
3	Channel #	1 ~ 6: Channel # for resistance calibration. 1 or 2: Channel # for capacitance calibration. Others: Reserved.
4	External Value (LSB Low)	External resistance or capacitance value depending on the calibration type. Unsigned 32-bit value, in unit of mΩ for resistance and pF for capacitance.
5	External Value (LSB Mid)	
6	External Value (MSB Low)	
7	External Value (MSB High)	

Acknowledge Message from Measurement uP

Message Size = 9 (if Status=READY); 5 (if ERROR); 4(if BUSY).
Message Type = 2

Message ID = 21

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. Otherwise, it is 0.
3	Measured Value (LSB Low)	Measured resistance or capacitance value depending on the calibration type. Unsigned 32-bit value, in unit of mΩ for resistance and pF for capacitance.
4	Measured Value (LSB Mid)	
5	Measured Value (MSB Low)	
6	Measured Value (MSB High)	

4.4.22 HST_eeprom_write

This command is mainly meant for HST lab evaluation purposes. In the hardware design, the EEPROM chip is 25LC128, which has capacity of 16 Kbytes with 16-bit address and 8-bit data. Its page size is 64-bytes. Each write cycle is able to write maximum of 64 bytes. However, user has to ensure that all data bytes are within one page boundary address. Physical page boundaries start at addresses that are integer multiple of page size – 1. Data bytes crossing page boundary addresses are ignored by the uP firmware.

Command Message from Host

Message Size = N + 3 + 3 (N is the number of data bytes to be written).
Message Type = 1
Message ID = 22

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	EEPROM Start Address (LSB)	16-bit Unsigned Value. EEPROM Start Address for write.
2	EEPROM Start Address (MSB)	
3	No. Of Byte (N)	Number of data bytes to be written into EEPROM. Unsigned Byte. Maximum # of bytes is 64.
4	Data Byte 1	Data bytes to be written.
5	Data Byte 2	
6	. . . . . .	
7	Data Byte N	

Acknowledge Message from Measurement uP

Message Size = 5.
Message Type = 2
Message ID = 22

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. Otherwise, it is 0.

4.4.23 HST_eeprom_read

This command is mainly meant for HST lab evaluation purposes. It reads back user specified number of data bytes from EEPROM starting from the user specified address.

Command Message from Host

Message Size = 6
Message Type = 1
Message ID = 23

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	EEPROM Start Address (LSB)	16-bit Unsigned Value. EEPROM Start Address for read.
2	EEPROM Start Address (MSB)	
3	No. Of Byte (N)	Number of data bytes to be read.

Acknowledge Message from Measurement uP

Message Size = N + 5.
Message Type = 2
Message ID = 23

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. Otherwise, it is 0.
3	Data Byte 1	Data bytes read from EEPROM.
4	Data Byte 2	
5	. . . . . .	
6	Data Byte N	

4.4.24 HST_dac_write

This command is mainly meant for HST lab evaluation purposes. It writes DAC register with user specified data.

Command Message from Host

Message Size = 6

Message Type = 1
Message ID = 24

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Register Address	Unsigned 8-bits. Refer to AD5668 datasheet; this is actually a combination of register and command. The higher nibble is the command and the lower nibble is the address. User has to ensure that this content is correct.
2	Data (LSB)	16-bit of data to be written to the specified register.
3	Data (MSB)	

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 24

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. Otherwise, it is 0.

4.4.25 HST_dac_read

This command is mainly meant for HST lab evaluation purposes. It reads the content from user specified DAC register. Actually the DAC IC Ad5668 does not support READ function. When this command is received, the uP just returns a value stored in its dummy register.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 25

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Register Address	Unsigned 8-bits. Refer to AD5668 datasheet; this is actually a combination of register and command. The uP will ignore the command nibble.

Acknowledge Message from Measurement uP

Message Size = 7
Message Type = 2
Message ID = 25

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. Otherwise, it is 0.
3	Data (LSB)	16-bit of data which was previously written to the specified register.
4	Data (MSB)	

4.4.26 HST_dac_output_enable

This command is mainly meant for HST lab evaluation purposes. It enables or disables all channel DAC output voltages with previous written values.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 26

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Enable Flag	Enable or Disable DAC output voltages. 0 – Disable; 1 – Enable. Other values are reserved. Unsigned 8-bit. Default value = 1, Enabled.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 26

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. Otherwise, it is 0.

4.4.27 HST_adc_write

This command is mainly meant for HST lab evaluation purposes. It writes to ADC register with user specified binary data.

The ADC register is specified with an 8-bit address. The data is specified by 24 bits. Depending on which register is written to, only 8-, or 16- or 24-bits are effectively written. Refer to the AD7173-8 datasheet for details. In Host PC GUI, it is highly recommended to use HEX value for both the address and data bytes.

Command Message from Host

Message Size = 8
Message Type = 1
Message ID = 27

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	ADC #	ADC index #. 1: ADC1 for measurement; 2: ADC2 for short detection.
2	Reg. Address	ADC register's address.
3	Data (LSB)	24-bit data to be written into ADC register.
4	Data (Mid)	
5	Data (MSB)	

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 27

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. Otherwise, it is 0.

4.4.28 HST_adc_read

This command is mainly meant for HST lab evaluation purposes. It reads ADC register with user specified address.

The ADC register is specified with an 8-bit address. The data read back can be 8-bits, 16-bits or 24-bits, depending on the register. Refer to the AD7173-8 datasheet for details. In Host PC GUI, it is highly recommended to use HEX value for both the address and data bytes.

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 28

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	ADC #	ADC index #. 1: ADC1 for measurement; 2: ADC2 for short detection. Other values: Reserved.
2	Reg. Address	ADC register's address.

Acknowledge Message from Measurement uP

Message Size = 8

Message Type = 2
Message ID = 28

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. Otherwise, it is 0.
3	Data (LSB)	24-bit data read back from specific ADC register. For an 8-bit register, the result is in LSB; Others bytes are set to 0. For a 16-bit register, the result is in LSB and Mid bytes; MSB is set to 0.
4	Data (Mid)	
5	Data (MSB)	

4.4.29 HST_get_adc_voltages

This command is mainly meant for HST bench test purposes. It reads channel input voltage from specified channel # and ADC.

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 29

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	ADC #	ADC Index #, 1 or 2. Other values are reserved. Unsigned 8-bits.
2	Channel #	Analog input channel #, Unsigned 8-bits. 1~16: The specific channel for voltage to be read out. 255 (0xFF): To read out voltages from all 16 channels. Other values: Reserved.

Acknowledge Message from Measurement uP

Message Size = 69
Message Type = 2
Message ID = 29

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. Otherwise, it is 0.
3	Voltage1 (LSB Low)	Read-back voltage from ADC input Channel 1, 0~2.5V. Unsigned 32-bit value, in unit of µV. If this channel is not selected, it will be zero.
4	Voltage1 (LSB High)	
5	Voltage1 (MSB Mid)	
6	Voltage1 (MSB High)	
7	Voltage2 (LSB Low)	Read-back voltage from ADC input Channel 2, 0~2.5V.

8	Voltage2 (LSB High)	Unsigned 32-bit value, in unit of µV. If this channel is not selected, it will be zero.
9	Voltage2 (MSB Mid)	
10	Voltage2 (MSB High)	
. . .	. . . . . .	. . . . . .
63	Voltage16 (LSB Low)	Read-back voltage from ADC input Channel 16, 0~2.5V. Unsigned 32-bit value, in unit of µV. If this channel is not selected, it will be zero.
64	Voltage16 (LSB High)	
65	Voltage16 (MSB Mid)	
66	Voltage16 (MSB High)	

4.4.30 HST_set_mux

Refer to the overview and functional blocks in Figure 3.1, there are two sets of 16-channel analog switches for DAC sources and ADC inputs. They are used for measurement switching between 10 HGAs and 6 sets of on-board resistors/capacitors.

This command is meant for evaluation purposes and used for manual setting of the analog switches or relays for capacitance measurements.

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 30

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Functional Block	Target functional block. Unsigned 8-bit. 1 – DAC switch; 2 – ADC switch; 3 – Capacitance Relay Array. Other values are reserved.
2	Channel #	Target channel to be switched to. For DAC and ADC, valid values are 1~16. For Capacitance Relay Array, valid values are 1~32.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 30

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. Otherwise, it is 0.

4.4.31 HST_set_temp_calibration

In HST electronics design, a Positive Thermist (PT-100 or PT-500) is used for temperature measurement. For each type of Thermist, there is standard data table available for temperature vs. resistance as shown for PT-100 below (as an example, only 0~100° is shown).

°C	0	1	2	3	4	5	6	7	8	9	°C
0	100.0000	100.3907	100.7814	101.1719	101.5623	101.9526	102.3427	102.7328	103.1227	103.5125	0
10	103.9022	104.2918	104.6813	105.0706	105.4599	105.8490	106.2380	106.6269	107.0156	107.4043	10
20	107.7928	108.1813	108.5696	108.9578	109.3458	109.7338	110.1216	110.5094	110.8970	111.2845	20
30	111.6718	112.0591	112.4463	112.8333	113.2202	113.6070	113.9937	114.3802	114.7667	115.1530	30
40	115.5392	115.9254	116.3113	116.6972	117.0830	117.4686	117.8541	118.2395	118.6248	119.0100	40
50	119.3951	119.7800	120.1648	120.5495	120.9341	121.3186	121.7030	122.0872	122.4713	122.8554	50
60	123.2392	123.6230	124.0067	124.3902	124.7737	125.1570	125.5402	125.9233	126.3063	126.6891	60
70	127.0718	127.4545	127.8370	128.2194	128.6016	128.9838	129.3658	129.7478	130.1296	130.5113	70
80	130.8928	131.2743	131.6556	132.0369	132.4180	132.7990	133.1799	133.5606	133.9413	134.3218	80
90	134.7022	135.0825	135.4627	135.8428	136.2227	136.6026	136.9823	137.3619	137.7414	138.1207	90
100	138.5000	138.8791	139.2582	139.6371	140.0159	140.3945	140.7731	141.1515	141.5299	141.9081	100

Figure 4.6 shows the hardware setup for temperature calibration.

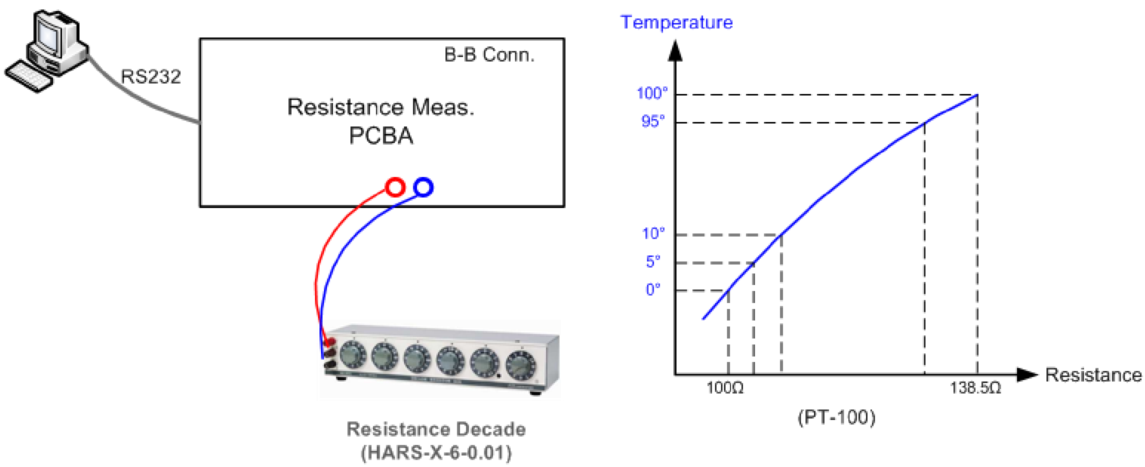


Figure 4.6 Setup for Temperature Calibration

The calibration procedures are described as below.

1. Connect the HARS-X-6-0.01 resistance decade to the desired channel to be calibrated (there are 3 temperature channels on the PCBA). The cable length and its resistance coefficient shall be as close to the actual application as possible.
2. Set the HARS-X-6-0.01 to a closet resistance value as indicated in the Table for a specific temperature.
3. Press the correct command button corresponding to the specific temperature on Host PC GUI.
4. The uP performs its calibration for this specific temperature. Upon completion, it responses to the user with READY.
5. Then, user adjust the HARS-X-6-0.01 value for calibration of next temperature.
6. Repeat the Steps 2~5 until all temperatures are calibrated.

In total, there are 20 temperature points to be calibrated, which are 5°, 10°, 15°, . . ., 95° and 100°. With these calibration data, the expected measurement accuracy for temperature range of 0~100° is 0.1°. Temperature beyond this range can be measured; however the accuracy is not guaranteed within 0.1°

Command Message from Host

Message Size = 5
Message Type = 1

Message ID = 31

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Channel #	The temperature channel index number under calibration. 1, 2 or 3. Other values are reserved. Unsigned 8-bits.
2	Temperature	The temperature corresponding to the input resistance value. Unsigned 8-bits Integer, in unit of ℃.

Acknowledge Message from Measurement uP

Message Size = 9
Message Type = 2
Message ID = 31

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. Otherwise, it is 0.
3	Voltage (LSB Low)	Read-back voltage from the specified temperature input channel, 0~2.5V. Unsigned 32-bit value, in unit of µV.
4	Voltage (LSB Mid)	
5	Voltage (MSB Low)	
6	Voltage (MSB High)	

4.4.32 HST_config_temp_meas

Temperature is a typically slow-change parameter for HST machine. Within the measurement uP, temperature input signals are sampled at regular interval of 1 second. These samples are processed with a digital low-pass filter before presenting it to user. This command is for user to configure time constant of the low-pass filter. All 3 temperature channels share the same time constant.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 32

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Time Constant	The time constant for temperature low-pass filter. Unsigned 8-bits, in unit of second. The default value is 10 seconds.

Acknowledge Message from Measurement uP

Message Size = 5

Message Type = 2
Message ID = 32

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. Otherwise, it is 0.

4.4.33 HST_get_temperature

This command is for user to read out existing measured temperature from a desired channel.

Command Message from Host

Message Size = **3**
Message Type = 1
Message ID = 33
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = **11**
Message Type = 2
Message ID = 33

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. Otherwise, it is 0.
3	CH1 Temperature (LSB)	Channel 1 Measured temperature. Usigned 16-bit value, in unit of 0.1℃.
4	CH1 Temperature (MSB)	
5	CH2 Temperature (LSB)	Channel 2 Measured temperature. Usigned 16-bit value, in unit of 0.1℃.
6	CH2 Temperature (MSB)	
7	CH3 Temperature (LSB)	Channel 3 Measured temperature. Usigned 16-bit value, in unit of 0.1℃.
8	CH3 Temperature (MSB)	

4.4.34 HST_get_cap_secondary_results

As discussed early, the purpose for capacitance measurement is to detect failures of Micro-Actuator (uACT) on HGA. From production process, there are three major potential failure modes: Open, Short and Cracking. From customer's experiences, the capacitance of uACT is sensitive to Open and Short failure modes. But it is not sensitive enough to the Cracking.

This is a "secrete" command to read out the measured Equivalent Series Resistance (ESR) on the uACT. This ESR value is actually a secondary result obtained from capacitance measurement if using LCR meter. During customer qualification process for the first HST prototype, users can decide whether they want to take this value for evaluation and to find out a better way for uACT failure detection.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 34
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 85 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 34

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	HGA1, uACT 1 ESR (LSB Low)	HGA1 uACT 1 ESR. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit resistance value, in unit of mΩ.
4	HGA1, uACT 1 ESR (LSB Mid)	
5	HGA1, uACT 1 ESR (MSB Low)	
6	HGA1, uACT 1 ESR (MSB High)	
7	HGA1, uACT 2 ESR (LSB Low)	HGA1 uACT 2 ESR. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit equivalent resistance value, in unit of mΩ.
8	HGA1, uACT 2 ESR (LSB Mid)	
9	HGA1, uACT 2 ESR (MSB Low)	
10	HGA1, uACT 2 ESR (MSB High)	
11~19	HGA2, ...	HGA2, uACT 1 and 2 ESR values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit equivalent resistance value, in unit of mΩ.
. . .	. . . . .	. . . . . .
75~82	HGA10, ...	HGA2, uACT 1 and 2 ESR values. If disabled or short-circuit detected, this value is set to 0. Unsigned 32-bit equivalent resistance value, in unit of mΩ.

4.4.35 HST_get_cap_reading

The command is meant for lab bench test during evaluation and debugging phase. It is for user to read the capacitance value for any device which is connected to the LCR meter at the moment. Typically, this command is used after setting the LCR to be connected with a specific channel using HST_set_mux command.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 35
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 9 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 35

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Capacitance (LSB Low)	Capacitance value read back from LCR meter. Unsigned 32-bit capacitance value, in unit of pF.
4	Capacitance (LSB Mid)	
5	Capacitance (MSB Low)	
6	Capacitance (MSB High)	

4.4.36 HST_start_self_test

On the HST electronics PCBAs, there are 6 high precision resistors and capacitors respectively. They can be used for auto-calibration when necessary during normal operations on site. On the other hand, they can be also used for diagnostic purposes to indicate health conditions of PCBAs after certain period of operation, such as component degrading, DC offset drifting and etc.

Upon receiving this self-test command, the uP starts following tests in sequence:

1. Check bias currents for the 6 DAC output channels.
2. Check the communication and functionality for the two ADCs.
3. Check the integrity of calibration data saved in EEPROM.
4. Measure the built-in high precision resistors and capacitors. Compare the measurement results with predefined values. If they are out of spec by 0.5% for resistance or 2% for capacitance, the self-test is considered as ERROR.
5. Check if the temperature sensor inputs are open or short (optional).

After completion of self-tests, the measured resistance and capacitance values are acknowledged to the Host PC. User may have to conduct more tests using debugging commands in order to find out detailed failure information. As general guideline, this failed PCBAs shall be sent back to design engineering for failure analysis or repair.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 36
Message Parameter = None

Acknowledge Message from Measurement uP

Message Size = 173 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 36

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
	Measured/Observed Resistance Values for 0Ω.	
3	CH1 Writer (LSB Low)	Channel 1 Writer resistance. Unsigned 32-bit resistance value, in unit of mΩ.
4	CH1 Writer (LSB Mid)	
5	CH1 Writer (MSB Low)	
6	CH1 Writer (MSB High)	
7	CH2 TA (LSB Low)	Channel 2 TA resistance. Unsigned 32-bit resistance value, in unit of mΩ.
8	CH2 TA (LSB Mid)	
9	CH2 TA (MSB Low)	
10	CH2 TA (MSB High)	
11	CH3 wH (LSB Low)	Channel 3 Write Heater resistance. Unsigned 32-bit resistance value, in unit of mΩ.
12	CH3 wH (LSB Mid)	
13	CH3 wH (MSB Low)	
14	CH3 wH (MSB High)	
15	CH4 rH (LSB Low)	Channel 4 Read Heater resistance. Unsigned 32-bit resistance value, in unit of mΩ
16	CH4 rH (LSB Mid)	
17	CH4 rH (MSB Low)	
18	CH4 rH (MSB High)	
19	CH5 Read1 (LSB Low)	Channel 5 Reader 1 resistance. Unsigned 32-bit resistance value, in unit of mΩ
20	CH5 Read1 (LSB Mid)	
21	CH5 Read1 (MSB Low)	
22	CH5 Read1 (MSB High)	
23	CH6 Read2 (LSB Low)	Channel 5 Reader 2 resistance. Unsigned 32-bit resistance value, in unit of mΩ
24	CH6 Read2 (LSB Mid)	
25	CH6 Read2 (MSB Low)	
26	CH6 Read2 (MSB High)	
	Measured/Observed Resistance Values (24 Bytes) for 10Ω.	
27~50	CH1~6 Resistances	CH1~6 measured resistance values as above for 10Ω. Unsigned 32-bit resistance values, in unit of mΩ
. . .	. . . . .	. . . . . .
	Measured/Observed Resistance Values (24 Bytes) for 10KΩ.	
123~146	CH1~6 Resistances	CH1~6 measured resistance values as above for 10KΩ. Unsigned 32-bit resistance values, in unit of mΩ
	Measured/Observed Capacitance Values (4 Bytes) for 100pF.	
147	Capacitance (LSB Low)	Measured capacitance value for 100pF Unsigned 32-bit resistance value, in unit of pF.
148	Capacitance (LSB Mid)	

149	Capacitance (MSB Low)	
150	Capacitance (MSB High)	
151~166	. . . . .	Measured capacitance values for 270pF, 470pF, 680pF and 820pF.
	Measured/Observed Capacitance Values (4 Bytes) for 10nF.	
167	Capacitance (LSB Low)	Measured capacitance value for 10nF Unsigned 32-bit resistance value, in unit of pF.
168	Capacitance (LSB Mid)	
169	Capacitance (MSB Low)	
170	Capacitance (MSB High)	

4.4.37 HST_get_firmware_version

This command is for user to get HST uP firmware version number.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 37
Message Parameters = None

Acknowledge Message from Measurement uP

Message Size = 7
Message Type = 2
Message ID = 37

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. Otherwise, it is 0.
3	Major Rev.	Major revision number.
4	Minor Rev.	Minor revision number.

4.4.38 HST_calibrate_offset

In the electronics hardware design, there are 12 differential amplification channels used for resistance measurements, 6 channels for sensing resistors and 6 channels for DUTs. Due component tolerances, these 12 channels have different offsets to be calibrated.

This command is to trigger the measurement uP to calibrate offset voltages which are used for resistance calculation later on during measurements.

These offsets are also automatically calibrated when HST_auto_calibration command is performed.

When performing manual calibration, these offset calibration is to triggered by user prior to any specific resistance calibration.

When user issue HST_save_calibration_data, the offset values are automatically saved into EEPROM.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 38
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = **57** (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 38

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Writer Sensing Offset. (LSB Low)	HGA Writer sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
4	Writer Sensing Offset. (LSB Mid)	
5	Writer Sensing Offset. (MSB Low)	
6	Writer Sensing Offset. (MSB High)	
7	TA Sensing Offset. (LSB Low)	HGA TA sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
8	TA Sensing Offset. (LSB Mid)	
9	TA Sensing Offset. (MSB Low)	
10	TA Sensing Offset. (MSB High)	
11	wH Sensing Offset. (LSB Low)	HGA wH sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
12	wH Sensing Offset. (LSB Mid)	
13	wH Sensing Offset. (MSB Low)	
14	rH Sensing Offset. (MSB High)	
15	rH Sensing Offset. (LSB Low)	HGA rH sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
16	rH Sensing Offset. (LSB Mid)	
17	rH Sensing Offset. (MSB Low)	
18	rH Sensing Offset. (MSB High)	
19	Read1 Sensing Offset. (LSB Low)	HGA Read1 sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
20	Read1 Sensing Offset. (LSB Mid)	
21	Read1 Sensing Offset. (MSB Low)	
22	Read1 Sensing Offset. (MSB High)	
23	Read2 Sensing Offset. (LSB Low)	HGA Read2 sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
24	Read2 Sensing Offset. (LSB Mid)	
25	Read2 Sensing Offset. (MSB Low)	

26	Read2 Sensing Offset. (MSB High)	
27	Writer DUT Offset. (LSB Low)	HGA Writer DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
28	Writer DUT Offset. (LSB Mid)	
29	Writer DUT Offset. (MSB Low)	
30	Writer DUT Offset. (MSB High)	
31	TA DUT Offset. (LSB Low)	HGA TA DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
32	TA DUT Offset. (LSB Mid)	
33	TA DUT Offset. (MSB Low)	
34	TA DUT Offset. (MSB High)	
35	wH DUT Offset. (LSB Low)	HGA wH DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
36	wH DUT Offset. (LSB Mid)	
37	wH DUT Offset. (MSB Low)	
38	rH DUT Offset. (MSB High)	
39	rH DUT Offset. (LSB Low)	HGA rH DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
40	rH DUT Offset. (LSB Mid)	
41	rH DUT Offset. (MSB Low)	
42	rH DUT Offset. (MSB High)	
43	Read1 DUT Offset. (LSB Low)	HGA Read1 DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
44	Read1 DUT Offset. (LSB Mid)	
45	Read1 DUT Offset. (MSB Low)	
46	Read1 DUT Offset. (MSB High)	
47	Read2 DUT Offset. (LSB Low)	HGA Read2 DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
48	Read2 DUT Offset. (LSB Mid)	
49	Read2 DUT Offset. (MSB Low)	
50	Read2 DUT Offset. (MSB High)	
51	INA Reference Voltage. (LSB Mid)	Reference Input Voltage to INA114 Amplifiers. Unsigned 32-bit offset voltage value, in unit of µV.
52	INA Reference Voltage. (MSB Low)	
53	INA Reference Voltage. (MSB High)	
54	INA Reference Voltage. (LSB Mid)	

4.4.39 HST_get_calibration_offset

After power-on reset, the calibrated offset values are automatically uploaded from EEPROM to uP RAM. User can issue this command to read out these values.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 39
Message Parameters = None.

Acknowledge Message from Measurement uP

Message Size = 57 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 39

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Writer Sensing Offset. (LSB Low)	HGA Writer sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
4	Writer Sensing Offset. (LSB Mid)	
5	Writer Sensing Offset. (MSB Low)	
6	Writer Sensing Offset. (MSB High)	
7	TA Sensing Offset. (LSB Low)	HGA TA sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
8	TA Sensing Offset. (LSB Mid)	
9	TA Sensing Offset. (MSB Low)	
10	TA Sensing Offset. (MSB High)	
11	wH Sensing Offset. (LSB Low)	HGA wH sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
12	wH Sensing Offset. (LSB Mid)	
13	wH Sensing Offset. (MSB Low)	
14	rH Sensing Offset. (MSB High)	
15	rH Sensing Offset. (LSB Low)	HGA rH sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
16	rH Sensing Offset. (LSB Mid)	
17	rH Sensing Offset. (MSB Low)	
18	rH Sensing Offset. (MSB High)	
19	Read1 Sensing Offset. (LSB Low)	HGA Read1 sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
20	Read1 Sensing Offset. (LSB Mid)	
21	Read1 Sensing Offset. (MSB Low)	
22	Read1 Sensing Offset. (MSB High)	
23	Read2 Sensing Offset. (LSB Low)	HGA Read2 sensing channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
24	Read2 Sensing Offset. (LSB Mid)	
25	Read2 Sensing Offset. (MSB Low)	
26	Read2 Sensing Offset. (MSB High)	
27	Writer DUT Offset. (LSB Low)	HGA Writer DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
28	Writer DUT Offset. (LSB Mid)	
29	Writer DUT Offset. (MSB Low)	
30	Writer DUT Offset. (MSB High)	

31	TA DUT Offset. (LSB Low)	HGA TA DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
32	TA DUT Offset. (LSB Mid)	
33	TA DUT Offset. (MSB Low)	
34	TA DUT Offset. (MSB High)	
35	wH DUT Offset. (LSB Low)	HGA wH DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
36	wH DUT Offset. (LSB Mid)	
37	wH DUT Offset. (MSB Low)	
38	rH DUT Offset. (MSB High)	
39	rH DUT Offset. (LSB Low)	HGA rH DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
40	rH DUT Offset. (LSB Mid)	
41	rH DUT Offset. (MSB Low)	
42	rH DUT Offset. (MSB High)	
43	Read1 DUT Offset. (LSB Low)	HGA Read1 DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
44	Read1 DUT Offset. (LSB Mid)	
45	Read1 DUT Offset. (MSB Low)	
46	Read1 DUT Offset. (MSB High)	
47	Read2 DUT Offset. (LSB Low)	HGA Read2 DUT channel offset. Unsigned 32-bit offset voltage value, in unit of µV.
48	Read2 DUT Offset. (LSB Mid)	
49	Read2 DUT Offset. (MSB Low)	
50	Read2 DUT Offset. (MSB High)	
51	INA Reference Voltage. (LSB Mid)	Reference Input Voltage to INA114 Amplifiers. Unsigned 32-bit offset voltage value, in unit of µV.
52	INA Reference Voltage. (MSB Low)	
53	INA Reference Voltage. (MSB High)	
54	INA Reference Voltage. (LSB Mid)	

4.4.40 HST_set_offset_relay

In the hardware design, there is a set of relays to bypass those sensing resistors during amplifier offset calibrations. This command is used to turn ON or OFF the relays. It is meant for hardware debugging purpose.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 40

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Relay ON/OFF Flag	0 – Relay is Open. 1 – Relay is Closed.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 40

Definition of Message Parameters

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

4.4.41 HST_start_short_detection

This command is meant for debugging purposes. After receiving this command, the measurement uP starts setting the DAC current source for 3 sets of different currents. Then it measures their corresponding voltages on each pad. If any specified pair has always the same voltages with 3 different current settings, it is concluded this pair is shorted together. After completion of short detection, the uP returns all measured pad voltages corresponding to the 3 sets of different current.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 41

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	HGA #	Index # of HGA, valid range from 1 to 10. Unsigned byte.

Acknowledge Message from Measurement uP

Message Size = 149 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 41

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	W+ Pad Voltage 1 (LSB Low)	Voltage on W+ Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
4	W+ Pad Voltage 1 (LSB Mid)	
5	W+ Pad Voltage 1 (MSB Low)	
6	W+ Pad Voltage 1 (MSB High)	

7	W- Pad Voltage 1 (LSB Low)	Voltage on W- Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
8	W- Pad Voltage 1 (LSB Mid)	
9	W- Pad Voltage 1 (MSB Low)	
10	W- Pad Voltage 1 (MSB High)	
11	TA+ Pad Voltage 1 (LSB Low)	Voltage on TA+ Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
12	TA+ Pad Voltage 1 (LSB Mid)	
13	TA+ Pad Voltage 1 (MSB Low)	
14	TA+ Pad Voltage 1 (MSB High)	
15	TA- Pad Voltage 1 (LSB Low)	Voltage on TA- Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
16	TA- Pad Voltage 1 (LSB Mid)	
17	TA- Pad Voltage 1 (MSB Low)	
18	TA- Pad Voltage 1 (MSB High)	
19	wH+ Pad Voltage 1 (LSB Low)	Voltage on wH+ Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
20	wH+ Pad Voltage 1 (LSB Mid)	
21	wH+ Pad Voltage 1 (MSB Low)	
22	wH+ Pad Voltage 1 (MSB High)	
23	wH- Pad Voltage 1 (LSB Low)	Voltage on wH- Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
24	wH- Pad Voltage 1 (LSB Mid)	
25	wH- Pad Voltage 1 (MSB Low)	
26	wH- Pad Voltage 1 (MSB High)	
27	rH+ Pad Voltage 1 (LSB Low)	Voltage on wH+ Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
28	rH+ Pad Voltage 1 (LSB Mid)	
29	rH+ Pad Voltage 1 (MSB Low)	
30	rH+ Pad Voltage 1 (MSB High)	
31	rH- Pad Voltage 1 (LSB Low)	Voltage on wH- Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
32	rH- Pad Voltage 1 (LSB Mid)	
33	rH- Pad Voltage 1 (MSB Low)	
34	rH- Pad Voltage 1 (MSB High)	
35	R1+ Pad Voltage 1 (LSB Low)	Voltage on R1+ Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
36	R1+ Pad Voltage 1 (LSB Mid)	
37	R1+ Pad Voltage 1 (MSB Low)	
38	R1+ Pad Voltage 1 (MSB High)	
39	R1- Pad Voltage 1 (LSB Low)	Voltage on R1- Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
40	R1- Pad Voltage 1 (LSB Mid)	
41	R1- Pad Voltage 1 (MSB Low)	
42	R1- Pad Voltage 1 (MSB High)	
43	R2+ Pad Voltage 1 (LSB Low)	Voltage on R2+ Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
44	R2+ Pad Voltage 1 (LSB Mid)	

45	R2+ Pad Voltage 1 (MSB Low)	
46	R2+ Pad Voltage 1 (MSB High)	
47	R2- Pad Voltage 1 (LSB Low)	Voltage on R2- Pad corresponding to DAC current setting 1. Unsigned 32-bit, in unit of µV.
48	R2- Pad Voltage 1 (LSB Mid)	
49	R2- Pad Voltage 1 (MSB Low)	
50	R2- Pad Voltage 1 (MSB High)	
51~98	Pad Voltages 2	Voltages on pads corresponding to DAC current setting 2.
99~146	Pad Voltages 3	Voltages on pads corresponding to DAC current setting 3.

4.4.42 HST_set_short_detection_current

This command is meant for debugging purposes. User selects the desired HAG index and Short Detection Current Index #

Command Message from Host

Message Size = 5
Message Type = 1
Message ID = 42

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	HGA #	Index # of HGA, valid range from 1 to 10. Unsigned Byte.
2	DAC Current Index	Index # for specific set of DAC currents for short detection. Valid value 1~3. Unsigned Byte.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 42

Definition of Message Parameters

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

4.4.43 HST_flex_cable_calibration

To achieve high accuracy resistance and capacitance measurements, it is important to have compensation for measurement cables. Including PCB traces, the cable resistance can be in range of 0~2 ohms for different channels and HGAs. The cable capacitance can be in range of 10~100pF depending on cable length and measurement frequency. Without compensation, these can introduce significant measurement errors.

This command is intended for cable calibration. There are two flex cables, one for up-tab and the other one for down-tab. Prior to this command, user has to attach a dedicated functional or calibration PCBAs (with all channel resistors set to 0Ω or high accuracy capacitance) to one of the flex cables. And then issue this command.

Once received this command, the measurement uP starts measuring cable resistances and capacitance on all channels of 10 HGAs. The measured results are acknowledged to Host PC and also stored in its internal memory. When user issues "HST_save_calibration_data", these data are automatically saved into EEPROM.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 43
The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Flex Cable Index #	Index # of flex cable. Unsigned Byte. 1 – Up Tab; 2 – Down Tab. Other values are reserved.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 43

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR

4.4.44 HST_get_cable_calibration_res_results

This command is meant for debugging purposes. It is used for user to read out cable resistance calibration data from EEPROM for verification.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 44

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Flex Cable Index #	Index # of flex cable. Unsigned Byte. 1 – Up Tab; 2 – Down Tab. Other values are reserved.

Acknowledge Message from Measurement uP

Message Size = 245 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 44

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	HGA1, Writer Res. (LSB Low)	HGA1 Writer Channel cable resistance. Unsigned 32-bit resistance value, in unit of mΩ.
4	HGA1, Writer Res. (LSB Mid)	
5	HGA1, Writer Res. (MSB Low)	
6	HGA1, Writer Res. (MSB High)	
7	HGA1, TA Res. (LSB Low)	HGA1 TA Channel cable resistance. Unsigned 32-bit resistance value, in unit of mΩ.
8	HGA1, TA Res. (LSB Mid)	
9	HGA1, TA Res. (MSB Low)	
10	HGA1, TA Res. (MSB High)	
11	HGA1, wH Res. (LSB Low)	HGA1 Write Heater Channel cable resistance. Unsigned 32-bit resistance value, in unit of mΩ.
12	HGA1, wH Res. (LSB Mid)	
13	HGA1, wH Res. (MSB Low)	
14	HGA1, rH Res. (MSB High)	
15	HGA1, rH Res. (LSB Low)	HGA1 Read Heater Channel cable resistance. Unsigned 32-bit resistance value, in unit of mΩ.
16	HGA1, rH Res. (LSB Mid)	
17	HGA1, rH Res. (MSB Low)	
18	HGA1, rH Res. (MSB High)	
19	HGA1, Read1 Res. (LSB Low)	HGA1 Reader 1 Channel cable resistance. Unsigned 32-bit resistance value, in unit of mΩ.
20	HGA1, Read1 Res. (LSB Mid)	
21	HGA1, Read1 Res. (MSB Low)	
22	HGA1, Read1 Res. (MSB High)	
23	HGA1, Read2 Res. (LSB Low)	HGA1 Reader 2 Channel cable resistance. Unsigned 32-bit resistance value, in unit of mΩ.
24	HGA1, Read2 Res. (LSB Mid)	
25	HGA1, Read2 Res. (MSB Low)	
26	HGA1, Read2 Res. (MSB High)	
27~50	HGA2, ...	HGA2, 6 Channel cable resistance values. Unsigned 32-bit resistance value, in unit of mΩ.
. . .	. . . . .	. . . . . .
219~242	HGA10, ...	HGA10, 6 Channel cable resistance values. Unsigned 32-bit resistance value, in unit of mΩ.

4.4.45 HST_set_cable_compensation

This command is meant for user manual adjustment for cable resistance or capacitance compensation.

Command Message from Host

Message Size = 11
Message Type = 1
Message ID = 45

The Command Parameters are defined as in the Table below.

Byte Index	Byte Definition	Description
1	Cable Index	1: Upper Tab Cable; 2: Down Tab Cable
2	HGA Index #	HGA Index #, 1~10.
3	Channel Index #	Resistance Channel Index #, 1~6.
4	Resistance or Capacitance Selection	1: Resistance; 2: Capacitance
5	Compensation Res/Cap. (LSB Low)	Cable resistance or capacitance compensation. Unsigned 32-bit resistance or capacitance value, in unit of mΩ or pF.
6	Compensation Res/Cap. (LSB Mid)	
7	Compensation Res/Cap. (MSB Low)	
8	Compensation Res/Cap. (MSB High)	

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 45

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR

4.4.46 HST_clear_all_cable_compensation

This command is meant for user to clear all cable compensation results and set their resistances to all 0.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 46
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 46

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR

4.4.47 HST_set_short_detection_threshold

For HGA pad short detection, one parameter, called "Short Detection Threshold Voltage", is to be set during machine commissioning. If voltage differences between two pads are always below this threshold value, then these two pads are considered as shorted together. This command is meant for user to set this threshold value.

Command Message from Host

Message Size = 7
Message Type = 1
Message ID = 47

The Command Parameters are defined as in the Table below.

Byte Index	Byte Definition	Description
1	Threshold Voltage. (LSB Low)	Short Detection Threshold Voltage. Usigned 32-bit value, in unit of µV.
2	Threshold Voltage. (LSB Mid)	
3	Threshold Voltage. (MSB Low)	
4	Threshold Voltage. (MSB High)	

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 47

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR

4.4.48 HST_get_short_detection_threshold

This command is to read out short detection threshold value set by user in EEPROM.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 48
Command Parameters = None

Acknowledge Message from Measurement uP

Message Size = 9
Message Type = 2
Message ID = 48

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Threshold Voltage. (LSB Low)	Short Detection Threshold Voltage. Usigned 32-bit value, in unit of µV.
4	Threshold Voltage. (LSB Mid)	
5	Threshold Voltage. (MSB Low)	
6	Threshold Voltage. (MSB High)	

4.4.49 HST_set_temp1_offset

In hardware design, temperature channel 1 is using an on-board sensor, which has some offset to be cancelled off in order to have accurate measurements. This command is meant for user to set the offset value. For example, if actual temperature is 25℃ and measured value shows 27.1℃, then the offset value shall be set to -2.1℃ or -21.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 49

The Command Parameters are defined as in the Table below.

Byte Index	Byte Definition	Description
1	Temp Offset Value	CH1 Temperature offset value for compensation. Signed value, in unit of 0.1 degree C.

Acknowledge Message from Measurement uP

Message Size = 5

Message Type = 2
Message ID = 49

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR

4.4.50 HST_get_temp1_offset

This command is to read out the temperature channel 1 offset value set by user in EEPROM.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 50
Command Parameter = None

Acknowledge Message from Measurement uP

Message Size = 6
Message Type = 2
Message ID = 50

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Temp Offset Value	CH1 Temperature offset value for compensation. Signed value, in unit of 0.1 degree C.

4.4.51 HST_get_cable_calibration_cap_results

This command is meant for debugging purposes. It is used for user to read out cable capacitance calibration data from EEPROM for verification.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 51

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Flex Cable Index #	Index # of flex cable. Unsigned Byte. 1 – Up Tab; 2 – Down Tab. Other values are reserved.

Acknowledge Message from Measurement uP

Message Size = 85 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 51

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

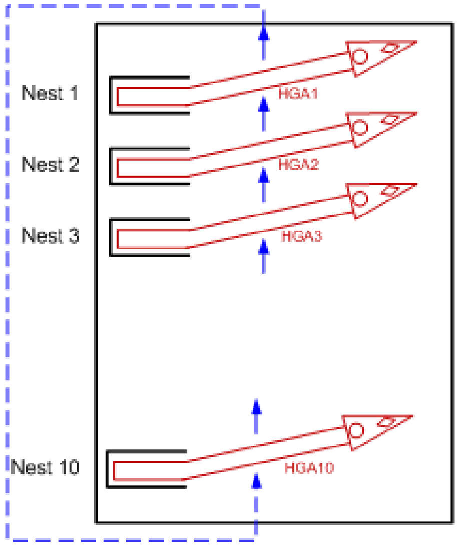
Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	HGA1, CH1 Cap. (LSB Low)	HGA1, Channel 1 cable capacitance value. Unsigned 32-bit capacitance value, in unit of pF.
4	HGA1, CH1 Cap. (LSB Mid)	
5	HGA1, CH1 Cap. (MSB Low)	
6	HGA1, CH1 Cap. (MSB High)	
7	HGA1, CH2 Cap. (LSB Low)	HGA1, Channel 2 cable capacitance value. Unsigned 32-bit capacitance value, in unit of pF.
8	HGA1, CH2 Cap. (LSB Mid)	
9	HGA1, CH2 Cap. (MSB Low)	
10	HGA1, CH2 Cap. (MSB High)	
11~18	HGA2, CH1 and CH2 Cap	HGA2, Channel 1 & 2 cable capacitance value. Unsigned 32-bit capacitance value, in unit of pF.
. . .	. . . . . .	. . . .
75~82	HGA10, CH1 & CH2 Cap.	HGA10, Channel 1 & 2 cable capacitance value. Unsigned 32-bit capacitance value, in unit of pF.

4.4.52 HST_set_precisor_cap_compensation

This command is used to set precisor compensation value for capacitance measurements. As discussed early, when the 10 HGAs are placed onto precisor, all the ground pads of HGAs are actually shorted together through precisor base plate (stainless steel). Due to this, the impact on capacitance measurement may be different from HGA to HGA depending on their physical location.

To determine the compensation values for each HGA (1~10), user has to place 10 HGAs onto precisor and perform a set of capacitance measurements in following sequences:

1. Perform 10 capacitance measurements and record the data.
2. Use the same set of HGA, rotate each HGA into next precisor nest position as shown below.



3. Perform 10 capacitance measurements for each HGA at 10 different precisor Nest position (1~10).
4. Calculate averaged value for each HGA at 10 different nest positions. Then we obtain a set of averaged measurement as below.

		Nest									
		1	2	3	4	5	6	7	8	9	10
	1	C1_1	C1_2	C1_3	C1_4	C1_5	C1_6	C1_7	C1_8	C1_9	C1_10
	2	C2_1	C2_2	C2_3	C2_4	C2_5	C2_6	C2_7	C2_8	C2_9	C2_10
	3	C3_1	C3_2	C3_3	C3_4	C3_5	C3_6	C3_7	C3_8	C3_9	C3_10
H	4	C4_1	C4_2	C4_3	C4_4	C4_5	C4_6	C4_7	C4_8	C4_9	C4_10
G	5	C5_1	C5_2	C5_3	C5_4	C5_5	C5_6	C5_7	C5_8	C5_9	C5_10
A	6	C6_1	C6_2	C6_3	C6_4	C6_5	C6_6	C6_7	C6_8	C6_9	C6_10
	7	C7_1	C7_2	C7_3	C7_4	C7_5	C7_6	C7_7	C7_8	C7_9	C7_10
	8	C8_1	C8_2	C8_3	C8_4	C8_5	C8_6	C8_7	C8_8	C8_9	C8_10
	9	C9_1	C9_2	C9_3	C9_4	C9_5	C9_6	C9_7	C9_8	C9_9	C9_10
	10	C10_1	C10_2	C10_3	C10_4	C10_5	C10_6	C10_7	C10_8	C10_9	C10_10

The precisor capacitance compensation value C_K (K=1, 2, ..., 10) is calculated as below:

$$C_{K\_i} = \frac{1}{10}\sum_{j=1}^{10} C_{i\_j} - C_{i\_j} \text{ and } C_K = \frac{1}{10}\sum_{i=1}^{10} C_{K\_i}$$

Where i, j and K is 1, 2, 3, ... , 10.

Command Message from Host

Message Size = 85
Message Type = 1
Message ID = 52

The Message Parameters are defined as in the Table below.

Byte Index	Byte Definition	Description
1	Precisor Index #	Index # of precisor. Unsigned Byte. 1 – Up Tab; 2 – Down Tab. Other values are reserved.
2	Enable Flag	Flag to disable precisor capacitance compensation. Unsigned Byte.

		1 – Enable the compensation. 0 – Disable the compensation.
3	HGA1, CH1 Compensation (LSB Low)	HGA1, Channel 1 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.
4	HGA1, CH1 Compensation. (LSB Mid)	
5	HGA1, CH1 Compensation. (MSB Low)	
6	HGA1, CH1 Compensation. (MSB High)	
7	HGA1, CH2 Compensation. (LSB Low)	HGA1, Channel 2 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.
8	HGA1, CH2 Compensation. (LSB Mid)	
9	HGA1, CH2 Compensation. (MSB Low)	
10	HGA1, CH2 Compensation. (MSB High)	
11~18	HGA2, CH1 and CH2 Compensation	HGA2, Channel 1 & 2 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.
. . .	. . . . . .	. . . .
75~82	HGA10, CH1 & CH2 Compensation.	HGA10, Channel 1 & 2 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 52

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR

4.4.53 HST_get_precisor_cap_compensation

This command is meant for debugging purposes. It is used for user to read out precisor capacitance compensation data from EEPROM for verification. After power-up, the uP automatically downloads the data from EEPROM.

Command Message from Host

Message Size = 4
Message Type = 1
Message ID = 53

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Precisor Index #	Index # of precisor. Unsigned Byte. 1 – Up Tab; 2 – Down Tab. Other values are reserved.

Acknowledge Message from Measurement uP

Message Size = 86 (if Status = READY); 4 (if BUSY); 5 (if ERROR).
Message Type = 2
Message ID = 53

The Message Parameters are defined as in the Table below.

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: READY; 1: BUSY; 2: ERROR
2	Error Code	Error Code if Status = ERROR
3	Enable Flag	Flag to disable precisor capacitance compensation. Unsigned Byte. 1 – Enable the compensation. 0 – Disable the compensation.
4	HGA1, CH1 Compensation (LSB Low)	HGA1, Channel 1 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.
5	HGA1, CH1 Compensation. (LSB Mid)	
6	HGA1, CH1 Compensation. (MSB Low)	
7	HGA1, CH1 Compensation. (MSB High)	
8	HGA1, CH2 Compensation. (LSB Low)	HGA1, Channel 2 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.
9	HGA1, CH2 Compensation. (LSB Mid)	
10	HGA1, CH2 Compensation. (MSB Low)	
11	HGA1, CH2 Compensation. (MSB High)	
12~19	HGA2, CH1 and CH2 Compensation	HGA2, Channel 1 & 2 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.
. . .	. . . . . .	. . . .
76~83	HGA10, CH1 & CH2 Compensation.	HGA10, Channel 1 & 2 Compensation capacitance value. Signed 32-bit capacitance value, in unit of pF.

4.4.54 HST_save_precisor_cap_compensation

This command is to save the precisor capacitance compensation data into EEPROM together with calculated checksum.

Command Message from Host

Message Size = 3
Message Type = 1
Message ID = 54
Message Parameters = None

Acknowledge Message from Measurement uP

Message Size = 5
Message Type = 2
Message ID = 54

Definition of Message Parameters

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

4.4.55 Unsolicited Status Command: HST_unsolicited_status

This is the only communication command initiated by the HST measurement uP and sent to Host PC.

Referring to Figure 4.1, commands "HST_get_operation_mode" and "HST_start_meas", the start of HST measurement can be triggered either by RS232 serial command from Host PC or rising-edge of GP_IN0 input from Motion Controller A3200. In second situation, the Host PC does not have the exact knowledge of when the measurement results are ready. For software programming convenience and save of machine cycle time, the uP will send an unsolicited information upon the completion of measurement triggered by GP_IN0. When the Host PC receives this command, it shall issue following commands "HST_get_res_results", "HST_get_cap_results" and "HST_get_bias_voltages" to get all measurement results from the uP.

Command Message from Host

Not Applicable.

Unsolicited Message from Measurement uP

Message Size = 5
Message Type = **3**
Message ID = **255**

Definition of Message Parameters

Byte Index	Byte Definition	Description
1	Status	0: Measurement is successfully completed. 2: Measurement is completed with ERROR.
2	Error Code	When Status = ERROR, this byte indicates the error code. Otherwise, it is 0.

4.4.56 Application Notes on Auto-Calibration and Manual-Calibration

When the PCBAs are received from vendors, the status of high precision resistors and capacitors are unknown in principle. They can be wrong in nominal values or accuracy and no proper way to check them on-board. Therefore, the first calibrations must be performed manually using the setup as shown in the Figure 4.5.

After manual calibration, a self-test command (HST_start_self_test) can be issued to verify if the on-board high precision resistors and capacitors are correct. If not, the user has to correct them before shipping out from the factory.

After confirmation of on-board precision resistors and capacitors, the PCBAs are mounted onto a machine and shipped to customer site. In case of component degrading or human errors (causing EEPROM corruption), re-calibration may be required on-site where the setup as shown in Figure 4.5 is not available. In this situation, user is provided with an option to make use of the Auto-Calibration command for re-calibration. Comparing with the Manual-Calibration, the Auto-Calibration may have less accuracy for capacitance measurement (1%). It is up to user to decide whether sending the PCBAs back to Design Engineering or doing re-calibration on-site.

4.5 Measurement Error Codes and Descriptions

After measurement, the uP returns a status byte to the Host PC. If an error occurred, the status byte indicates ERROR and it is also associated with a specific error code in following byte. All error codes are list in Table 4.5 below.

Error Code	Definition & Description
0	No error and the command is executed successfully.
1	Host RS232 communication is timeout.
2	Host command framing is out of synchronization.
3	No “ETX” byte is detected in Host command frame.
4	Host RS232 command checksum error.
5	Wrong command parameter from Host.
6	Illegal command from Host.
7	Fail to write to ADC register.
8	Fail to read from EEPROM.
9	Fail to write to EEPROM.
10	Signature bytes are corrupted in EEPROM.
11	Wrong checksum of calibration data in EEPROM.
12	Communication timeout with LCR meter.
13	Measurement error from LCR meter.
14	HST calibrated is disabled.
15	ADC input is out of range.
16~255	Reserved for future.

For ease of continuous development, it is recommended to display both error code and description on the Host PC. If an unknown error code is received, the Host PC displays the error code only.

Because of multiple functional blocks in the HST system design, the error codes shall be organized in such a way that the user can tell the source of error in terms of functional blocks just from the error codes.

4.6 Hardware IO Triggering for Start of Measurements

The start of measurement can be triggered by either software RS232 command “HST_start_meas” or a hardware IO input, depending on the Operation Mode selection DIP switch (SW2, bit0 OP_MODE0) setting as discussed in Section 4.4.8.

When OP_MODE0 (Bit 0) is set 0:
The start of measurement is always initiated by RS232 command from Host PC and the hardware IO inputs are ignored. Upon completion of all measurements, the uP will acknowledge with Ready or Error message to the Host PC. After that, the Host PC can issue a command “HST_get_all_meas_results” to read out all test results via RS232.

When OP_MODE0 (Bit 0) is set to 1:
The start of measurement can be triggered by either hardware IO inputs or RS232 command from Host PC. The hardware IO trigger is meant for normal production to shorten the communication cycle time. The RS232 is meant for **off-line** machine maintenance purposes, such as calibration, trouble-shooting and etc. This arrangement opens the door for potential conflicts where two start-of-measurement commands come together from both hardware and RS232 at the same moment. **This situation may cause unpredictable behavior and should be avoided!!**

Communication Protocol for IO Triggering

There are 2 GPIO inputs and 2 GPIO outputs are used for communications, i.e.

GP_IN0 Input to measurement uP. It is an edge-sensitive signal. A falling-edge will trigger the start of measurement. Upon completion of measurement, it shall be set to high by the Host PC or AeroTech PLC controller and prepare for next measurement.

GP_IN1 Input to the measurement uP as command parameter for start of measurement. If GP_IN1 = 1, it means UP_TAB; If GP_IN1 = 0, it means DOWN_TAB.

GP_OUT0 Output from uP to indicate completion of measurement. 0 for measurement Busy and 1 for Completion.

GP_OUT1 Output from uP to indicate measurement status. 1 for measurement completed with No Error; 0 for measurement completed with Error.

Figure 4.7 below shows the Timing Chart for IO triggering.

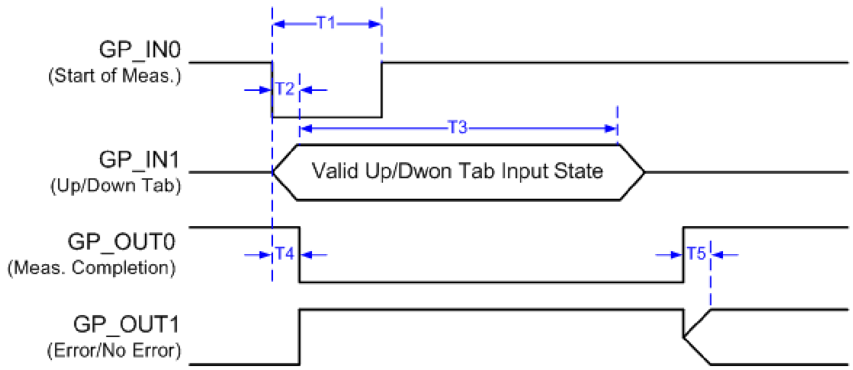


Figure 4.7 Timing Chart for IO Triggering Signals

Notes: High or Low are the signal logic status input/output to/from the measurement uP. It is not necessary the same as AeroTech, depending on sink or sourcing IO.

The Timing requirements are described below.

Name	Description	Requirements
T1	Minimum negative pulse duration from AeroTech PLC.	>=20ms
T2	Maximum delay of a valid Up/Down Tab selection.	<=10ms
T3	Minimum time of Up/Down Tab signal has to be valid.	>=20ms
T4	Maximum delay of a valid status of measurement completion.	<=10ms
T5	Maximum delay between meas. completion and error status.	<=10ms

5. Proposal for Production Functional Tester

From discussion with customers, the forecast for HST machine is probably 100 over 5 years, i.e. 20 machines per year. With such a low volume, it is not economically wise to invest thousands of dollars to develop fully automated functional tester to test a few PCBAs which are worth of hundreds of dollars. However, the quality can't be compromised. Therefore, it is proposed to have a low-cost design for the Functional Tester, which is intended to be used by design engineers rather than production operators. The functional tests are considered as outgoing quality check for every PCBA shipping out to customers. They shall be performed after all calibrations have been done.

The functional tests are split into two-stage tests: PCBA Test and Machine Test.

Functional Test at PCBA Level

This test is solely at PCBA level. Figure 5.1 shows the concept design.

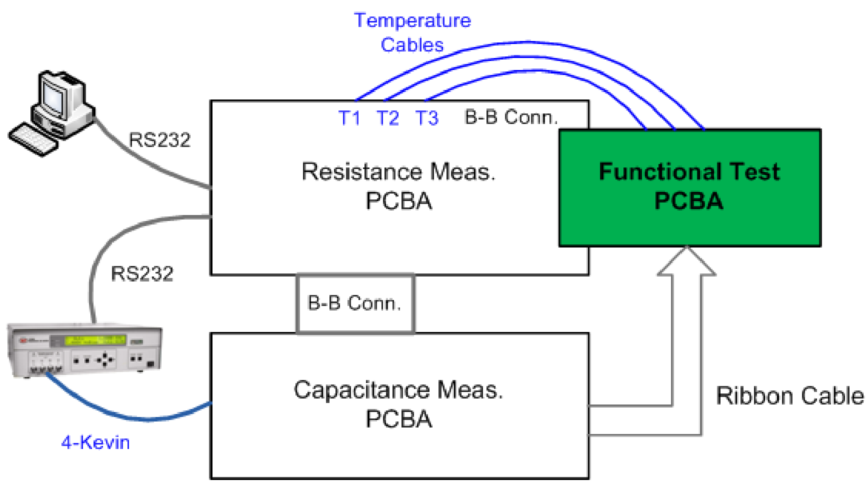


Figure 5.1 Concept Design for Functional Test at PCBA Level

This test setup is similar to the manual calibration setup as shown in the Figure 4.5. Rather than using resistance/capacitance decades as variable inputs, fixed high precision resistors and capacitors are mounted on the functional test PCBA. The resistor and capacitor values are chosen to be in the middle of HGA nominal ranges.

For the 3-channel temperature measurement circuits, there are three high precision resistors, 100Ω, 120Ω and 140Ω connected to Channel 1, 2 and 3. They are corresponding to 0°, ~50+°, and ~100+°.

Through using the API commands defined in Chapter 4, all the resistors, capacitors and temperatures can be measured. Compare these measurement results with the desired values, they are within specified tolerance ranges, the functional tests are considered Pass.

The circuit design on the Functional Test PCBA is proposed as below:

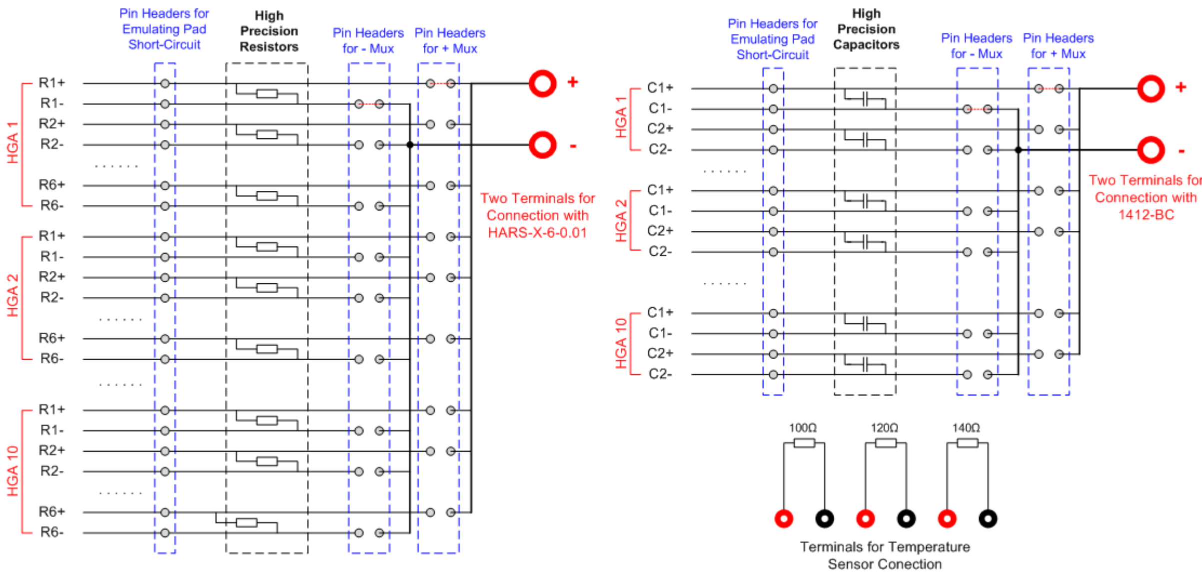


Figure 5.2 Proposed Circuit Design for Functional Test PCBA

Functional Test at Machine Level

After the success of functional tests at PCBA level, we can proceed with this functional test at machine level. Figure 5.3 below shows the concept of design.

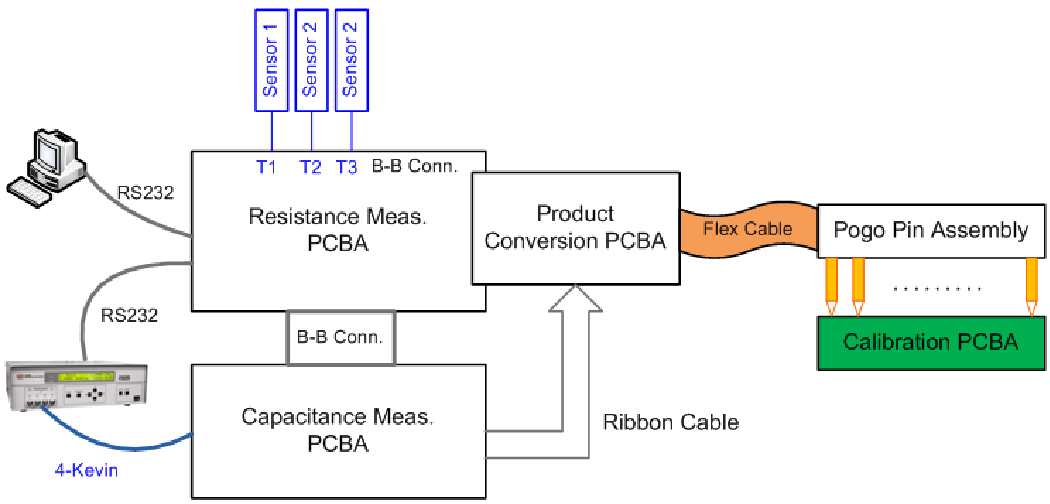


Figure 5.3 Concept Design for Functional Tests at Machine Level

As shown in the Figure 5.2 and 5.3, the major difference between these two functional tests is that the machine level test does include the Product Conversion PCBA, Flex Cable, Pogo Pin Assembly, Pin Contacts with Calibration PCBA.

The purpose of calibration PCBA is to emulate 10-HGAs in production. There are some known precision resistor and capacitors on the PCBA. This calibration PCBA is situated in a mechanical housing and can be manually clipped onto the Pogo Pin assembly. Once the above is setup ready, user can trigger a measurement command through GUI on Host PC. From the measurement results, user can compare these results with the results from PCBA level tests and make conclusions on following:

1. How much are the differences between these two measurements and are they acceptable?

2. How much extra resistance and capacitance are contributed by the Conversion PCBA, Flex cable and Pogo Pin assembly.

The tests can be done with servo both ON and OFF for evaluation of servo noise impacts on measurement results.

6. GUI Requirements for HST Bench Test Tool

Rather than application software for customers, this Host GUI is mainly used as an internal **Standalone Bench Test Tool** for electronics evaluation test during development and functional test in mass production. It shares the same sets of API commands and communication protocols as HST application software for customers. Therefore, most of source codes or functional modules can be re-used for each other if the software is designed properly.

As it is used as an internal software test tool, “Parcel-Compliant” is nice to have but not compulsory. The GUI requirements specified in this document are mainly meant for functional purposes. The software designers/programmers are allowed to change the GUI appearances, but not to change functionality requirements.

6.1 Overview of GUI Design

All communications between Host PC and uP are through API commands. All API commands can be basically classified into three groups: Bench Test, Functional Test, Configuration & Setup Commands, Calibration Commands and Debug Commands. Figure 6.1 below shows the overview for GUI design.

Functional Requirements

1. Software Tool name/revision and the firmware revision shall be displayed at the top of GUI. At every moment of launching this software tool, it shall first inquire the firmware version from measurement uP. If there is no acknowledgement within 2 seconds, it is considered no RS232 connection and the firmware revision is unknown.

2. Five functional TABs or groups are required for different test purposes, i.e.

Bench Tests:	Mainly used for development evaluation and debugging.
Functional Tests:	Used as OQC (Outgoing Quality Check) for PCBAs before shipments.
Configuration/Setup:	Used for test configuration/setup for different products in future.
Calibration:	Used for lab PCBAs measurement calibrations.
Debug:	Used for failure analysis and trouble-shooting in case of test failures.

3. Access to the Calibration and Debug Tabs shall be protected by passwords. This is to prevent user from mistakenly overwriting the calibrated data stored in EEPROM. Only authorised personnel are allowed to access the commands in these two Tabs.

4. “Bench Tests” is the default main GUI presented to user every time when this Bench Test Tool is launched.

5. Data in “Configuration & Setup” shall be automatically saved in Host PC hard disk as a HST configuration file in a dedicated directory at every moment when this Bench Test Tool application is closed. They shall be automatically downloaded into measurement uP using API commands “HST_config_res_meas”, “HST_config_cap_meas”, “HST_config_short_detection”, “HST_hga_enable” and “HST_meas_channel_enable”.

6. At every moment when this Bench Test Tool application is launched, the previously saved configuration data shall be automatically uploaded from hard disk and displayed on the “Configuration & Setup” Tab.

7. No data retention is required for “Bench Tests”, “Functional Tests”, “Calibration” and “Debug” Tabs. All data on these GUI pages are cleared to be 0 or blank when this Bench Test Tool application is launched.

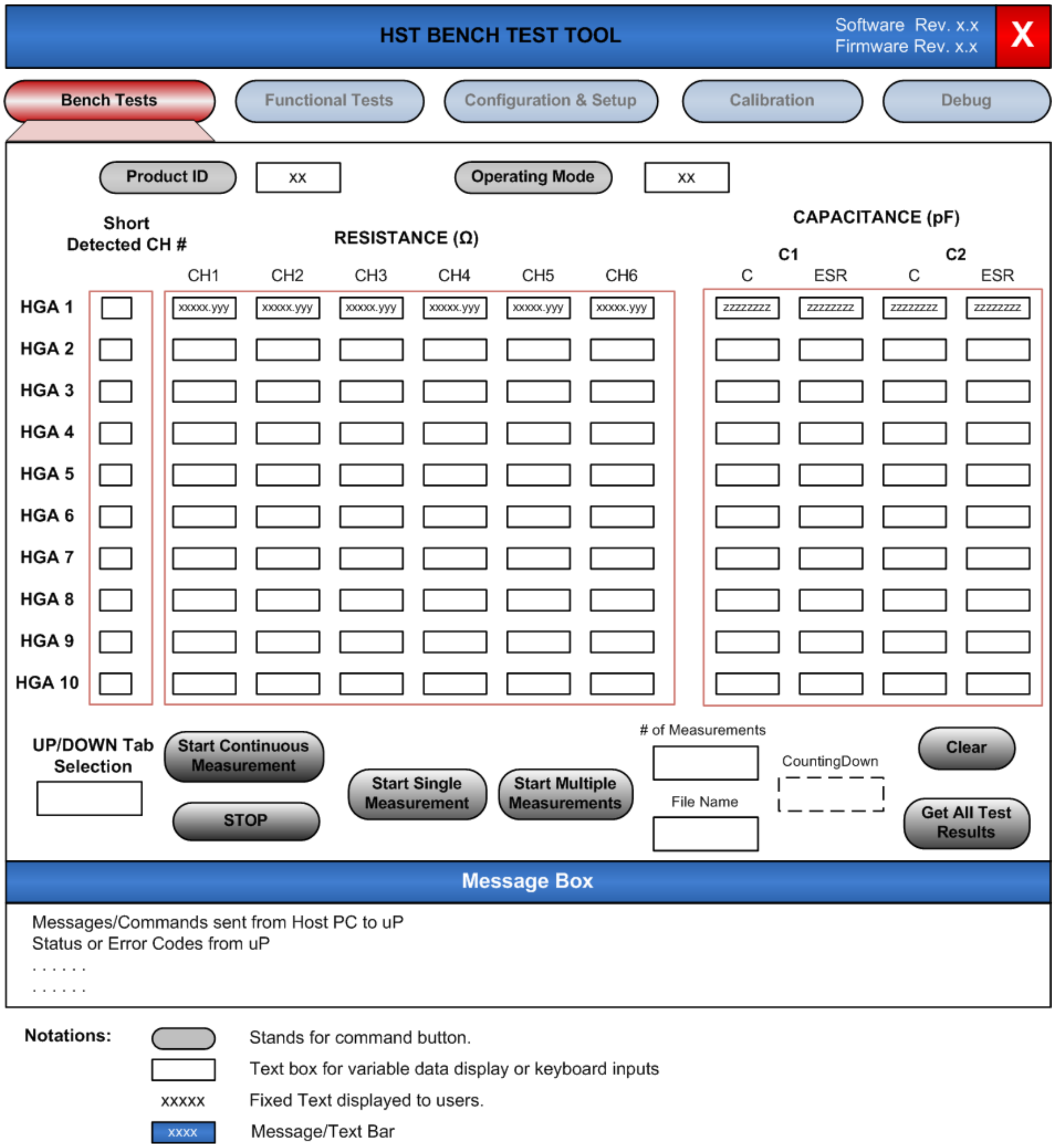


Figure 6.1 Overview of GUI Design

6.2 GUI Design and Requirements for Bench Tests

As shown in the Figure 6.1, this GUI is the default page after launch of this Bench Test Tool application. There are 6 command buttons and their behaviors are defined as below.

Product ID: This command is associated with the API command “HST_get_product_id”. The product ID # returned from uP shall be displayed in the test box next to it. In future, this ID # will be replaced with a text string of product name. The size of text box shall be 20 characters.

Operating Mode: This command is associated with API command “HST_get_operation_mode”. A test string “Trigger by RS232” or “Trigger by IO” shall be displayed in the text box depending on the response message from measurement uP.

UP/DOWN Tab Selection:

This is an input parameter associated with “HST_start_meas” command. A drop-down manual shall be provided for user selection. There are two options: UP Tab and DOWN Tab. The UP Tab and DOWN Tab correspond to command input parameter 1 and 2 respectively. When launching this Bench Test Tool, the selection can be always default to UP Tab.

Start Continuous Measurement:

When this command button is triggered, the Host PC will continuously send “HST_start_meas” command to the uP and wait for measurement results. The results are displayed and updated into the message box for every single command as shown in the Figure 6.1. It is recommended to have about 50ms interval delay between two consecutive measurement commands.

Display of Short-Circuit Detection --- 0 stands for no short circuit is detected. If short circuits are detected on multiple channels, only the first detected channel number is displayed in the message box. The measurement results for Resistance and Capacitance are all set to 0.

Display of Resistance Value --- Floating point value. The basic unit is ohm (Ω). Three digits after decimal point and maximum value shall be 100KΩ.

Display of Capacitance Value --- There are two parameters measured for each channel, i.e. Primary and Secondary. The primary is the capacitance, which is an integer value. The basic unit is pF. The maximum value is 1000nF (10^6pF). The secondary is the Equivalent Series Resistance (ESR, refer to command “HST_get_cap_secondary_results”), which is an integer value with basic unit of mΩ. The maximum value shall be <1KΩ.

STOP: This command is for user to terminate the process of continuous or multiple measurement command. Once received, this Bench Test application software shall stop sending new “HST_start_meas” command to uP once existing command is acknowledged by the uP.

Start Single Measurement:

This command is similar to the Start Continuous Measurements, except that the Host PC sends “HST_start_meas” once only.

Start Multiple Measurements:

When this command is requested, typically the user wants to log the measurement results in a text file. Later on, this file can be exported into Microsoft EXCEL for further statistics analysis. There are two parameters for user to key-in. One is the number of measurements and the other one is the log-file name. By default, the extension name is *****.txt. Depending on the user specified # of measurements, this test may take long time to complete. Therefore, it is important to have a **Counting Down** message box to indicate the number of measurement cycles left. In this way, the user knows that the measurement is still alive and how much time to complete. Besides written to a text file, the results shall be updated into text/data boxes in the GUI.

If user wants to terminate this test, the STOP button can be used. In this case, results shall be recorded in the log-file. The format of text log-file is proposed as below.

Test #	HGA #	Short Detection	Resistance						Capacitance			
			Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	C1		C2	
									C	ESR	C	ESR
1	1											
	2											
	3											
	4											
	5											
	6											
	7											
	8											
	9											
	10											
……												
N	1											
	2											

Results for each HGA shall be preceded with ASCII code of Line-Feed and terminated with ASCII code of Carriage Return.

Clear

Clear command is just to erase all measurement data to be blank.

Get All Test Results

This command button is functional only with measurement uP firmware V0.18 and onwards. It is linked to the API command “HST_get_all_meas_results”. When this command is issued to uP, all test results: pad short detection, resistances and capacitances, 445 bytes in total, are acknowledged to the Host PC in one-shot. The Host PC shall have enough RS232 receiving buffer size to ensure that no overflow occurs during transmission. After receiving all test results, the corresponding text boxes for results shall be updated. The ESR results are not included in “HST_get_all_meas_results”, therefore, the values displayed on GUI shall remain the same as historical data. The intention of this command button is to verify the uP firmware functionality.

Notes: The existing implementation for getting test results with four different commands can remain as it is now.

Message Box:

The message box is meant for user to visualize the test activities in progress.

1. For any message or command sent from Host PC to measurement uP, the command shall be displayed with converted text message in this box.
2. The command status READY or Error Code (in case of ERROR) shall be displayed in this box.
3. The measurement results are not displayed in this box.

6.3 GUI Design and Requirements for Functional Tests

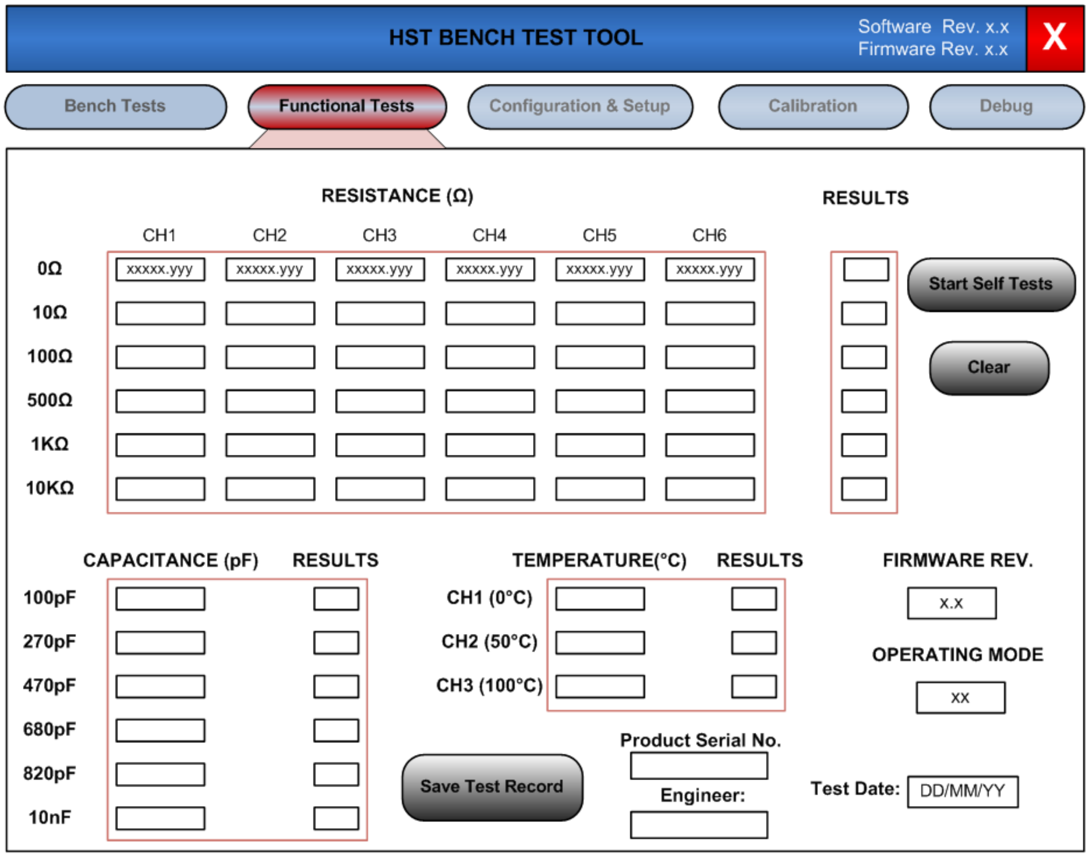


Figure 6.2 GUI Design for Functional Tests

This GUI serves as a Functional Tester for resistance and capacitance PCBAs after completion of all necessary calibration, debugging and bench tests. As discussed early, there 6 sets of high precision resistors and capacitors built-in on boards. Through the Self-Test command “HST_start_self_tests”, measurement results are feedback from uP. The Host PC shall verify that whether measured results are within ±(0.5%+0.25Ω) of specified values for resistance or ±(1.5%+10pF) of specified values for capacitance. If yes, the result is considered Pass; otherwise, it is considered Fail. The Pass or Fail shall be filled into the result message box as shown above.

For temperature measurement, the Pass or Fail criteria is ±0.5℃ of specified value.

A Clear command button is required for user to clear all measurement results to be blank and user can restart the self-test if necessary.

Besides, the Host PC shall read out the Operating Mode, uP Firmware Revision and current Data information as records for Functional Tests.

Upon completion of Functional Tests, user may want to save test results as an OQC quality record. In this case, user can press the button “Save Test Record” and then is requested to key in the product serial number and test engineer’s name.

The OQC test report shall be saved in a separate folder different from the Test Configuration File.

6.4 GUI Design and Requirements for Configuration & Setup

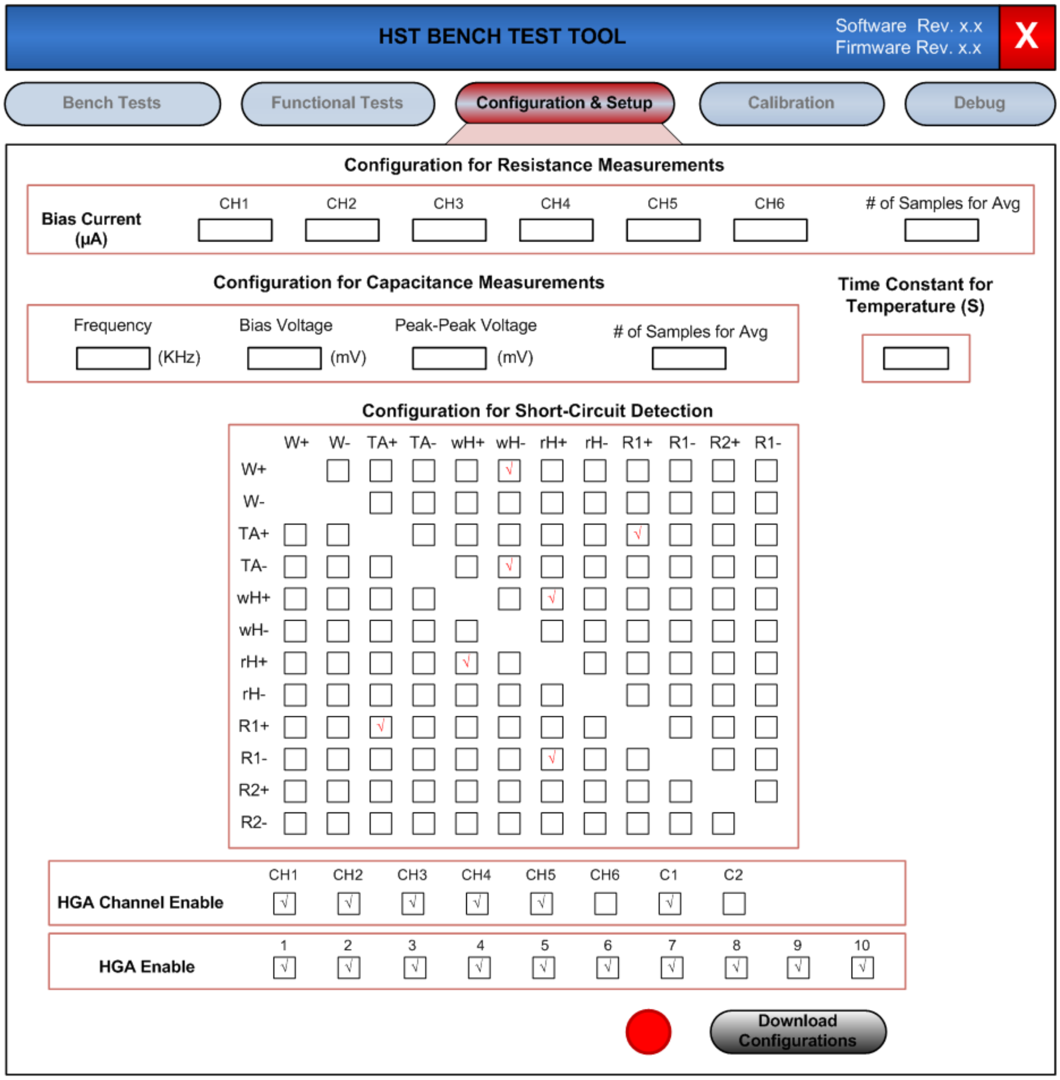


Figure 6.3 GUI Design and Requirements for Configuration and Setup

The configuration and setup for HST measurements are basically classified into 5 categories:

Configuration for Resistance Measurements
Configuration for Capacitance Measurements
Configuration for Short Circuit Detections
HGA Channel Enable
HGA Enable

Configuration for Resistance Measurement

Referring to API command "HST_config_res_meas", this is just to setup a desired bias current for each resistance measurement channel. User can use the message boxes as shown in the Figure 6.3 to key-in the exact values. The maximum values are limited to 30mA, i.e. 30000µA. If a value is out of range, the measurement uP will correct it to a default value.

Configuration for Capacitance Measurements

Refer to API command "HST_config_cap_meas" for details. The recommended default values are:

Frequency = 60KHz; Bias Voltage = 0; Peak-Peak Voltage = 1V (1000mV); # of Sample for Avg. = 4.

Depending on cycle time evaluation results, the number of samples for averaging may be changed.

Configuration for Temperature Measurements

Refer to API command "HST_config_meas", user can configure a time constant for temperature measurements. This time constant will be applied to all three temperature channels.

Configuration for Short Circuit Detections

As shown in the Figure 6.3, a 2-D matrix of Check-Box is created for user to configure pad-pairing for short-circuit detection. If a check-box is checked, the corresponding two pads need to be tested. For example, W+ and wH- have to be tested if they are shorted or not. The check-boxes in diagonal are removed because it does not make sense that a pad to be paired with itself for short detection. For ease of software programming and implementation, below are the rules for this configuration:

1. In a row, maximum of one check-box is allowed to be checked.
2. In a column, maximum of two check-boxes are allowed to be checked, because that a pad has maximum of two adjacent pads paractically.
3. In a row, if there is no check-box is checked, it means that no short-circuit detection is required for the corresponding pad in the row.
4. This configuration is product-dependent and shall be set by qualified engineers.

In the Figure 6.3, the check-boxes are checked for GrenadaBP2 (or Iris) product.

HGA Channel Enable

There are 8 check-boxes in total for HGA Channel Enable Configuration. Depending a specific product, corresponding channels shall be checked. For GrenadaBP2 product, 5 resistance channels and 1 capacitance channel are required.

HGA Enable

10 HGAs at maximum can be tested at one-shot. However, user can select or de-select tests on specific HGAs if necessary. If a check-box is unchecked, tests on the corresponding HGA are disabled.

Download of Configuration Data

As shown in the Figure 6.3, there is a command button "Download Configuration" associated with a LED indication. If there is any data or status change in any configuration message boxes or check-boxes, the LED shall become **RED** indicating to the user that latest changes are to be downloaded into measurement uP. Once user press the "Download Configuration" command button, the Host PC shall issue a set of commands (refer to "HST_config_res_meas", "HST_config_cap_meas", "HST_config_short_detection", "HST_meas_channel_enable", ...) to download these data into measurement uP. After successful download, the LED shall become **GREEN**.

Attentions to User: After hardware reset or power on/off the measurement PCBA, the configuration data displayed on this Bench Test Tool are no longer the same as what inside measurement uP. To be sure, although the LED is GREEN, it is highly recommended to re-download the configuration data into the uP.

6.5 GUI Design and Requirements for PCBA Calibrations

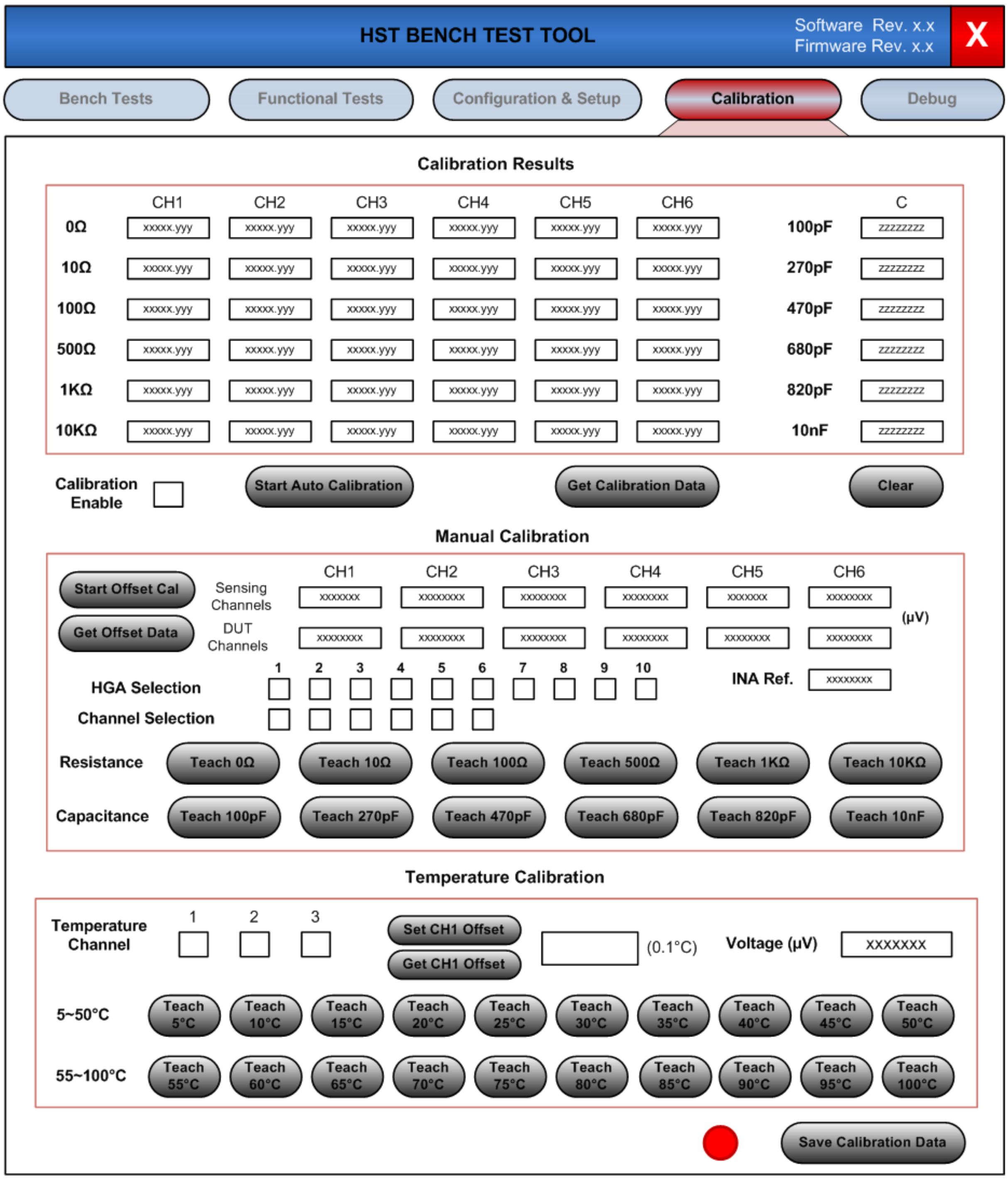


Figure 6.4 GUI Design and Requirements for HST Calibrations

The purpose of this Calibration GUI is to meet user's following needs:

1. If the PCBAs are calibrated before, the user can use GUI to read out all calibration data saved in EEPROM on PCBA board.
2. Calibration or re-calibration for the PCBA. The calibrations include: Resistance, Capacitance and Temperature.

The API commands associated with these calibrations are:

HST_calibration_enable, HST_start_auto_calibration, HST_save_calibration_data, HST_get_calibration_data, HST_manual_set_calibration, HST_set_temp_calibration, and etc.

Calibration Results

As discussed early in both hardware design and API commands, there are 6 high precision resistors and capacitors built-in on board. User can make use of these for auto calibrations; or make use of external precision resistor and capacitor decades for manual calibrations. The calibrated results (from uP's acknowledgement messages) are dynamically displayed into each corresponding text box as shown in the Figure 6.4.

For example, user pressed the command button "Start Auto Calibration". After that, the results from this Auto-Calibration are displayed in the text boxes. Later-on, if the user is not happy with these values, he can press the "Teach 0Ω" command button with channel selection = 1, the first text box corresponding to 0Ω and CH1 shall be updated with new Taught values.

In summary, the text boxes are shared between commands "Start Auto Calibration", "Get Calibration Data" and commands for Manual Calibration.

To user, these data in the text boxes are Read Only.

A "Clear" command button is required for user to clear the text boxes to be blank.

A Check-box is required to enable or disable calibration. All calibration command buttons are allowed only if the calibration is enabled. Otherwise, the uP will response Error to the Host PC.

Manual Calibration

This meant for user to perform manual calibrations using external precision resistor and capacitor decades. Prior to manual calibration for any external specific resistance value, user has to perform Offset Calibration first for all sensing channels and DUT channels. If an Auto Calibration has been performed before, user can use Get Offset Data command button to read out previous calibrated offset values. The offset values from command either Start Offset Calibration or Get Offset Data are all displayed in the text boxes as shown in the Manual Calibration section of Figure 6.4.

There are two sets of check-boxes: HGA Selection and Channel Selection.

HGA Selection:	To specify which HGA input is currently connected with external resistor or capacitor decade. There are 10 possible selections. Only one can be selected.
Channel Selection:	To specify which channel is currently connected with external resistor or capacitor decade. There are 6 possible selections for resistor and 2 for capacitor. Only one can be selected.

There are two sets of 6 "Teach xx" command buttons for resistance and capacitance calibration. When one of the button is pressed, the results from uP shall be updated in a corresponding test box in Calibration Results.

Temperature Calibration

There are 3 temperature channels in total. User has to use the check-box as shown in Figure 6.4 to select one of them at a time.

There are 20 "Teach xx" command buttons in total, which cover the temperature range of 5~100℃ with interval of 5℃. Each time after user issues one of the commands, the measured voltage result fro uP is updated in the "Voltage" text box as shown. To complete a calibration for a specific temperature channel, all 20 button commands have to be issued at least once.

Besides the "Teach xx" buttons, another two command buttons are added associated with a text box, i.e. "**Set CH1 Offset**" and "**Get CH1 Offset**". These two commands correspond to the two API commands: "HST_set_temp1_offset" and "HST_get_temp1_offset". In the hardware design, temperature Channel 1 (CH1) uses an on-board sensor which is different from other two channel sensors. These two commands are used for user to manually key in CH1 offset correction value vs. the other two channels. The offset correction value is a Signed Value (8-bit). Use can use the Text Box to set the offset value. If the "Get CH1 Offset" button is clicked, the Text Box shall be updated with the actual value read from uP.

Save Calibration Data

This command is intend for uP to save all calibration data in to EEPROM. There is a LED associated with the "Save Calibration Data" command button. If there is any data change in calibration results or a new calibration command button is pressed, the LED shall become **RED**. After successful "Save Calibration Data" command, this LED shall turn **GREEN**.

6.6 GUI Design and Requirements for Cable Calibration

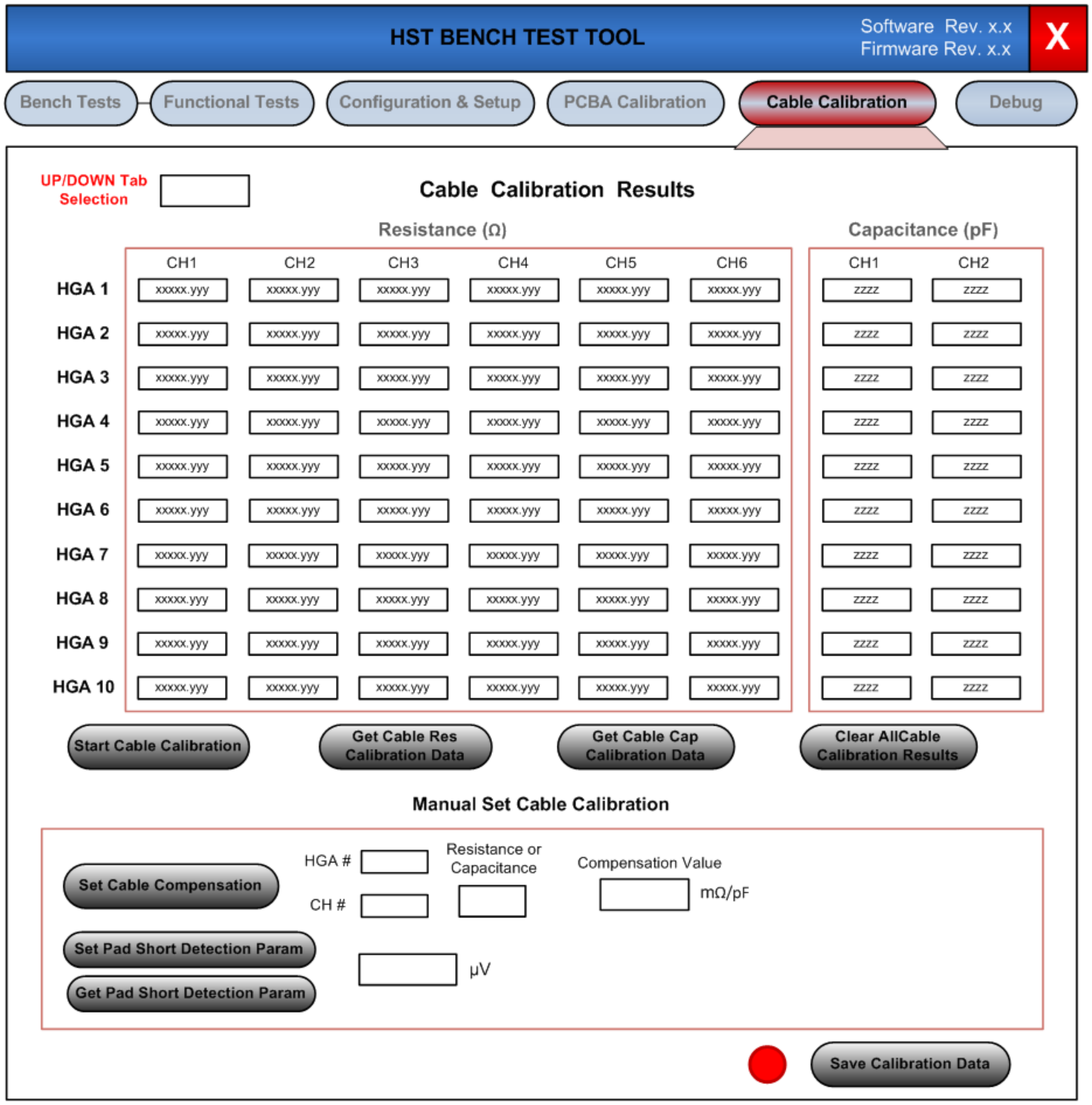


Figure 6.5 GUI Design Requirements for Measurement Cable Calibration

To achieve measurement accuracy of 0.25% for resistance and 0.5% for capacitance, it is essential to have calibration for measurement cables. The calibration is basically to measure the cable resistance and capacitance on each channel of each HGA. A set of resistances and capacitances obtained from the calibration are to be stored in the EEPROM on measurement uP board. Later on, these calibration data are used for the correction of actual measurement results on HGAs.

Prior to this cable calibration, user has to:

1. Complete the PCBA calibration using “Start Auto Calibration” command.
2. Attach the Functional PCBA or Calibration PCBA (all with 0Ω resistors and 100pF high accuracy capacitors on board to the end of measurement Flex Cable.

The GUI requirements are shown in the Figure 6.5 above.

UP/DOWN Tab Selection

Before any cable calibration, user first has to select which cable to be calibrated, UP Tab or Down Tab. This selection serves as the command input parameter for all command buttons on this calibration page: “Start Cable Calibration”, “Get Cable Calibration Data” and “Set Cable Compensation”.

When launching this Bench Test Tool, the cable selection can be default to UP Tab and explicitly displayed in its Text Box.

Start Cable Calibration

This is a command button associated with “HST_flex_cable_calibration” command. After successful completion of execution, the uP returns all cable measurement results in unit of mΩ for resistance and pF for capacitance. For resistance, the results shall be displayed in unit of Ω with 3 digits after decimal point. For capacitance, the result shall be displayed in unit of pF.

Get Cable Res Calibration Data

This command button is associated with “HST_get_cable_res_calibration_results” command. Once this command is received from Host PC, the measurement uP returns the resistance calibration data which are currently stored in its RAM. The data can be either from EEPROM read at power-on or latest cable calibration. The results of this command shall be updated and displayed in the message boxes accordingly as shown in the Figure 6.5.

Get Cable Cap Calibration Data

This command button is associated with “HST_get_cable_cap_calibration_results” command. Once this command is received from Host PC, the measurement uP returns the capacitance calibration data which are currently stored in its RAM. The data can be either from EEPROM read at power-on or latest cable calibration. The results of this command shall be updated and displayed in the message boxes accordingly as shown in the Figure 6.5.

Clear Cable Calibration Results

This command button is associated with “HST_clear_all_cable_compensation” command. When this command button is activated by user, the Host PC just sends the command to uP only. After receiving this command, the uP will reset all cable calibration resistance and capacitance to 0 internally.

If user pressed the command button “Get Cable Calibration Data” after this command, the data in calibration results Test Boxes shall be set all 0.

Set Cable Compensation

This command button is meant for user to manually adjust the cable calibration resistance or capacitance if necessary. It is associated with “HST_set_cable_compensation” API command. Prior to issuing this command to uP, user has to specify the HGA index, Channel index and the selection of Resistance or Capacitance. When this command is received, the uP will set the internal specific cable calibration variable to user specified value.

After this command, if the user presses the command button “Get Cable Calibration Data”, the new results will be returned from uP and updated in the message boxes.

Set Pad Short Detection Param

This command button is associated with API command "HST_set_short_detection_threshold". It is meant for user to optimize the threshold voltage for pad short detection when necessary in future. In the Test Box, the user can specify the desired voltage in unit of µV and click this command button. Then this new value is saved into the RAM of measurement uP. To save this new value for future tests, the user has to press "Save Calibration Data" to EEPROM.

To decide a correct value for short detection, it requires certain experience and domain knowledge. In order to prevent an unconscious click of this command button, the GUI shall initialize this text box to a default value of 2000, i.e. 2000µV.

Get Pad Short Detection Param

This command button is associated with the API command "HST_get_short_detection_threshold". When this button is clicked, a value will be returned from the measurement uP. The Test Box shall be updated with the newly returned value from uP.

Save Calibration Data

This command button is the same as the one in previous calibration Tab Page as shown in the Figure 6.4. It is nice but not compulsory for this Cable Calibration Tab Page, because user can use the same command button on previous calibration Tab Page to save the calibration data.

6.7 GUI Design and Requirements for Precisor Capacitance Compensation

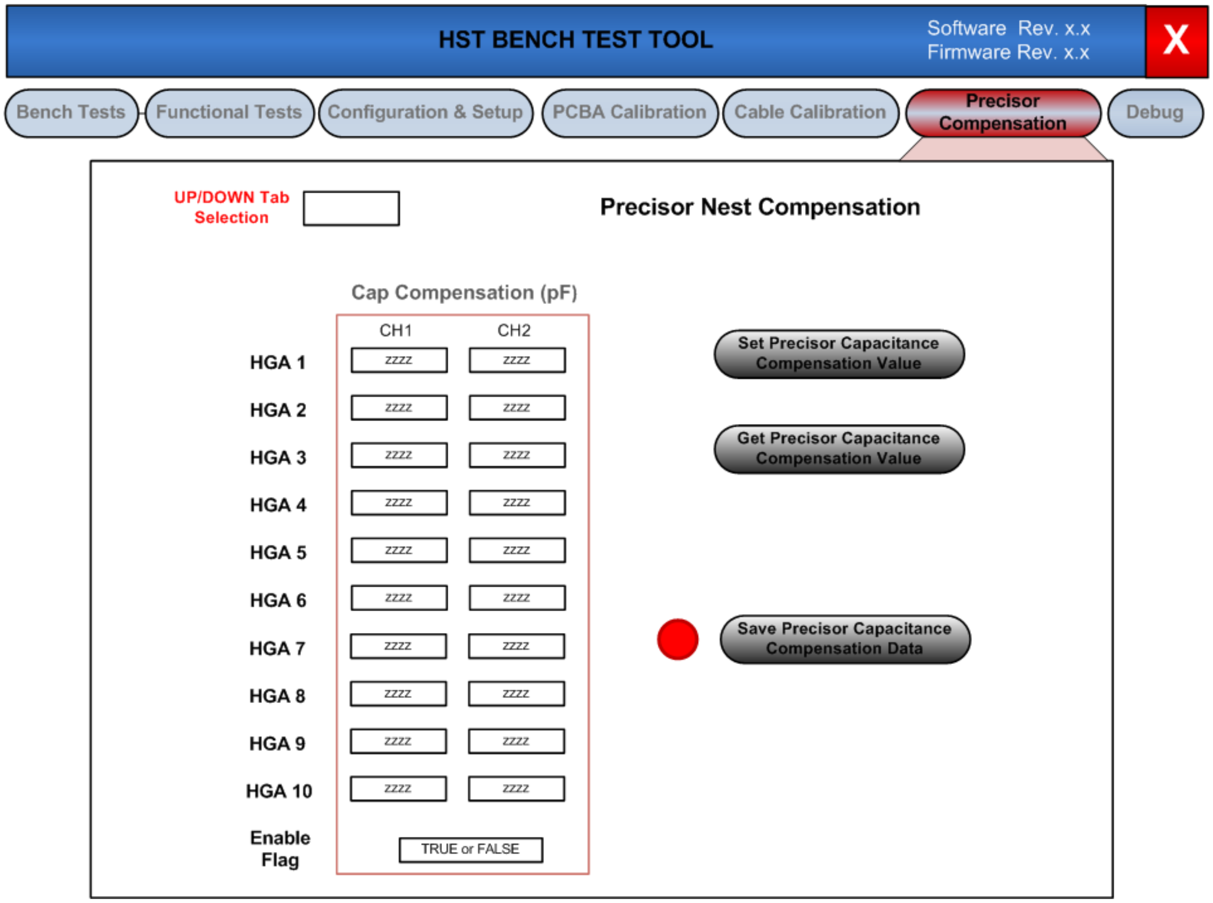


Figure 6.6 GUI Design Requirements for Precisor Nest Compensation

When 10 HGAs are loaded onto precisor nests, the ground pads of all HGAs become actually short-connected together through precisor base plate (stainless steel). This causes different offsets on capacitance measurements on each HGA. These offsets have to be compensated during capacitance measurements in order to achieve ±10pF accuracy.

UP/DOWN Tab Selection

Before setting any precisor compensation values, user first has to select Precisor Tab is, UP Tab or Down Tab. This selection serves as the command input parameter for all command buttons on this page: “Set Precisor Capacitance Compensation” and “Set Precisor Capacitance Compensation”.

When launching this Bench Test Tool, the precisor tab selection can be default to UP Tab and explicitly displayed in its Text Box.

Set Precisor Capacitance Compensation

This command button is associated with the API command “HST_set_precisor_cap_compensation”. When this button is pressed, the Host PC sends the API command to measurement uP and the command parameters are from corresponding test boxes on the left. The values in the text boxes are keyed in by user and they are **signed** values.

Get Precisor Capacitance Compensation

This command button is associated with the API command “HST_get_precisor_cap_compensation”. It is meant for user to read out the latest compensation values stored in measurement uP. The returned values from uP shall be displayed accordingly in each of the text boxes on the left.

Enable Flag

This is similar to the test box for compensation capacitance, except that it is 8-bit Unsigned Byte value. User has to set it to either 1 or 0. When user press the command button “HST_set_precisor_cap_compensation” or “HST_get_precisor_cap_compensation”, this value is downloaded to or read from the measurement uP.

Save Precisor Capacitance Compensation Data

This command button is associated with API command ““HST_save_precisor_cap_compensation”. When this button is triggered, the host PC sends the API command to measurement uP to save all latest capacitance compensation data into its EEPROM.

6.8 GUI Design and Requirements for Function Debugging

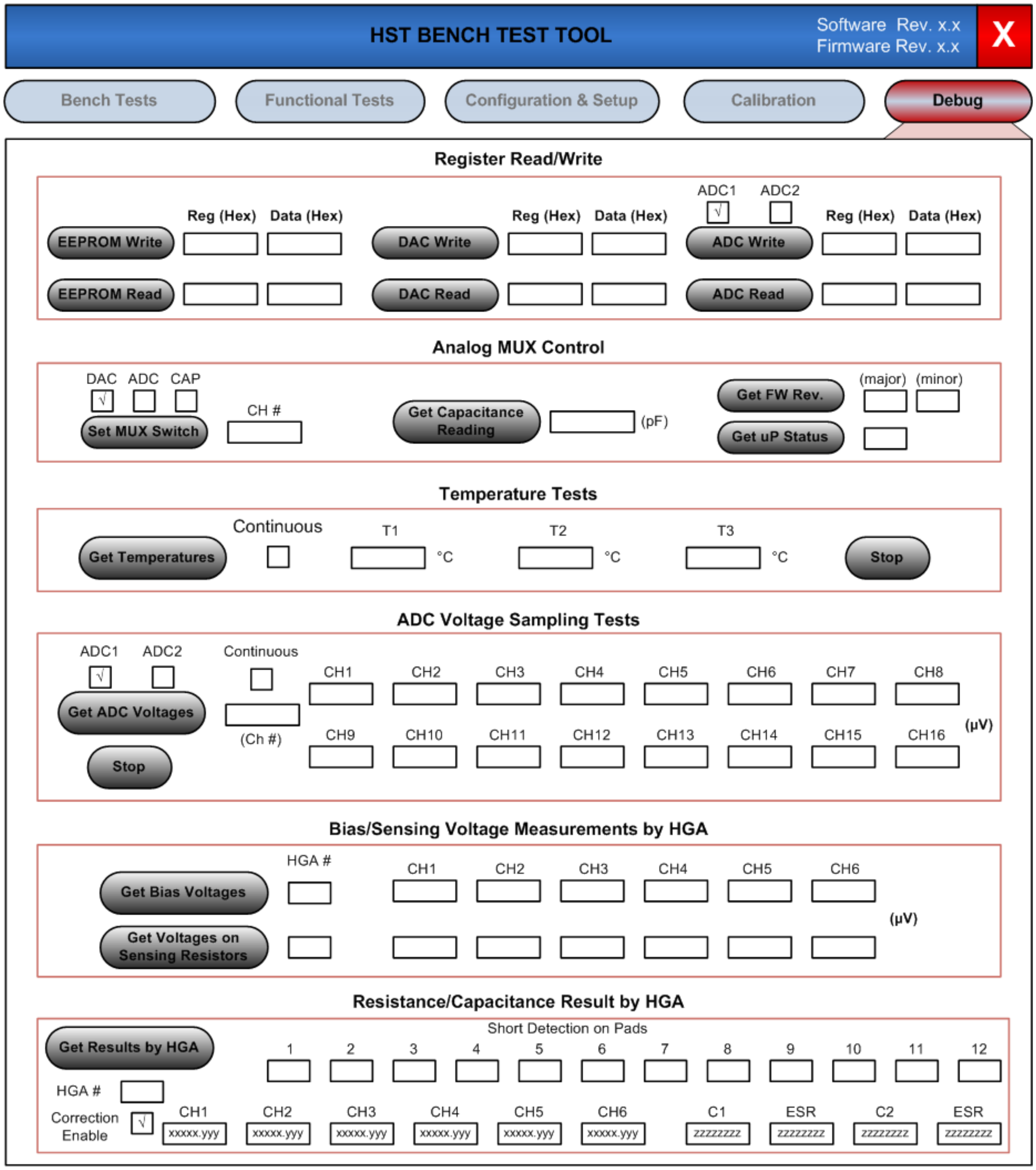


Figure 6.7 GUI Design and Requirements for Function Debugging

When integrating Software, Firmware, PCBAs and other devices all together, it is not guaranteed that everything is correct at first time. The main purpose of this Debug GUI is for user to test all functions at low hardware levels.

Basically all necessary functions on this GUI are classified into 5 groups: Register Read/Write, Analog Mux Control, Temperature Tests, ADC Voltage Sampling Tests and Bias/Sensing Resistor Measurements by HGA.

Register Read/Write

This is the most basic functionality to be tested in order for other functions to work. There is no any protection for the contents key-in by user. It is the user's responsibility for any consequence. The GUI here is just to make sure that the data and address are correctly passed onto the measurement uP.

EEPROM Read/Write: For Write operation, user has to key-in both register address and data byte. For Read operation, only the register address has to be key-in by user. The text/data box shall be Read Only and is for display of acknowledgment data from uP. For convenience, all address and data shall be key-in and displayed in Hex format.

DAC Read/Write: This is similar to EEPROM Read/Write.

ADC Read/Write: There are two ADCs in hardware design, ADC1 for resistance measurement and ADC2 for short-circuit detection. User has to select the target ADC using the two check-boxes before sending any commands. The two selection check-boxes are exclusive to each other, i.e. if one is selected, then the other one is automatically de-selected.

Associated API commands are HST_eerpom_read, HST_eeprom_write, HST_dac_read, HST_dac_write, HST_adc_read and HST_adc_write.

Analog MUX Control

There are three switching functional blocks for DAC, ADC and Capacitance Relays. Before activation of the command button "Set Mux Switch", user has to select one of the functional blocks using the check-boxes. These three check-boxes are exclusive to each other, if one is selected; the other two are automatically de-selected. Only one can be selected at a time. After selection, the user has to key in the desired channel # to be switched to. Refer to API command HST_set_mux for details.

Besides the MUX control, other three commands, "Get Capacitance Reading", "Get FW Revision" and "Get uP Status" are grouped in this GUI section. The text/data message boxes for these three commands shall be Read-Only.

Temperature Tests

When user activates the command button "Get Temperatures", the Host PC shall read back temperatures from all the three channels using API command HST_get_temperature.

A check-box is available for user to enable continuous read back temperatures from the three channels. This meant for evaluation on the digital low-pass filter design in uP. If continuous measurement is enabled, the Host PC shall read back temperatures from uP continuously at regular basis of every 200ms.

A "Stop" button is for user to terminate the continuous temperature measurement process if it was initiated.

ADC Voltage Sampling Tests

The associated API command is HST_get_adc_voltages.

Before activating the command button "Get ADC Voltages", user has to select the target ADC using the two check-boxes, ADC1 and ADC2. These two are exclusive to each other and only one is allowed at a time.

After that, user has to specify the channel number. The channel number can be 1~16 or 255. If a value between 1~16 is selected, only one channel input voltage is sampled by uP and other channel input values are forced to 0 by uP. If the channel number is 255, the uP will reads all 16 channel inputs and acknowledged back to the Host PC. The Host PC just need to display these data into their corresponding text/data boxes. The text/data boxes "CH1~16" shall be Read-Only to user. It is recommended to have a drop-down selection manual for user to select channel number, rather than using keyboard.

Besides a single measurement, there is a check-box "Continuous" for user to enable continuous measurements for analog inputs. Once this box is checked, the Host PC continuously sending API command HST_get_adc_voltages to uP at regular basis of 50~200ms. The read back results are dynamically displayed in the text/data message boxes accordingly.

Associated with the Continuous Measurements, a "Stop" command button is required to terminate continuous measurement process.

Bias/Sensing Voltage Measurements by HGA

There are two command buttons available for user to read out the measured voltages on DUT and sensing resistors. Typically these commands shall be used after "Start Single Measurement" or "Start Continuous Measurements" on the Bench Tests GUI Tab.

Before activation of these command buttons, user has to key-in the desired HGA index #. Valid value is 1~10. Text/data message boxes are for displaying measurement results from uP and these boxes shall be Read-Only for user.

Resistance/Capacitance Results by HGA

As discussed in Calibration, there are two types of measurement results, one is the result calculated from raw data; and the other one is the result corrected with calibration data. The purpose of this GUI is to evaluate the calibration effectiveness. As shown in the Figure 6.5, user has to select the desired HGA for evaluation, and also inform the measurement electronics what type of results are interested, raw or corrected results, using a check-box in the GUI.

6.9 Notes on Bench Test Tool GUI Programming

RS232 Buffer Size and Flushing on Host PC

The buffer size for RS232 on Host PC shall be configured as 256 bytes at minimum. Upon launching of this HST Bench Test Tool application, the buffer contents shall be flushed and make sure no inherited data inside the buffer.

HST Configuration and Setup Data

Referring to the GUI design for HST Configuration & Setup, all these data shall be automatically saved into a *.ini file on hard disk at the moment when this HST Bench Test Tool application software is closed.

At the moment of launching this application software, these configuration/Setup data shall be read from the *.ini file and automatically loaded into the GUI for display.

Initialization of GUI Display

Except for the Configuration/Setup data, text/data message boxes shall be initialized to be blank, NULL or 0, whichever is appropriate.

All check-boxes can be optionally initialized to default status but must be displayed correctly.

Drop-down Manual for Limited Choices of Inputs

For some parameters (such as channel index #) which have limited input variables, a drop-down choice manual is preferred than using keyboard.

7. Evaluation and Test Plan

The evaluation tests can be basically classified into three groups: Software tests at API level, Hardware level (functional tests for components and circuits), and the measurement performance evaluation at integration level.

7.1 Software Tests at API Level

When the Host PC software and uP firmware are ready, all the 34 API commands as defined in Chapter 4 have to be tested and verified without error. The main focuses are protocol, command ID, command parameters, acknowledged status, acknowledged parameters and etc.

The test tools to be used are HyperTerminal, R32C117 uP E30 Emulator, and etc.

7.2 Functional Tests for Component and Circuits

This hardware level evaluation includes:

1. Power Supply
   All power supplies must be within ±5% of their designed values. Not short connection between VCC_5V and VDD_5V.

2. uP Crystal Clock Frequency
   It shall be within ±1% of its designed value: 16MHz. uP internal shall be around 96MHz.

3. Power-on and Hardware Reset Time
   It shall be within range of 100~200ms.

4. SPI Clock Frequency
   The SPI clock frequency for EEPROM, DAC and ADC shall be about 1MHz, or 1Mbps.

5. Serial flash via Host PC RS232 interface.

6. All reference voltages for DAC, ADC and instrument amplifiers shall be within ±0.5% of their designed value. The noise ripples shall be below 20µV when no SPI communication.

7. Write and Read function check with EEPROM, DAC and ADCs through SPI interfaces.

8. Functional checks on DAC, ADCs and Analog Switches.

9. Sampling check on all critical resistors and capacitors.

10. Functional check on GPIOs for both sinking and sourcing configurations.

Test tools to be used are: multi-meters, oscilloscope with differential probes, standalone GUI software and etc.

7.3 Performance Evaluation at Integration Level

1. Bias current range, resolution and accuracy for all 6 DAC channels.

2. Resistance measurement range and accuracy.

3. Capacitance measurement range and accuracy.

4. Temperature measurement range and accuracy.

5. Individual cycle time for resistance measurements and capacitance measurements.

6. Total cycle time for sequential resistance and capacitance measurements.

7. Total cycle time for con-current resistance and capacitance measurements.

8. Repeatability measurements on resistance and capacitance.

9. Temperature test (Optional).
   Place the whole PCBAs into a temperature chamber at 45°C. Repeat the accuracy tests for resistance and capacitance.