

# **Lab-Volt 9063 Software Development Kit (SDK) Documentation**

Revision 7

## *Table of Contents*

### **Quick Start**

What can I do with this SDK.....	4
Contents of the SDK.....	4
Activation of the SDK.....	4
How to use the SDK .....	5
Minimum requirements to use the SDK.....	5
Prerequisites.....	5
Error codes should be handled properly .....	6
Controlling multiple 9063 .....	6

### **Diagrams (How to use the SDK)**

Operating the SDK .....	7
Controlling Inputs (E1, I1, ...) .....	8
Controlling Digital Outputs .....	9
Controlling Analog Outputs .....	10

### **Constants**

AO1, AO2, DO1, DO2, STOP, START .....	11
---------------------------------------	----

### **Enums**

ErrorCode.....	12
FunctionType .....	13
Input.....	14
InputRange.....	15
PPR.....	16
TrigSlope .....	17
TrigSource .....	18
Waveform .....	19

### **Functions**

AcquireData .....	20
CloseDevice.....	22
GetEncoderValue .....	23
GetFirmwareVersion.....	24
InitDevice .....	25
LV9063EntryPoint .....	26
SendAcqTable .....	27
SendAnalogOutputsTable .....	28
SendDigitalOutputsTable .....	29
SendRelayTable.....	30
SetAcqTable .....	31
SetAllInputsRange.....	32
SetAnalogOutputAmplitude .....	33
SetAnalogOutputFrequency .....	34
SetAnalogOutputFunction .....	35
SetAnalogOutputStartStop .....	36

SetAnalogOutputValue .....	37
SetAnalogOutputWaveform .....	38
SetDB9 .....	39
SetDigitalOutputs .....	40
SetDigitalOutputState .....	41
SetEncoderPPR .....	42
SetFunctionGenerator .....	43
SetInput .....	44
SetInputRangeE .....	45
SetInputRangeI .....	46
SetNbInputs .....	47
SetNbPackets .....	48
SetSamplingFreq .....	49
SetSync .....	50
SetTrigDelay .....	51
SetTrigHardware .....	52
SetTrigLevel .....	53
SetTrigSlope .....	54
SetTrigSource .....	55
SetTrigTable .....	56

## Quick Start

### What can I do with this SDK

The 9063 SDK offers the possibility to control various inputs and outputs of the Data Acquisition and Control Interface, Model 9063. Using the functions in the SDK you can acquire data using voltages and currents inputs. You can also acquire data using the Analog inputs and Encoder value. The SDK gives you access to control the Analog and Digital outputs of the Data Acquisition and Control Interface.

### Contents of the SDK

- DLL Files to communicate with the Data Acquisition and Control Interface
- Documentation about the functions included in the SDK
- LabVIEW examples
- MATLAB examples

### Activation of the SDK

A function set (license) is needed to allow the communication between third-party programming tools (NI LabVIEW, Mathworks MATLAB,...) and the Lab-Volt Data Acquisition and Control Interface (DACI), Model 9063.

To know which available function sets (licenses) are activated in the 9063, connect it to a computer via a usb cable and run LVDAC-EMS. In the LVDAC-EMS Start-Up window, check if the function 9069-90 : SDK appears in the list.

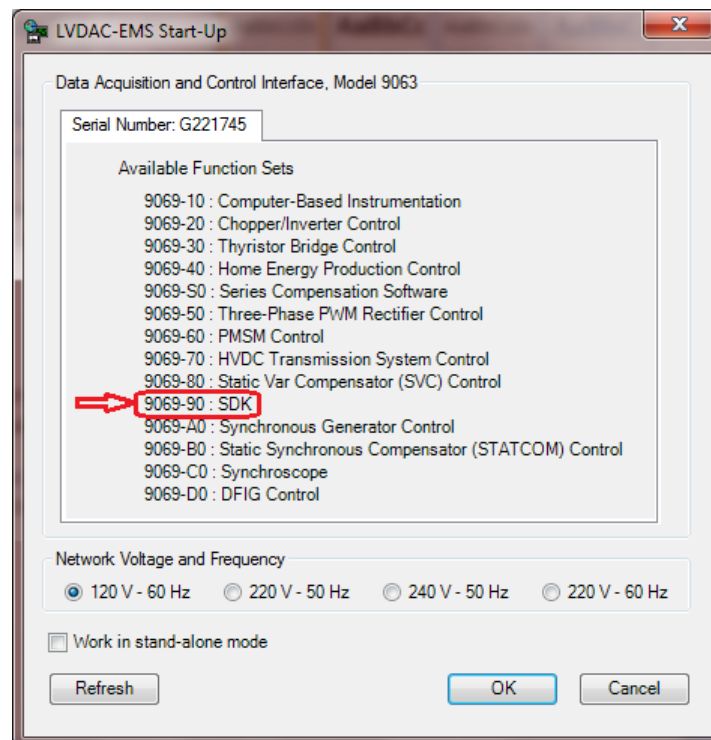


Figure 1: LVDAC-EMS Start-Up Windows

If the function SDK is unavailable on the 9063, a license file for 9063 SDK (Software Development Kit), Model 9069-90 must be ordered via a Lab-Volt dealer. It must then be activated in the DACI using the Function Set Activation Wizard available in LVDAC-EMS under Tools menu.

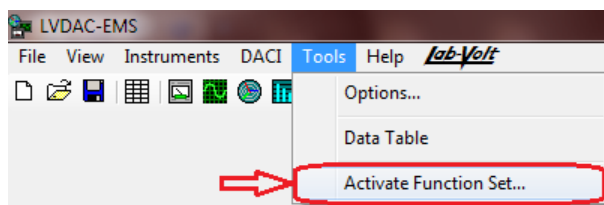


Figure 2: Function Set Activation Wizard

## Minimum requirements to use the SDK

- License for 9063 SDK (Software Development Kit), Model 9069-90
- LabVIEW 2011 or later
  - **32-bit version only is supported.**
- MATLAB 7.12 R2011a or later
  - **32-bit version only is supported.**
  - To learn how to install a 32-bit version on a 64-bit Windows machine, click on the link below:  
<http://www.mathworks.com/support/solutions/en/data/1-579TVF/index.html?solution=1-579TVF>
- Microsoft Visual Studio 2010

## Prerequisites

- Windows XP (32-bit) or Windows 7 (32 or 64-bit)
- Microsoft .Net Framework 4.0

## How to use the SDK

The SDK is built with Microsoft Visual Studio 2012 using Microsoft .Net Framework 4.0. The Library can be used with different development tools that support Microsoft .Net Framework 4.0 including MATLAB, LabVIEW and Microsoft Visual Studio. All you have to do is to load the library and start using functions.

**The files LV9063SDK.dll and WinUsbWrapper.dll should be in the same folder.**

*To MATLAB and LabVIEW users :*

Depending on the which version you are using, you might need to manually specify the right Common Language Runtime (CLR). Using Notepad, open the *matlab.exe.config* (or *labview.exe.config*) file located in the same folder as *matlab.exe* (or *labview.exe*). If the file doesn't exist, create it and copy the following lines in it :

```
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0"/>
  </startup>
</configuration>
```

If the file already exists, there are two things to look for in it:

1. Make sure the **supportedRuntime version** starts with “**v4.0**”.
2. Make sure the **useLegacyV2RuntimeActivationPolicy** is set to **true**.

In the event that one or both of these conditions are not met, swap the appropriate lines with the ones provided on the previous page. Please note that the « supportedRuntime version » parameter may look different, but as long as it starts with « **v4.0** », it is OK:

- `<supportedRuntime version="v4.0.2027.3100"/>` OK
- `<supportedRuntime version="v4.0.30319"/>` OK
- `<supportedRuntime version="v2.0"/>` NOT OK

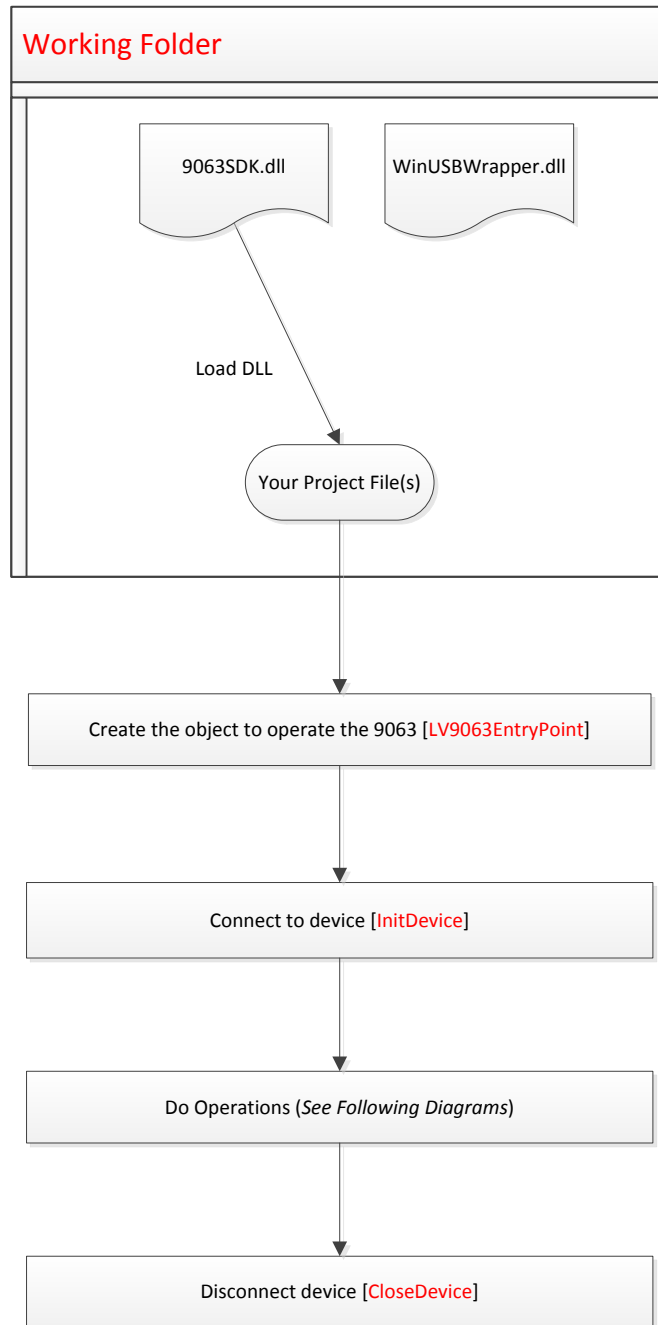
## Error codes should be handled properly

For ease of reading, the examples provided in the documentation disregard almost all error codes returned by the various functions. For a better idea on how to handle error codes properly, please refer to the examples (MATLAB, LabVIEW or C#) provided in the SDK folder. The default location for these files is “*C:\Program Files (x86)\Lab-Volt\LVDAC-EMS\SDK\9063 SDK*”.

## Controlling multiple 9063

To control multiple 9063, you must create a `LV9063EntryPoint` instance for each device you want to control and initialize them to the proper 9063 index. See example below:

```
LV9063EntryPoint device1 = new LV9063EntryPoint();
LV9063EntryPoint device2 = new LV9063EntryPoint();
device1.InitDevice(0);    // device1 is now associated with the 9063 at index 0.
device2.InitDevice(1);    // device2 is now associated with the 9063 at index 1.
```

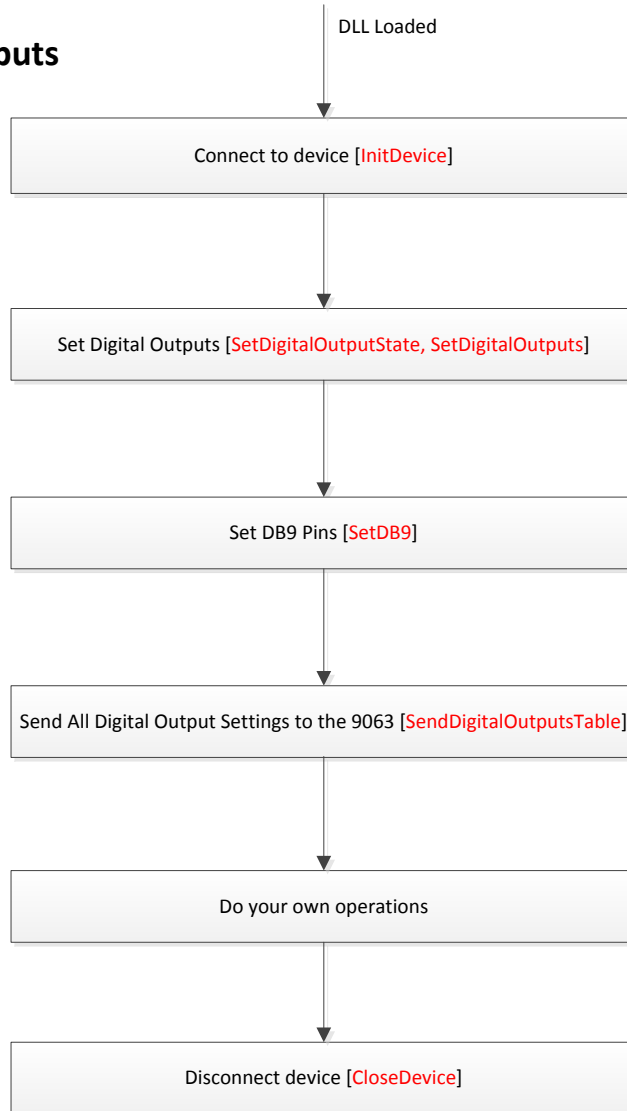
*Diagrams (Using the SDK)***Operating the 9063 SDK**

**Controlling Inputs (E1, I1, ...)**





## Controlling Digital Outputs



## Controlling Analog Outputs



## *Constants*

### **Description**

These are the constants that can be used with the 9063 SDK.

### **Values Description**

AO1	Analog Output #1.
AO2	Analog Output #2.
DO1	Digital Output #1.
DO2	Digital Output #2.
STOP	Stop an analog output.
START	Start an analog output.

## *ErrorCode*

### Description

This is an enumeration of different error codes that may occur when calling a function of the SDK. Most of the SDK functions will return an error code of type *ErrorCode*.

### Prototype

```
public enum ErrorCode
{
    None = 0,
    Communication = -1,
    OutOfBounds = -2,
    DeviceID = -3,
    InvalidParameter = -4,
    UnavailablePackage = -5
}
```

### Values Descriptions

None	There are no errors.
Communication	Communication error. Check the connexion between the computer and the 9063 module.
OutOfBounds	Index error. The index provided to the function is invalid.
DeviceID	This error occurs if the module index does not exist when calling the <i>InitDevice</i> function.
InvalidParameter	The parameter passed to the function is invalid.
UnavailablePackage	This error occurs when calling a function that is not available for the 9063 module currently in use.

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();
if (dll.InitDevice(0) != LV9063EntryPoint.ErrorCode.None)
{
    System.Windows.Forms.MessageBox.Show("Communication failed");
    return;
}
```

## *FunctionType*

### **Description**

This is an enumeration to adjust the type of function used by analog outputs.

### **Prototype**

```
public enum FunctionType
{
    CommandButton,
    FunctionGenerator
}
```

### **Values Descriptions**

CommandButton	The analog output is controlled by a command button.
FunctionGenerator	The analog output is used as a function generator.

### **Example**

```
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,
    LV9063EntryPoint.FunctionType.FunctionGenerator);
dll.SetAnalogOutputAmplitude(LV9063EntryPoint.AO1, 5f);
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,
    LV9063EntryPoint.Waveform.Sine);
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);
dll.SendAnalogOutputsTable();
```

## *Input*

### **Description**

This is an enumeration of all the inputs of the 9063.

### **Prototype**

```
public enum Input
{
    E1 = 0,
    I1 = 1,
    E2 = 2,
    I2 = 3,
    E3 = 4,
    I3 = 5,
    E4 = 6,
    I4 = 7,
    AI1 = 8,
    AI2 = 9,
    AI3 = 10,
    AI4 = 11,
    AI5 = 12,
    AI6 = 13,
    AI7 = 14,
    AI8 = 15
}
```

### **Values Descriptions**

E...	Voltage inputs.
I...	Current inputs.
AI...	Analog inputs.

### **Example**

```
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
```

## *InputRange*

### Description

This is an enumeration for the range of the selected input.

### Prototype

```
public enum InputRange
{
    Low,
    High
}
```

### Values Descriptions

Low	Low input ranges (-80V/+80V for voltage inputs, -4A/+4A for current inputs)
High	High input ranges (-800V/+800V for voltage inputs, -40A/+40A for current inputs)

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.I1); // Set input #2 as I1
dll.SetInputRangeE(LV9063EntryPoint.Input.E1, LV9063EntryPoint.InputRange.Low);
dll.SetInputRangeI(LV9063EntryPoint.Input.I1, LV9063EntryPoint.InputRange.High);
dll.SendRelayTable();
dll.SetSamplingFreq(1000);
dll.SendAcqTable();
dll.AcquireData(data);
```

## *Encoder*

### **Description**

This is an enumeration to specify the pulse per revolution of the encoder in use.

### **Prototype**

```
public enum Encoder
{
    encoder360 = 360,
    encoder512 = 512,
    encoder1024 = 1024,
    encoder2048 = 2048
}
```

### **Values Descriptions**

encoder360	360 pulses per revolution.
encoder512	512 pulses per revolution.
encoder1024	1024 pulses per revolution.
encoder2048	2048 pulses per revolution.

### **Example**

```
int value = 0;
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetEncoderPPR(LV9063EntryPoint.Encoder.encoder360);
dll.GetEncoderValue(ref value);
```



## *TrigSlope*

### Description

This is an enumeration for the triggering slope.

### Prototype

```
public enum TrigSlope
{
    Rising,
    Falling
}
```

### Values Descriptions

Rising	Rising edge (positive slope).
Falling	Falling edge (negative slope).

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetTrigHardware(true);
dll.SetTrigSource(LV9063EntryPoint.TrigSource.E1); // Set E1 as triggering source
dll.SetTrigLevel(1.5f); // Set 1.5 V as triggering level
dll.SetTrigSlope(LV9063EntryPoint.TrigSlope.Rising); // Positive slope
dll.SetSamplingFreq(1000);
dll.SendAcqTable();
dll.AcquireData(data);
```

## *TrigSource*

### Description

This is an enumeration for the triggering source when using the trigger functions.

### Prototype

```
public enum TrigSource
{
    Off = 0,
    E1 = 1,
    E2 = 2,
    E3 = 3,
    E4 = 4,
    I1 = 5,
    I2 = 6,
    I3 = 7,
    I4 = 8,
    AI1 = 13,
    AI2 = 14,
    AI3 = 15,
    AI4 = 16,
    AI5 = 17,
    AI6 = 18,
    AI7 = 19,
    AI8 = 20
}
```

### Values Descriptions

Off	No triggering source.
E...	Trigger on the corresponding voltage input.
I...	Trigger on the corresponding current input.
AI...	Trigger on the corresponding analog input.

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetTrigHardware(true);
dll.SetTrigSource(LV9063EntryPoint.TrigSource.E1);
```

## Waveform

### Description

This is an enumeration to adjust the type of wave form used by analog outputs.

### Prototype

```
public enum Waveform
{
    Sine,
    Square,
    Triangle,
    SawTooth
}
```

### Values Descriptions

Sine	Sine wave form.
Square	Square wave form.
Triangle	Triangle wave form.
SawTooth	Saw tooth wave form.

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,
    LV9063EntryPoint.FunctionType.FunctionGenerator);
dll.SetAnalogOutputAmplitude(LV9063EntryPoint.AO1, 5f);
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,
    LV9063EntryPoint.Waveform.Square);
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);
dll.SendAnalogOutputsTable();
```

## AcquireData

### Description

Acquire the data points from the 9063 module.

### Prototype

```
public ErrorCode AcquireData(float[] data, bool formatted)
```

### Parameters

**data** Output parameter that will store the data points received from the 9063 module. If this parameter is not initialized or is of the wrong size, it will be automatically initialized / resized to the correct size. The received data is organized in the array in this manner: for example, if the inputs E1, E2 and I1 were specified before calling SendAcqTable(), then the data in the array will be [E1<sub>0</sub>, E2<sub>0</sub>, I1<sub>0</sub>, E1<sub>1</sub>, E2<sub>1</sub>, I1<sub>1</sub>, E1<sub>2</sub>, E2<sub>2</sub>, I1<sub>2</sub>,...].

Ex.: For 2 inputs (channels) with 1024 samples per input (nbPackets = 8 for SetAcqTable), the correct size of the array is 1024 samples \* 2 inputs = 2048 array elements.

**formatted** Flag to indicate if the data must be raw or formatted, where:  
 True = receive formatted values (V or A),  
 False = receive raw values (12 bits ADC value)

**Table 1: Raw values interpretation**

	Range	Min		Max	
		Raw (12bits ADC) Value	Formatted Value	Raw (12bits ADC) Value	Formatted Value
<b>Voltage Inputs (E1 - E4)</b>	High	0	-800 V	4095	+800 V
	Low	0	-80 V	4095	+80 V
<b>Current Inputs (I1 - I4)</b>	High	0	-40 A	4095	+40 A
	Low	0	-4 A	4095	+4 A
<b>Analog Inputs (AI-1 – AI-8)</b>	N/A	0	-10 V	4095	+10 V

### Return Values

ErrorCode.None  
 ErrorCode.Communication

## *AcquireData...*

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(3); // Set for 3 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetInput(2, LV9063EntryPoint.Input.I1); // Set input #3 as I1
dll.SetSamplingFreq(1000);
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

SetInput

SetNbInputs

SetNbPackets

SetSamplingFreq

## *CloseDevice*

### **Description**

Terminate communication with the 9063 module.

### **Prototype**

```
public ErrorCode CloseDevice()
```

### **Parameters**

This function has no parameters.

### **Return Values**

```
ErrorCode.None  
ErrorCode.Communication
```

### **Example**

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
// ..  
// Do something  
// ..  
dll.CloseDevice();
```

### **See Also**

LV9063EntryPoint  
InitDevice

## *GetEncoderValue*

### **Description**

Read the speed in RPM determined from the encoder.

### **Prototype**

```
public ErrorCode GetEncoderValue(ref int value)
```

### **Parameters**

value            Output parameter that will store the speed in RPM.

### **Return Values**

ErrorCode.None

ErrorCode.Communication

### **Example**

```
int value = 0;
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetEncoderPPR(LV9063EntryPoint.Encoder.encoder360);
dll.GetEncoderValue(ref value);
```

### **See Also**

SetEncoderPPR

## *GetFirmwareVersion*

### **Description**

Read the firmware version of the 9063 module.

### **Prototype**

```
public ErrorCode GetFirmwareVersion(ref int version)
```

### **Parameters**

version            Output parameter that will store the firmware version of the 9063 module.

### **Return Values**

ErrorCode.None  
ErrorCode.Communication

### **Example**

```
int version = 0;  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.GetFirmwareVersion(ref version);
```



## *InitDevice*

### **Description**

Initialize the communication with the 9063 module.

### **Prototype**

```
public ErrorCode InitDevice(int moduleIndex)
```

### **Parameters**

moduleIndex            Index of the selected 9063 module.

### **Return Values**

```
ErrorCode.None  
ErrorCode.DeviceID  
ErrorCode.UnavailablePackage
```

### **Example**

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
if (dll.InitDevice(0) != LV9063EntryPoint.ErrorCode.None)  
{  
    System.Windows.Forms.MessageBox.Show("Communication failed");  
    return;  
}
```

### **See Also**

LV9063EntryPoint  
CloseDevice  
ErrorCode

## *LV9063EntryPoint*

### **Description**

Class constructor. Use this constructor to create your object to control the 9063 module.

### **Prototype**

```
public LV9063EntryPoint()
```

### **Parameters**

This constructor has no parameters.

### **Return Values**

Returns a new object of type LV9063EntryPoint. DLL must be loaded into your project before you can use this function.

### **Example**

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);
```

## *SendAcqTable*

### Description

Send the command to set the acquisition table. Call this function after setting the inputs, the sampling frequency or the number of packets or after calling the *SetAcqTable* function.

### Prototype

```
public ErrorCode SendAcqTable ()
```

### Parameters

This function has no parameters.

### Default Values

NbInputs	1
Input	Input #1 as <code>LV9063EntryPoint.Input.E1</code>
NbPackets	2
SamplingFrequency	1000 Hz
TrigHardware	<code>false</code>
TrigLevel	0 V
TrigSlope	<code>LV9063EntryPoint.TrigSlope.Rising</code>
TrigDelay	0
TrigSource	<code>LV9063EntryPoint.TrigSource.OFF</code>
Sync	<code>false</code>

### Return Values

```
ErrorCode.None  
ErrorCode.Communication
```

### Example

```
float[] data = new float[16384];  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetNbInputs(2); // Set for 2 inputs  
dll.SetInput(0, LV9063EntryPoint.Input.I1); // Set input #1 as I1  
dll.SetInput(1, LV9063EntryPoint.Input.I2); // Set input #2 as I2  
dll.SetSamplingFreq(1000);  
dll.SendAcqTable();  
dll.AcquireData(data);
```

### See Also

SetNbInputs  
SetInput  
SetSamplingFreq  
SetAcqTable

## *SendAnalogOutputsTable*

### Description

Send the analog outputs table to the 9063 module.

### Prototype

```
public ErrorCode SendAnalogOutputsTable()
```

### Parameters

This function has no parameters.

### Default Values

AnalogOutputFunction	LV9063EntryPoint.FunctionType.CommandButton
AnalogOutputWaveform	LV9063EntryPoint.Waveform.Sine
AnalogOutputAmplitude	0 V
AnalogOutputFrequency	0.01 Hz
AnalogOutputValue	0 (-10V)
AnalogOutputStartStop	STOP

### Return Values

```
ErrorCode.None  
ErrorCode.Communication
```

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.FunctionType.FunctionGenerator);  
dll.SetAnalogOutputAmplitude(LV9063EntryPoint.AO1, 5f);  
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.Waveform.Sine);  
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);  
dll.SendAnalogOutputsTable();
```

### See Also

SetAnalogOutputAmplitude  
SetAnalogOutputFrequency  
SetAnalogOutputFunction  
SetAnalogOutputStartStop  
SetAnalogOutputValue  
SetAnalogOutputWaveform

## *SendDigitalOutputsTable*

### Description

Send the command to the 9063 to set the digital outputs.

### Prototype

```
public ErrorCode SendDigitalOutputsTable()
```

### Parameters

This function has no parameters.

### Default Values

DigitalOutputState	false
DB9Value	0

### Return Values

```
ErrorCode.None  
ErrorCode.Communication
```

### Example

```
byte db9Value;  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetDigitalOutputState(LV9063EntryPoint.DO1, true);  
db9Value = 0x4A; // Set control pins 1, 3 and SYNC  
dll.SetDB9(db9Value);  
dll.SendDigitalOutputsTable();
```

### See Also

SetDigitalOutputState  
SetDB9

## *SendRelayTable*

### Description

Send the command to set the relay table.

### Prototype

```
public ErrorCode SendRelayTable()
```

### Parameters

This function has no parameters.

### Return Values

`ErrorCode.None`

`ErrorCode.Communication`

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.I1); // Set input #1 as I1
dll.SetInput(1, LV9063EntryPoint.Input.I2); // Set input #2 as I2
dll.SetInputRangeI(LV9063EntryPoint.Input.I1, LV9063EntryPoint.InputRange.Low);
dll.SetInputRangeI(LV9063EntryPoint.Input.I2, LV9063EntryPoint.InputRange.High);
dll.SendRelayTable();
dll.SetSamplingFreq(1000);
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

[AcquireData](#)

[SetNbInputs](#)

[SetInput](#)

[SetInputRangeE](#)

[SetInputRangeI](#)

## SetAcqTable

### Description

Set the acquisition table with the parameters for each input.

### Prototype

```
public ErrorCode SetAcqTable(float samplingFreq, Input[] inputs, int nbPackets)
```

### Parameters

- samplingFreq** New sampling frequency in Hertz. Frequency must be between 0.02 Hz and 600 KHz.
- inputs** Array of variable size. Contains the input IDs. See [Input](#) enum.
- nbPackets** New number of packets of 256 samples per input.  
Ex.: For 2 inputs (channels) with 1024 samples per input, nbPackets is (1024 samples / 256) \* 2 inputs = **8** packets.

### Return Values

[ErrorCode](#).None  
[ErrorCode](#).OutOfBounds

### Example

```
float[] data = new float[16384];
LV9063EntryPoint.Input[] inputs = { LV9063EntryPoint.Input.E1,
    LV9063EntryPoint.Input.E2, LV9063EntryPoint.Input.I1}; // Set inputs #1 as E1,
    #2 as E2 and #3 as I1
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetAcqTable(1000f, inputs, 4); // Set sampling freq. at 1000Hz, set the inputs
    and set 4 packets
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

[AcquireData](#)  
[SetInput](#)  
[SetNbInputs](#)

## *SetAllInputsRange*

### Description

Set the range for all the *E* and *I* inputs.

### Prototype

```
public ErrorCode SetAllInputsRange(InputRange[] ranges)
```

### Parameters

ranges            An array for the 8 inputs range, organized as:  
                  [E1, E2, E3, E4, I1, I2, I3, I4]

### Return Values

`ErrorCode.None`

`ErrorCode.OutOfBounds`

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(8); // Set for 8 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetInput(2, LV9063EntryPoint.Input.E3); // Set input #3 as E3
dll.SetInput(3, LV9063EntryPoint.Input.E4); // Set input #4 as E4
dll.SetInput(4, LV9063EntryPoint.Input.I1); // Set input #5 as I1
dll.SetInput(5, LV9063EntryPoint.Input.I2); // Set input #6 as I2
dll.SetInput(6, LV9063EntryPoint.Input.I3); // Set input #7 as I3
dll.SetInput(7, LV9063EntryPoint.Input.I4); // Set input #8 as I4
dll.SetAllInputsRange(new LV9063EntryPoint.InputRange[] {
    LV9063EntryPoint.InputRange.Low, LV9063EntryPoint.InputRange.Low,
    LV9063EntryPoint.InputRange.Low, LV9063EntryPoint.InputRange.High,
    LV9063EntryPoint.InputRange.Low, LV9063EntryPoint.InputRange.Low,
    LV9063EntryPoint.InputRange.Low, LV9063EntryPoint.InputRange.High });
    // Set E1 as Low range, E2 as Low range,
    // E3 as Low range, E4 as High range,
    // I1 as Low range, I2 as Low range,
    // I3 as Low range and I4 as High range
dll.SendRelayTable();
dll.SetSamplingFreq(1000);
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

- AcquireData
- SetInput
- SetNbInputs
- SetInputRangeE
- SetInputRangeI
- SendRelayTable



## *SetAnalogOutputAmplitude*

### Description

Set the amplitude in Volt for the specified analog output (*Function Generator* only).

### Prototype

```
public ErrorCode SetAnalogOutputAmplitude(int index, float amplitude)
```

### Parameters

index	Index of the analog output. Can be AO1 or AO2, where: AO1 = Analog output 1, AO2 = Analog output 2.
amplitude	New amplitude value in Volt for the analog output at <i>index</i> . The amplitude must be between -10 V to 10 V.

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.FunctionType.FunctionGenerator);  
dll.SetAnalogOutputAmplitude(LV9063EntryPoint.AO1, 5f);  
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.Waveform.Sine);  
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);  
dll.SendAnalogOutputsTable();
```

### See Also

SetAnalogOutputFrequency  
SetAnalogOutputFunction  
SetAnalogOutputStartStop  
SendAnalogOutputsTable  
SetAnalogOutputValue  
SetAnalogOutputWaveform

## *SetAnalogOutputFrequency*

### Description

Set the frequency in Hertz for the specified analog output (*Function Generator* only).

### Prototype

```
public ErrorCode SetAnalogOutputFrequency(int index, float frequency)
```

### Parameters

index	Index of the analog output. Can be AO1 or AO2, where: AO1 = Analog output 1, AO2 = Analog output 2.
frequency	New frequency value in Hertz for the analog output at <i>index</i> . The frequency must be between 0 Hz to 100 Hz.

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.FunctionType.FunctionGenerator);  
dll.SetAnalogOutputFrequency(LV9063EntryPoint.AO1, 60f);  
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.Waveform.Sine);  
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);  
dll.SendAnalogOutputsTable();
```

### See Also

SetAnalogOutputAmplitude  
SetAnalogOutputFunction  
SetAnalogOutputStartStop  
SendAnalogOutputsTable  
SetAnalogOutputValue  
SetAnalogOutputWaveform

## *SetAnalogOutputFunction*

### Description

Set the function type for the specified analog output.

### Prototype

```
public ErrorCode SetAnalogOutputFunction(int index, FunctionType function)
```

### Parameters

index	Index of the analog output. Can be AO1 or AO2, where: AO1 = Analog output 1, AO2 = Analog output 2.
function	New function type for the analog output at <i>index</i> . See <a href="#">FunctionType</a> enum.

### Return Values

[ErrorCode](#).None  
[ErrorCode](#).OutOfBounds

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,
    LV9063EntryPoint.FunctionType.FunctionGenerator);
dll.SetAnalogOutputAmplitude(LV9063EntryPoint.AO1, 5f);
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,
    LV9063EntryPoint.Waveform.Sine);
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);
dll.SendAnalogOutputsTable();
```

### See Also

SetAnalogOutputAmplitude  
SetAnalogOutputFrequency  
SetAnalogOutputStartStop  
SendAnalogOutputsTable  
SetAnalogOutputValue  
SetAnalogOutputWaveform

## *SetAnalogOutputStartStop*

### Description

Set the start/stop status of the specified analog output (*Function Generator* only).

### Prototype

```
public ErrorCode SetAnalogOutputStartStop(int index, int status)
```

### Parameters

index	Index of the analog output. Can be AO1 or AO2, where: AO1 = Analog output 1, AO2 = Analog output 2.
status	New status for the analog output at <i>index</i> . Can be STOP or START, where: STOP = Stopped state, START = Started state.

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.FunctionType.FunctionGenerator);  
dll.SetAnalogOutputAmplitude(LV9063EntryPoint.AO1, 5f);  
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,  
    LV9063EntryPoint.Waveform.Sine);  
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);  
dll.SendAnalogOutputsTable();
```

### See Also

SetAnalogOutputAmplitude  
SetAnalogOutputFrequency  
SetAnalogOutputFunction  
SendAnalogOutputsTable  
SetAnalogOutputValue  
SetAnalogOutputWaveform

## SetAnalogOutputValue

### Description

Set the value of the specified analog output (*Command Button* only).

### Prototype

```
public ErrorCode SetAnalogOutputValue(int index, int value)
```

### Parameters

index	Index of the analog output. Can be AO1 or AO2, where: AO1 = Analog output 1, AO2 = Analog output 2.
value	New value for the analog output at <i>index</i> . Can be in the range [0, 4095], where: 0 = -10 V, ... 2047 = 0 V, ... 4095 = 10 V.

### Return Values

`ErrorCode.None`  
`ErrorCode.OutOfBounds`

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,
    LV9063EntryPoint.FunctionType.CommandButton);
dll.SetAnalogOutputValue(LV9063EntryPoint.AO1, 3072); // ~5 Volts
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);
dll.SendAnalogOutputsTable();
```

### See Also

SetAnalogOutputAmplitude  
SetAnalogOutputFrequency  
SetAnalogOutputFunction  
SetAnalogOutputStartStop  
SendAnalogOutputsTable  
SetAnalogOutputWaveform

## *SetAnalogOutputWaveform*

### Description

Set the waveform for the specified analog output.

### Prototype

```
public ErrorCode SetAnalogOutputWaveform(int index, Waveform waveform)
```

### Parameters

index	Index of the analog output. Can be AO1 or AO2, where: AO1 = Analog output 1, AO2 = Analog output 2.
waveform	New waveform type for the analog output at <i>index</i> . See <a href="#">Waveform</a> enum.

### Return Values

[ErrorCode](#).None  
[ErrorCode](#).OutOfBounds

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetAnalogOutputFunction(LV9063EntryPoint.AO1,
    LV9063EntryPoint.FunctionType.FunctionGenerator);
dll.SetAnalogOutputAmplitude(LV9063EntryPoint.AO1, 5f);
dll.SetAnalogOutputWaveform(LV9063EntryPoint.AO1,
    LV9063EntryPoint.Waveform.Sine);
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);
dll.SendAnalogOutputsTable();
```

### See Also

[SetAnalogOutputAmplitude](#)  
[SetAnalogOutputFrequency](#)  
[SetAnalogOutputFunction](#)  
[SetAnalogOutputStartStop](#)  
[SendAnalogOutputsTable](#)  
[SetAnalogOutputValue](#)

## *SetDB9*

### Description

Set the state of the pins on the DB9 port, where a pin set to 0 equals to 0 V and a pin set to 1 equals to 5 V.

### Prototype

```
public ErrorCode SetDB9(byte value)
```

### Parameters

value	New value for the pins of the DB9, where:	
	Bit0 = Control Pin 1,	Bit1 = Control Pin 2,
	Bit2 = Control Pin 3,	Bit3 = Control Pin 4,
	Bit4 = Control Pin 5,	Bit5 = Control Pin 6,
	Bit6 = SYNC Pin 7.	

### Return Values

```
ErrorCode.None
```

### Example

```
byte db9Value;
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetDigitalOutputState(LV9063EntryPoint.DO1, true);
db9Value = 0x4A; // "0100 1010" Set control pins 2, 4 and SYNC to 5V
dll.SetDB9(db9Value);
dll.SendDigitalOutputsTable();
```

### See Also

SetDigitalOutputState  
SendDigitalOutputsTable

## *SetDigitalOutputs*

### Description

Set the TTL level of both digital outputs.

### Prototype

```
public ErrorCode SetDigitalOutputs(bool d1, bool d2)
```

### Parameters

- d1            New TTL level for the digital output 1, where:  
              *True* = high (+5 V),  
              *False* = low (0 V).
- d2            New TTL level for the digital output 2, where:  
              *True* = high (+5 V),  
              *False* = low (0 V).

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
byte db9Value;  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetDigitalOutputs(false, true); // Digital output #1 to low and #2 to high  
dll.SendDigitalOutputsTable();
```

### See Also

SetDigitalOutputState  
SendDigitalOutputsTable



## *SetDigitalOutputState*

### Description

Set the state of the specified digital output.

### Prototype

```
public ErrorCode SetDigitalOutputState(int index, bool state)
```

### Parameters

index	Index of the digital output, where: 0 = Digital output 1, 1 = Digital output 2.
state	New state for the digital output at <i>index</i> , where: <i>True</i> = high (+5 V), <i>False</i> = low (0 V).

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
byte db9Value;  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetDigitalOutputState(LV9063EntryPoint.DO1, true);  
db9Value = 0x4A; // "0100 1010" Set control pins 2, 4 and SYNC  
dll.SetDB9(db9Value);  
dll.SendDigitalOutputsTable();
```

### See Also

SendDigitalOutputsTable  
SetDB9

## *SetEncoderPPR*

### **Description**

Set the number of pulses per revolution of the encoder.

### **Prototype**

```
public ErrorCode SetEncoderPPR(Encoder ppr)
```

### **Parameters**

ppr                    New number of pulses per revolution of the encoder.

### **Return Values**

`ErrorCode.None`

### **Example**

```
int val = 0;
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetEncoderPPR(LV9063EntryPoint.Encoder.encoder360);
dll.GetEncoderValue(ref val);
```

### **See Also**

GetEncoderValue

## *SetFunctionGenerator*

### Description

Set the specified analog output to create a function generator.

### Prototype

```
public ErrorCode SetFunctionGenerator(int index, Waveform waveform, float amplitude, float frequency)
```

### Parameters

index	Index of the analog output, where: AO1 = Analog output 1, AO2 = Analog output 2.
waveform	New waveform type for the analog output at <i>index</i> . See <a href="#">Waveform</a> enum.
amplitude	New amplitude value in Volt for the analog output at <i>index</i> . Amplitude can be between 0 to 10 V
frequency	New frequency value in Hertz for the analog output at <i>index</i> . Frequency can be between 0 to 100 Hz.

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetFunctionGenerator(LV9063EntryPoint.AO1, LV9063EntryPoint.Waveform.Sine,  
    5f, 60f);  
dll.SetAnalogOutputStartStop(LV9063EntryPoint.AO1, LV9063EntryPoint.START);  
dll.SendAnalogOutputsTable();
```

### See Also

SetEncoder  
GetEncoderValue

## *SetInput*

### Description

Set an input to read data from the 9063.

### Prototype

```
public ErrorCode SetInput(int index, Input input)
```

### Parameters

index	Index of the input in the input vector. The index value must be between 0 and one less than the number of inputs set by the “SetNbInputs” function. For example, if you use “SetNbInputs(3)”, then the valid index values are 0, 1 and 2.
input	ID of the input. See <a href="#">Input</a> enum.

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
float[] data = new float[16384];  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetNbInputs(2); // Set for 2 inputs  
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1  
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2  
dll.SetSamplingFreq(1000f);  
dll.SendAcqTable();  
dll.AcquireData(data);
```

### See Also

AcquireData  
SetNbInputs  
SetInputRangeE  
SetInputRangeI

## *SetInputRangeE*

### Description

Set the range for the specified voltage input.

### Prototype

```
public ErrorCode SetInputRangeE(Input input, InputRange range)
```

### Parameters

input	ID of the voltage input. See <a href="#">Input</a> enum.
range	New range for the specified voltage input. See <a href="#">InputRange</a> enum.

### Return Values

[ErrorCode](#).None  
[ErrorCode](#).OutOfBounds

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.I1); // Set input #2 as I1
dll.SetInputRangeE(LV9063EntryPoint.Input.E1, LV9063EntryPoint.InputRange.Low);
dll.SetInputRangeI(LV9063EntryPoint.Input.I1, LV9063EntryPoint.InputRange.Low);
dll.SendRelayTable();
dll.SetSamplingFreq(1000f);
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

[AcquireData](#)  
[SetInput](#)  
[SetNbInputs](#)  
[SetInputRangeI](#)  
[SendRelayTable](#)

## *SetInputRangeI*

### Description

Set the range for the specified current input.

### Prototype

```
public ErrorCode SetInputRangeI(Input index, InputRange range)
```

### Parameters

index	ID of the current input. See Input enum.
range	New range for the specified current input. See InputRange enum.

### Return Values

ErrorCode.None  
 ErrorCode.OutOfBounds

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.I1); // Set input #2 as I1
dll.SetInputRangeE(LV9063EntryPoint.Input.E1, LV9063EntryPoint.InputRange.Low);
dll.SetInputRangeI(LV9063EntryPoint.Input.I1, LV9063EntryPoint.InputRange.Low);
dll.SendRelayTable();
dll.SetSamplingFreq(1000f);
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

AcquireData  
 SetInput  
 SetNbInputs  
 SetInputRangeE  
 SendRelayTable

## *SetNbInputs*

### Description

Set the number of inputs to read data from the 9063.

### Prototype

```
public ErrorCode SetNbInputs(int nbInputs)
```

### Parameters

nbInputs      New number of inputs.

### Return Values

ErrorCode.None  
ErrorCode.OutOfBounds

### Example

```
float[] data = new float[16384];  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetNbInputs(2); // Set for 2 inputs  
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1  
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2  
dll.SetSamplingFreq(1000f);  
dll.SendAcqTable();  
dll.AcquireData(data);
```

### See Also

AcquireData  
SetInput  
SetInputRangeE  
SetInputRangeI

## SetNbPackets

### Description

Set the number of packets containing 256 samples. The total number of samples is divided evenly between all enabled inputs, where:

$$\frac{\text{nbPackets} * 256 \text{ samples}}{\text{nbInputs}} = \text{number of samples per input}$$

### Prototype

```
public ErrorCode SetNbPackets(int nbPackets)
```

### Parameters

nbPackets      New number of packets of 256 samples.

### Return Values

```
ErrorCode.None  
ErrorCode.OutOfBounds
```

### Example

```
int nbPackets = 4;  
float[] data = new float[16384];  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetNbInputs(2); // Set for 2 inputs  
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1  
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2  
dll.SetNbPackets(nbPackets); // Set 4 packets of 256 samples = 1024 samples or  
                             // 512 samples per input  
  
dll.SetSamplingFreq(1000f);  
dll.SendAcqTable();  
dll.AcquireData(data);
```

### See Also

AcquireData  
SetNbInputs  
SetInput



## *SetSamplingFreq*

### Description

Set the sampling frequency in Hertz.

### Prototype

```
public ErrorCode SetSamplingFreq(float samplingFreq)
```

### Parameters

samplingFreq New sampling frequency in Hertz. Sampling frequency must be between 0.02 Hz to 600 KHz.

### Return Values

ErrorCode.None  
ErrorCode.OutOfBounds

### Example

```
float[] data = new float[16384];  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetNbInputs(2); // Set for 2 inputs  
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1  
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2  
dll.SetSamplingFreq(1000f); // 1000 Hz  
dll.SendAcqTable();  
dll.AcquireData(data);
```

### See Also

AcquireData  
SetInput  
SetNbInputs

## *SetSync*

### Description

Set the status of the *SYNC* signal as the triggering signal.

### Prototype

```
public ErrorCode SetSync(bool status)
```

### Parameters

status            New status of the *SYNC* signal, where:  
                  *True* = use the *SYNC* signal for the triggering signal,  
                  *False* = do not use the *SYNC* signal for the triggering signal.

### Return Values

```
ErrorCode.None
```

### Example

```
float[] data = new float[16384];  
LV9063EntryPoint dll = new LV9063EntryPoint();  
dll.InitDevice(0);  
dll.SetNbInputs(2); // Set for 2 inputs  
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1  
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2  
dll.SetTrigHardware(true);  
dll.SetSync(true); // Use the SYNC signal as the triggering signal  
dll.SetSamplingFreq(1000f); // 1000 Hz  
dll.SendAcqTable();  
dll.AcquireData(data);
```

### See Also

AcquireData  
SetNbInputs  
SetInput

## *SetTrigDelay*

### Description

Set the triggering delay (hardware triggering only).

### Prototype

```
public ErrorCode SetTrigDelay(int delay)
```

### Parameters

delay            New triggering delay. Must be in the range [0, 65535], where:

- 0 = first triggering point for which there is at least 50% of the points after this point and at least 0% of the points before this point,
- ...
- 6553 = first triggering point for which there is at least 45% of the points after this point and at least 5% of the points before this point,
- ...
- 65535 = first triggering point for which there is at least 0% of the points after this point and at least 50% of the points before this point.

### Return Values

`ErrorCode.None`  
`ErrorCode.OutOfBounds`

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetTrigHardware(true);
dll.SetTrigDelay(0);
dll.SetSamplingFreq(1000f); // 1000 Hz
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

AcquireData  
SetTrigHardware  
SetTrigLevel  
SetTrigSlope  
SetTrigSource

## *SetTrigHardware*

### Description

Set the status of hardware triggering.

### Prototype

```
public ErrorCode SetTrigHardware(bool status)
```

### Parameters

status            New status of hardware triggering, where:  
                   *True* = use hardware triggering,  
                   *False* = do not use hardware triggering.

### Return Values

`ErrorCode.None`

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetTrigHardware(true);
dll.SetTrigSource(LV9063EntryPoint.TrigSource.E1); // Set E1 as triggering source
dll.SetTrigLevel(2); // Set 2 V as triggering level
dll.SetSamplingFreq(1000f); // 1000 Hz
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

AcquireData  
 SetTrigDelay  
 SetTrigLevel  
 SetTrigSlope  
 SetTrigSource

## *SetTrigLevel*

### Description

Set the triggering level (hardware triggering only).

### Prototype

```
public ErrorCode SetTrigLevel(float level)
```

### Parameters

level            Triggering level in the units of the triggering source (V for voltage, A for current).

### Return Values

`ErrorCode.None`

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetTrigHardware(true);
dll.SetTrigSource(LV9063EntryPoint.TrigSource.E1); // Set E1 as triggering source
dll.SetTrigLevel(1.5f); // Set 1.5 V as triggering level
dll.SetSamplingFreq(1000f); // 1000 Hz
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

AcquireData  
SetTrigDelay  
SetTrigHardware  
SetTrigSlope  
SetTrigSource

## SetTrigSlope

### Description

Set the edge (*falling* or *rising*) of the triggering slope (hardware triggering only).

### Prototype

```
public ErrorCode SetTrigSlope(TrigSlope slope)
```

### Parameters

slope            Use *Rising* for a rising edge triggering (positive slope) and *Falling* for a falling edge triggering (negative slope). See [TrigSlope](#) enum.

### Return Values

[ErrorCode](#).None

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetTrigHardware(true);
dll.SetTrigSource(LV9063EntryPoint.TrigSource.E1); // Set E1 as triggering source
dll.SetTrigLevel(1.5f); // Set 1.5 V as triggering level
dll.SetTrigSlope(LV9063EntryPoint.TrigSlope.Rising); // Positive slope
dll.SetSamplingFreq(1000f); // 1000 Hz
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

AcquireData  
SetTrigDelay  
SetTrigHardware  
SetTrigLevel  
SetTrigSource

## *SetTrigSource*

### Description

Set the triggering source (hardware triggering only).

### Prototype

```
public ErrorCode SetTrigSource(TrigSource source)
```

### Parameters

source            ID of the triggering source. See [TrigSource](#) enum.

### Return Values

[ErrorCode.None](#)

[ErrorCode.OutOfBounds](#)

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetNbInputs(2); // Set for 2 inputs
dll.SetInput(0, LV9063EntryPoint.Input.E1); // Set input #1 as E1
dll.SetInput(1, LV9063EntryPoint.Input.E2); // Set input #2 as E2
dll.SetTrigHardware(true);
dll.SetTrigSource(LV9063EntryPoint.TrigSource.E1); // Set E1 as triggering source
dll.SetTrigLevel(1.5f); // Set 1.5 V as triggering level
dll.SetSamplingFreq(1000f); // 1000 Hz
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

[AcquireData](#)

[SetTrigDelay](#)

[SetTrigHardware](#)

[SetTrigLevel](#)

[SetTrigSlope](#)

## SetTrigTable

### Description

Set the triggering table (hardware triggering only).

### Prototype

```
public ErrorCode SetTrigTable(TrigSource source, float level, int delay, TrigSlope slope)
```

### Parameters

source	ID of the triggering source. See <a href="#">TrigSource</a> enum.
level	Triggering level in the units of the triggering source (V for voltage, A for current).
delay	New triggering delay. Can be in the range [0, 65535], where: 0 = first triggering point for which there is at least 50% of the points after this point and at least 0% of the points before this point, ... 6553 = first triggering point for which there is at least 45% of the points after this point and at least 5% of the points before this point, ... 65535 = first triggering point for which there is at least 0% of the points after this point and at least 50% of the points before this point.
slope	Use <i>Rising</i> for a rising edge triggering (positive slope) and <i>Falling</i> for a falling edge triggering (negative slope). See <a href="#">TrigSlope</a> enum.

### Return Values

[ErrorCode](#).None  
[ErrorCode](#).OutOfBounds

### Example

```
float[] data = new float[16384];
LV9063EntryPoint dll = new LV9063EntryPoint();
dll.InitDevice(0);
dll.SetAcqTable(1000f, new LV9063EntryPoint.Input[] { LV9063EntryPoint.Input.E1,
    LV9063EntryPoint.Input.E2 }, 4);
dll.SetTrigHardware(true);
dll.SetTrigTable(LV9063EntryPoint.TrigSource.E1, 1.5f, 0, LV9063EntryPoint.
    TrigSlope.Rising);
dll.SendAcqTable();
dll.AcquireData(data);
```

### See Also

[SetTrigDelay](#)  
[SetTrigHardware](#)  
[SetTrigLevel](#)  
[SetTrigSlope](#)  
[SetTrigSource](#)