

# FUTEK USB DLL Version 2.1.2000.0

## Programmer's Guide



10 Thomas, Irvine, CA 92618, USA  
Toll Free: (800) 23-FUTEK  
Telephone: (949) 465-0900  
Fax: (949) 465-0905  
[futek@futek.com](mailto:futek@futek.com)  
[www.futek.com](http://www.futek.com)

## Table of Contents

Warning .....	- 5 -
Command Overview.....	- 5 -
Type of Board .....	- 5 -
Command Response.....	- 8 -
Common Errors.....	- 12 -
Command Specific Errors.....	- 15 -
Variables Overview.....	- 18 -
DeviceHandle.....	- 18 -
DeviceStatus.....	- 18 -
ErrorDescription .....	- 18 -
Calibration Overview.....	- 19 -
Calibration Example .....	- 19 -
Connection Commands.....	- 21 -
Open_Device_Connection.....	- 21 -
Close_Device_Connection.....	- 22 -
Data Link Commands .....	- 23 -
Slave_Activity_Inquiry.....	- 23 -
Set Commands.....	- 24 -
Set_Sensor_Identification_Number .....	- 24 -
Set_Unit_Code.....	- 25 -
Set_Decimal_Point.....	- 26 -
Set_Calibration_Register .....	- 28 -
Set_Loading_Point.....	- 30 -
Set_Load_of_Loading_Point .....	- 33 -
Set_Sensor_Configuration.....	- 36 -
Set_Load_Switch.....	- 38 -
Set_Number_of_Loading_Points .....	- 39 -
Set_ADC_Configuration.....	- 40 -
Set_ADC_Configuration2.....	- 44 -
Set_Average_Setting.....	- 47 -
Set_Sensitivity .....	- 50 -
Set_Bridge_Resistance .....	- 51 -
Set_Direction .....	- 52 -
Set_Zero_Correction .....	- 54 -
Set_Shunt_Value .....	- 55 -
Set_Calibration_Code .....	- 56 -
Set_Calibration_Day .....	- 57 -
Set_Calibration_Month.....	- 58 -
Set_Calibration_Year .....	- 59 -
Set_Pulses_Per_Rotation .....	- 60 -
Get Commands.....	- 61 -
Get_Offset_Value .....	- 61 -
Get_Fullscale_Value .....	- 62 -
Normal_Data_Request.....	- 63 -
Fast_Data_Request.....	- 64 -
Version_of_Board.....	- 67 -

---

Reset_Board.....	68 -
Reset_Board2.....	69 -
Get_Display_Page.....	70 -
Get_Internal_Register.....	71 -
Get_DataLogging.....	73 -
Get_Rotation_Values.....	74 -
Get_Device_Count.....	75 -
Get_Device_Serial_Number.....	76 -
Get_Loading_Point.....	77 -
Get_Offset_Load .....	78 -
Get_Load_of_Loading_Point .....	79 -
Get_Fullscale_Load.....	80 -
Get_Number_of_Loading_Points .....	81 -
Get_Sensitivity .....	82 -
Get_Bridge_Resistance .....	83 -
Get_ADC_PGA_Setting.....	84 -
Get_ADC_Sampling_Rate_Setting .....	87 -
Get_Sensor_Identification_Number .....	91 -
Get_Unit_Code.....	92 -
Get_Direction .....	93 -
Get_Shunt_Value .....	95 -
Get_Decimal_Point .....	96 -
Get_Calibration_Code .....	98 -
Get_Calibration_Day .....	99 -
Get_Calibration_Month.....	100 -
Get_Calibration_Year .....	101 -
Get_Pulses_Per_Rotation .....	102 -
Get_Average_Setting.....	103 -
Get_Type_of_Board .....	106 -
Get_Hardware_Version.....	108 -
Get_Firmware_Version.....	109 -
Get_Firmware_Year.....	110 -
Get_Firmware_Month .....	112 -
Backup Commands .....	114 -
Create_Back_Up.....	114 -
Restore_Back_Up .....	116 -
Control Commands .....	118 -
Change_Battery_Enter.....	118 -
Change_Tare_Up .....	119 -
Change_Display_Back .....	120 -
Change_Reset_Left .....	121 -
Change_Menu .....	122 -
Change_Unit_Right .....	123 -
Change_Shunt_Exit .....	124 -
Change_Hold_Down .....	125 -
Debugging Commands .....	126 -
Read_Memory_Register .....	126 -
Write_Memory_Register .....	127 -

---

Read_EEPROM_Register.....	- 128 -
Write_EEPROM_Register.....	- 129 -
Read_TEDS_Register.....	- 130 -
Write_TEDS_Register .....	- 131 -
Read_Channel_Register.....	- 132 -
Write_Channel_Register.....	- 133 -
Register Addresses .....	- 134 -
EEPROM Register .....	- 134 -
Additional Information.....	- 138 -
Unit Codes.....	- 138 -
Release Information.....	- 140 -
Copyright.....	- 140 -
Disclaimer .....	- 140 -
Contact Information .....	- 140 -

## Warning

 The use of certain commands can affect the factory calibration of the USB Device. This Programmer's Guide provides a wide range of commands some of which can directly write to memory. Once the memory is overwritten, it will be lost forever. Please refer to the [Command Overview](#) for commands that affect memory. If you are unsure how to use a command or if you are unsure of its affects on memory, please feel free to contact us.

## Command Overview

The Command Overview provides a comprehensive listing of the available commands and includes the type of boards supported, command responses, common errors and the command specific errors.

### Type of Board

The Type of Board table provides a comprehensive listing of the available commands and the various types of boards that support each command. The Memory column indicates which commands write to the EEPROM of the USB Device.

Commands	Type of Board	USB100	USB200	USB110	USB210	USB120	USB220	USB320	USB230	USB410	IHH500	IPM650	Memory
<b>Connection Commands</b>													
Open_Device_Connection	•	•	•	•	•	•	•	•	•	•	•	•	
Close_Device_Connection	•	•	•	•	•	•	•	•	•	•	•	•	
<b>Data Link Commands</b>													
Slave_Activity_Inquiry	•	•	•	•	•	•	•	•	•	•	•	•	
<b>Set Commands</b>													
Set_Sensor_Identification_Number	•	•	•	•	•	•	•	•	•	•	•	•	
Set_Unit_Code	•	•	•	•	•	•	•	•	•	•			
Set_Decimal_Point	•	•	•	•	•	•	•	•	•	•			
Set_Calibration_Register											•	•	
Set_Loading_Point	•	•	•	•	•	•	•	•	•	•	•	•	
Set_Load_of_Loading_Point	•	•	•	•	•	•	•	•	•	•	•	•	
Set_Sensor_Configuration	•	•	•	•	•	•	•	•	•	•	•	•	
Set_Load_Switch	•	•	•	•	•	•	•		•		•	•	
Set_Number_of_Loading_Points	•	•	•	•	•	•	•	•	•	•	•	•	
Set_ADC_Configuration	•	•	•	•	•	•	•	•	•	•			
Set_ADC_Configuration2	•	•	•	•	•	•	•	•	•	•	•	•	
Set_Average_Setting	•	•	•	•	•	•	•	•	•	•	•	•	
Set_Sensitivity	•	•	•	•	•	•	•	•	•	•			

Commands	Type of Board	USB100	USB200	USB110	USB210	USB120	USB220	USB320	USB230	USB410	IHH500	IPM650	Memory
Set_Bridge_Resistance	•	•	•	•	•	•	•	•	•	•			•
Set_Direction	•	•	•	•	•	•	•	•	•	•			•
Set_Zero_Correction	•	•	•	•	•	•	•	•	•	•			•
Set_Shunt_Value	•	•	•	•	•	•	•	•	•	•			•
Set_Calibration_Code	•	•	•	•	•	•	•	•	•	•			•
Set_Calibration_Day	•	•	•	•	•	•	•	•	•	•			•
Set_Calibration_Month	•	•	•	•	•	•	•	•	•	•			•
Set_Calibration_Year	•	•	•	•	•	•	•	•	•	•			•
Set_Pulses_Per_Rotation										•			•
<b>Get Commands</b>													
Get_Offset_Value	•	•	•	•	•	•	•	•	•	•	•	•	•
Get_Fullscale_Value	•	•	•	•	•	•	•	•	•	•	•	•	•
Normal_Data_Request	•	•	•	•	•	•	•	•	•	•	•	•	•
Fast_Data_Request				•	•	•	•	•	•				
Version_of_Board	•	•	•	•	•	•	•	•	•	•	•	•	•
Reset_Board	•	•	•	•	•	•	•	•	•	•	•	•	•
Reset_Board2	•	•	•	•	•	•	•	•	•	•	•	•	•
Get_Display_Page											•	•	
Get_Internal_Register											•	•	
Get_DataLogging											•	•	
Get_Rotation_Values											•	•	•
Get_Device_Count	•	•	•	•	•	•	•	•	•	•	•	•	•
Get_Device_Serial_Number	•	•	•	•	•	•	•	•	•	•	•	•	•
Get>Loading_Point	•	•	•	•	•	•	•	•	•	•			
Get_Offset_Load	•	•	•	•	•	•	•	•	•	•			
Get_Load_of>Loading_Point	•	•	•	•	•	•	•	•	•	•			
Get_Fullscale_Load	•	•	•	•	•	•	•	•	•	•			
Get_Number_of>Loading_Points	•	•	•	•	•	•	•	•	•	•			
Get_Sensitivity	•	•	•	•	•	•	•	•	•	•			
Get_Bridge_Resistance	•	•	•	•	•	•	•	•	•	•			
Get_ADC_PGA_Setting	•	•	•	•	•	•	•	•	•	•			
Get_ADC_Sampling_Rate_Setting	•	•	•	•	•	•	•	•	•	•			
Get_Sensor_Identification_Number	•	•	•	•	•	•	•	•	•	•			
Get_Unit_Code	•	•	•	•	•	•	•	•	•	•			
Get_Direction	•	•	•	•	•	•	•	•	•	•			

Commands	Type of Board	USB100	USB200	USB110	USB210	USB120	USB220	USB320	USB230	USB410	IHH500	IPM650	Memory
Get_Shunt_Value	•	•	•	•	•	•	•	•	•	•			
Get_Decimal_Point	•	•	•	•	•	•	•	•	•	•			
Get_Calibration_Code	•	•	•	•	•	•	•	•	•	•			
Get_Calibration_Day	•	•	•	•	•	•	•	•	•	•			
Get_Calibration_Month	•	•	•	•	•	•	•	•	•	•			
Get_Calibration_Year	•	•	•	•	•	•	•	•	•	•			
Get_Pulses_Per_Rotation										•			
Get_Average_Setting	•	•	•	•	•	•	•	•	•	•			
Get_Type_of_Board	•	•	•	•	•	•	•	•	•	•	•	•	•
Get_Hardware_Version	•	•	•	•	•	•	•	•	•	•	•	•	•
Get_Firmware_Version	•	•	•	•	•	•	•	•	•	•	•	•	•
Get_Firmware_Year	•	•	•	•	•	•	•	•	•	•	•	•	•
Get_Firmware_Month	•	•	•	•	•	•	•	•	•	•	•	•	•
<b>Backup Commands</b>													
Create_Back_Up				•	•	•	•	•	•	•			•
Restore_Back_Up				•	•	•	•	•	•	•			•
<b>Control Commands</b>													
Change_Battery_Enter										•	•		
Change_Tare_Up										•	•		
Change_Display_Back										•	•		
Change_Reset_Left										•	•		
Change_Menu										•	•		
Change_Unit_Right										•	•		
Change_Shunt_Exit										•	•		
Change_Hold_Down										•	•		
<b>Debugging Commands</b>													
Read_Memory_Register	•	•	•	•	•	•	•	•	•	•	•	•	
Write_Memory_Register	•	•	•	•	•	•	•	•	•	•	•	•	•
Read EEPROM_Register	•	•	•	•	•	•	•	•	•	•	•	•	
Write EEPROM_Register	•	•	•	•	•	•	•	•	•	•	•	•	•
Read TEDS_Register										•	•		
Write TEDS_Register										•	•		•
Read_Channel_Register										•	•		
Write_Channel_Register										•	•		•

## ***Command Response***

The Command Response table provides a comprehensive listing of the available commands and the response of each command. The responses have been separated for commands that were successful and unsuccessful. If the command was sent and received successfully, then the response will typically be either 0 or a number in a given range. Please refer to the sections related to a specific command for more information. If the command was sent and received unsuccessfully, then the response will be "Error". Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

Commands	Command Response	If Successful: Return Value	If Unsuccessful: "Error"
<b>Connection Commands</b>			
Open_Device_Connection			
Close_Device_Connection			
<b>Data Link Commands</b>			
Slave_Activity_Inquiry	0	•	
<b>Set Commands</b>			
Set_Sensor_Identification_Number	0	•	
Set_Unit_Code	0	•	
Set_Decimal_Point	0	•	
Set_Calibration_Register	0	•	
Set_Loading_Point	0	•	
Set_Load_of_Loading_Point	0	•	
Set_Sensor_Configuration	0	•	
Set_Load_Switch	0	•	
Set_Number_of_Loading_Points	0	•	
Set_ADC_Configuration	0	•	
Set_ADC_Configuration2	0	•	
Set_Average_Setting	0	•	
Set_Sensitivity	0	•	
Set_Bridge_Resistance	0	•	
Set_Direction	0	•	
Set_Zero_Correction	0	•	
Set_Shunt_Value	0	•	

Commands	Command Response	If Successful: Return Value	If Unsuccessful: "Error"
Set_Calibration_Code	0	•	
Set_Calibration_Day	0	•	
Set_Calibration_Month	0	•	
Set_Calibration_Year	0	•	
Set_Pulses_Per_Rotation	0	•	
<b>Get Commands</b>			
Get_Offset_Value	Number	•	
Get_Fullscale_Value	Number	•	
Normal_Data_Request	Number	•	
Fast_Data_Request	String	•	
Version_of_Board	0	•	
Reset_Board	0	•	
Reset_Board2	0	•	
Get_Display_Page	0	•	
Get_Internal_Register	Number	•	
Get_DataLogging	0	•	
Get_Rotation_Values	0	•	
Get_Device_Count	Number	•	
Get_Device_Serial_Number	Number	•	
Get>Loading_Point	Number	•	
Get_Offset_Load	Number	•	
Get_Load_of>Loading_Point	Number	•	
Get_Fullscale_Load	Number	•	
Get_Number_of>Loading_Points	Number	•	
Get_Sensitivity	Number	•	
Get_Bridge_Resistance	Number	•	
Get_ADC_PGA_Setting	Number	•	
Get_ADC_Sampling_Rate_Setting	Number	•	
Get_Sensor_Identification_Number	Number	•	
Get_Unit_Code	Number	•	
Get_Direction	Number	•	

Commands	Command Response	If Successful: Return Value	If Unsuccessful: "Error"
Get_Shunt_Value	Number	•	
Get_Decimal_Point	Number	•	
Get_Calibration_Code	Number	•	
Get_Calibration_Day	Number	•	
Get_Calibration_Month	Number	•	
Get_Calibration_Year	Number	•	
Get_Pulses_Per_Rotation	Number	•	
Get_Average_Setting	Number	•	
Get_Type_of_Board	Number	•	
Get_Hardware_Version	Number	•	
Get_Firmware_Version	Number	•	
Get_Firmware_Year	Number	•	
Get_Firmware_Month	Number	•	
<b>Backup Commands</b>			
Create_Back_Up	0	•	
Restore_Back_Up	0	•	
<b>Control Commands</b>			
Change_Battery_Enter	0	•	
Change_Tare_Up	0	•	
Change_Display_Back	0	•	
Change_Reset_Left	0	•	
Change_Menu	0	•	
Change_Unit_Right	0	•	
Change_Shunt_Exit	0	•	
Change_Hold_Down	0	•	
<b>Debugging Commands</b>			
Read_Memory_Register	Number	•	
Write_Memory_Register	0	•	
Read_EEPROM_Register	Number	•	
Write_EEPROM_Register	0	•	
Read_TEDS_Register	Number	•	

Commands	Command Response	If Successful: Return Value	If Unsuccessful: "Error"
Write_TEDS_Register		0	•
Read_Channel_Register		Number	•
Write_Channel_Register		0	•

## Common Errors

The Common Errors table provides a comprehensive listing of the available commands and the common errors that can be returned by the ErrorDescription when a command is unsuccessful. Please refer to the [Command Specific Errors](#) table for additional errors that can be returned by the ErrorDescription.

Commands	ErrorDescription	"Timeout Error"	"Packet Number Error"	"Packet Size Error"	"Packet Frame Error"	"Packet Command Error"	"Command Number Error"	"Command Error"	"End Of Packet Error"	"TEDS Error"	"Invalid Type of Board"
<b>Connection Commands</b>											
Open_Device_Connection											
Close_Device_Connection											
<b>Data Link Commands</b>											
Slave_Activity_Inquiry	•	•	•	•	•	•	•	•	•		
<b>Set Commands</b>											
Set_Sensor_Identification_Number	•	•	•	•	•	•	•	•	•		
Set_Unit_Code	•	•	•	•	•	•	•	•	•	•	
Set.Decimal_Point	•	•	•	•	•	•	•	•	•	•	
Set_Calibration_Register	•	•	•	•	•	•	•	•	•	•	
Set_Loading_Point	•	•	•	•	•	•	•	•	•	•	
Set_Load_of_Loading_Point	•	•	•	•	•	•	•	•	•	•	
Set_Sensor_Configuration	•	•	•	•	•	•	•	•	•	•	
Set_Load_Switch	•	•	•	•	•	•	•	•	•	•	
Set_Number_of_Loading_Points	•	•	•	•	•	•	•	•	•	•	
Set_ADC_Configuration	•	•	•	•	•	•	•	•	•	•	•
Set_ADC_Configuration2	•	•	•	•	•	•	•	•	•	•	
Set_Average_Setting	•	•	•	•	•	•	•	•	•	•	
Set_Sensitivity	•	•	•	•	•	•	•	•	•	•	
Set_Bridge_Resistance	•	•	•	•	•	•	•	•	•	•	
Set_Direction	•	•	•	•	•	•	•	•	•	•	
Set_Zero_Correction	•	•	•	•	•	•	•	•	•	•	
Set_Shunt_Value	•	•	•	•	•	•	•	•	•	•	
Set_Calibration_Code	•	•	•	•	•	•	•	•	•	•	
Set_Calibration_Day	•	•	•	•	•	•	•	•	•	•	
Set_Calibration_Month	•	•	•	•	•	•	•	•	•	•	

Commands	ErrorDescription	"Timeout Error"	"Packet Number Error"	"Packet Size Error"	"Packet Frame Error"	"Packet Command Error"	"Command Number Error"	"Command Error"	"End Of Packet Error"	"TEDS Error"	"Invalid Type of Board"
Set_Calibration_Year	•	•	•	•	•	•	•	•	•	•	•
Set_Pulses_Per_Rotation	•	•	•	•	•	•	•	•	•	•	•
<b>Get Commands</b>											
Get_Offset_Value	•	•	•	•	•	•	•	•	•	•	
Get_Fullscale_Value	•	•	•	•	•	•	•	•	•	•	
Normal_Data_Request	•	•	•	•	•	•	•	•	•	•	
Fast_Data_Request	•	•	•	•	•	•	•	•	•	•	
Version_of_Board	•	•	•	•	•	•	•	•	•	•	
Reset_Board	•	•	•	•	•	•	•	•	•	•	
Reset_Board2	•	•	•	•	•	•	•	•	•	•	
Get_Display_Page	•	•	•	•	•	•	•	•	•	•	
Get_Internal_Register	•	•	•	•	•	•	•	•	•	•	
Get_DataLogging	•	•	•	•	•	•	•	•	•	•	
Get_Rotation_Values	•	•	•	•	•	•	•	•	•	•	
Get_Device_Count											
Get_Device_Serial_Number											
Get>Loading_Point	•	•	•	•	•	•	•	•	•	•	
Get_Offset_Load	•	•	•	•	•	•	•	•	•	•	
Get_Load_of>Loading_Point	•	•	•	•	•	•	•	•	•	•	
Get_Fullscale_Load	•	•	•	•	•	•	•	•	•	•	
Get_Number_of>Loading_Points	•	•	•	•	•	•	•	•	•	•	
Get_Sensitivity	•	•	•	•	•	•	•	•	•	•	
Get_Bridge_Resistance	•	•	•	•	•	•	•	•	•	•	
Get_ADC_PGA_Setting	•	•	•	•	•	•	•	•	•	•	
Get_ADC_Sampling_Rate_Setting	•	•	•	•	•	•	•	•	•	•	
Get_Sensor_Identification_Number	•	•	•	•	•	•	•	•	•	•	
Get_Unit_Code	•	•	•	•	•	•	•	•	•	•	
Get_Direction	•	•	•	•	•	•	•	•	•	•	
Get_Shunt_Value	•	•	•	•	•	•	•	•	•	•	
Get_Decimal_Point	•	•	•	•	•	•	•	•	•	•	
Get_Calibration_Code	•	•	•	•	•	•	•	•	•	•	

Commands	ErrorDescription	"Timeout Error"	"Packet Number Error"	"Packet Size Error"	"Packet Frame Error"	"Packet Command Error"	"Command Number Error"	"Command Error"	"End Of Packet Error"	"TEDS Error"	"Invalid Type of Board"
Get_Calibration_Day	●	●	●	●	●	●	●	●	●	●	●
Get_Calibration_Month	●	●	●	●	●	●	●	●	●	●	●
Get_Calibration_Year	●	●	●	●	●	●	●	●	●	●	●
Get_Pulses_Per_Rotation	●	●	●	●	●	●	●	●	●	●	●
Get_Average_Setting	●	●	●	●	●	●	●	●	●	●	●
Get_Type_of_Board	●	●	●	●	●	●	●	●	●	●	●
Get_Hardware_Version	●	●	●	●	●	●	●	●	●	●	●
Get_Firmware_Version	●	●	●	●	●	●	●	●	●	●	●
Get_Firmware_Year	●	●	●	●	●	●	●	●	●	●	●
Get_Firmware_Month	●	●	●	●	●	●	●	●	●	●	●
<b>Backup Commands</b>											
Create_Back_Up	●	●	●	●	●	●	●	●	●	●	●
Restore_Back_Up	●	●	●	●	●	●	●	●	●	●	●
<b>Control Commands</b>											
Change_Battery_Enter	●	●	●	●	●	●	●	●	●	●	●
Change_Tare_Up	●	●	●	●	●	●	●	●	●	●	●
Change_Display_Back	●	●	●	●	●	●	●	●	●	●	●
Change_Reset_Left	●	●	●	●	●	●	●	●	●	●	●
Change_Menu	●	●	●	●	●	●	●	●	●	●	●
Change_Unit_Right	●	●	●	●	●	●	●	●	●	●	●
Change_Shunt_Exit	●	●	●	●	●	●	●	●	●	●	●
Change_Hold_Down	●	●	●	●	●	●	●	●	●	●	●
<b>Debugging Commands</b>											
Read_Memory_Register	●	●	●	●	●	●	●	●	●	●	●
Write_Memory_Register	●	●	●	●	●	●	●	●	●	●	●
Read EEPROM_Register	●	●	●	●	●	●	●	●	●	●	●
Write EEPROM_Register	●	●	●	●	●	●	●	●	●	●	●
Read_TEDS_Register	●	●	●	●	●	●	●	●	●	●	●
Write_TEDS_Register	●	●	●	●	●	●	●	●	●	●	●
Read_Channel_Register	●	●	●	●	●	●	●	●	●	●	●
Write_Channel_Register	●	●	●	●	●	●	●	●	●	●	●

## ***Command Specific Errors***

The Command Specific Errors table provides a comprehensive listing of the available commands and the command specific errors that can be returned by the ErrorDescription when a command is unsuccessful.

Commands	Command Specific Errors
<b>Connection Commands</b>	
Open_Device_Connection	
Close_Device_Connection	
<b>Data Link Commands</b>	
Slave_Activity_Inquiry	
<b>Set Commands</b>	
Set_Sensor_Identification_Number	"Invalid Serial Number"
Set_Unit_Code	"Invalid Unit Code"
Set.Decimal_Point	"Invalid Decimal Point"
Set_Calibration_Register	"Invalid Register"
Set_Loading_Point	"Invalid Loading Point"
	"Invalid Loading Point"
Set_Load_of_Loading_Point	"Invalid Load Value"
Set_Sensor_Configuration	"Invalid Sensor Configuration"
Set_Load_Switch	"Invalid Simulated Load"
Set_Number_of_Loading_Points	"Invalid Number Of Loading Points"
Set_ADC_Configuration	"Invalid Sampling Rate"
Set_ADC_Configuration2	"Invalid ADC Configuration"
Set_Average_Setting	"Invalid Average Setting"
Set_Sensitivity	"Invalid Sensitivity"
Set_Bridge_Resistance	"Invalid Bridge Resistance"
Set_Direction	"Invalid Direction"
Set_Zero_Correction	"Invalid Point"
Set_Shunt_Value	"Invalid Shunt Value"
Set_Calibration_Code	"Invalid Calibration Code"
Set_Calibration_Day	"Invalid Calibration Day"
Set_Calibration_Month	"Invalid Calibration Month"
Set_Calibration_Year	"Invalid Calibration Year"
Set_Pulses_Per_Rotation	"Invalid Pulses Per Rotation"
<b>Get Commands</b>	
Get_Offset_Value	
Get_Fullscale_Value	
Normal_Data_Request	

<b>Commands</b>	<b>Command Specific Errors</b>
Fast_Data_Request	"Invalid Firmware Version"
Version_of_Board	
Reset_Board	
Reset_Board2	
Get_Display_Page	
Get_Internal_Register	"Invalid Register"
Get_DataLogging	"Invalid Counter"
Get_Rotation_Values	
Get_Device_Count	
Get_Device_Serial_Number	
Get>Loading_Point	"Invalid Point"
Get_Offset_Load	
Get_Load_of>Loading_Point	"Invalid Point"
Get_Fullscale_Load	
Get_Number_of>Loading_Points	
Get_Sensitivity	
Get_Bridge_Resistance	
Get_ADC_PGA_Setting	
Get_ADC_Sampling_Rate_Setting	
Get_Sensor_Identification_Number	
Get_Unit_Code	
Get_Direction	
Get_Shunt_Value	
Get_Decimal_Point	
Get_Calibration_Code	
Get_Calibration_Day	
Get_Calibration_Month	
Get_Calibration_Year	
Get_Pulses_Per_Rotation	
Get_Average_Setting	
Get_Type_of_Board	
Get_Hardware_Version	
Get_Firmware_Version	
Get_Firmware_Year	
Get_Firmware_Month	
<b>Backup Commands</b>	
Create_BackUp	"Invalid Page Number"
Restore_BackUp	"Invalid Page Number"

Commands	Command Specific Errors
<b>Control Commands</b>	
Change_Battery_Enter	
Change_Tare_Up	
Change_Display_Back	
Change_Reset_Left	
Change_Menu	
Change_Unit_Right	
Change_Shunt_Exit	
Change_Hold_Down	
<b>Debugging Commands</b>	
Read_Memory_Register	"Invalid Address"
	"Invalid Address"
Write_Memory_Register	"Invalid Data"
Read_EEPROM_Register	"Invalid Address"
	"Invalid Address"
	"Invalid High Byte"
Write_EEPROM_Register	"Invalid Low Byte"
	"Invalid High Address"
Read_TEDS_Register	"Invalid Low Address"
	"Invalid Data"
	"Invalid High Address"
Write_TEDS_Register	"Invalid Low Address"
	"Invalid Address"
Read_Channel_Register	"Invalid Channel Number"
	"Invalid Address"
	"Invalid Data"
Write_Channel_Register	"Invalid Channel Number"

## Variables Overview

The Variables Overview provides a comprehensive listing of the available variables and includes the DeviceHandle, DeviceStatus and ErrorDescription.

### *DeviceHandle*

The DeviceHandle is a pointer to a variable that contains the handle of the device. The DeviceHandle must be used with many of the commands to read from and/or write to the device.

### *DeviceStatus*

The DeviceStatus is a variable that contains a numerical value associated with the status of the device. Please refer to the DeviceStatus table for additional errors that can be returned by the ErrorDescription.

DeviceStatus				
Hexadecimal	Decimal	Octal	Binary	Status
00	0	00	00000000	OK
01	1	01	00000001	Invalid Handle
02	2	02	00000010	Device Not Found
03	3	03	00000011	Device Not Opened
04	4	04	00000100	IO Error
05	5	05	00000101	Insufficient Resources
06	6	06	00000110	Invalid Parameter
07	7	07	00000111	Invalid Baud Rate
08	8	10	00001000	Device Not Opened For Erase
09	9	11	00001001	Device Not Opened For Write
0A	10	12	00001010	Failed to Write Device
0B	11	13	00001011	EEPROM Read Failed
0C	12	14	00001100	EEPROM Write Failed
0D	13	15	00001101	EEPROM Erased Failed
0E	14	16	00001110	EEPROM Not Present
0F	15	17	00001111	EEPROM Not Programmed
10	16	20	00010000	Invalid Arguments
11	17	21	00010001	Not Supported
12	18	22	00010010	Other Error
13	19	23	00010011	Device List Not Ready

### *ErrorDescription*

The ErrorDescription is a string that describes the “Error” generated by the current command being used. Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## Calibration Overview

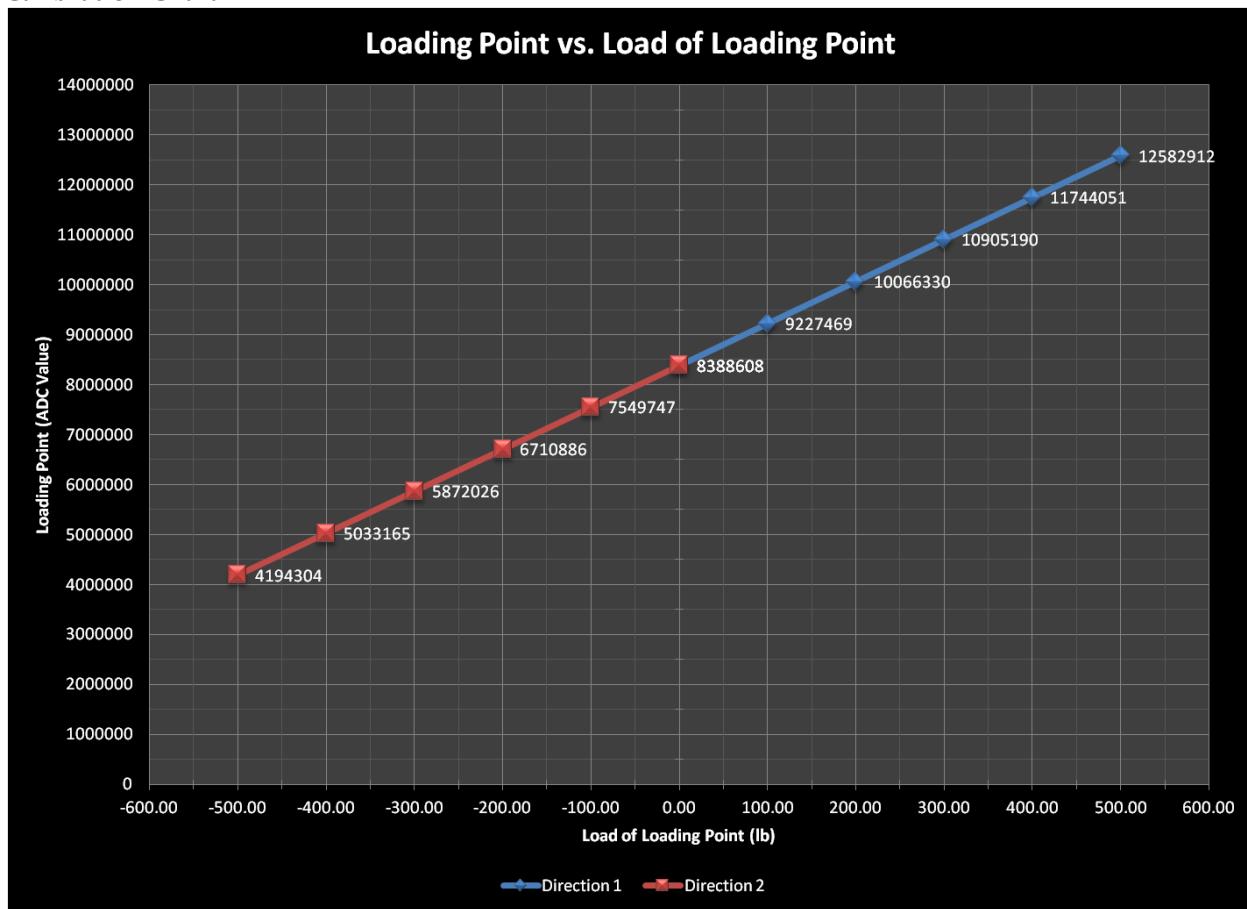
The Calibration Overview provides a graphical representation of the calibration data that is stored in the EEPROM of the USB Device.

### Calibration Example

#### Summary

The following example provides a graphical representation of a 5-point calibration that was performed in two directions using a USB210. Each direction contains data related to the physical load (Load of Loading Point (lb)) and the input voltage (Loading Point (ADC Value)). Depending on the type of board used, there can be several loading points applied during calibration.

#### Calibration Chart



**Calibration Data**

Direction 1		
Point	Load of Loading Point (lb)	Loading Point (ADC Value)
Loading Point 0	0.00	8388608
Loading Point 1	100.00	9227469
Loading Point 2	200.00	10066330
Loading Point 3	300.00	10905190
Loading Point 4	400.00	11744051
Loading Point 15	500.00	12582912

Direction 2		
Point	Load of Loading Point (lb)	Loading Point (ADC Value)
Loading Point 7	0.00	8388608
Loading Point 8	-100.00	7549747
Loading Point 9	-200.00	6710886
Loading Point 10	-300.00	5872026
Loading Point 11	-400.00	5033165
Loading Point 14	-500.00	4194304

Please refer to the [EEPROM Register](#) table for more information.

## Connection Commands

### *Open\_Device\_Connection*

#### Summary

Open\_Device\_Connection is used to open a connection with the USB Device. This subroutine will assign the DeviceHandle and DeviceStatus. The DeviceHandle will be required for subsequent use with many of the other commands.

#### Definition

```
Public Sub Open_Device_Connection(ByVal SerialNumber As String)
```

Parameters	Data Type	Description
SerialNumber	String	Device Serial Number

Variables	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
<a href="#">DeviceStatus</a>	String	Status of the USB Device

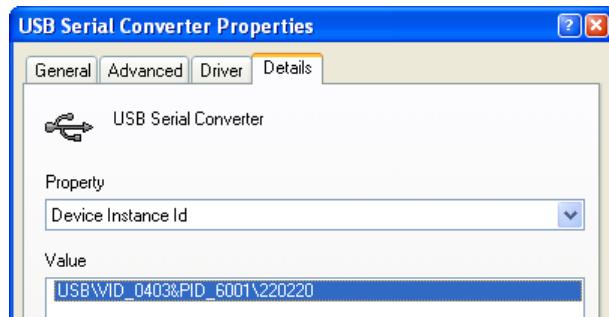
#### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
DeviceHandle	Handle	0
DeviceStatus	0	Status

#### Comments

The SerialNumber of the USB Device will most likely be one of two values: (1) the serial number that is engraved on the USB Device or (2) the serial number of the sensor that it was mated with during a system calibration. In Microsoft Windows XP, this can be verified by going to Computer Management → Device Manager → Universal Serial Bus Controllers → USB Serial Converter → Properties. Go to the Details Tab and select the Device Instance Id as the Property. The Value has the following format:

USB\VID\_0403&PID\_6001\SerialNumber



## *Close\_Device\_Connection*

### Summary

Close\_Device\_Connection is used to close a connection with the USB Device. This subroutine will assign the DeviceStatus.

### Definition

**Public Sub** Close\_Device\_Connection(**ByVal** DeviceHandle **As String**)

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
Variables	Data Type	Description
<a href="#">DeviceStatus</a>	String	Status of the USB Device

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
DeviceStatus	0	Status

### Comments

Close\_Device\_Connection should be used when you have finished communicating with the USB Device. It is always recommended to close the device connection prior to closing an application.

## Data Link Commands

### *Slave\_Activity\_Inquiry*

#### **Summary**

Slave\_Activity\_Inquiry is used to verify the communication data link with the USB Device. The command does not change any settings in the EEPROM or assign any variables a value. It is for testing / debugging purposes only.

#### **Definition**

Public Function Slave\_Activity\_Inquiry(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

#### **Return Type**

String

#### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Slave_Activity_Inquiry	0	"Error"

#### **Comments**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## Set Commands

### *Set\_Sensor\_Identification\_Number*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

#### Summary

Set\_Sensor\_Identification\_Number is used to set the serial number stored in the EEPROM of the USB Device.

#### Definition

Public Function Set\_Sensor\_Identification\_Number(**ByVal** DeviceHandle **As String**, **ByVal** SerialNumber **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
SerialNumber	Integer	Serial Number to assign in memory

#### Return Type

String

#### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Sensor_Identification_Number	0	"Error"

#### Comments

The SerialNumber used with this command must be a numerical value between zero and +16,777,215. It cannot contain any alpha or special characters.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Unit\_Code*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Set\_Unit\_Code is used to set the value for Engineering Units stored in the EEPROM of the USB Device.

### **Definition**

Public Function Set\_Unit\_Code(ByVal DeviceHandle As String, ByVal UnitCode As Integer) As String

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device
<u>UnitCode</u>	Integer	Unit Code to assign in memory

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Unit_Code	0	"Error"

### **Comments**

Please refer to the [Unit Codes](#) table for more information.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Decimal\_Point*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

*Set\_Decimal\_Point* is used to set the decimal point format stored in the EEPROM of the USB Device.

### Definition

Public Function Set\_Decimal\_Point(**ByVal** DeviceHandle **As String**, **ByVal** DecimalPoint **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
DecimalPoint	Integer	Decimal Point Format to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Decimal_Point	0	"Error"

### Comments

The commands with numerical values that should be taken into account with the DecimalPoint include:

[Set\\_Load\\_of>Loading Point](#) and [Get\\_Load\\_of>Loading Point](#). The DecimalPoint should be used to format a numerical value using the following expression:

---

### For example:

If the Offset Value = 1000 and the Decimal Point = 2, then:

DecimalPoint				
Hexadecimal	Decimal	Octal	Binary	Decimal Point Format
00	0	0000	00000000	0.
01	1	0001	00000001	0.0
02	2	0002	00000010	0.00
03	3	0003	00000011	0.000

DecimalPoint				
Hexadecimal	Decimal	Octal	Binary	Decimal Point Format
04	4	0004	00000100	0.0000
05	5	0005	00000101	0.00000
06	6	0006	00000110	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Calibration\_Register*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

`Set_Calibration_Register` is used to set the various Calibration Registers including: Resistance, Voltage, Current and Temperature.

### Definition

**Public Function** `Set_Calibration_Register(ByVal DeviceHandle As String, ByVal Register As Byte, ByVal HighByte As Byte, ByVal LowByte As Byte) As String`

Parameters	Data Type	Description
<code>DeviceHandle</code>	String	Handle assigned to the USB Device
<code>Register</code>	Byte	Calibration Register
<code>HighByte</code>	Byte	High Byte of data
<code>LowByte</code>	Byte	Low Byte of data

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
<code>Set_Calibration_Register</code>	0	"Error"

### Comments

The Calibration Registers can be seen in the Register table below. The HighByte and LowByte are used to pass data to the register. The HighByte and LowByte should be equal to zero for each of the Registers except for the following:

50% Load + Multimeter Reading (Voltage in Volts)

50% load + Multimeter Reading (Current in mA)

Temperatures + 00.00 (°C)

Register				
Hexadecimal	Decimal	Octal	Binary	Calibration Register
00	0	0000	00000000	Initialize Resistance Calibration
01	1	0001	00000001	Resistance (0 Ω)
02	2	0002	00000010	Resistance (10 kΩ)
03	3	0003	00000011	Finalize Resistance Calibration
04	4	0004	00000100	Initialize Analog Outputs (Voltage & Current)

Register				
Hexadecimal	Decimal	Octal	Binary	Calibration Register
05	5	0005	00000101	10% Load (Voltage)
06	6	0006	00000110	90% Load (Voltage)
07	7	0007	00000111	50% Load + Multimeter Reading (Voltage in Volts)
08	8	0010	00001000	0% Load (Current)
09	9	0011	00001001	100% Load (Current)
0A	10	0012	00001010	50% load + Multimeter Reading (Current in mA)
0B	11	0013	00001011	Apply Analog Outputs Calibration Values
0C	12	0014	00001100	Temperatures + 00.00 (°C)
0D	13	0015	00001101	Reset Calibration
0E	14	0016	00001110	Bridge Input (2.00 mV/V)
0F	15	0017	00001111	Initialize Sensor and System Profile
10	16	0020	00010000	Initialize Analog Input (Voltage)
11	17	0021	00010001	Voltage Input (0.00VDC)
12	18	0022	00010010	Voltage Input (2.00VDC)
13	19	0023	00010011	Finalize Analog Input (Voltage) Calibration
14	20	0024	00010100	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Loading\_Point*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

Set\_Loading\_Point is used to set the loading point value stored in the EEPROM of the USB Device.

### Definition

Public Function Set\_Loading\_Point(**ByVal** DeviceHandle **As String**, **ByVal** LoadingPoint **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
LoadingPoint	Integer	Loading Point to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Loading_Point	0	"Error"

### Comments

Set\_Loading\_Point should be used when applying a physical load to the sensor. This can be used to perform a live system calibration of the sensor and the USB Device.

For FUTEK Model Numbers USB100 and USB200 only.

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
00	0	0000	00000000	Loading Point 0
01	1	0001	00000001	Loading Point 1
02	2	0002	00000010	Loading Point 2
03	3	0003	00000011	Loading Point 3
04	4	0004	00000100	Loading Point 4
05	5	0005	00000101	Loading Point 5
06	6	0006	00000110	Loading Point 6
07	7	0007	00000111	Loading Point 7
08	8	0010	00001000	Loading Point 8
09	9	0011	00001001	Loading Point 9
0A	10	0012	00001010	Loading Point 10

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
0B	11	0013	00001011	Loading Point 11
0C	12	0014	00001100	Loading Point 12
0D	13	0015	00001101	Loading Point 13
0E	14	0016	00001110	Loading Point 14
0F	15	0017	00001111	Loading Point 15
10	16	0020	00010000	Loading Point 16
11	17	0021	00010001	Loading Point 17
12	18	0022	00010010	Loading Point 18
13	19	0023	00010011	Loading Point 19
14	20	0024	00010100	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320 and USB230 only.

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
00	0	0000	00000000	Loading Point 0
01	1	0001	00000001	Loading Point 1
02	2	0002	00000010	Loading Point 2
03	3	0003	00000011	Loading Point 3
04	4	0004	00000100	Loading Point 4
05	5	0005	00000101	Loading Point 5
06	6	0006	00000110	Loading Point 6
07	7	0007	00000111	Loading Point 7
08	8	0010	00001000	Loading Point 8
09	9	0011	00001001	Loading Point 9
0A	10	0012	00001010	Loading Point 10
0B	11	0013	00001011	Loading Point 11
0C	12	0014	00001100	Loading Point 12
0D	13	0015	00001101	Loading Point 13
0E	14	0016	00001110	Loading Point 14
0F	15	0017	00001111	Loading Point 15
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers IHH500 and IPM650 only.

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
00	0	0000	00000000	Loading Point 0
01	1	0001	00000001	Loading Point 1
02	2	0002	00000010	Loading Point 2
03	3	0003	00000011	Loading Point 3
04	4	0004	00000100	Loading Point 4
05	5	0005	00000101	Loading Point 5
06	6	0006	00000110	Loading Point 6
07	7	0007	00000111	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Load\_of>Loading\_Point*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

Set\_Load\_of>Loading\_Point is used to set the load value related to the loading point stored in the EEPROM of the USB Device.

### Definition

Public Function Set\_Load\_of>Loading\_Point(**ByVal** DeviceHandle **As String**, **ByVal** LoadingPoint **As Integer**, **ByVal** LoadValue **As Integer**) **As String**

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device
LoadingPoint	Integer	Loading Point to assign in memory
LoadValue	Integer	Load Value to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Load_of>Loading_Point	0	"Error"

### Comments

The LoadValue used with this command must be a numerical value between zero and +65,535.

### For FUTEK Model Numbers USB100 and USB200 only.

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
00	0	0000	00000000	Loading Point 0
01	1	0001	00000001	Loading Point 1
02	2	0002	00000010	Loading Point 2
03	3	0003	00000011	Loading Point 3
04	4	0004	00000100	Loading Point 4
05	5	0005	00000101	Loading Point 5
06	6	0006	00000110	Loading Point 6
07	7	0007	00000111	Loading Point 7
08	8	0010	00001000	Loading Point 8
09	9	0011	00001001	Loading Point 9

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
0A	10	0012	00001010	Loading Point 10
0B	11	0013	00001011	Loading Point 11
0C	12	0014	00001100	Loading Point 12
0D	13	0015	00001101	Loading Point 13
0E	14	0016	00001110	Loading Point 14
0F	15	0017	00001111	Loading Point 15
10	16	0020	00010000	Loading Point 16
11	17	0021	00010001	Loading Point 17
12	18	0022	00010010	Loading Point 18
13	19	0023	00010011	Loading Point 19
14	20	0024	00010100	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320 and USB230 only.

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
00	0	0000	00000000	Loading Point 0
01	1	0001	00000001	Loading Point 1
02	2	0002	00000010	Loading Point 2
03	3	0003	00000011	Loading Point 3
04	4	0004	00000100	Loading Point 4
05	5	0005	00000101	Loading Point 5
06	6	0006	00000110	Loading Point 6
07	7	0007	00000111	Loading Point 7
08	8	0010	00001000	Loading Point 8
09	9	0011	00001001	Loading Point 9
0A	10	0012	00001010	Loading Point 10
0B	11	0013	00001011	Loading Point 11
0C	12	0014	00001100	Loading Point 12
0D	13	0015	00001101	Loading Point 13
0E	14	0016	00001110	Loading Point 14
0F	15	0017	00001111	Loading Point 15
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers IHH500 and IPM650 only.

LoadingPoint				
Hexadecimal	Decimal	Octal	Binary	Point
00	0	0000	00000000	Loading Point 0
01	1	0001	00000001	Loading Point 1
02	2	0002	00000010	Loading Point 2
03	3	0003	00000011	Loading Point 3
04	4	0004	00000100	Loading Point 4
05	5	0005	00000101	Loading Point 5
06	6	0006	00000110	Loading Point 6
07	7	0007	00000111	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Sensor\_Configuration*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

*Set\_Sensor\_Configuration* is used to set the sensor configuration for Input / Output Type, Nominal Sensitivity and Bridge Resistance.

### Definition

Public Function Set\_Sensor\_Configuration(**ByVal** DeviceHandle **As String**, **ByVal** SensorConfiguration **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
SensorConfiguration	Integer	Sensor Configuration for Input / Output Type, Sensitivity and Bridge Resistance

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Sensor_Configuration	0	"Error"

### Comments

The SensorConfiguration is an 8-bit number. The bits are stored in the following format RTTTSSBB.

R = Reserved

TTT = Input / Output Type (applies to FUTEK Model Number IHH500 and IPM650 only)

SS = Nominal Sensitivity

BB = Nominal Bridge Resistance

SensorConfiguration						
Hexadecimal	Decimal	Octal	Binary	Bit	Bit Format	Configuration
00	0	0000	00000000	R	0 XXX XX XX	Reserved
00	0	0000	00000000	TTT	X 000 XX XX	Bridge
01	1	0001	00000001	TTT	X 001 XX XX	Bridge and Pulse
02	2	0002	00000010	TTT	X 010 XX XX	Voltage Output
03	3	0003	00000011	TTT	X 011 XX XX	Voltage Output and Pulse
04	4	0004	00000100	TTT	X 100 XX XX	Current Output
05	5	0005	00000101	TTT	X 101 XX XX	Current Output and Pulse

SensorConfiguration						
Hexadecimal	Decimal	Octal	Binary	Bit	Bit Format	Configuration
06	6	0006	00000110	TTT	X 110 XX XX	Undefined
07	7	0007	00000111	TTT	X 111 XX XX	Undefined
00	0	0000	00000000	SS	X XXX 00 XX	0.5 mV/V
01	1	0001	00000001	SS	X XXX 01 XX	1.0 mV/V
02	2	0002	00000010	SS	X XXX 10 XX	2.0 mV/V
03	3	0003	00000011	SS	X XXX 11 XX	4.0 mV/V
00	0	0000	00000000	BB	X XXX XX 00	120 ohms
01	1	0001	00000001	BB	X XXX XX 01	350 ohms
02	2	0002	00000010	BB	X XXX XX 10	700 ohms
03	3	0003	00000011	BB	X XXX XX 11	1000 ohms

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Load\_Switch*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

Set\_Load\_Switch is used to enable or disable the simulated load of the USB Device.

### Definition

Public Function Set\_Load\_Switch(ByVal DeviceHandle As String, ByVal SimulatedLoad As Integer) As String

Parameters	Data Type	Description
DeviceHandle	String	Handle assigned to the USB Device
SimulatedLoad	Integer	Simulated Load to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Load_Switch	0	"Error"

### Comments

Set\_Load\_Switch can be used to simulate an input to the USB Device. The simulated load provides a known, electrical, unbalancing of a strain gage bridge similar to a shunt calibration. The simulated load will be in addition to any input from the sensor, therefore all preloads should be removed prior to enabling the simulated load.

SimulatedLoad				
Hexadecimal	Decimal	Octal	Binary	Load
00	0	0000	00000000	Disable
01	1	0001	00000001	Enable
02	2	0002	00000010	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Number\_of>Loading\_Points*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

Set\_Number\_of>Loading\_Points is used to set the number of loading point values stored in the EEPROM of the USB Device.

### Definition

**Public Function** Set\_Number\_of>Loading\_Points(**ByVal** DeviceHandle **As String**, **ByVal** NumberOfLoadingPoints **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
NumberOfLoadingPoints	Integer	Number of Loading Points to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Number_of>Loading_Points	0	"Error"

### Comments

The NumberOfLoadingPoints is used to assign the total number of points that are recorded and stored in memory at the time of calibration. Please refer to [Set\\_Loading\\_Point](#) and [Set\\_Load\\_of>Loading\\_Point](#) because there should be a LoadingPoint Value and a LoadValue for each LoadingPoint. The valid range for NumberOfLoadingPoints depends on the FUTEK Model Number.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_ADC\_Configuration*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

Set\_ADC\_Configuration is used to set the sampling rate of the ADC Configuration for the USB Device.

### Definition

Public Function Set\_ADC\_Configuration(**ByVal** DeviceHandle **As String**, **ByVal** SamplingRate **As Byte**)  
**As String**

#### Parameters      Data Type      Description

<u>DeviceHandle</u>	String	Handle assigned to the USB Device
SamplingRate	Byte	Sampling Rate to assign to the ADC

#### Return Type

String

#### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_ADC_Configuration	0	"Error"

### Comments

The SamplingRate is used to assign the ADC Configuration that will be used when converting data from analog to digital. This setting applies to the conversion rate of the hardware. The rate at which data is recorded or collected by software will vary depending on the design of the software and the performance of the hardware that is running it.

For FUTEK Model Numbers USB100 and USB200 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	External Input, 50/60 Hz Rejection, 5 Samples Per Second
01	1	0001	00000001	External Input, 50/60 Hz Rejection, 10 Samples Per Second
02	2	0002	00000010	External Input, 50 Hz Rejection, 5 Samples Per Second
03	3	0003	00000011	External Input, 50 Hz Rejection, 10 Samples Per Second
04	4	0004	00000100	External Input, 60 Hz Rejection, 5 Samples Per Second
05	5	0005	00000101	External Input, 60 Hz Rejection, 10 Samples Per Second
06	6	0006	00000110	Reserved, Do Not Use
07	7	0007	00000111	Reserved, Do Not Use
08	8	0010	00001000	Temperature Input, 50/60 Hz Rejection, 5 Samples Per Second

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
09	9	0011	00001001	Reserved, Do Not Use
0A	10	0012	00001010	Temperature Input, 50 Hz Rejection, 5 Samples Per Second
0B	11	0013	00001011	Reserved, Do Not Use
0C	12	0014	00001100	Temperature Input, 60 Hz Rejection, 5 Samples Per Second
0D	13	0015	00001101	Reserved, Do Not Use
0E	14	0016	00001110	Reserved, Do Not Use
0F	15	0017	00001111	Reserved, Do Not Use
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers USB110 and USB210 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	2.5 Samples Per Second
01	1	0001	00000001	10 Samples Per Second
02	2	0002	00000010	20 Samples Per Second
03	3	0003	00000011	25 Samples Per Second
04	4	0004	00000100	30 Samples Per Second
05	5	0005	00000101	50 Samples Per Second
06	6	0006	00000110	60 Samples Per Second
07	7	0007	00000111	80 Samples Per Second
08	8	0010	00001000	100 Samples Per Second
09	9	0011	00001001	150 Samples Per Second
0A	10	0012	00001010	200 Samples Per Second
0B	11	0013	00001011	250 Samples Per Second
0C	12	0014	00001100	300 Samples Per Second
0D	13	0015	00001101	500 Samples Per Second
0E	14	0016	00001110	750 Samples Per Second
0F	15	0017	00001111	1000 Samples Per Second
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers USB120, USB220, USB320, IHH500 and IPM650 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	5 Samples Per Second
01	1	0001	00000001	10 Samples Per Second
02	2	0002	00000010	15 Samples Per Second
03	3	0003	00000011	20 Samples Per Second
04	4	0004	00000100	25 Samples Per Second
05	5	0005	00000101	30 Samples Per Second
06	6	0006	00000110	50 Samples Per Second
07	7	0007	00000111	60 Samples Per Second
08	8	0010	00001000	100 Samples Per Second
09	9	0011	00001001	300 Samples Per Second
0A	10	0012	00001010	600 Samples Per Second
0B	11	0013	00001011	960 Samples Per Second
0C	12	0014	00001100	1200 Samples Per Second
0D	13	0015	00001101	1600 Samples Per Second
0E	14	0016	00001110	2400 Samples Per Second
0F	15	0017	00001111	4800 Samples Per Second
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Number USB230 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	2.5 Samples Per Second
01	1	0001	00000001	5 Samples Per Second
02	2	0002	00000010	10 Samples Per Second
03	3	0003	00000011	15 Samples Per Second
04	4	0004	00000100	25 Samples Per Second
05	5	0005	00000101	30 Samples Per Second
06	6	0006	00000110	50 Samples Per Second
07	7	0007	00000111	60 Samples Per Second
08	8	0010	00001000	100 Samples Per Second
09	9	0011	00001001	500 Samples Per Second
0A	10	0012	00001010	1000 Samples Per Second
0B	11	0013	00001011	2000 Samples Per Second

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
0C	12	0014	00001100	3750 Samples Per Second
0D	13	0015	00001101	7500 Samples Per Second
0E	14	0016	00001110	15000 Samples Per Second
0F	15	0017	00001111	15000 Samples Per Second
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_ADC\_Configuration2*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

*Set\_ADC\_Configuration2* is used to set the PGA setting and the sampling rate of the ADC Configuration for the USB Device.

### Definition

**Public Function** *Set\_ADC\_Configuration2*(**ByVal** DeviceHandle **As String**, **ByVal** ADCConfiguration **As Integer**) **As String**

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device
ADCConfiguration	Integer	Sampling Rate to assign to the ADC

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
<i>Set_ADC_Configuration2</i>	0	"Error"

### Comments

The ADCConfiguration is used to assign the Programmable Gain Amplifier (PGA) setting and the sampling rate that will be used when converting data from analog to digital. This setting applies to the conversion rate of the hardware. The rate at which data is recorded or collected by software will vary depending on the design of the software and the performance of the hardware that is running it.

The ADCConfiguration is an 8-bit number.

The high nibble represents the PGA setting and the low nibble represents the sampling rate setting. Please refer to [Set\\_ADC\\_Configuration](#) for more information about the sampling rate.

### For FUTEK Model Numbers USB100 and USB200 only.

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
0	0	00	0000	Gain = 1
1	1	01	0001	Reserved, Do Not Use
2	2	02	0010	Reserved, Do Not Use
3	3	03	0011	Reserved, Do Not Use
4	4	04	0100	Reserved, Do Not Use

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
5	5	05	0101	Reserved, Do Not Use
6	6	06	0110	Reserved, Do Not Use
7	7	07	0111	Reserved, Do Not Use
8	8	10	1000	Reserved, Do Not Use
9	9	11	1001	Reserved, Do Not Use
A	10	12	1010	Reserved, Do Not Use
B	11	13	1011	Reserved, Do Not Use
C	12	14	1100	Reserved, Do Not Use
D	13	15	1101	Reserved, Do Not Use
E	14	16	1110	Reserved, Do Not Use
F	15	17	1111	Reserved, Do Not Use

For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320, IHH500 and IPM650 only.

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
0	0	00	0000	Gain = 1
1	1	01	0001	Gain = 2
2	2	02	0010	Gain = 4
3	3	03	0011	Gain = 8
4	4	04	0100	Gain = 16
5	5	05	0101	Gain = 32
6	6	06	0110	Gain = 64
7	7	07	0111	Gain = 128
8	8	10	1000	Reserved, Do Not Use
9	9	11	1001	Reserved, Do Not Use
A	10	12	1010	Reserved, Do Not Use
B	11	13	1011	Reserved, Do Not Use
C	12	14	1100	Reserved, Do Not Use
D	13	15	1101	Reserved, Do Not Use
E	14	16	1110	Reserved, Do Not Use
F	15	17	1111	Reserved, Do Not Use

For FUTEK Model Numbers USB230 only.

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
0	0	00	0000	Gain = 1
1	1	01	0001	Gain = 2

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
2	2	02	0010	Gain = 4
3	3	03	0011	Gain = 8
4	4	04	0100	Gain = 16
5	5	05	0101	Gain = 32
6	6	06	0110	Gain = 64
7	7	07	0111	Gain = 64
8	8	10	1000	Reserved, Do Not Use
9	9	11	1001	Reserved, Do Not Use
A	10	12	1010	Reserved, Do Not Use
B	11	13	1011	Reserved, Do Not Use
C	12	14	1100	Reserved, Do Not Use
D	13	15	1101	Reserved, Do Not Use
E	14	16	1110	Reserved, Do Not Use
F	15	17	1111	Reserved, Do Not Use

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Average\_Setting*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

Set\_Average\_Setting is used to enable or disable the averaging setting of the USB Device.

### Definition

Public Function Set\_Average\_Setting(ByVal DeviceHandle As String, ByVal AverageSetting As Integer)  
As String

Parameters	Data Type	Description
DeviceHandle	String	Handle assigned to the USB Device
AverageSetting	Integer	Average Setting to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Average_Setting	0	"Error"

### Comments

The AverageSetting can be used to disable averaging or to enable a moving or mean average. The number of sample that are used in the average can be selected from 2 to 256 samples.

For FUTEK Model Numbers USB110 and USB210 only.

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
00	0	0000	00000000	Disable Averaging	Average = 2
01	1	0001	00000001	Disable Averaging	Average = 4
02	2	0002	00000010	Disable Averaging	Average = 8
03	3	0003	00000011	Disable Averaging	Average = 16
04	4	0004	00000100	Disable Averaging	Average = 32
05	5	0005	00000101	Disable Averaging	Average = 64
06	6	0006	00000110	Disable Averaging	Average = 128
07	7	0007	00000111	Disable Averaging	Average = 256
08	8	0010	00001000	Undefined	Undefined
...	...	...	...	...	...
0F	15	0017	00001111	Undefined	Undefined
10	16	0020	00010000	Moving Average	Average = 2

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
11	17	0021	00010001	Moving Average	Average = 4
12	18	0022	00010010	Moving Average	Average = 8
13	19	0023	00010011	Moving Average	Average = 16
14	20	0024	00010100	Moving Average	Average = 32
15	21	0025	00010101	Moving Average	Average = 64
16	22	0026	00010110	Moving Average	Average = 128
17	23	0027	00010111	Moving Average	Average = 256
18	24	0030	00011000	Undefined	Undefined
...	...	...	...	...	...
1F	31	0037	00011111	Undefined	Undefined
20	32	0040	00100000	Mean Average	Average = 2
21	33	0041	00100001	Mean Average	Average = 4
22	34	0042	00100010	Mean Average	Average = 8
23	35	0043	00100011	Mean Average	Average = 16
24	36	0044	00100100	Mean Average	Average = 32
25	37	0045	00100101	Mean Average	Average = 64
26	38	0046	00100110	Mean Average	Average = 128
27	39	0047	00100111	Mean Average	Average = 256
28	40	0050	00101000	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

For FUTEK Model Numbers USB120, USB220, USB320, USB230 and IHH500 only.

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
00	0	0000	00000000	Disable Averaging	Average = 2
01	1	0001	00000001	Disable Averaging	Average = 4
02	2	0002	00000010	Disable Averaging	Average = 8
03	3	0003	00000011	Disable Averaging	Average = 16
04	4	0004	00000100	Disable Averaging	Average = 32
05	5	0005	00000101	Disable Averaging	Average = 64
06	6	0006	00000110	Disable Averaging	Average = 128
07	7	0007	00000111	Disable Averaging	Average = 256
08	8	0010	00001000	Undefined	Undefined
...	...	...	...	...	...
0F	15	0017	00001111	Undefined	Undefined
10	16	0020	00010000	Moving Average	Average = 2
11	17	0021	00010001	Moving Average	Average = 4
12	18	0022	00010010	Moving Average	Average = 8

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
13	19	0023	00010011	Moving Average	Average = 16
14	20	0024	00010100	Moving Average	Average = 32
15	21	0025	00010101	Moving Average	Average = 64
16	22	0026	00010110	Moving Average	Average = 128
17	23	0027	00010111	Moving Average	Average = 256
18	24	0030	00011000	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Sensitivity*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

`Set_Sensitivity` is used to set the nominal sensitivity value stored in the EEPROM of the USB Device.

### Definition

Public Function `Set_Sensitivity(ByVal DeviceHandle As String, ByVal Sensitivity As Integer) As String`

Parameters	Data Type	Description
<code>DeviceHandle</code>	String	Handle assigned to the USB Device
<code>Sensitivity</code>	Integer	Nominal Sensitivity to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
<code>Set_Sensitivity</code>	0	"Error"

### Comments

The Sensitivity is a nominal value that represents the sensor's rated output expressed in mV/V. The Sensitivity value is for reference only and does not affect the calibration of the device.

Sensitivity				
Hexadecimal	Decimal	Octal	Binary	Nominal Sensitivity
00	0	0000	00000000	0.5 mV/V
01	1	0001	00000001	1.0 mV/V
02	2	0002	00000010	2.0 mV/V
03	3	0003	00000011	4.0 mV/V
04	4	0004	00000100	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Bridge\_Resistance*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

`Set_Bridge_Resistance` is used to set the nominal bridge resistance value stored in the EEPROM of the USB Device.

### Definition

**Public Function** `Set_Bridge_Resistance(ByVal DeviceHandle As String, ByVal BridgeResistance As Integer) As String`

Parameters	Data Type	Description
<code>DeviceHandle</code>	String	Handle assigned to the USB Device
<code>BridgeResistance</code>	Integer	Nominal Bridge Resistance to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
<code>Set_Bridge_Resistance</code>	0	"Error"

### Comments

The `BridgeResistance` is a nominal value that represents the sensor's bridge resistance expressed in ohms ( $\Omega$ ). The `BridgeResistance` value is for reference only and does not affect the calibration of the device.

BridgeResistance				
Hexadecimal	Decimal	Octal	Binary	Nominal Resistance
00	0	0000	00000000	120 $\Omega$
01	1	0001	00000001	350 $\Omega$
02	2	0002	00000010	700 $\Omega$
03	3	0003	00000011	1000 $\Omega$
04	4	0004	00000100	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Direction*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### Summary

Set\_Direction is used to set the direction value stored in the EEPROM of the USB Device.

### Definition

Public Function Set\_Direction(**ByVal** DeviceHandle **As String**, **ByVal** Direction **As Integer**) **As String**

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device
Direction	Integer	Direction to assign in memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Direction	0	"Error"

### Comments

The Direction is a numerical value indicating the direction type and the number of directions used during calibration.

Direction					
Hexadecimal	Decimal	Octal	Binary	Direction 1	Direction 2
00	0	0000	00000000	Compression	Undefined
01	1	0001	00000001	Tension	Undefined
02	2	0002	00000010	CW	Undefined
03	3	0003	00000011	CCW	Undefined
04	4	0004	00000100	Pressure	Undefined
05	5	0005	00000101	Vacuum	...
06	6	0006	00000110	Undefined	Undefined
...	...	...	...	...	...
0F	15	0017	00001111	Undefined	Undefined
10	16	0020	00010000	Compression	Tension
11	17	0021	00010001	Tension	Compression
12	18	0022	00010010	CW	CCW
13	19	0023	00010011	CCW	CW

Direction					
Hexadecimal	Decimal	Octal	Binary	Direction 1	Direction 2
14	20	0024	00010100	Pressure	Vacuum
15	21	0025	00010101	Vacuum	Pressure
16	22	0026	00010110	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Zero\_Correction*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

*Set\_Zero\_Correction* is used to set the new loading point value based on a zero correction for fixturing.

### **Definition**

**Public Function** Set\_Zero\_Correction(**ByVal** DeviceHandle **As String**, **ByVal** Point **As Integer**, **ByVal** ZeroCorrectionValue **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
Point	Integer	Loading Point to perform Zero Correction on
ZeroCorrectionValue	Integer	New Value to assign to a Loading Point

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Zero_Correction	0	"Error"

### **Comments**

The Point is a numerical value representing the loading point that will be modified. The ZeroCorrectionValue is a numerical value that will be assigned to the loading point. This command should be used when the loading point value includes an offset due to the calibration fixture. The ZeroCorrectionValue should represent the loading point value after the value of the fixturing is subtracted. This command should be used after [Set\\_Loading\\_Point](#) has assigned the loading point value. Please refer to [Get\\_Loading\\_Point](#) in order to determine the value that is currently assigned to the loading point.

### **For example:**

If the loading point value = 8,388,608 without any fixturing and the loading point value = 8,388,908 with the fixturing, then the difference is 300 counts. Therefore, each loading point that is taken with the fixturing should then have 300 counts subtracted from its loading point value. The result of the subtraction is the value that should be assigned as the ZeroCorrectionValue.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Shunt\_Value*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

*Set\_Shunt\_Value* is used to set the shunt value stored in the EEPROM of the USB Device.

### **Definition**

Public Function Set\_Shunt\_Value(ByVal DeviceHandle As String, ByVal ShuntValue As Integer) As String

#### **Parameters**

Parameter	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
ShuntValue	Integer	Value associated with a simulated load

#### **Return Type**

String

#### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Shunt_Value	0	"Error"

#### **Comments**

The ShuntValue is a numerical value representing the simulated load that is applied to the USB Device. This command should be used when there is zero load applied to the sensor. This command should be used after [Set\\_Load\\_Switch](#) has been enabled. Please remember to disable the simulated load when finished.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Calibration\_Code*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Set\_Calibration\_Code is used to set the calibration code stored in the EEPROM of the USB Device.

### **Definition**

Public Function Set\_Calibration\_Code(**ByVal** DeviceHandle As String, **ByVal** CalibrationCode **As Integer**)  
**As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
CalibrationCode	Integer	Calibration Code to assign in memory

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Calibration_Code	0	"Error"

### **Comments**

The CalibrationCode is a numerical value representing the operator who performed the calibration of the USB Device. The CalibrationCode must be a numerical value between zero and +255.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Calibration\_Day*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Set\_Calibration\_Day is used to set the calibration day stored in the EEPROM of the USB Device.

### **Definition**

Public Function Set\_Calibration\_Day(**ByVal** DeviceHandle **As String**, **ByVal** CalibrationDay **As Integer**)  
**As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
CalibrationDay	Integer	Calibration Day to assign in memory

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Calibration_Day	0	"Error"

### **Comments**

The CalibrationDay is a numerical value representing the day of the month when the calibration of the USB Device was performed. The CalibrationDay must be a numerical value between +1 and +31.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Calibration\_Month*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Set\_Calibration\_Month is used to set the calibration month stored in the EEPROM of the USB Device.

### **Definition**

Public Function Set\_Calibration\_Month(ByVal DeviceHandle As String, ByVal CalibrationMonth As Integer) As String

#### **Parameters**

	<b>Data Type</b>	<b>Description</b>
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
CalibrationMonth	Integer	Calibration Month to assign in memory

#### **Return Type**

String

#### **Return Values**

<b>Return Values</b>		
<b>Command/Variable</b>	<b>If Successful</b>	<b>If Unsuccessful</b>
Set_Calibration_Month	0	“Error”

#### **Comments**

The CalibrationMonth is a numerical value representing the month when the calibration of the USB Device was performed. The CalibrationMonth must be a numerical value between +1 and +12.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Calibration\_Year*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Set\_Calibration\_Year is used to set the calibration year stored in the EEPROM of the USB Device.

### **Definition**

Public Function Set\_Calibration\_Year(**ByVal** DeviceHandle **As String**, **ByVal** CalibrationYear **As Integer**)  
**As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
CalibrationYear	Integer	Calibration Year to assign in memory

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Calibration_Year	0	"Error"

### **Comments**

The CalibrationYear is a numerical value representing the year when the calibration of the USB Device was performed. The CalibrationYear must be a numerical value between zero and +255. The Calibration year is zero indexed from the year 2000.

### **For Example:**

If the CalibrationYear is 11, then the year when the calibration was performed is 2011.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Set\_Pulses\_Per\_Rotation*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Set\_Pulses\_Per\_Rotation is used to set the number of pulses per rotation stored in the EEPROM of the USB Device.

### **Definition**

Public Function Set\_Pulses\_Per\_Rotation(**ByVal** DeviceHandle **As String**, **ByVal** PulsesPerRotation **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
PulsesPerRotation	Integer	Pulses Per Rotation to assign in memory

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Pulses_Per_Rotation	0	"Error"

### **Comments**

The PulsesPerRotation is a numerical value representing the number of pulses per rotation associated with the encoder of a torque sensor. The PulsesPerRotation must be a numerical value between zero and +65,535.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## Get Commands

### *Get\_Offset\_Value*

#### **Summary**

Get\_Offset\_Value is used to get the offset value stored in the EEPROM of the USB Device.

#### **Definition**

Public Function Get\_Offset\_Value(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

#### **Return Type**

String

#### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Offset_Value	Number	"Error"

#### **Comments**

Get\_Offset\_Value is used to get the offset value stored in the EEPROM of the USB Device. The returned value should be a numerical value between -16,777,216 and +16,777,215.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Fullscale\_Value*

### **Summary**

Get\_Fullscale\_Value is used to get the fullscale value stored in the EEPROM of the USB Device.

### **Definition**

**Public Function** Get\_Fullscale\_Value(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Description
<u>DeviceHandle</u>	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Fullscale_Value	Number	"Error"

### **Comments**

Get\_Fullscale\_Value is used to get the fullscale value stored in the EEPROM of the USB Device. The returned value should be a numerical value between -16,777,216 and +16,777,215.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Normal\_Data\_Request*

### **Summary**

Normal\_Data\_Request is used to get the current ADC value from the USB Device.

### **Definition**

**Public Function** Normal\_Data\_Request(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Normal_Data_Request	Number	"Error"

### **Comments**

Normal\_Data\_Request is used to get the current ADC value from the USB Device. The returned value should be a numerical value between -16,777,216 and +16,777,215.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Fast\_Data\_Request*

### Summary

Fast\_Data\_Request is used to get the current ADC values from the USB Device.

### Definition

```
Public Function Fast_Data_Request(ByVal DeviceHandle As String, ByVal DelimitedStyle As Integer,
ByVal ChannelNumber As Integer, ByVal BoardType As String, ByVal SamplingRate As String, ByVal
FirmwareVersion As String) As String
```

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
DelimitedStyle	Integer	Assigns the Delimiter and Terminator
ChannelNumber	Integer	Channel Number of the device
BoardType	String	Board Type of the device
SamplingRate	String	Sampling Rate of the device
FirmwareVersion	String	Firmware Version of the device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Fast_Data_Request	String	"Error"

### Comments

Fast\_Data\_Request is used to get the current ADC Values from the USB Device. The returned value should be a string of data which will vary in size depending on how many samples are in the buffer. If there are no ADC Values in the buffer, then the response will be an empty string.

The DelimitedStyle is be used to create a comma delimited or tab delimited format.

The ChannelNumber is used in order to maintain different counters for each USB Device.

The BoardType can be found using [Get Type of Board](#).

The SamplingRate can be found using [Get ADC Sampling Rate Setting](#).

The FirmwareVersion can be found using [Get Firmware Version](#).

**For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320 and USB230 only.**

**Firmware Version 4 and higher only.**

DelimitedStyle					
Hexadecimal	Decimal	Octal	Binary	Delimiter	Terminator
00	0	0000	00000000	Comma	Comma
01	1	0001	00000001	Tab	Tab

DelimitedStyle					
Hexadecimal	Decimal	Octal	Binary	Delimiter	Terminator
02	2	0002	00000010	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

**For Example:**

If the DelimitedStyle is zero, then the return value will have the following format:

*Counter [,] Number of Samples [,] ADC Value [,] Place Holder [,] Place Holder [,] Place Holder [,] Date [,] Time*

```
0,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
1,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
2,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
3,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
4,14,13214080,,,Tuesday, October 18,2011,2:29:27 PM
5,14,13214592,,,Tuesday, October 18,2011,2:29:27 PM
6,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
7,14,13214080,,,Tuesday, October 18,2011,2:29:27 PM
8,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
9,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
10,14,13213824,,,Tuesday, October 18,2011,2:29:27 PM
11,14,13214080,,,Tuesday, October 18,2011,2:29:27 PM
12,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
13,14,13214336,,,Tuesday, October 18,2011,2:29:27 PM
```

If the DelimitedStyle is one, then the return value will have the following format:

*Counter [tab] Number of Samples [tab] ADC Value [tab] Place Holder [tab] Place Holder [tab] Place Holder [tab] Date [tab] Time*

0	14	13214336	Tuesday, October 18,2011	2:29:27 PM
1	14	13214336	Tuesday, October 18,2011	2:29:27 PM
2	14	13214336	Tuesday, October 18,2011	2:29:27 PM
3	14	13214336	Tuesday, October 18,2011	2:29:27 PM
4	14	13214080	Tuesday, October 18,2011	2:29:27 PM
5	14	13214592	Tuesday, October 18,2011	2:29:27 PM
6	14	13214336	Tuesday, October 18,2011	2:29:27 PM
7	14	13214080	Tuesday, October 18,2011	2:29:27 PM
8	14	13214336	Tuesday, October 18,2011	2:29:27 PM
9	14	13214336	Tuesday, October 18,2011	2:29:27 PM
10	14	13213824	Tuesday, October 18,2011	2:29:27 PM
11	14	13214080	Tuesday, October 18,2011	2:29:27 PM
12	14	13214336	Tuesday, October 18,2011	2:29:27 PM
13	14	13214336	Tuesday, October 18,2011	2:29:27 PM

The three Place Holders are used to reserve the location to insert the calculated tracking, peak and valley values based on the ADC Value.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Version\_of\_Board*

### **Summary**

Version\_of\_Board is used to get the version of board from the USB Device.

### **Definition**

**Public Function** Version\_of\_Board(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Version_of_Board	0	"Error"

### **Comments**

Version\_of\_Board is used to get the version of board from the USB Device.

The type of board can be found using [Get Type of Board](#).

The hardware version can be found using [Get Hardware Version](#).

The firmware version can be found using [Get Firmware Version](#).

The firmware year can be found using [Get Firmware Year](#).

The firmware month can be found using [Get Firmware Month](#).

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Reset\_Board***

### **Summary**

Reset\_Board is used to reset the board for the USB Device.

### **Definition**

Public Function Reset\_Board(**ByVal** DeviceHandle **As String**) **As String**

<b>Parameters</b>	<b>Data Type</b>	<b>Description</b>
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

<b>Return Values</b>		
<b>Command/Variable</b>	<b>If Successful</b>	<b>If Unsuccessful</b>
Reset_Board	0	"Error"

### **Comments**

Reset\_Board is used to reset the board for the USB Device. This command may take several seconds to respond.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Reset\_Board2***

### **Summary**

Reset\_Board2 is used to reset the board for the USB Device.

### **Definition**

Public Function Reset\_Board2(**ByVal** DeviceHandle **As String**) **As String**

<b>Parameters</b>	<b>Data Type</b>	<b>Description</b>
<a href="#"><u>DeviceHandle</u></a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

<b>Return Values</b>		
<b>Command/Variable</b>	<b>If Successful</b>	<b>If Unsuccessful</b>
Reset_Board2	0	"Error"

### **Comments**

Reset\_Board2 is used to reset the board for the USB Device. This command may take longer than 10 seconds to respond.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Display\_Page*

### **Summary**

Get\_Display\_Page is used to get the display page from the USB Device.

### **Definition**

**Public Function** Get\_Display\_Page(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Display_Page	0	"Error"

### **Comments**

Get\_Display\_Page is used to get the display page from the USB Device. This command assigns the values for each line of the LCD to the following variables: LDCLine1, LCDLine2, LCDLine3 and LCDLine4. This command assigns a string of data containing all four lines of the LCD to the following variable: DisplayPage.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Internal\_Register*

### Summary

Get\_Internal\_Register is used to get the internal register from the USB Device.

### Definition

Public Function Get\_Internal\_Register(**ByVal** DeviceHandle **As String**, **ByVal** Register **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
Register	Integer	This is the Internal Register to Read From

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Internal_Register	Number	"Error"

### Comments

Get\_Internal\_Register is used to get the internal register from the USB Device. The returned value should be a numerical value between zero and +16,777,215.

### This command applies to:

FUTEK Model Number IHH500 only.

Register				
Hexadecimal	Decimal	Octal	Binary	Terminator
00	0	0000	00000000	Data Logging Status
01	1	0001	00000001	Tare Register
02	2	0002	00000010	Offset 1 Register
03	3	0003	00000011	FULLSCALE 1 Register
04	4	0004	00000100	FULLSCALE 2 Register
05	5	0005	00000101	Sensor Capacity
				Decimal Point (High Byte) Unit (Middle Byte) Direction (Low Byte)
06	6	0006	00000110	
				Input Enable (High Byte) Output Enable (Middle Byte) Number of Digits (Low Byte)
07	7	0007	00000111	

Register				
Hexadecimal	Decimal	Octal	Binary	Terminator
08	8	0010	00001000	Offset 2 Register
09	9	0011	00001001	Peak Register
0A	10	0012	00001010	Valley Register
0B	11	0013	00001011	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Input Enable	
Bit	Flag
0	Power Enable Flag
1	Bridge Enable Flag
2	Voltage Input Enable Flag
3	Current Input Enable Flag
4	Pulse Enable Flag
5	Shunt Enable Flag
6	TEDS Enable Flag
7	RTD Enable Flag

Output Enable	
Bit	Flag
0	USB Enable Flag
1	ASCII Enable Flag
2	Relay1 Input Enable Flag
3	Relay2 Input Enable Flag
4	Voltage Output Enable Flag
5	Current Output Enable Flag
6	Auto TEDS Enable Flag
7	Reserved

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Data\_Logging*

### Summary

Get\_Data\_Logging is used to get the data logging value from the USB Device.

### Definition

**Public Function** Get\_Data\_Logging(**ByVal** DeviceHandle **As String**, **ByVal** Counter **As Integer**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
Counter	Integer	Data Logging Counter

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Data_Logging	0	"Error"

### Comments

Get\_Data\_Logging is used to get the data logging value from the USB Device. This command assigns the values for each of the following variables: DataLogging\_Counter, DataLogging\_Value1, DataLogging\_Value2, DataLogging\_Value3 and DataLogging\_Value4.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Get\_Rotation\_Values***

### **Summary**

Get\_Rotation\_Values is used to get the rotation values from the USB Device.

### **Definition**

**Public Function** Get\_Rotation\_Values(**ByVal** DeviceHandle **As String**) **As String**

<b>Parameters</b>	<b>Data Type</b>	<b>Description</b>
<a href="#"><u>DeviceHandle</u></a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

<b>Return Values</b>		
<b>Command/Variable</b>	<b>If Successful</b>	<b>If Unsuccessful</b>
Get_Rotation_Values	0	"Error"

### **Comments**

Get\_Rotation\_Values is used to get the rotation values from the USB Device. This command assigns the values for each of the following variables: AngleValue and RPMValue.

**For FUTEK Model Numbers USB410, IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Device\_Count*

### **Summary**

Get\_Device\_Count is used to get the number of USB Devices currently connected to the computer.

### **Definition**

**Public Function** Get\_Device\_Count() **As String**

<b>Parameters</b>	<b>Data Type</b>	<b>Description</b>
N/A	N/A	N/A

<b>Variables</b>	<b>Data Type</b>	<b>Description</b>
<a href="#">DeviceStatus</a>	String	Status of the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Device_Count	Number	"Error"
DeviceStatus	0	Status

### **Comments**

Get\_Device\_Count is used to get the number of USB Devices currently connected to the computer.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Device\_Serial\_Number*

### Summary

Get\_Device\_Serial\_Number is used to get the serial number of USB Device at the specified index.

### Definition

Public Function Get\_Device\_Serial\_Number(ByVal Index As String) As String

Parameters	Data Type	Description
Index	String	Zero Index location of the USB Device

Variables	Data Type	Description
<a href="#">DeviceStatus</a>	String	Status of the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Device_Count	Number	"Error"
DeviceStatus	0	Status

### Comments

Get\_Device\_Serial\_Number is used to get the serial number of USB Device at the specified index.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get>Loading Point*

### Summary

Get>Loading\_Point is used to get the loading point value stored in the EEPROM of the USB Device.

### Definition

Public Function Get>Loading\_Point( ByVal DeviceHandle As String, ByVal Point As Integer) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
Point	Integer	Loading Point to retrieve from memory

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get>Loading_Point	Number	"Error"

### Comments

Get>Loading\_Point is used to get the loading point value stored in the EEPROM of the USB Device. The returned value should be a numerical value between -16,777,216 and +16,777,215.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Offset\_Load*

### **Summary**

Get\_Offset\_Load is used to get the offset load stored in the EEPROM of the USB Device.

### **Definition**

**Public Function** Get\_Offset\_Load(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Offset_Load	Number	"Error"

### **Comments**

Get\_Offset\_Load is used to get the offset load stored in the EEPROM of the USB Device. The returned value should be a numerical value between zero and +65,535.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Get\_Load\_of>Loading\_Point***

### **Summary**

Get\_Load\_of>Loading\_Point is used to get the load value related to the loading point stored in the EEPROM of the USB Device.

### **Definition**

Public Function Get\_Load\_of>Loading\_Point(**ByVal** DeviceHandle **As String**, **ByVal** Point **As Integer**)  
**As String**

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device
Point	Integer	Loading Point to retrieve from memory

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Load_of>Loading_Point	Number	"Error"

### **Comments**

Get\_Load\_of>Loading\_Point is used to get the load value related to the loading point stored in the EEPROM of the USB Device. The returned value should be a numerical value between zero and +65,535.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Get\_Fullscale\_Load***

### **Summary**

Get\_Fullscale\_Load is used to get the fullscale load stored in the EEPROM of the USB Device.

### **Definition**

**Public Function** Get\_Fullscale\_Load(**ByVal** DeviceHandle **As String**) **As String**

<b>Parameters</b>	<b>Data Type</b>	<b>Description</b>
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

<b>Return Values</b>		
<b>Command/Variable</b>	<b>If Successful</b>	<b>If Unsuccessful</b>
Get_Fullscale_Load	Number	"Error"

### **Comments**

Get\_Fullscale\_Load is used to get the fullscale load stored in the EEPROM of the USB Device. The returned value should be a numerical value between zero and +65,535.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Number\_of>Loading\_Points*

### Summary

Get\_Number\_of>Loading\_Points is used to get the number of loading point values stored in the EEPROM of the USB Device.

### Definition

Public Function Get\_Number\_of>Loading\_Points(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Number_of>Loading_Points	Number	"Error"

### Comments

Get\_Number\_of>Loading\_Points is used to get the number of loading point values stored in the EEPROM of the USB Device. The returned value should be a numerical value between zero and +255.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Sensitivity*

### Summary

Get\_Sensitivity is used to get the nominal sensitivity value stored in the EEPROM of the USB Device.

### Definition

Public Function Get\_Sensitivity(ToDeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Sensitivity	Number	"Error"

### Comments

Get\_Sensitivity is used to get the nominal sensitivity value stored in the EEPROM of the USB Device. The sensitivity is a nominal value that represents the sensor's rated output expressed in mV/V. The sensitivity value is for reference only and does not affect the calibration of the device

Sensitivity				
Hexadecimal	Decimal	Octal	Binary	Nominal Sensitivity
00	0	0000	00000000	0.5 mV/V
01	1	0001	00000001	1.0 mV/V
02	2	0002	00000010	2.0 mV/V
03	3	0003	00000011	4.0 mV/V
04	4	0004	00000100	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Bridge\_Resistance*

### Summary

Get\_Bridge\_Resistance is used to get the nominal bridge resistance value stored in the EEPROM of the USB Device.

### Definition

Public Function Get\_Bridge\_Resistance(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Bridge_Resistance	Number	"Error"

### Comments

Get\_Bridge\_Resistance is used to get the nominal bridge resistance value stored in the EEPROM of the USB Device. The bridge resistance is a nominal value that represents the sensor's bridge resistance expressed in ohms ( $\Omega$ ). The bridge resistance value is for reference only and does not affect the calibration of the device

BridgeResistance				
Hexadecimal	Decimal	Octal	Binary	Nominal Resistance
00	0	0000	00000000	120 $\Omega$
01	1	0001	00000001	350 $\Omega$
02	2	0002	00000010	700 $\Omega$
03	3	0003	00000011	1000 $\Omega$
04	4	0004	00000100	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_ADC\_PGA\_Setting*

### Summary

Get\_ADC\_PGA\_Setting is used to get the ADC PGA (Programmable Gain Amplifier) setting for the USB Device.

### Definition

Public Function Get\_ADC\_PGA\_Setting(ToDeviceHandle As String) As String

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_ADC_PGA_Setting	Number	"Error"

### Comments

Get\_ADC\_PGA\_Setting is used to get the ADC PGA (Programmable Gain Amplifier) setting for the USB Device. The returned value should be a numerical value between zero and +15.

For FUTEK Model Numbers USB100 and USB200 only.

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
0	0	00	0000	Gain = 1
1	1	01	0001	Reserved, Do Not Use
2	2	02	0010	Reserved, Do Not Use
3	3	03	0011	Reserved, Do Not Use
4	4	04	0100	Reserved, Do Not Use
5	5	05	0101	Reserved, Do Not Use
6	6	06	0110	Reserved, Do Not Use
7	7	07	0111	Reserved, Do Not Use
8	8	10	1000	Reserved, Do Not Use
9	9	11	1001	Reserved, Do Not Use
A	10	12	1010	Reserved, Do Not Use
B	11	13	1011	Reserved, Do Not Use
C	12	14	1100	Reserved, Do Not Use
D	13	15	1101	Reserved, Do Not Use

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
E	14	16	1110	Reserved, Do Not Use
F	15	17	1111	Reserved, Do Not Use

For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320, IHH500 and IPM650 only.

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
0	0	00	0000	Gain = 1
1	1	01	0001	Gain = 2
2	2	02	0010	Gain = 4
3	3	03	0011	Gain = 8
4	4	04	0100	Gain = 16
5	5	05	0101	Gain = 32
6	6	06	0110	Gain = 64
7	7	07	0111	Gain = 128
8	8	10	1000	Reserved, Do Not Use
9	9	11	1001	Reserved, Do Not Use
A	10	12	1010	Reserved, Do Not Use
B	11	13	1011	Reserved, Do Not Use
C	12	14	1100	Reserved, Do Not Use
D	13	15	1101	Reserved, Do Not Use
E	14	16	1110	Reserved, Do Not Use
F	15	17	1111	Reserved, Do Not Use

For FUTEK Model Numbers USB230 only.

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
0	0	00	0000	Gain = 1
1	1	01	0001	Gain = 2
2	2	02	0010	Gain = 4
3	3	03	0011	Gain = 8
4	4	04	0100	Gain = 16
5	5	05	0101	Gain = 32
6	6	06	0110	Gain = 64
7	7	07	0111	Gain = 64
8	8	10	1000	Reserved, Do Not Use
9	9	11	1001	Reserved, Do Not Use
A	10	12	1010	Reserved, Do Not Use

PGA Setting				
Hexadecimal	Decimal	Octal	Binary	Gain
B	11	13	1011	Reserved, Do Not Use
C	12	14	1100	Reserved, Do Not Use
D	13	15	1101	Reserved, Do Not Use
E	14	16	1110	Reserved, Do Not Use
F	15	17	1111	Reserved, Do Not Use

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_ADC\_Sampling\_Rate\_Setting*

### Summary

Get\_ADC\_Sampling\_Rate\_Setting is used to get the ADC sampling rate setting for the USB Device.

### Definition

Public Function Get\_ADC\_Sampling\_Rate\_Setting(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_ADC_Sampling_Rate_Setting	Number	"Error"

### Comments

Get\_ADC\_Sampling\_Rate\_Setting is used to get the ADC sampling rate setting for the USB Device. The returned value should be a numerical value between zero and +255.

For FUTEK Model Numbers USB100 and USB200 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	External Input, 50/60 Hz Rejection, 5 Samples Per Second
01	1	0001	00000001	External Input, 50/60 Hz Rejection, 10 Samples Per Second
02	2	0002	00000010	External Input, 50 Hz Rejection, 5 Samples Per Second
03	3	0003	00000011	External Input, 50 Hz Rejection, 10 Samples Per Second
04	4	0004	00000100	External Input, 60 Hz Rejection, 5 Samples Per Second
05	5	0005	00000101	External Input, 60 Hz Rejection, 10 Samples Per Second
06	6	0006	00000110	Reserved, Do Not Use
07	7	0007	00000111	Reserved, Do Not Use
08	8	0010	00001000	Temperature Input, 50/60 Hz Rejection, 5 Samples Per Second
09	9	0011	00001001	Reserved, Do Not Use
0A	10	0012	00001010	Temperature Input, 50 Hz Rejection, 5 Samples Per Second
0B	11	0013	00001011	Reserved, Do Not Use
0C	12	0014	00001100	Temperature Input, 60 Hz Rejection, 5 Samples Per Second
0D	13	0015	00001101	Reserved, Do Not Use
0E	14	0016	00001110	Reserved, Do Not Use

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
0F	15	0017	00001111	Reserved, Do Not Use
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers USB110 and USB210 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	2.5 Samples Per Second
01	1	0001	00000001	10 Samples Per Second
02	2	0002	00000010	20 Samples Per Second
03	3	0003	00000011	25 Samples Per Second
04	4	0004	00000100	30 Samples Per Second
05	5	0005	00000101	50 Samples Per Second
06	6	0006	00000110	60 Samples Per Second
07	7	0007	00000111	80 Samples Per Second
08	8	0010	00001000	100 Samples Per Second
09	9	0011	00001001	150 Samples Per Second
0A	10	0012	00001010	200 Samples Per Second
0B	11	0013	00001011	250 Samples Per Second
0C	12	0014	00001100	300 Samples Per Second
0D	13	0015	00001101	500 Samples Per Second
0E	14	0016	00001110	750 Samples Per Second
0F	15	0017	00001111	1000 Samples Per Second
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Numbers USB120, USB220, USB320, IHH500 and IPM650 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	5 Samples Per Second
01	1	0001	00000001	10 Samples Per Second
02	2	0002	00000010	15 Samples Per Second
03	3	0003	00000011	20 Samples Per Second
04	4	0004	00000100	25 Samples Per Second
05	5	0005	00000101	30 Samples Per Second

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
06	6	0006	00000110	50 Samples Per Second
07	7	0007	00000111	60 Samples Per Second
08	8	0010	00001000	100 Samples Per Second
09	9	0011	00001001	300 Samples Per Second
0A	10	0012	00001010	600 Samples Per Second
0B	11	0013	00001011	960 Samples Per Second
0C	12	0014	00001100	1200 Samples Per Second
0D	13	0015	00001101	1600 Samples Per Second
0E	14	0016	00001110	2400 Samples Per Second
0F	15	0017	00001111	4800 Samples Per Second
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

For FUTEK Model Number USB230 only.

SamplingRate				
Hexadecimal	Decimal	Octal	Binary	ADC Configuration
00	0	0000	00000000	2.5 Samples Per Second
01	1	0001	00000001	5 Samples Per Second
02	2	0002	00000010	10 Samples Per Second
03	3	0003	00000011	15 Samples Per Second
04	4	0004	00000100	25 Samples Per Second
05	5	0005	00000101	30 Samples Per Second
06	6	0006	00000110	50 Samples Per Second
07	7	0007	00000111	60 Samples Per Second
08	8	0010	00001000	100 Samples Per Second
09	9	0011	00001001	500 Samples Per Second
0A	10	0012	00001010	1000 Samples Per Second
0B	11	0013	00001011	2000 Samples Per Second
0C	12	0014	00001100	3750 Samples Per Second
0D	13	0015	00001101	7500 Samples Per Second
0E	14	0016	00001110	15000 Samples Per Second
0F	15	0017	00001111	15000 Samples Per Second
10	16	0020	00010000	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Sensor\_Identification\_Number*

### **Summary**

Get\_Sensor\_Identification\_Number is used to get the serial number stored in the EEPROM of the USB Device.

### **Definition**

Public Function Get\_Sensor\_Identification\_Number(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Sensor_Identification_Number	Number	"Error"

### **Comments**

Get\_Sensor\_Identification\_Number is used to get the serial number stored in the EEPROM of the USB Device. The returned value should be a numerical value between zero and +16,777,215.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Get\_Unit\_Code***

### **Summary**

Get\_Unit\_Code is used to get the value for Engineering Units stored in the EEPROM of the USB Device.

### **Definition**

Public Function Get\_Unit\_Code(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Unit_Code	Number	"Error"

### **Comments**

Get\_Unit\_Code is used to get the value for Engineering Units stored in the EEPROM of the USB Device. The returned value should be a numerical value between zero and +255.

Please refer to the [Unit Codes](#) table for more information.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Direction*

### Summary

Get\_Direction is used to get the direction value stored in the EEPROM of the USB Device.

### Definition

Public Function Get\_Direction(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Direction	Number	"Error"

### Comments

Get\_Direction is used to get the direction value stored in the EEPROM of the USB Device. The direction is a numerical value indicating the direction type and the number of directions used during calibration. The returned value should be a numerical value between zero and +255.

Direction					
Hexadecimal	Decimal	Octal	Binary	Direction 1	Direction 2
00	0	0000	00000000	Compression	Undefined
01	1	0001	00000001	Tension	Undefined
02	2	0002	00000010	CW	Undefined
03	3	0003	00000011	CCW	Undefined
04	4	0004	00000100	Pressure	Undefined
05	5	0005	00000101	Vacuum	...
06	6	0006	00000110	Undefined	Undefined
...	...	...	...	...	...
0F	15	0017	00001111	Undefined	Undefined
10	16	0020	00010000	Compression	Tension
11	17	0021	00010001	Tension	Compression
12	18	0022	00010010	CW	CCW
13	19	0023	00010011	CCW	CW
14	20	0024	00010100	Pressure	Vacuum
15	21	0025	00010101	Vacuum	Pressure

Direction					
Hexadecimal	Decimal	Octal	Binary	Direction 1	Direction 2
16	22	0026	00010110	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Shunt\_Value*

### **Summary**

Get\_Shunt\_Value is used to get the shunt value stored in the EEPROM of the USB Device.

### **Definition**

**Public Function** Get\_Shunt\_Value(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Shunt_Value	Number	"Error"

### **Comments**

Get\_Shunt\_Value is used to get the shunt value stored in the EEPROM of the USB Device. The shunt value is a numerical value representing the simulated load that is applied to the USB Device. The returned value should be a numerical value between zero and +16,777,215.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Decimal\_Point*

### Summary

Get\_Decimal\_Point is used to get the decimal point format stored in the EEPROM of the USB Device.

### Definition

Public Function Get\_Decimal\_Point(ToDeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Decimal_Point	Number	"Error"

### Comments

Get\_Decimal\_Point is used to get the decimal point format stored in the EEPROM of the USB Device. The decimal point should be used to format a numerical value using the following expression:

---

### For example:

If the Offset Value = 1000 and the Decimal Point = 2, then:

---



---



---

DecimalPoint				
Hexadecimal	Decimal	Octal	Binary	Decimal Point Format
00	0	0000	00000000	0.
01	1	0001	00000001	0.0
02	2	0002	00000010	0.00
03	3	0003	00000011	0.000
04	4	0004	00000100	0.0000
05	5	0005	00000101	0.00000
06	6	0006	00000110	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Calibration\_Code*

### **Summary**

Get\_Calibration\_Code is used to get the calibration code stored in the USB Device.

### **Definition**

**Public Function** Get\_Calibration\_Code(**ByVal** DeviceHandle **As String**) **As String**

<b>Parameters</b>	<b>Data Type</b>	<b>Description</b>
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

<b>Return Values</b>		
<b>Command/Variable</b>	<b>If Successful</b>	<b>If Unsuccessful</b>
Get_Calibration_Code	Number	"Error"

### **Comments**

Get\_Calibration\_Code is used to get the calibration code stored in the USB Device. The calibration code is a numerical value representing the operator who performed the calibration of the USB Device. The calibration code must be a numerical value between zero and +255.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Calibration\_Day*

### **Summary**

Get\_Calibration\_Day is used to get the calibration day stored in the EEPROM of the USB Device.

### **Definition**

**Public Function** Get\_Calibration\_Day(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Calibration_Day	Number	"Error"

### **Comments**

Get\_Calibration\_Day is used to get the calibration day stored in the EEPROM of the USB Device. The calibration day is a numerical value representing the day of the month when the calibration of the USB Device was performed. The returned value should be a numerical value between zero and +255.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Get\_Calibration\_Month***

### **Summary**

Get\_Calibration\_Month is used to get the calibration month stored in the EEPROM of the USB Device.

### **Definition**

**Public Function Get\_Calibration\_Month(**ByVal DeviceHandle **As String)** As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Calibration_Month	Number	"Error"

### **Comments**

Get\_Calibration\_Month is used to get the calibration month stored in the EEPROM of the USB Device. The calibration month is a numerical value representing the month when the calibration of the USB Device was performed. The returned value should be a numerical value between zero and +255.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Calibration\_Year*

### **Summary**

Get\_Calibration\_Year is used to get the calibration year stored in the EEPROM of the USB Device.

### **Definition**

**Public Function** Get\_Calibration\_Year(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Calibration_Year	Number	"Error"

### **Comments**

Get\_Calibration\_Year is used to get the calibration year stored in the EEPROM of the USB Device. The calibration year is a numerical value representing the year when the calibration of the USB Device was performed. The Calibration year is zero indexed from the year 2000. The returned value should be a numerical value between zero and +255.

### **For Example:**

If the calibration year is 11, then the year when the calibration was performed is 2011.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Get\_Pulses\_Per\_Rotation***

### **Summary**

Get\_Pulses\_Per\_Rotation is used to get the number of pulses per rotation stored in the EEPROM of the USB Device.

### **Definition**

Public Function Get\_Pulses\_Per\_Rotation(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Pulses_Per_Rotation	Number	"Error"

### **Comments**

Get\_Pulses\_Per\_Rotation is used to get the number of pulses per rotation stored in the EEPROM of the USB Device. The pulses per rotation is a numerical value representing the number of pulses per rotation associated with the encoder of a torque sensor. The returned value should be a numerical value between zero and +65,535.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Average\_Setting*

### Summary

Get\_Average\_Setting is used to get the averaging method and averaging setting stored in the EEPROM of the USB Device.

### Definition

Public Function Get\_Average\_Setting(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
DeviceHandle	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Set_Average_Setting	Number	"Error"

### Comments

Get\_Average\_Setting is used to get the averaging method and averaging setting stored in the EEPROM of the USB Device. The averaging method can be set to disable averaging or to enable a moving or mean average. The number of sample that are used in the average can be selected from 2 to 256 samples. The returned value should be a numerical value between zero and +255.

For FUTEK Model Numbers USB110 and USB210 only.

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
00	0	0000	00000000	Disable Averaging	Average = 2
01	1	0001	00000001	Disable Averaging	Average = 4
02	2	0002	00000010	Disable Averaging	Average = 8
03	3	0003	00000011	Disable Averaging	Average = 16
04	4	0004	00000100	Disable Averaging	Average = 32
05	5	0005	00000101	Disable Averaging	Average = 64
06	6	0006	00000110	Disable Averaging	Average = 128
07	7	0007	00000111	Disable Averaging	Average = 256
08	8	0010	00001000	Undefined	Undefined
...	...	...	...	...	...
0F	15	0017	00001111	Undefined	Undefined
10	16	0020	00010000	Moving Average	Average = 2
11	17	0021	00010001	Moving Average	Average = 4

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
12	18	0022	00010010	Moving Average	Average = 8
13	19	0023	00010011	Moving Average	Average = 16
14	20	0024	00010100	Moving Average	Average = 32
15	21	0025	00010101	Moving Average	Average = 64
16	22	0026	00010110	Moving Average	Average = 128
17	23	0027	00010111	Moving Average	Average = 256
18	24	0030	00011000	Undefined	Undefined
...	...	...	...	...	...
1F	31	0037	00011111	Undefined	Undefined
20	32	0040	00100000	Mean Average	Average = 2
21	33	0041	00100001	Mean Average	Average = 4
22	34	0042	00100010	Mean Average	Average = 8
23	35	0043	00100011	Mean Average	Average = 16
24	36	0044	00100100	Mean Average	Average = 32
25	37	0045	00100101	Mean Average	Average = 64
26	38	0046	00100110	Mean Average	Average = 128
27	39	0047	00100111	Mean Average	Average = 256
28	40	0050	00101000	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

For FUTEK Model Numbers USB120, USB220, USB320, USB230 and IHH500 only.

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
00	0	0000	00000000	Disable Averaging	Average = 2
01	1	0001	00000001	Disable Averaging	Average = 4
02	2	0002	00000010	Disable Averaging	Average = 8
03	3	0003	00000011	Disable Averaging	Average = 16
04	4	0004	00000100	Disable Averaging	Average = 32
05	5	0005	00000101	Disable Averaging	Average = 64
06	6	0006	00000110	Disable Averaging	Average = 128
07	7	0007	00000111	Disable Averaging	Average = 256
08	8	0010	00001000	Undefined	Undefined
...	...	...	...	...	...
0F	15	0017	00001111	Undefined	Undefined
10	16	0020	00010000	Moving Average	Average = 2
11	17	0021	00010001	Moving Average	Average = 4
12	18	0022	00010010	Moving Average	Average = 8
13	19	0023	00010011	Moving Average	Average = 16

AverageSetting					
Hexadecimal	Decimal	Octal	Binary	Averaging Method	Number of Samples
14	20	0024	00010100	Moving Average	Average = 32
15	21	0025	00010101	Moving Average	Average = 64
16	22	0026	00010110	Moving Average	Average = 128
17	23	0027	00010111	Moving Average	Average = 256
18	24	0030	00011000	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Type\_of\_Board*

### Summary

Get\_Type\_of\_Board is used to get the type of board (FUTEK model number) that represents the USB Device.

### Definition

Public Function Get\_Type\_of\_Board(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Type_of_Board	Number	"Error"

### Comments

Get\_Type\_of\_Board is used to get the type of board (FUTEK model number) that represents the USB Device. The returned value should be a numerical value between zero and +255.

Type of Board				
Hexadecimal	Decimal	Octal	Binary	Model Number
00	0	0000	00000000	USB100/USB200
01	1	0001	00000001	USB110
02	2	0002	00000010	USB210
03	3	0003	00000011	USB220
04	4	0004	00000100	USB230
05	5	0005	00000101	IHH500
06	6	0006	00000110	USB120
07	7	0007	00000111	USB320
08	8	0010	00001000	USB410
09	9	0011	00001001	USB240
0A	10	0012	00001010	IPM650
0B	11	0013	00001011	Undefined
...	...	...	...	...
FF	255	0377	11111111	Undefined

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Get\_Hardware\_Version***

### **Summary**

Get\_Hardware\_Version is used to get the hardware version of the USB Device.

### **Definition**

**Public Function** Get\_Hardware\_Version(**ByVal** DeviceHandle **As String**) **As String**

<b>Parameters</b>	<b>Data Type</b>	<b>Description</b>
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Hardware_Version	Number	"Error"

### **Comments**

Get\_Hardware\_Version is used to get the hardware version of the USB Device. The hardware version specifies the revision of the hardware. The returned value should be a numerical value between zero and +255.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Firmware\_Version*

### **Summary**

Get\_Firmware\_Version is used to get the firmware version of the USB Device.

### **Definition**

**Public Function** Get\_Firmware\_Version(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Firmware_Version	Number	"Error"

### **Comments**

Get\_Firmware\_Version is used to get the firmware version of the USB Device. The firmware version specifies the revision of the firmware. The returned value should be a numerical value between zero and +255.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Firmware\_Year*

### Summary

Get\_Firmware\_Year is used to get the year that the firmware was created for the USB Device.

### Definition

Public Function Get\_Firmware\_Year(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Firmware_Year	Number	"Error"

### Comments

Get\_Firmware\_Year is used to Get the Year that the Firmware was created for the USB Device. The returned value should be a numerical value between zero and +15.

FirmwareYear				
Hexadecimal	Decimal	Octal	Binary	Year
00	0	0000	00000000	2000
01	1	0001	00000001	2001
02	2	0002	00000010	2002
03	3	0003	00000011	2003
04	4	0004	00000100	2004
05	5	0005	00000101	2005
06	6	0006	00000110	2006
07	7	0007	00000111	2007
08	8	0010	00001000	2008
09	9	0011	00001001	2009
0A	10	0012	00001010	2010
0B	11	0013	00001011	2011
0C	12	0014	00001100	2012
0D	13	0015	00001101	2013
0E	14	0016	00001110	2014
0F	15	0017	00001111	2015

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Get\_Firmware\_Month*

### Summary

Get\_Firmware\_Month is used to get the month that the firmware was created for the USB Device.

### Definition

Public Function Get\_Firmware\_Month(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### Return Type

String

### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Get_Firmware_Month	Number	"Error"

### Comments

Get\_Firmware\_Month is used to get the month that the firmware was created for the USB Device. The returned value should be a numerical value between zero and +15.

FirmwareMonth				
Hexadecimal	Decimal	Octal	Binary	Month
00	0	0000	00000000	Invalid
01	1	0001	00000001	January
02	2	0002	00000010	February
03	3	0003	00000011	March
04	4	0004	00000100	April
05	5	0005	00000101	May
06	6	0006	00000110	June
07	7	0007	00000111	July
08	8	0010	00001000	August
09	9	0011	00001001	September
0A	10	0012	00001010	October
0B	11	0013	00001011	November
0C	12	0014	00001100	December
0D	13	0015	00001101	Invalid
0E	14	0016	00001110	Invalid
0F	15	0017	00001111	Invalid

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## Backup Commands

### *Create\_Back\_Up*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

#### Summary

Create\_Back\_Up is used to create a backup of the calibration information stored in the EEPROM of the USB Device.

#### Definition

Public Function Create\_Back\_Up(**ByVal** DeviceHandle **As String**, **ByVal** PageNumber **As Integer**) **As String**

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device
PageNumber	Integer	Page Number in memory

#### Return Type

String

#### Return Values

Return Values		
Command/Variable	If Successful	If Unsuccessful
Create_Back_Up	0	"Error"

#### Comments

Create\_Back\_Up is used to create a backup of the calibration information stored in the EEPROM of the USB Device. This command should be used with extreme caution since it will overwrite entire pages of the EEPROM. The data in the destination page will be overwritten and will not be retrievable once this command is executed.

**For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320, USB230 and USB410 only.  
Firmware Version 4 and higher only.**

PageNumber				
Hexadecimal	Decimal	Octal	Binary	Destination Page
00	0	0000	00000000	Page 0 is reserved as the Source Page
01	1	0001	00000001	Page 1
02	2	0002	00000010	Page 2
03	3	0003	00000011	Page 3
04	4	0004	00000100	Invalid

PageNumber				
Hexadecimal	Decimal	Octal	Binary	Destination Page
...	...	...	...	...
0F	15	0017	00001111	Invalid

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Restore\_Back\_Up***

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Restore\_Back\_Up is used to restore a backup of the calibration information stored in the EEPROM of the USB Device.

### **Definition**

**Public Function** Restore\_Back\_Up(**ByVal** DeviceHandle **As String**, **ByVal** PageNumber **As Integer**) **As String**

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device
PageNumber	Integer	Page Number in memory

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Restore_Back_Up	0	"Error"

### **Comments**

Restore\_Back\_Up is used to restore a backup of the calibration information stored in the EEPROM of the USB Device. This command should be used with extreme caution since it will overwrite entire pages of the EEPROM. The data in the destination page will be overwritten and will not be retrievable once this command is executed.

**For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320, USB230 and USB410 only.  
Firmware Version 4 and higher only.**

PageNumber				
Hexadecimal	Decimal	Octal	Binary	Source Page
00	0	0000	00000000	Page 0 is reserved as the Destination Page
01	1	0001	00000001	Page 1
02	2	0002	00000010	Page 2
03	3	0003	00000011	Page 3
04	4	0004	00000100	Invalid
...	...	...	...	...
0F	15	0017	00001111	Invalid

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## Control Commands

### *Change\_Battery\_Enter*

#### **Summary**

Change\_Battery\_Enter is used to control the Battery / Enter button on the IHH500 Keypad or the Channel / Enter button on the IPM650 Keypad.

#### **Definition**

Public Function Change\_Battery\_Enter(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<u>DeviceHandle</u>	String	Handle assigned to the USB Device

#### **Return Type**

String

#### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Battery_Enter	0	"Error"

#### **Comments**

Change\_Battery\_Enter is used to control the Battery / Enter button on the IHH500 Keypad or the Channel / Enter button on the IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

For FUTEK Model Numbers IHH500 and IPM650 only.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Change\_Tare\_Up*

### **Summary**

Change\_Tare\_Up is used to control the Tare / Up button on the IHH500 or IPM650 Keypad.

### **Definition**

**Public Function** Change\_Tare\_Up(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Tare_Up	0	"Error"

### **Comments**

Change\_Tare\_Up is used to control the Tare / Up button on the IHH500 or IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Change\_Display\_Back*

### **Summary**

Change\_Display\_Back is used to control the Display / Back button on the IHH500 or IPM650 Keypad.

### **Definition**

**Public Function** Change\_Display\_Back(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Display_Back	0	"Error"

### **Comments**

Change\_Display\_Back is used to control the Display / Back button on the IHH500 or IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Change\_Reset\_Left*

### **Summary**

Change\_Reset\_Left is used to control the Reset / Left button on the IHH500 or IPM650 Keypad.

### **Definition**

Public Function Change\_Reset\_Left(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Reset_Left	0	"Error"

### **Comments**

Change\_Reset\_Left is used to control the Reset / Left button on the IHH500 or IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Change\_Menu*

### **Summary**

Change\_Menu is used to control the Menu button on the IHH500 or IPM650 Keypad.

### **Definition**

Public Function Change\_Menu(ByVal DeviceHandle As String) As String

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Menu	0	"Error"

### **Comments**

Change\_Menu is used to control the Menu button on the IHH500 or IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Change\_Unit\_Right*

### **Summary**

Change\_Unit\_Right is used to control the Unit / Right button on the IHH500 or IPM650 Keypad.

### **Definition**

**Public Function** Change\_Unit\_Right(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Unit_Right	0	"Error"

### **Comments**

Change\_Unit\_Right is used to control the Unit / Right button on the IHH500 or IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Change\_Shunt\_Exit*

### **Summary**

Change\_Shunt\_Exit is used to control the Shunt / Exit button on the IHH500 or IPM650 Keypad.

### **Definition**

**Public Function** Change\_Shunt\_Exit(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Shunt_Exit	0	"Error"

### **Comments**

Change\_Shunt\_Exit is used to control the Shunt / Exit button on the IHH500 or IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Change\_Hold\_Down*

### **Summary**

Change\_Hold\_Down is used to control the Hold / Down button on the IHH500 or IPM650 Keypad.

### **Definition**

**Public Function** Change\_Hold\_Down(**ByVal** DeviceHandle **As String**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Change_Hold_Down	0	"Error"

### **Comments**

Change\_Hold\_Down is used to control the Hold / Down button on the IHH500 or IPM650 Keypad. This command can be used to remotely control the instrument from a computer. The instrument will behave as if the actual button on the keypad were pressed.

**For FUTEK Model Numbers IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## Debugging Commands

### *Read\_Memory\_Register*

#### **Summary**

Read\_Memory\_Register is used to read a single byte of data from the microcontroller's Memory Register.

#### **Definition**

Public Function Read\_Memory\_Register(**ByVal** DeviceHandle **As String**, **ByVal** Address **As Byte**) **As String**

Parameters	Data Type	Description
<a href="#"><u>DeviceHandle</u></a>	String	Handle assigned to the USB Device
Address	Byte	Memory Address

#### **Return Type**

String

#### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Read_Memory_Register	Number	"Error"

#### **Comments**

Read\_Memory\_Register is used to read a single byte of data from the microcontroller's Memory Register. The Address is used to specify the address of the microcontroller's Memory Register. The returned value should be a numerical value between zero and +255.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Write\_Memory\_Register***

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Write\_Memory\_Register is used to write a single byte of data to the microcontroller's Memory Register.

### **Definition**

**Public Function** Write\_Memory\_Register(**ByVal** DeviceHandle **As String**, **ByVal** Address **As Byte**, **ByVal** Data **As Byte**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
Address	Byte	Memory Address
Data	Byte	Single Byte of Data

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Write_Memory_Register	0	"Error"

### **Comments**

Write\_Memory\_Register is used to write a single byte of data to the microcontroller's Memory Register. The Address is used to specify the address of the microcontroller's Memory Register. The Data is the value to be written.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Read\_EEPROM\_Register***

### **Summary**

Read\_Memory\_Register is used to read two bytes of data from the microcontroller's EEPROM Register.

### **Definition**

**Public Function** Read\_EEPROM\_Register(**ByVal** DeviceHandle **As String**, **ByVal** Address **As Byte**) **As String**

#### **Parameters**

	<b>Data Type</b>	<b>Description</b>
<a href="#"><u>DeviceHandle</u></a>	String	Handle assigned to the USB Device
Address	Byte	Memory Address

#### **Variables**

	<b>Data Type</b>	<b>Description</b>
DataHighByte	Byte	High Byte of Data
DataLowByte	Byte	Low Byte of Data

#### **Return Type**

String

#### **Return Values**

<b>Return Values</b>		
<b>Command/Variable</b>	<b>If Successful</b>	<b>If Unsuccessful</b>
Read_Memory_Register	Number	"Error"

#### **Comments**

Read\_Memory\_Register is used to read two bytes of data from the microcontroller's EEPROM Register. The Address is used to specify the starting address of the microcontroller's EEPROM Register. The data can be retrieved from the DataHighByte and DataLowByte once the command is finished.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## ***Write\_EEPROM\_Register***

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Write\_Memory\_Register is used to write two bytes of data to the microcontroller's EEPROM Register.

### **Definition**

**Public Function** Write\_EEPROM\_Register(**ByVal** DeviceHandle **As String**, **ByVal** Address **As Byte**, **ByVal** HighByte **As Byte**, **ByVal** LowByte **As Byte**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
Address	Byte	Memory Address
HighByte	Byte	High Byte of Data
LowByte	Byte	Low Byte of Data

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Write_Memory_Register	0	"Error"

### **Comments**

Write\_Memory\_Register is used to write two bytes of data to the microcontroller's EEPROM Register. The Address is used to specify the starting address of the microcontroller's EEPROM Register.

Please refer to the [EEPROM Register](#) table for more information.

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Read\_TEDS\_Register*

### **Summary**

Read\_TEDS\_Register is used to read a single byte of data from the TEDS Register.

### **Definition**

**Public Function** Read\_TEDS\_Register(**ByVal** DeviceHandle **As String**, **ByVal** HighAddress **As Byte**,  
**ByVal** LowAddress **As Byte**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
HighAddress	Byte	Memory Address
LowAddress	Byte	Memory Address

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Read_TEDS_Register	Number	"Error"

### **Comments**

Read\_TEDS\_Register is used to read a single byte of data from the TEDS Register. The HighAddress and LowAddress are used to specify the addresses of the TEDS Register. The returned value should be a numerical value between zero and +255.

**For FUTEK Model Number IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Write\_TEDS\_Register*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Write\_TEDS\_Register is used to write a single byte of data to the TEDS Register.

### **Definition**

**Public Function** Write\_TEDS\_Register(**ByVal** DeviceHandle **As String**, **ByVal** HighAddress **As Byte**,  
**ByVal** LowAddress **As Byte**, **ByVal** Data **As Byte**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
HighAddress	Byte	Memory Address
LowAddress	Byte	Memory Address
Data	Byte	Single Byte of Data

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Write_TEDS_Register	0	"Error"

### **Comments**

Write\_TEDS\_Register is used to write a single byte of data to the TEDS Register. The HighAddress and LowAddress are used to specify the addresses of the TEDS Register. The Data is the value to be written.

### **For FUTEK Model Number IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Read\_Channel\_Register*

### **Summary**

Read\_Channel\_Register is used to read a single byte of data from the Channel Register.

### **Definition**

**Public Function** Read\_Channel\_Register(**ByVal** DeviceHandle **As String**, **ByVal** ChannelNumber **As Byte**,  
**ByVal** Address **As Byte**) **As String**

Parameters	Data Type	Description
<a href="#">DeviceHandle</a>	String	Handle assigned to the USB Device
ChannelNumber	Byte	Channel Number of the USB Device
Address	Byte	Memory Address

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Read_Channel_Register	Number	"Error"

### **Comments**

Read\_Channel\_Register is used to read a single byte of data from the Channel Register. The Address is used to specify the address of the Channel Register. The returned value should be a numerical value between zero and +255.

**For FUTEK Model Number IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## *Write\_Channel\_Register*

**Warning:**  The use of this command can affect the factory calibration of the USB Device.

### **Summary**

Write\_Channel\_Register is used to write a single byte of data to the Channel Register.

### **Definition**

Public Function Write\_Channel\_Register(ByVal DeviceHandle As String, ByVal ChannelNumber As Byte, ByVal Address As Byte, ByVal Data As Byte) As String

Parameters	Data Type	Description
DeviceHandle	String	Handle assigned to the USB Device
ChannelNumber	Byte	Channel Number of the USB Device
Address	Byte	Memory Address
Data	Byte	Single Byte of Data

### **Return Type**

String

### **Return Values**

Return Values		
Command/Variable	If Successful	If Unsuccessful
Write_Channel_Register	0	"Error"

### **Comments**

Write\_Channel\_Register is used to write a single byte of data to the Channel Register. The Address is used to specify the address of the Channel Register. The Data is the value to be written.

### **For FUTEK Model Number IHH500 and IPM650 only.**

Please refer to the [Common Errors](#) or the [Command Specific Errors](#) tables for more information.

## Register Addresses

### *EEPROM Register*

For FUTEK Model Numbers USB100 and USB200 only.

EEPROM Register					
Starting Address Hexadecimal	Ending Address Hexadecimal	Number of Bytes	Register Contents	One Direction	Two Directions
00	03	4	Loading Point 0	Offset 1	Offset 1
04	07	4	Loading Point 1	Direction 1	Direction 1
08	0B	4	Loading Point 2	Direction 1	Direction 1
0C	0F	4	Loading Point 3	Direction 1	Direction 1
10	13	4	Loading Point 4	Direction 1	Direction 1
14	17	4	Loading Point 5	Direction 1	Direction 1
18	1B	4	Loading Point 6	Direction 1	Direction 1
1C	1F	4	Loading Point 7	Direction 1	Direction 1
20	23	4	Loading Point 8	Direction 1	Direction 1
24	27	4	Loading Point 9	Direction 1	Offset 2
28	2B	4	Loading Point 10	Direction 1	Direction 2
2C	2F	4	Loading Point 11	Direction 1	Direction 2
30	33	4	Loading Point 12	Direction 1	Direction 2
34	37	4	Loading Point 13	Direction 1	Direction 2
38	3B	4	Loading Point 14	Direction 1	Direction 2
3C	3F	4	Loading Point 15	Direction 1	Direction 2
40	43	4	Loading Point 16	Direction 1	Direction 2
44	47	4	Loading Point 17	Direction 1	Direction 2
48	4B	4	Loading Point 18	Direction 1	Fullscale 2
4C	4F	4	Loading Point 19	Fullscale 1	Fullscale 1
50	51	2	Load of Loading Point 0	Offset 1	Offset 1
52	53	2	Load of Loading Point 1	Direction 1	Direction 1
54	55	2	Load of Loading Point 2	Direction 1	Direction 1
56	57	2	Load of Loading Point 3	Direction 1	Direction 1
58	59	2	Load of Loading Point 4	Direction 1	Direction 1
5A	5B	2	Load of Loading Point 5	Direction 1	Direction 1
5C	5D	2	Load of Loading Point 6	Direction 1	Direction 1
5E	5F	2	Load of Loading Point 7	Direction 1	Direction 1
60	61	2	Load of Loading Point 8	Direction 1	Direction 1
62	63	2	Load of Loading Point 9	Direction 1	Offset 2
64	65	2	Load of Loading Point 10	Direction 1	Direction 2

EEPROM Register					
Starting Address Hexadecimal	Ending Address Hexadecimal	Number of Bytes	Register Contents	One Direction	Two Directions
66	67	2	Load of Loading Point 11	Direction 1	Direction 2
68	69	2	Load of Loading Point 12	Direction 1	Direction 2
6A	6B	2	Load of Loading Point 13	Direction 1	Direction 2
6C	6D	2	Load of Loading Point 14	Direction 1	Direction 2
6E	6F	2	Load of Loading Point 15	Direction 1	Direction 2
70	71	2	Load of Loading Point 16	Direction 1	Direction 2
72	73	2	Load of Loading Point 17	Direction 1	Direction 2
74	75	2	Load of Loading Point 18	Direction 1	Fullscale 2
76	77	2	Load of Loading Point 19	Fullscale 1	Fullscale 1
78	78	1	Number of Loading Points		
79	79	1	Sensor Configuration		
7A	7A	1	ADC Configuration		
7B	7D	3	Sensor Identification Number		
7E	7E	1	Unit Code		
7F	7F	1	Direction		

For FUTEK Model Numbers USB110, USB210, USB120, USB220, USB320 and USB230 only.

EEPROM Register					
Starting Address Hexadecimal	Ending Address Hexadecimal	Number of Bytes	Register Contents	One Direction	Two Directions
00	03	4	Loading Point 0	Offset 1	Offset 1
04	07	4	Loading Point 1	Direction 1	Direction 1
08	0B	4	Loading Point 2	Direction 1	Direction 1
0C	0F	4	Loading Point 3	Direction 1	Direction 1
10	13	4	Loading Point 4	Direction 1	Direction 1
14	17	4	Loading Point 5	Direction 1	Direction 1
18	1B	4	Loading Point 6	Direction 1	Direction 1
1C	1F	4	Loading Point 7	Direction 1	Offset 2
20	23	4	Loading Point 8	Direction 1	Direction 2
24	27	4	Loading Point 9	Direction 1	Direction 2
28	2B	4	Loading Point 10	Direction 1	Direction 2
2C	2F	4	Loading Point 11	Direction 1	Direction 2
30	33	4	Loading Point 12	Direction 1	Direction 2
34	37	4	Loading Point 13	Direction 1	Direction 2
38	3B	4	Loading Point 14	Direction 1	Fullscale 2
3C	3F	4	Loading Point 15	Fullscale 1	Fullscale 1

EEPROM Register					
Starting Address Hexadecimal	Ending Address Hexadecimal	Number of Bytes	Register Contents	One Direction	Two Directions
40	41	2	Load of Loading Point 0	Offset 1	Offset 1
42	43	2	Load of Loading Point 1	Direction 1	Direction 1
44	45	2	Load of Loading Point 2	Direction 1	Direction 1
46	47	2	Load of Loading Point 3	Direction 1	Direction 1
48	49	2	Load of Loading Point 4	Direction 1	Direction 1
4A	4B	2	Load of Loading Point 5	Direction 1	Direction 1
4C	4D	2	Load of Loading Point 6	Direction 1	Direction 1
4E	4F	2	Load of Loading Point 7	Direction 1	Offset 2
50	51	2	Load of Loading Point 8	Direction 1	Direction 2
52	53	2	Load of Loading Point 9	Direction 1	Direction 2
54	55	2	Load of Loading Point 10	Direction 1	Direction 2
56	57	2	Load of Loading Point 11	Direction 1	Direction 2
58	59	2	Load of Loading Point 12	Direction 1	Direction 2
5A	5B	2	Load of Loading Point 13	Direction 1	Direction 2
5C	5D	2	Load of Loading Point 14	Direction 1	Fullscale 2
5E	5F	2	Load of Loading Point 15	Fullscale 1	Fullscale 1
60	60	1	Number of Loading Points		
61	61	1	Reserved		
62	62	1	Sensor Configuration		
63	63	1	Reserved		
64	64	1	ADC Configuration		
65	65	1	Reserved		
66	68	3	Sensor Identification Number		
69	69	1	Reserved		
6A	6A	1	Unit Code		
6B	6B	1	Reserved		
6C	6C	1	Direction		
6D	6D	1	Reserved		
6E	71	4	Shunt Value		
72	72	1	Decimal Point		
73	73	1	Reserved		
74	74	1	Calibration Code		
75	75	1	Reserved		
76	76	1	Calibration Day		
77	77	1	Reserved		
78	78	1	Calibration Month		
79	79	1	Reserved		

EEPROM Register					
Starting Address Hexadecimal	Ending Address Hexadecimal	Number of Bytes	Register Contents	One Direction	Two Directions
7A	7A	1	Calibration Year		
7B	7B	1	Reserved		
7C	7D	2	Pulses Per Rotation		
7E	7F	2	Undefined		

## Additional Information

### *Unit Codes*

UnitCodes					
Hexadecimal	Decimal	Octal	Binary	Abbreviation	Engineering Units
00	0	0000	00000000	atm	atmosphere
01	1	0001	00000001	bar	bar
02	2	0002	00000010	dyn	dyne
03	3	0003	00000011	ft-H <sub>2</sub> O	foot of water (H <sub>2</sub> O)
04	4	0004	00000100	ft-lb	foot pound
05	5	0005	00000101	g	gram
06	6	0006	00000110	g-cm	gram centimeter
07	7	0007	00000111	g-mm	gram millimeter
08	8	0010	00001000	in-H <sub>2</sub> O	inches of water (H <sub>2</sub> O)
09	9	0011	00001001	in-lb	inch pound
0A	10	0012	00001010	in-oz	inch ounce
0B	11	0013	00001011	kdyn	kilodyne
0C	12	0014	00001100	kg	kilogram
0D	13	0015	00001101	kg-cm	kilogram centimeter
0E	14	0016	00001110	kg/cm <sup>2</sup>	kilogram per centimeter squared
0F	15	0017	00001111	kg-m	kilogram meter
10	16	0020	00010000	klb	kilopound
11	17	0021	00010001	kN	kiloNewton
12	18	0022	00010010	kPa	kiloPascal
13	19	0023	00010011	kpsi	thousand pounds per square inch
14	20	0024	00010100	lb	pound
15	21	0025	00010101	Mdyn	megadyne
16	22	0026	00010110	mmHG	millimeter of mercury (torr)
17	23	0027	00010111	mN-m	milliNewton meter
18	24	0030	00011000	MPa	megaPascal
19	25	0031	00011001	MT	metric ton
1A	26	0032	00011010	N	Newton
1B	27	0033	00011011	N-cm	Newton centimeter
1C	28	0034	00011100	N-m	Newton meter
1D	29	0035	00011101	N-mm	Newton millimeter
1E	30	0036	00011110	oz	Ounces
1F	31	0037	00011111	psi	pound per square inch
20	32	0040	00100000	Pa	Pascal

UnitCodes					
Hexadecimal	Decimal	Octal	Binary	Abbreviation	Engineering Units
21	33	0041	00100001	ST	short ton (US)
22	34	0042	00100010	mV/V	millivolts per volt
23	35	0043	00100011	µA	microampere
24	36	0044	00100100	mA	milliampere
25	37	0045	00100101	A	ampere
26	38	0046	00100110	mm	millimeter
27	39	0047	00100111	cm	centimeter
28	40	0050	00101000	dm	decimeter
29	41	0051	00101001	m	meter
2A	42	0052	00101010	km	kilometer
2B	43	0053	00101011	in	inch
2C	44	0054	00101100	ft	foot
2D	45	0055	00101101	yd	yard
2E	46	0056	00101110	mi	mile
2F	47	0057	00101111	µg	microgram
30	48	0060	00110000	mg	milligram
31	49	0061	00110001	LT	long ton (UK)
32	50	0062	00110010	mbar	millibar
33	51	0063	00110011	°C	degree Celsius
34	52	0064	00110100	°F	degree Fahrenheit
35	53	0065	00110101	°K	Kelvin
36	54	0066	00110110	°Ra	degree Rankine
37	55	0067	00110111	kN-m	kiloNewton meter
38	56	0070	00111000	g-m	gram meter
39	57	0071	00111001	nV	nanovolt
3A	58	0072	00111010	µV	microvolt
3B	59	0073	00111011	mV	millivolt
3C	60	0074	00111100	V	volt
3D	61	0075	00111101	kV	kilovolt
3E	62	0076	00111110	NONE	NONE
3F	63	0077	00111111	Undefined	Undefined
...	...	...	...	...	...
FF	255	0377	11111111	Undefined	Undefined

## Release Information

FUTEK USB DLL Version 0.9.0.0 Programmer's Guide – Initial Release March 2009

FUTEK USB DLL Version 1.1.0.0 Programmer's Guide – Initial Release July 2009

FUTEK USB DLL Version 2.0.0.0 Programmer's Guide – Initial Release January 2011

FUTEK USB DLL Version 2.1.2000.0 Programmer's Guide – Initial Release November 2011

## Copyright

Copyright © FUTEK Advanced Sensor Technology, Inc. 2009-2011.

## Disclaimer

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied.

This document provides preliminary information that may be subject to change without notice.

## Contact Information

**FUTEK Advanced Sensor Technology, Inc.**

10 Thomas, Irvine, CA 92618, USA

Toll Free: (800) 23-FUTEK

Telephone: (949) 465-0900

Fax: (949) 465-0905

Email (All Inquires): [futek@futek.com](mailto:futek@futek.com)

Website URL: [www.futek.com](http://www.futek.com)