

ISONIC utPod SDK

Version 1.0

Programmer's Manual

Current SDK is suitable for ISONIC utPod firmware version 1.40 and later and ISONIC utPod for PC SW package version 2.5 released on Dec 2, 2011 or later

© 2011, Sonotron NDT

Table of Contents

1 Overview	3
1.1 ISONIC utPod SDK Files List	3
1.2 Principle of operation	4
2 ISONIC utPod SDK Functions	5
2.1 UTPodCreateHandle	5
2.2 UTPodReleaseHandle	6
2.3 UTPodShow	6
2.4 UTPodEnable	7
2.5 UTPodActivateScreen	7
2.6 UTPodStartAScans	8
2.7 UTPodStopAScans	8
2.8 UTPodGetParamsDump	9
2.9 UTPodSetParamValue	9
2.10 UTPodSetFreezeState	10
2.11 UTPODSaveFile	11
2.12 UTPodLoadFile	11
2.13 UTPodShutdown	12
2.14 UTPODGetSDKVersion	12
2.15 UTPodGetSWVersion	13
3 ISONIC utPod SDK Notifications	14
3.1 UTPODMSG_CONNECTED	14
3.2 UTPODMSG_CONNECTIONFAILED	15
3.3 UTPODMSG_DISCONNECTED	15
3.4 UTPODMSG_SCREENCHANGE	15
3.5 UTPODMSG_ASCAN	16
3.6 UTPODMSG_PARAMSDUMP	17
3.7 UTPODMSG_PARAMCHANGE	17
3.8 UTPODMSG_SAVEDATA	17
3.9 UTPODMSG_LOADFAILED	18
3.10 UTPODMSG_VERSION	18
4 Structures	19
4.1 ASCANDATA structure	19
4.2 PARAMSDUMP Structure	20
4.3 PARAMCHANGE Structure	20
4.4 SAVEDATA Structure	20
5 Constants	21
5.1 MEASURE_XXX constants	21
5.2 ISONIC utPod Parameters (UTPAR_XXX Constants)	22

1 Overview

ISONIC utPod SDK allows creation of third-party software applications for remote controlling of **ISONIC utPod for PC software**. **ISONIC utPod SDK** is language-independent and can be used with any programming language allowing use of dynamic-link libraries.

1.1 ISONIC utPod SDK Files List

ISONIC utPod SDK package includes the following files:

- **utPodSDK.dll** – dynamic-link library containing all necessary functions for remote controlling of the **ISONIC utPod for PC software**
- **utPodSDK.lib** – library file
- **utPodSDK.h** – C header file
- **utPodSDK.pdf** – Programmer's manual
- **utPodSDK Test** – sample Visual Studio 2008 MFC project demonstrating all of the **ISONIC utPod SDK** features

1.2 Principle of operation

ISONIC utPod SDK uses Windows TCP sockets for communication with the **ISONIC utPod for PC software**, allowing the third-party software to connect both locally and through the network. The application creates an utPod connection handle using **UTPodCreateHandle** function, which is used in all subsequent calls to **ISONIC utPod SDK** functions. At the end of operation the application should release the utPod connection handle using **UTPodReleaseHandle** function. The application controls the **ISONIC utPod for PC software** using corresponding **ISONIC utPod SDK** functions. The responses for the **ISONIC utPod for PC software** are sent through application-defined Windows message.

ISONIC utPod SDK allows to perform the following operations:

- Show/Hide the **ISONIC utPod for PC software** window
- Enable/Disable the **ISONIC utPod for PC software** window user input
- Activate different work modes of the **ISONIC utPod for PC software**
- Change various parameters
- Receive A-Scans and Measurements
- Save and load **ISONIC utPod** files
- Shut down the **ISONIC utPod for PC software** and **ISONIC utPod unit**

Note: ISONIC utPod SDK requires ISONIC utPod for PC version 2.5 or higher. The version of the ISONIC utPod for PC software is shown at the top-right corner of the Settings screen. Versions not showing version number at the Settings screen are lower than 2.5 and therefore cannot be used with ISONIC utPod SDK.

2 ISONIC utPod SDK Functions

2.1 UTPodCreateHandle

Unicode function format

HANDLE WINAPI UTPodCreateHandle(LPCWSTR IPAddress,HWND NotifyWindow,UINT NotifyMessage)

ANSI function format

HANDLE WINAPI UTPodCreateHandle(LPCSTR IPAddress,HWND NotifyWindow,UINT NotifyMessage)

Creates utPod connection handle which should be used in all subsequent calls to ISONIC utPod SDK functions.

Parameters

- *IPAddress* – null-terminated string containing IP address or computer name of the computer running the **ISONIC utPod for PC software**. If set to NULL, **ISONIC utPod SDK** will connect to the **ISONIC utPod for PC software** on local machine
- *NotifyWindow* – handle to window which will receive notifications form the ISONIC utPod SDK through Windows message identified by *NotifyMessage*
- *NotifyMessage* – application-defined Windows message ID

Return Value

NULL if the function fails. Otherwise returns utPod connection handle which should be used in all subsequent calls to ISONIC utPod SDK functions.

The application must wait for **UTPODMSG_CONNECTED** notification prior to using utPod connection handle in subsequent **ISONIC utPod SDK** calls.

Calling application is responsible for releasing the handle using the **UTPodReleaseHandle** function.

Note: do not pass this handle to any Windows handles function (CloseHandle etc.)

2.2 UTPodReleaseHandle

Function format

VOID WINAPI UTPodReleaseHandle(HANDLE UTPodHandle)

Disconnects from the **ISONIC utPod for PC software** and releases the utPod connection handle returned by **UTPodCreateHandle** function

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**

Return value

None

Note: utPod connection handle specified by *UTPodHandle* should not be used after calling this function

2.3 UTPodShow

Function format:

VOID WINAPI UTPodShow(HANDLE UTPodHandle,BOOL ShowFlag)

Shows or hides the **ISONIC utPod for PC** window.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**
- *ShowFlag* – if TRUE, the **ISONIC utPod for PC** is shown on the screen. If FALSE, the **ISONIC utPod for PC** is hidden.

Return value

None

2.4 UTPodEnable

Function format:

VOID WINAPI UTPodEnable(HANDLE UTPodHandle,BOOL EnableFlag)

Enables or disables the **ISONIC utPod for PC** user input (mouse and keyboard).

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**
- *EnableFlag* – if TRUE, the **ISONIC utPod for PC** user input is enabled. If FALSE, the **ISONIC utPod for PC** user input is disabled.

Return value

None

2.5 UTPodActivateScreen

Function format:

VOID WINAPI UTPodActivateScreen(HANDLE UTPodHandle,SHORT ScreenID)

Activates the **ISONIC utPod for PC** screen specified by *ScreenID* parameter. If the specified screen is already active, the command is ignored. **UTPODMSG_SCREENCHANGE** notification is sent to application upon command completion.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**
- *ScreenID* – identifier of the screen to be activated. Can be set to one of the following:

Constant	Numeric Value	Screen Activated
SCREEN_FLAWDETECTOR	0	Flaw detector screen
SCREEN_THICKNESSCALIB	1	Thickness Calibration screen
SCREEN_THICKNESSMEAS	2	Thickness Measurement screen
SCREEN_CORROCALIB	3	Corrosion Calibration screen
SCREEN_CORROMEAS	4	Corrosion Measurement screen
SCREEN_OTHER	5	Main ISONIC utPod for PC screen

Return value

None

2.6 UTPodStartAScans

Function format:

VOID WINAPI UTPodStartAScans(HANDLE UTPodHandle)

Instructs the **ISONIC utPod for PC** software to start sending A-Scans and measurements read from the **ISONIC utPod** device. The **ISONIC utPod SDK** application will start sending A-Scans and measurements to the application through **UTPODMSG_ASCAN** notification.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**

Return value

None

2.7 UTPodStopAScans

Function format:

VOID WINAPI UTPodStopAScans(HANDLE UTPodHandle)

Instructs the **ISONIC utPod for PC** software to stop sending A-Scans and measurements read from the **ISONIC utPod** device.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**

Return value

None

2.8 UTPodGetParamsDump

Function format:

VOID WINAPI UTPodGetParamsDump(HANDLE UTPodHandle);

Requests from the **ISONIC utPod for PC** a list of all current parameters values relevant to **ISONIC utPod SDK**. The response is sent to the application through **UTPODMSG_PARAMS_DUMP** notification. This command will be ignored if the **ISONIC utPod for PC** is currently not in Flaw Detector, Thickness or Corrosion operation modes.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**

Return value

None

2.9 UTPodSetParamValue

Function format:

VOID WINAPI UTPodSetParamValue(HANDLE UTPodHandle,SHORT ParamID,FLOAT ParamValue);

Sets value of parameter specified by *ParamID* to the value specified by *ParamValue*. This command will be ignored if the **ISONIC utPod for PC** is currently not in Flaw Detector, Thickness or Corrosion operation modes. Upon successful change of parameter value, ISONIC utPod SDK will send **UTPODMSG_PARAMCHANGE** notification, specifying actual value to which the specified parameter was set. This command will be ignored if the **ISONIC utPod for PC** is currently not in Flaw Detector, Thickness or Corrosion operation modes. Note: parameter value sent with **UTPODMSG_PARAMCHANGE** notification may differ from value sent by application.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**
- *ParamID* – parameter identifier. See p. 5.2 for the list of parameter identifier
- *ParamValue* – desired parameter value. See p. 5.2 for details

Return value

None

2.10 UTPodSetFreezeState

Function format:

VOID WINAPI UTPodSetFreezeState(HANDLE UTPodHandle,SHORT FreezeState);

Changes current freeze state of the **ISONIC utPod for PC** software. This command will be ignored if the **ISONIC utPod for PC** is currently not in Flaw Detector, Thickness or Corrosion operation modes.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**
- *FreezeState* – desired freeze state. Can be one of the following:

Constant	Numeric Value	Meaning	Notes
FRZ_UNFREEZE	0	Live A-Scan	
FRZ_FREEZE	1	Frozen A-Scan	Can be sent only when current freeze state is FRZ_UNFREEZE
FRZ_FRZPEAK	2	Freeze Peak mode	Can be sent only when current freeze state is FRZ_FREEZE or FRZ_FRZPEAKFREEZE
FRZ_FRZPEAKFREEZE	3	Frozen Freeze peak	Can be sent only when current freeze state is FRZ_FRZPEAK

Return value

None

Note: **ISONIC utPod SDK** continues sending **UTPODMSG_ASCAN** notification even if the **ISONIC utPod for PC** is in **FRZ_FREEZE** or **FRZ_FRZPEAKFREEZE** state. In this cases the A-Scan sent with notification remains unchanged, but some measurements can change while in "frozen" state.

2.11 UTPODSaveFile

Function format:

VOID WINAPI UTPODSaveFile(HANDLE UTPodHandle);

Requests from the **ISONIC utPod for PC** software raw data for saving file in **ISONIC utPod** format. ISONIC utPod SDK will send this data through **UTPODMSG_SAVEDATA** notification. This command will be ignored if the **ISONIC utPod for PC** is currently not in Flaw Detector, Thickness or Corrosion operation modes.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**

Return value

None

2.12 UTPodLoadFile

Function format:

VOID WINAPI UTPodLoadFile(HANDLE UTPodHandle,DWORD DataLength,LPBYTE FileData);

Loads raw data from **ISONIC utPod** file into the **ISONIC utPod for PC** software. Upon successful file load, **UTPODMSG_PARAMS DUMP** will be sent to the application by **ISONIC utPod SDK**. If the file load fails, **UTPODMSG_LOADFAILED** notification will be sent by to the application by **ISONIC utPod SDK**. This command will be ignored if the **ISONIC utPod for PC** is currently not in Flaw Detector, Thickness or Corrosion operation modes.

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**
- *DataLength* - size, in bytes, of data buffer specified by *FileData* parameter
- *FileData* – buffer containing raw ISONIC utPod file data

Return value

None

Note: It is important to use this function for loading files appropriate for the current ISONIC utPod for PC operation mode. For example, do not use this function for loading files saved in Thickness mode when ISONIC utPod PC is in Flaw detector mode

2.13 UTPodShutdown

Function format:

VOID WINAPI UTPodShutdown(HANDLE UTPodHandle,BOOL ShutdownDevice);

Shuts down the **ISONIC utPod for PC** Software

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**
- *ShutdownDevice* – if TRUE, the **ISONIC utPod** device connected to **ISONIC utPod for PC** software will be shut down. Otherwise, the device will remain switched on. **Note:** if the Automatic Shutdown option of the **ISONIC utPod for PC** is activated, the device will shut down regardless of this parameter value

Return value

None

2.14 UTPODGetSDKVersion

Function format:

DWORD WINAPI UTPodGetSDKVersion();

Retrieves current version of utPodSDK.dll

Parameters

None

Return Value

32-bit version value, high-order word specifies major version number, low-order word specifies minor version number

2.15 UTPodGetSWVersion

Function format:

VOID WINAPI UTPodGetSWVersion(HANDLE UTPodHandle);

Requests software version of the ISONIC utPod for PC software. The software version is returned through UTPODMSG_VERSION notification

Parameters

- *UTPodHandle* – utPod connection handle returned by **UTPodCreateHandle**

Return Value

None

3 ISONIC utPod SDK Notifications

The ISONIC utPod SDK communicates with the calling application through the single application-defined Windows message, identified by *NotifyMessage* parameter passed through the **UTPodCreateHandle** function. The message is sent to the window identified by *NotifyWindow* parameter of the **UTPodCreateHandle** function. The calling application should provide appropriate message handler for processing data sent from **ISONIC utPod SDK**. **wParam** of the message contains notification code, while **lParam** of the message contains notification-specific data. The following sections describe all possible notification codes specified by **wParam**.

3.1 UTPODMSG_CONNECTED

Numeric value: 0

Sent by **ISONIC utPod SDK** upon successful connection to the **ISONIC utPod for PC Software**.

lParam Content

Integer value identifying currently active screen of the **ISONIC utPod for PC Software**. Can be one of the following:

Constant	Numeric Value	Screen Description
SCREEN_FLAWDETECTOR	0	Flaw detector screen
SCREEN_THICKNESSCALIB	1	Thickness Calibration screen
SCREEN_THICKNESSMEAS	2	Thickness Measurement screen
SCREEN_CORROCALIB	3	Corrosion Calibration screen
SCREEN_CORROMEAS	4	Corrosion Measurement screen
SCREEN_OTHER	5	Any other screen

3.2 UTPODMSG_CONNECTIONFAILED

Numeric Value: 1

Sent by **ISONIC utPod SDK** when the connection to the **ISONIC utPod for PC Software** fails. Possible reasons: the **ISONIC utPod for PC Software** is not running on the target machine or already connected to another software.

The application **must** call **UTPodReleaseHandle** function upon receiving this notification.

IParam Content

Not used

3.3 UTPODMSG_DISCONNECTED

Numeric Value: 2

Sent by **ISONIC utPod SDK** when the **ISONIC utPod for PC Software** window is closed.

The application **must** call **UTPodReleaseHandle** function upon receiving this notification.

IParam Content

Not used

3.4 UTPODMSG_SCREENCHANGE

Numeric Value: 3

Sent by **ISONIC utPod SDK** when the active screen of the **ISONIC utPod for PC** is changed by user or by **ISONIC utPod SDK** command.

IParam Content

See **UTPODMSG_CONNECTED**

3.5 UTPODMSG_ASCAN

Numeric Value: 4

Sent by **ISONIC utPod SDK** when a new A-Scan and measurements set are read from the **ISONIC utPod** device.

IParam Content

Pointer to **ASCANDATA** structure containing A-Scan and all currently relevant measurements. See p. 4.1 for detailed description of the **ASCANDATA** structure

Example

The following example demonstrates processing UTPODMSG_ASCAN notification inside *NotifyMessage* handler:

```
afx_msg LRESULT <<AppWindowClass>>::OnUTPodMsg(WPARAM wParam,LPARAM lParam)
{
    LPASCANDATA aData;
    float tofVal;
    switch(wParam)
    {
        .....
        case UTPODMSG_ASCAN:
            aData=(LPASCANDATA)lParam; //cast lParam to ASCANDATA pointer
            ProcessAScan(aData->AScan,aData->PointsCount); //call A-Scan processing function
            //read TOF in gate A if available
            if (aData->MeasFlag & (1 << MEASURE_TOFA)) //check if gate A TOF measurement is available
            {
                if (aData->AVisible) //read measurement only if gate A is visible on the screen
                    tofVal=aData->MeasValues[MEASURE_TOFA];
            }
            break;
        .....
    }
}
```


3.6 UTPODMSG_PARAMSDUMP

Numeric Value: 5

This notification is sent by **ISONIC utPod SDK** in the following cases:

- In response to calling **UTPodGetParamsDump** function
- User loads file in **ISONIC utPod for PC** software
- The **ISONIC utPod SDK** application loads file using **UTPodLoadFile** function

IParam Content

Pointer to **PARAMSDUMP** structure containing a list of all parameters values relevant to **ISONIC utPod SDK**. See p. 4.2 for detailed description of the **PARAMSDUMP** structure

3.7 UTPODMSG_PARAMCHANGE

Numeric Value: 6

This notification is sent by **ISONIC utPod SDK** in the following cases:

- In response to calling **UTPodSetParamValue** function
- User changes parameter in **ISONIC utPod for PC** software

IParam Content

Pointer to **PARAMCHANGE** structure containing a list of all parameters values relevant to **ISONIC utPod SDK**. See p. 4.3 for detailed description of the **PARAMCHANGE** structure

3.8 UTPODMSG_SAVEDATA

Numeric Value: 7

This notification is sent by **ISONIC utPod SDK** in response to calling **UTPodSaveFile** function.

IParam Content

Pointer to **SAVEDATA** structure containing a list of all parameters values relevant to **ISONIC utPod SDK**. See p. 4.4 for detailed description of the **SAVEDATA** structure.

Important note: the data sent by ISONIC utPod SDK must be saved as is, without any modification. Otherwise, the ISONIC utPod device will not be able to load the saved file properly.

3.9 UTPODMSG_LOADFAILED

Numeric Value: 8

Sent by **ISONIC utPod SDK** when loading of the file data sent through **UTPODLoadFile** function fails.

IParam Content

None

3.10 UTPODMSG_VERSION

Numeric Value: 9

Sent by **ISONIC utPod SDK** in response to calling **UTPodGetSWVersion** function.

IParam Content

32-bit **ISONIC utPod for PC** version value, high-order word specifies major version number, low-order word specifies minor version number

4 Structures

4.1 ASCANDATA structure

Structure definition

```
typedef struct tagASCANDATA
{
    SHORT MeasUnit;
    SHORT FreezeState;
    SHORT Rectification;
    SHORT PointsCount;
    SHORT AScan[320];
    BOOL AVisible;
    BOOL BVisible;
    DWORD MeasFlag;
    FLOAT MeasValues[22];
} ASCANDATA, *LPASCANDATA;
```

Structure members

- *MeasUnit* – current **ISONIC utPod device** measurement unit. Zero value corresponds to Metric, 1 – to Imperial
- *FreezeState* – current freeze state. See **UTPodSetFreezeState** function description for possible values of this member
- *Rectification* – current value of Rectification parameter. Can be one of the following:
 - 0 – Full
 - 1 – PosHalf
 - 2 – NegHalf
 - 3 – RF
- *PointsCount* – number of valid points in A-Scan points array specified by *AScan* member. Maximum value is 320. At low ranges can be less than 320 but always greater than 1
- *AScan* – an array of maximum 320 short values describing current A-Scan. If *Rectification* is not RF, zero value designates zero line while 150 designates 100% of screen or higher. If *Rectification* is RF, value of 75 designates the aero line, 0 designates -100% of screen or lower, 150 designates 100% of screen or higher
- *AVisible* – specifies whether gate A is visible on the screen. TRUE if the gate A is visible and FALSE otherwise. See *MeasFlags* member description for explanation
- *BVisible* – specifies whether gate B is visible on the screen. TRUE if the gate B is visible and FALSE otherwise. See *MeasFlags* member description for explanation
- *MeasFlag* – 32-bit mask designating availability of measurement values in *MeasValues* array. Each bit corresponds to MEASURE_XXX constant (see p. 5.1 for a list of possible measurement values). If bit set to 1, the corresponding measurement value is available in *MeasValues* array, otherwise the corresponding measurement value should be ignored. The *MeasFlag* mask should be used in conjunction with *AVisible* and *BVisible* members. For example, if gate A is switched on, all bits corresponding to measurements related to gate A will be set to 1, but if at the same time gate A is out of the screen, *AVisible* member will be set to FALSE and all gate A related measurements should be ignored. Refer the example showing proper measurement reading in the description of **UTPODMSG_ASCAN** notification
- *MeasValues* – array of 22 float values, each corresponding to MEASURE_XXX constant. See p. 5.1 a list possible measurement values

4.2 PARAMSDUMP Structure

Structure definition

```
typedef struct tagPARAMSDUMP
{
    SHORT MeasUnit;
    BOOL LFDevice;
    FLOAT Parameters[24];
} PARAMSDUMP, *LPPARAMSDUMP;
```

Structure members

- *MeasUnit* - current **ISONIC utPod device** measurement unit. Zero value corresponds to Metric, 1 – to Imperial
- *LFDevice* – if this member is TRUE, the **ISONIC utPod device** currently connected to **ISONIC utPod for PC software** is low-frequency **ISONIC utPod** version. Otherwise, it is standard **ISONIC utPod device**.
- *Parameters* – an array of 24 float values. Each value in array corresponds to UTPAR_XXX constant. See p. 5.2 for a list of **ISONIC utPod** parameters in this array and their possible values

4.3 PARAMCHANGE Structure

Structure definition

```
typedef struct tagPARAMCHANGE
{
    SHORT ParamIndex;
    FLOAT ParamValue;
} PARAMCHANGE, *LPPARAMCHANGE;
```

Structure members

- *ParamIndex* – identifier of the changed parameter. See p. 5.2 for the list of parameter identifiers
- *ParamValue* – new parameter value

4.4 SAVEDATA Structure

Structure definition

```
typedef struct tagSAVEDATA
{
    DWORD DataLength;
    LPBYTE FileData;
} SAVEDATA, *LPSAVEDATA;
```

Structure members

- *DataLength* – size, in bytes, of data buffer specified by *FileData* member
- *FileData* – buffer containing raw data to be saved in a file

5 Constants

5.1 MEASURE_XXX constants

MEASURE_XXX constants are used to designate measurement values in ASCANDATA structure. The following table describes each constant and corresponding measurement value:

Constant	Numeric Value	Meas. Value	Unit	Notes
MEASURE_OFF	0	OFF	None	
MEASURE_TOFA	1	T(A)	μs	
MEASURE_SA	2	s(A)	mm or in	
MEASURE_TA	3	t(A)	mm or in	Available if probe angle is greater than zero
MEASURE_AA	4	a(A)	mm or in	Available if probe angle is greater than zero
MEASURE_VA	5	V(A)	dB	
MEASURE_HA	6	H(A)	%	
MEASURE_TOFB	7	T(B)	μs	
MEASURE_SB	8	s(B)	mm or in	
MEASURE_TB	9	t(B)	mm or in	Available if probe angle is greater than zero
MEASURE_AB	10	a(B)	mm or in	Available if probe angle is greater than zero
MEASURE_VB	11	V(B)	dB	
MEASURE_HB	12	H(B)	%	
MEASURE_DELTATOF	13	ΔT	μs	
MEASURE_DELTAS	14	Δs	mm or in	
MEASURE_DELTAT	15	Δt	mm or in	Available if probe angle is greater than zero
MEASURE_DELTAA	16	Δa	mm or in	Available if probe angle is greater than zero
MEASURE_DELTAV	17	ΔV	dB	
MEASURE_DVCA	18	ΔVC(A)	dB	
MEASURE_DVCB	19	ΔVC(B)	dB	
MEASURE_D	20	D	mm or in	Available only in Thickness or Corrosion modes. Negative if no measurement was obtained
MEASURE_DELAY	21	In Thickness mode shows measured delay in μs. If delay is off, this measurement is set to negative value. In Corrosion mode – 0 designates “0 to 1st” mode, 1 designates “Multiecho” mode		

5.2 ISONIC utPod Parameters (UTPAR_XXX Constants)

Constant	Numeric value	Meaning	Unit	Min Value	Max Value	Min Step	Notes
UTPAR_GAIN	0	Gain	dB	0	100	0.5	
UTPAR_DISPDELAY	1	Display Delay	µs	0	400	0.01	
UTPAR_RANGE	2	Range	mm or in	M: 0.5 I: 0.02	Variable	M: 0.1 I: 0.001	
UTPAR_USVELOCITY	3	Ultrasound Velocity	m/s or in/µs	M: 300 I: 11.8	M: 20000 I: 787.4	M: 1 I: 0.1	
UTPAR_RECTIFY	4	Rectification	0 – Full, 1 – PosHalf, 2 – NegHalf, 3 - RF				
UTPAR_ASWITCH	5	Gate A switch	0- Gate off, 1 – Gate on				In Thickness and Calibration modes, is read-only and always set to 1
UTPAR_ASTART	6	Gate A start	mm or in	0	M: 7500 I: 295.3	M: 0.1 I: 0.001	
UTPAR_AWIDTH	7	Gate A width	mm or in	0	M: 7500 I: 295.3	M: 0.1 I: 0.001	
UTPAR_ATHRESHOLD	8	Gate A threshold	%	5	95	1	When Rectification is RF can be negative (-5 to -95)
UTPAR_BSWITCH	9	Gate B switch	0- Gate off, 1 – Gate on				In Thickness and Calibration modes, is read-only and always set to 0
UTPAR_BSTART	10	Gate B start	mm or in	0	M: 7500 I: 295.3	M: 0.1 I: 0.001	Not used in Thickness and Calibration modes
UTPAR_BWIDTH	11	Gate B width	mm or in	0	M: 7500 I: 295.3	M: 0.1 I: 0.001	Not used in Thickness and Calibration modes
UTPAR_BTHRESHOLD	12	Gate B threshold	%	5	95	1	When Rectification is RF can be negative (-5 to -95). Not used in Thickness and Calibration modes
UTPAR_FILTER	13	Filter	0: 0.1-13 MHz 1: 1-3 MHz 2: 3-5 MHz 3: 5-7 MHz 4: 0.5-25 MHz (0.03-1 for LF devices)				Read-only In Thickness and Calibration modes

Constant	Numeric value	Meaning	Unit	Min Value	Max Value	Min Step	Notes
			5: OFF				
UTPAR_THICKNESS	14	Geometry thickness	mm or in	M: 5 I: 0.2	M: 500 I: 20	M:1 I: 0.05	Negative value designates “Ignore thickness” setting switched on. Not used in Thickness and Calibration modes
UTPAR_DIAMETER	15	Geometry diameter	mm or in	M: 50 I: 2	M: 4000 I: 160	M:1 I: 0.05	Negative value designates “Flat object” setting switched on Not used in Thickness and Calibration modes
UTPAR_ANGLE	16	Probe angle	degrees	0	89	1	Not used in Thickness and Calibration modes
UTPAR_XVALUE	17	Probe X-value	mm or in	0	M: 50 I: 2	M: 0.5 I: 0.001	Not used in Thickness and Calibration modes
UTPAR_PULSERMODE	18	Pulser mode	0 – Single, 1 - Dual				Read-only In Thickness and Calibration modes
UTPAR_PULSEWIDTH	19	Pulse width	ns	50	S: 600 LF: 10000	10	
UTPAR_PROBEDELAY	20	Probe delay	µs	0	70	0.01	Not used in Thickness mode
UTPAR_MEASMODE	21	Measurement mode	0 – Flank, 1 - Top				Not used in Thickness and Calibration modes
UTPAR_PRF	22	PRF	Hz	15	2000	1	
UTPAR_FIRINGLEVEL	23	Firing Level		1	12	1	

Note: In “Min Value”, “Max Value” and “Min Step” columns **M** means Metric measurement unit, **I** means Imperial measurement unit, **LF** means **ISONIC utPod Low-Frequency device** in Flaw Detector mode