Restricted

Siemens Healthineers
Business Area Ultrasound

Title: Temperature Measurement Software Detailed Design Description

Part Number: 11344281-EPH-002-04

Revision Data

Rev	ECO #	Change Description	Printed Name
04	**702365**	**1) CAPA update(voltage sequence)** **2) To modify and add control parameter for rev4.0**	**Hwang, InSeop**

This document is under Engineering Change Order control. The official ECO released document is maintained in SAP. Per Change Control Procedure 08266240, any printed or electronic document external to SAP is reference copy only and must be confirmed in SAP as the most recent version or if still active. Employees, supervisors, and managers own this responsibility for documents they own or use.

Revision History

Rev	Change Description	Author/Printed Name	Date Made/ECO#
04	1) CAPA update(voltage sequence) 2) To modify and add control parameter for rev4.0	Hwang, InSeop	2019.09.17 / 702365
03	To add VTx control, the automation mode change for WF3 on temperature measurement software rev 3.0	Hwang, InSeop	2017.01.30 / 660066
02	To add transmit channel modulation control on temperature measurement software rev 2.0	Hwang, InSeop	2017.01.18 / 660065
01	Initial SAP released on temperature measurement software rev 1.0	Hwang, InSeop	2017.01.16 / 657217

Table of Contents

REVISION HISTORY **2**
1.0 PURPOSE **4**
2.0 SCOPE **4**
3.0 SOFTWARE OVERVIEW **4**
4.0 DESIGN DESCRIPTION **5**
4.1 MEASUREMENT PROGRAM DESCRIPTION 5
4.2 USER INTERFACE PROGRAM DESCRIPTIONS 9
4.3 DATABASE PROGRAM DESCRIPTIONS 12
4.4 TELNET PROGRAM DESCRIPTIONS 13
5.0 TYPE DEFINITIONS **16**
6.0 QUALITY RECORDS **17**
7.0 REFERENCE **17**

1.0 PURPOSE

The purpose of this document is to describe the temperature measurement software that is used by Siemens Healthineers, Ultrasound Group.

2.0 SCOPE

This software is used to measure the temperature of transducer of ultrasound imaging systems in the research/development and in the production phases.

3.0 SOFTWARE OVERVIEW

3.1 Hardware Requirements

This software was written for an IBM PC-compatible computer.

3.2 Builds and Installation

This program was written with the LabVIEW and Database Connectivity Toolkit components of National Instruments Professional Developer Suite. For use in measurements, it is converted to an executable using the LabVIEW Application Builder. The computer which runs the executable does not need to have LabVIEW or the Database Connectivity Toolkit installed, but for each computer on which it is installed, the free LabView run-time engine must be installed. To install the executable:

- In the build directory, open the “disks” folder and run Setup.exe. Complete the installation.
- If the program has not been installed on this computer before and the install software does not put the following three files into the system folder, open the “data” folder within the program folder, and copy “lvlib.dll”, “lvsql08.dll” and “lvutl08.dll” to the system folder.
- Configure ODBC to access the acoustic database: open the control panel for ODBC data sources, and add a source under the “System DSN” tab. Select the “SQL Server” driver and follow the instructions for entering the database parameters.

3.3 User Characteristics

The software is designed to be used by an engineer or technician who is trained to perform temperature measurements.

3.4 Program Execution

The program is executed by double-clicking on its icon or selecting it in the Start menu.

4.0 DESIGN DESCRIPTION

4.1 MEASUREMENT PROGRAM DESCRIPTION

4.1.1 Thermometer data receive Programs

4.1.1.1 Data compensation.vi

This program compensates the temperature data by thermometer as using previous data if current data is missing due to communication failure.

4.1.2 Miscellaneous Programs

4.1.2.1 Thermal PRF check by PID.vi

This subroutine measures the temperature of transducer after sending PRF and voltage to ultrasound system. It consists of three sub-programs as below(Target temp.vi, PID unit_re.vi and PID Command.vi)

4.1.2.2 Target temp.vi

This subroutine supports the target temperature per specific time.

4.1.2.3 PID unit_re.vi

This subroutine calculates PRF by PID algorithm(The difference: Target temperature – Current temperature) for increasing the transducer surface temperature.

* PID: Proportional, Integral and Derivative control

4.1.2.4 PID Command.vi

This subroutine sends PRF and voltage to ultrasound system if the control in Target temp.vi is “True”.

4.1.3 Temperature Measurement Programs

4.1.3.1 Thermal measure Volt&SA.vi

This program measures the surface temperature of transducer and the ambient temperature each step. One step is voltage-temperature measurement. Another is scanRange & aperture measurement. Also measurement data point is able to change for accuracy. After the measurement is done, the result data and other information transfer to SQL database.

4.1.4 Parameter Control Programs

4.1.4.1 Unlock and limitDeg.vi

This program changes the status of parameter or temperature limitation value in ultrasound system. First of all, unlock is unlock all parameter for a possible change parameter. Second of all, probetempriselimitdeg is modified to be low value for more accuracy data while it prevents the temperature is going up during the change of parameter to ultrasound system.

4.1.4.2 GetFov.vi

This subroutine gets the RoiLateralMin and RoiLateralSpan from ultrasound system for using valuable parameter(scan_range & aperture step)

4.1.4.3 ProbeType.vi

This program gets probetype by ultrasound system for using valuable parameter each probetype.

4.1.4.4 SetZForInte_modify_Bmode_temperature_PRF&Volt.vi

This program runs voltage-dependent temperature measurements for the given list of frequency. For each frequency, the program controls frequency, channelmodulation, waveformstyle, ElevAperIndex, cycle, txelement, PRF, IsCPAEn and pulseVoltage on non-scanning mode.

1) For Waveform(0) with RLE, this subroutine controls mode change, TxPulseRLE, waveformsyle and system frequency.
2) The change of PRF is based on event control. After change PRF, software read and check data for verification.

If an error or warning was found on the previous measurement, a dialog is displayed for a user to select which measurement to run next; if there is no response, the default is to repeat a measurement once, then move on. If no errors are found, the program checks for user interrupts. The next measurement is set to the user interrupt value if an interrupt is found, or if not, the next value in the input list is taken. The program also selects spot check measurements at a given interval.

4.1.4.5 SetZForInte_modify_Bmode_temperature_SA Test.vi

This program runs scanrange & aperture(SA) dependent temperature measurements for the given list of scanrange & aperture(SA). For each SA step, the program controls frequency, channelmodulation, waveformstyle, ElevAperIndex, RoiLaterSpan, RoiLaterMin, cycle, txelement, PRF, IsCPAEn and pulseVoltage on scanning mode.

1) For Waveform(0) with RLE, this subroutine controls TxPulseRLE, waveformsyle and system frequency.

If an error or warning was found on the previous measurement, a dialog is displayed for a user to select which measurement to run next; if there is no response, the default is to repeat a measurement once, then move on. If no errors are found, the program checks for user interrupts. The next measurement is set to the user interrupt value if an interrupt is found, or if not, the next value in the input list is taken. The program also selects spot check measurements at a given interval.

4.1.4.6 AutoTemp.vi

This program executes a list of temperature measurements for different ultrasound system settings. Given a list of measSetId for the measurements to be made, the program first retrieves the corresponding measurement conditions from the database. An interrupt window is displayed for a user to reorder/stop measurements, and measurements are run using TempMeasOneSetZ.vi. After each one, PickNextMeasZ.vi is used to select the next measurement. After the measurements end, the result IDs (temperatureId) and any errors or warnings are displayed.

4.1.4.7 TopTemperatureMeasure.vi

This is the top level control and user interface program for automatic temperature measurements. It displays a menu for the user to select an action, and calls the corresponding subroutine. The menu options include:

- Database: to save data value specific location
- measSetId(Voltage): non-scanning temperature measurement for voltage dependence
- measSetId(SA modeling): Scanning temperature measurement for scanrange&aperture dependence
- Z address: Ultrasound IP address
- S/N(System, Transducer) & S/W: measurement information as system serial number, transducer serial number and software version.
- PID control: The selection of PID target temperature
- System Selection: measurement system type
- VISA resource and Thermometer type: Thermometer type and communication port
- Log folder and Logfile name: measurement log history

4.1.4.8 calcVoltagesZ.vi

This subroutine calculates the list of voltages for voltage-dependent measurements.

```
    For i = 0 to numMeasVoltage-1,
    V[i] = maxTxVoltageVolt^((totalVoltagePt-1-i)/(totalVoltagePt-1))
```

Any voltages which are greater than ceilTxVoltageVolt are removed and replaced by ceilTxVoltageVolt. For the prevention of transducer damage and CAPA action, the sequence of voltage is modified from low voltage to high voltage. Since a momentary high voltage may occur to the damage of transducer.

4.1.4.9 GetElementNum.vi

This subroutine calculates the list of element for scanrange & aperture dependent measurements.

4.1.4.10 Cal Fov.vi

This subroutine calculates the list of scanrange for scanrange & aperture dependent measurements.

4.1.4.11 Calculate PRF.vi

This subroutine calculates the list of PRF for temperature measurements. It helps to prevent the damage of transducer.

4.1.4.12 VTxIndex_control.vi

This subroutine sets the environment of voltage before temperature measurement.

4.1.5 The Verification of parameter Programs

4.1.5.1 Command_verification.vi

This subroutine run to verify all parameter by comparison between input parameter and read parameter on non-scanning mode. It consists of two parts:

1) Verification step
   channelmodulation, RLE(waveform 0) data, TxWaveformSyle, FocusRangeCm, NumTxcycle, NumTxElement, SampleRateHz, Pulsevoltage, Txfrequency and ElevAperSwitchDepth.
2) Result and action step
   If the comparison data is not match, system freeze is first. Second is running of end of telnet. Last sequence is pop up

message “parameter are not match”. For matching case, run measurement sequence.

4.1.5.2 Command_verification_SA test.vi

This subroutine run to verify all parameter by comparison between input parameter and read parameter on scanning mode. It consists of several two parts:

1) Verification step
channelmodulation, RLE(waveform 0) data, TxWaveformSyle, FocusRangeCm, NumTxcycle, NumTxElement, SampleRateHz, Pulsevoltage, Txfrequency, ScanRange and ElevAperSwitchDepth.
2) Result and action step
If the comparison data is not match, system freeze is first. Second is running of end of telnet. Last sequence is pop up message “parameter are not match”. For matching case, run measurement sequence.

4.2 User interface program descriptions

4.2.1 Menu Input and Display Subroutines

The programs shown in

Table 1 are windows which take a user’s input for an action to be performed by a subroutine, run that subroutine, and display any output. Each of these windows may be selected from the TopTemperatureMeasure.

Table 1: Menu input and display window programs

Input/display program	Subroutine called
changeTempdatabase.vi	AutoTemp.vi
InputLimitVoltage.vi	AutoTemp.vi
Setting_target_temp.vi	AutoTemp.vi
Thermal PRF check by PID.vi	SetZForInte_modify_Bmode_temperature_PRF&Volt.vi SetZForInte_modify_Bmode_temperature_SA Test.vi Target temp.vi Data compensation.vi PID unit_re.vi PID Command.vi
Thermal measure Volt&SA.vi	SetZForInte_modify_Bmode_temperature_PRF&Volt.vi SetZForInte_modify_Bmode_temperature_SA Test.vi

	Data compensation.vi MeasResSummToInsertZ JH.vi

4.2.2 Input Subroutines for Starting Complete Temperature Measurements

4.2.2.1 change database.vi

This window accepts user input for the database name, user name and password.

4.2.2.2 InputLimitVoltage.vi

This window accepts user input for measurement constants, including input limit voltage and CW mode selection.

4.2.2.3 Setting_target_temp.vi

This window sets the target temperature each transducer type, initial PRF on PID step and PID On/Off switch.

4.2.3 Input/Display Subroutines Used During Temperature Measurements

4.2.3.1 InterruptMeasZ.vi

This program displays the list of measSetIds being measured, and allows the user to select which measurement will be run next. It is designed to be called once at the beginning of a measurement, to display the ID list, and once at the end to read any interrupts. When it is called, it displays the input ID list updated with markers showing the current measurement and next measurement. For reading interrupts, it may be called with an empty input ID list, and the displayed list doesn’t change. It returns the user interrupt state and next measurement.

4.2.3.2 DisplayResIdErrorZ.vi

This program displays results of measurements which have already been completed, while further measurements are being run. The measurement setting and result IDs (measSetId and temperatureId) are displayed along with any errors or warnings found. When the program is called, the input results are added to the previous list, which is maintained using an uninitialized shift register. If the “First iteration” input is true, the previous list is cleared.

4.2.3.3 TimedDialogZ.vi

This subroutine displays a dialog message and a list of options for a user, and returns the user selection, or returns –1 if no user responds in a given time. A user should select an option and click Enter before the displayed time remaining reaches zero, or click on “more time”. The “options” input sets the default user option, and the “dialog sec” input sets the initial time allowed (and the increment for “more time”).

4.2.3.4 ErrorHandlerZ.vi

This program is a revised version of the standard Labview error program, “General Error Handler.vi”. The original program uses the standard Labview error cluster, and informs the user of error occurrences and details. The modified version allows different actions to be selected in response to an error, and waits for a limited time for user input before selecting a default action. If type of dialog = 2, it presents a timed dialog box for user to select a response, with time given by the dialog sec input. Dialog box options can be set by wiring dialog prompts input (default options are Continue (0) or Stop (1)). The user selection is returned in selection/timeout:

-1	timeout
0	no error
1	first option selected
2, 3, ...	additional user options

The error and response are also sent to the log file.

4.2.3.5 DisplayAndLogZ.vi

This subroutine writes messages to a log file and/or to a display window. The action input has the following options:

0 Write message to file and display window
1 Write to file only
2 Write to display window only
3 Clear display window; write message to file and window

The log file write is omitted if the log file name equals the name shown in the “skip file write” field on the front panel.

4.3 Database program descriptions

The subroutines in this section call the Labview Database Connectivity (SQL) Library programs to read from and write to the acoustic database (see Reference 3). They include many copies of templates to convert between Labview clusters and SQL statements.

4.3.1 Programs for saving data to database

These programs generally use the same procedure for saving data to the database: they use a template subroutine to convert a cluster of data to a SQL statement, then use SQL library programs to execute the statement and retrieve the ID number which references the stored data.

4.3.1.1 Save measurement results

The results of one measurement set are saved by thermal measure Volt&SA.vi. The program saves summary data to the table temperature on database, and uses subroutines to save other data clusters to other tables (generally within loops since multiple points are saved). Each save routine also calls a cluster conversion routine, which creates an insert SQL statement. The save and conversion subroutines and their database tables are detailed in

Table 2.

Table 2: Subroutines for saving measurement results, and data saved.

Save subroutine	Cluster conversion	Database table
Thermal measure Volt&SA.vi	MeasResSummToInsertZ JH.vi	temperature

4.3.2 Read data from database

This program retrieves data by using a template subroutine to convert a cluster definition to a select SQL statement, then using SQL library subroutines to execute the statement. The result is converted to a cluster of the defined type using another template subroutine.

4.3.2.1 getMeasSetforTemperature.vi

This program retrieves an array of measurement settings and a cluster of accessory parameters from the database. The input measSetId list may include multiple IDs separated by commas, and ranges specified with dashes. One cluster of data is retrieved from meas_setting for each ID in the list, and an array of these clusters is returned.

4.3.3 Other database subroutines

Two of the Labview Database Connectivity (SQL) Library subroutines were modified slightly for this application, and saved as new subroutines. The template subroutines call these programs for conversion between clusters and SQL statements.

4.3.3.1 Parse Control Data Z.vi

This program is a modification of the SQL Library's subroutine Parse Control Data.vi; the only change is that it saves floating point numbers in exponential format rather than fractional. This makes in easier to insure that enough precision is used for very small numbers.

4.3.3.2 Parse Cluster Z.vi

This subroutine is a modification of the SQL Library's subroutine Parse Cluster.vi. The only change is that it calls Parse Control Data Z.vi rather than Parse Control Data.vi.

4.4 Telnet program descriptions

The programs in this section are used to set up the Z system for temperature measurements, via a Telnet interface. The Z system parameters used, and their location in the menu structure, are shown in Table 3. For K2 project, the name of location is changed from *"beamforming"* to *"Fe"*

Table 3: Z system parameters used for setting up temperature measurements

menu	name	description
unlock	unlock	Unlock all parameter on Z system
Hardware Control Layer. Xp. Common.	ProbeTempRiseLimitDeg	To control the limitation of probe temperature degree.
Hardware Control Layer. Beamforming(Fe). Common.	SysTxFreqHzA	List of possible system transmit frequencies
	PulseVoltageA	Transmit voltage
Hardware Control Layer. Beamforming(Fe). BImage.	PulseVoltageSelA	voltage channel (0 or 1)
	TxFrequencyIndexA	Transmit frequency index
Hardware Control Layer. Beamforming(Fe). DImage.	PulseVoltageSelA	voltage channel (0 or 1)
	TxFrequencyIndexA	Transmit frequency index
Hardware Control Layer. Xp. BImage.	BeamStyleIndexA	Current operation beamstyle index
Hardware Control Layer.	TxWaveformStyleA	waveformstyle

Xp. BImage.	PulseRepetitionRate	PRF
Hardware Control Layer. Xp. DImage.	TxWaveformStyleA	waveformstyle
	PulseRepetitionRate	PRF
Hardware Control Layer. Seq. BImage.	PrtMinSecA	Pulse repetition time(sec)
Imaging Control Layer. ProbeModel. DProbe.	IsTxChannelModulationEn	channel modulation
Imaging Control Layer. ProbeModel. CommonProbe.	ProbeType	Probe Type
Imaging Control Layer. ProbeModel. BProbe.	NumTxFreq	Number of transmit frequency
	SysTxfreqIndexA	List of possible system transmit frequencies
	IsTxChannelModulationEn	channel modulation
Imaging Control Layer. ImageModel. BImages. BImages0.	RoiLateralMin RoiLateralSpan	Current operation of Roi size
Imaging Control Layer. ImageModel. Common.	ExamIndex	Current operation of exam index
Imaging Control Layer. ImageModel. DImage.	NumTxCyclesA	cycles
	NumTxElementsA	elements
	SampleRateHz	samplerateHz
debug. Ic. Ic Events. Common Events.	BModeEvent	change B mode
	THIButtonEvent	Change THI mode
	DModeEvent	Change D mode event
	PresetSelectEvent opt PresetSelectEvent	Preset select event
	FreezeEvent	Freeze event
debug. Ic.	BTxFreqIndexEvent opt	B transmit frequency event option
	BTxFreqIndexEvent	B transmit frequency event

Ic Events. B Events.	DepthValueEvent	Depth value event

4.4.1 tcpReadFromZ.vi

This program reads a response from the Z system after a Telnet command has been sent. The Z response will start with a status number. The program reads until either a number has been received, or until the timeout (input) is reached. Then, the program continues reading until no more bytes are received for 20 milliseconds; however, if more than 250 characters are received, it reads until an end sequence occurs. The end sequence is two successive end-of-line marks (\r\n\r\n). Finally, it checks the status number: values outside the range 200-299 are reported as errors.

4.4.2 tcpCmdArrayToZ_modify.vi

This program sends an array of commands to the Z system, one at a time, via Telnet, and returns an array of Z system responses. The input array of delays contains a delay time corresponding to each command. (Delays of 0 are used for menu and read commands.) After each command is sent, the response is read using tcpReadFromZ.vi, then the program waits for the corresponding delay time.

4.4.3 tcpSysTxFreqFromZ.vi

This program reads all allowed transmit frequencies from the Z system, and finds the one closest to the input frequency. It creates an array of commands to read all system frequencies and sends them using tcpCmdArrayToZ_modify. It returns an array of all frequencies, and the index and value of the one closest to the input frequency.

5.0 Type Definitions

Type definitions are used for clusters or other controls which appear in multiple subroutines. This allows the controls to be updated in many locations at once by changing the type definition. Changes should always be made to the type definition, not to individual instances, and each instance should be set to automatically update from the definition. The type definitions which exchange data with the database must match the database field names and types.

Table 4 lists the type definitions used.

Table 4: Type Definitions

Name	Description (DB = database)
Database.ctl	DB login parameters
measSetting.ctl	Tx settings for one measurement (DB format)
meas_temperature _result.ctl	Result temperature data(DB format)
ZAddress.ctl	Z system Telnet parameters

6.0 Quality Records

The location of documents is SAP P41, and retention period for the documents is defined in Quality Record & Retention Procedure [3900789].

7.0 Reference

No.	Document Name	Doc. No.
1	Temperature Measurement Software Requirements Specification	11344281-EPH-001
2	National Instruments Professional Development Suite documentation, (includes LabVIEW, Database Connectivity, Application Builder, and other pertinent information)	Public document - filed reference copy available.
3	Acoustic Database Description	4848375

SAP-EDM Signature Information

- generated automatically by SAP system **P41** -

Appendix to Document: **11344281 EPH 002 04 , ECO: 702365**
Sheet generated at : **2019-11-06T05:24:45 UTC**
Originator **: SIEMENS Healthcare, P41**
Signatures related to this document and performed in SAP:

Meaning	UTC date and time	surname, given name of signee
AUTHOR	**2019-11-06T05:22:24**	**HWANG, INSEOP**
APPROVAL	**2019-11-06T05:23:40**	**HWANG, INSEOP**