



# HRS SDK PROGRAMMING GUIDE

RELEASE DATE: 2019/11/07

Customize Your Tests with HRS SDK and Make Testing More Efficient

[www.whaleteq.com](http://www.whaleteq.com)



<b>1. OVERVIEW .....</b>	<b>3</b>
1.1. CONTENTS.....	3
1.2. CAUTIONS .....	4
<b>2. API FUNCTIONS FOR C/C++ .....</b>	<b>4</b>
2.1. FUNDAMENTAL FUNCTIONS .....	4
2.1.1. InitPPG .....	4
2.1.2. StartPPG.....	5
2.1.3. IsConnected.....	5
2.1.4. ClosePPG .....	5
2.2. PD SAMPLING RELATED FUNCTIONS .....	6
2.2.1. IsWiderPDRange .....	6
2.2.2. RegisterPDSamplingCB.....	7
2.2.3. RegisterPDSamplingPeakCB.....	7
2.2.4. SetSampleSetting .....	8
2.2.5. SetSampleChannel .....	8
2.2.6. SendLedOff.....	9
2.3. MAIN MODE .....	10
2.3.1. SendCommand.....	10
2.4. FINE TUNE MODE (ONLY SUPPORTED ON HRS200) .....	10
2.4.1. RegisterTimeAmpCB (HRS200 Only).....	10
2.4.2. OutputFunction (HRS200 Only) .....	11
2.4.3. SendFineTune (HRS200 Only) .....	11
2.4.4. SendFineTuneNoiseAmp (HRS200 Only).....	12
2.5. PLAY RAW DATA (ONLY SUPPORTED ON HRS200).....	12
2.5.1. LoadDynamic (HRS200 Only).....	12
2.5.2. GetDynamicArray (HRS200 Only).....	13
2.5.3. PlayDynamic (HRS200 Only).....	13
2.5.4. ResampleDynamic (HRS200 Only).....	14
<b>3. API FUNCTIONS FOR C#.....</b>	<b>15</b>
3.1. FUNDAMENTAL FUNCTIONS .....	15
3.1.1. InitPPG .....	15



3.1.2.	<i>StartPPG</i> .....	15
3.1.3.	<i>IsConnected</i> .....	16
3.1.4.	<i>ClosePPG</i> .....	16
3.2.	<b>PD SAMPLING RELATED FUNCTIONS</b> .....	16
3.2.1.	<i>IsWiderPDRange</i> .....	17
3.2.2.	<i>RegisterPDSamplingCB</i> .....	18
3.2.3.	<i>RegisterPDSamplingPeakCB</i> .....	18
3.2.4.	<i>SetSampleSetting</i> .....	19
3.2.5.	<i>SetSampleChannel</i> .....	19
3.2.6.	<i>SendLedOff</i> .....	20
3.3.	<b>MAIN MODE</b> .....	20
3.3.1.	<i>SendCommand</i> .....	20
3.4.	<b>FINE TUNE MODE (ONLY SUPPORTED ON HRS200)</b> .....	21
3.4.1.	<i>OutputFunction (HRS200 Only)</i> .....	21
3.4.2.	<i>RegisterTimeAmpCB (HRS200 Only)</i> .....	22
3.4.3.	<i>SendFineTune (HRS200 Only)</i> .....	22
3.4.4.	<i>SendFineTuneNoiseAmp (HRS200 Only)</i> .....	23
3.5.	<b>PLAY RAW DATA (ONLY SUPPORTED ON HRS200)</b> .....	23
3.5.1.	<i>LoadDynamic (HRS200 Only)</i> .....	23
3.5.2.	<i>GetDynamicArray (HRS200 Only)</i> .....	24
3.5.3.	<i>PlayDynamic (HRS200 Only)</i> .....	24
3.5.4.	<i>ResampleDynamic (HRS200 Only)</i> .....	25
4.	<b>CONTACT WHALETEQ</b> .....	26

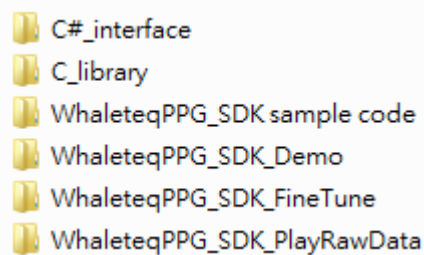


# 1. Overview

The SDK is designed for Whaleteq customer to control the output signal of HRS. Therefore, WhaleTeq customer could develop their own production tool or other tools with special requirements

## 1.1. Contents

The HRS SDK package includes



- **C#\_interface:**  
WhaleTeqHRS\_SDK.cs/dll for C# developer.
- **C\_library**  
WhaleTeqHRS\_SDK.h/lib/dll for C/C++ developer.
- **WhaleteqPPG\_SDK sample code**  
A folder that include the source code of 3 demo programs: "Demo", "Fine Tune", and "Play Raw Data".
- **WhaleteqPPG\_SDK\_Demo**  
A program to demonstrate how to implement customized AP using the SDK.
- **WhaleteqPPG\_SDK\_FineTune**  
A program to demonstrate how to implement HRS Fine Tune mode by using the SDK.
- **WhaleteqPPG\_SDK\_PlayRawData**  
A program to demonstrate how to implement HRS Play Raw Data mode by using the SDK.



## 1.2. Cautions

- The supported API functions are different between HRS200 and HRS100+. The functions related to "Fine Tune" and "Play Raw Data" mode can't be used on HRS100+.
- The PD range of HRS is enhanced in July 2017. For implementation detail about different PD range of HRS, please reference section 2.2 (C/C++) or 3.2 (C#).

## 2. API functions for C/C++

For C/C++ developer, reference *C\_library\WhaleTeqPPG\_SDK\_release.h* for the APIs.

### 2.1. Fundamental Functions

This part contents the fundamental functions for communicate with HRS.

#### 2.1.1. InitPPG

##### Prototype

```
WHALETEQPPG_SDK_API bool InitPPG(DeviceConnectedCB connCB);
```

##### Parameter

**DeviceConnectedCB:**

Boolean function for checking whether USB is connected.

The prototype for this function is:

```
"typedef void (__stdcall *DeviceConnectedCB)(bool DeviceConnectedCB)"
```

##### Description

This API initials HRS, and must be called first. "Hot Plug" support with "connCB" parameter informing device connecting and un-connecting.

##### Return

**TRUE:** The device is connected.

**FALSE:** connected device not found.



### 2.1.2. StartPPG

#### Prototype

```
WHALETEQPPG_SDK_API bool StartPPG();
```

#### Parameter

None.

#### Description

The first API for "Hot Plug" unsupported environment. (e.g. LabView) Same as InitPPG() with connCB as NULL.

#### Return

**TRUE:** Success.

**FALSE:** Fail.

### 2.1.3. IsConnected

#### Prototype

```
WHALETEQPPG_SDK_API bool IsConnected();
```

#### Parameter

None.

#### Description

This API checks whether HRS device is connected.

#### Return

**TRUE:** Success.

**FALSE:** Fail.

### 2.1.4. ClosePPG

#### Prototype

```
WHALETEQPPG_SDK_API void ClosePPG();
```

#### Parameter

None.

#### Description

Last calling API when system exiting for releasing resource.



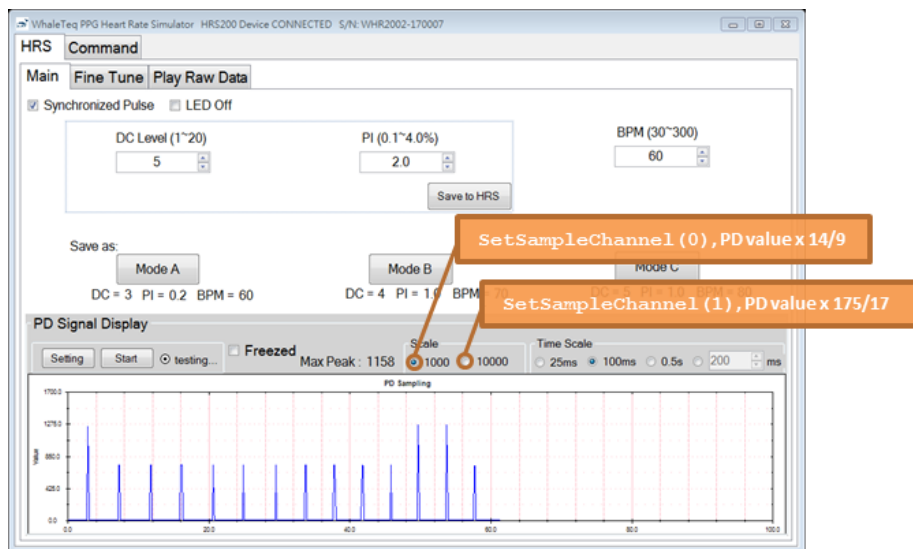
## Return

None.

## 2.2. PD Sampling Related Functions

The PD range of HRS is enhanced in July 2017. If there are different version of HRSs in hand, or would like to get the same PD value with HRS AP, please follow below implementation detail:

- Normal PD range (**IsWiderPDRange = FALSE**)
  - Get PD value without extra adjustment.
- Wider PD range (**IsWiderPDRange = TRUE**)
  - For scale 1000: call **SetSampleChannel(0)**, PD value x 14/9
  - For scale 10000: call **SetSampleChannel(1)**, PD value x 175/17



### 2.2.1. IsWiderPDRange

#### Prototype

```
WHALETEQPPG_SDK_API bool IsWiderPDRange();
```

#### Parameter

None.

#### Description



The API checks whether the wider PD range is supported on connected HRS device.

### Return

**TRUE:** "Wider PD range" is **supported** on the connected HRS device.

**FALSE:** "Wider PD range" is **unsupported** on the connected HRS device.

## 2.2.2. RegisterPDSamplingCB

### Prototype

```
WHALETEQPPG_SDK_API void RegisterPDSamplingCB(PDSamplingCB cb);
```

### Parameter

**PDSamplingCB:**

Callback function for real time PD Sampling.

The prototype for this function is:

```
"typedef void (__stdcall *PDSamplingCB)(int ValueNum, unsigned short*
ValueArray)"
```

### Description

This API register callback to get real time PD sampling data. About 0.00003 second (30us) interval between each 2 sampling data

### Return

None.

## 2.2.3. RegisterPDSamplingPeakCB

### Prototype

```
WHALETEQPPG_SDK_API void RegisterPDSamplingPeakCB
(PDSamplingPeakCB cb, double cycleSecond);
```

### Parameter

**PDSamplingPeakCB:**

Callback function for real time PD Sampling Peak value.

The prototype for this function is:

```
"typedef void (__stdcall *PDSamplingPeakCB)(int PeakValue)"
```

**cycleSecond:**

The duration of each cycle Maximum Peak detection.





### Description

This API register callback to get real time PD sampling Maximum Peak value in cycle seconds.

### Return

None.

## 2.2.4. SetSampleSetting

### Prototype

```
WHALETEQPPG_SDK_API bool SetSampleSetting(bool Enable, bool HighResolution);
```

### Parameter

#### Enable:

**TRUE:** Enable HRS PD Sampling. HRS receives real time lightness change from DUT (e.g. PPG module, smart watch) in this setting.

**FALSE:** Disable HRS PD Sampling. HRS will not receive lightness change from DUT (e.g. PPG module, smart watch) in the setting.

#### HighResolution:

**TRUE:** Set HRS as "HighResolution" mode. It only can be set when LED Off (by "SendLedOff()"). In "HighResolution" mode, HRS will return high resolution PD Sampling value.

**FALSE:** Set HRS as normal resolution mode.

### Description

This API control PD Sampling functionality.

### Return

**TRUE:** Success.

**FALSE:** Fail.

## 2.2.5. SetSampleChannel

### Prototype

```
WHALETEQPPG_SDK_API void SetSampleChannel(int Channel);
```

### Parameter

0: Set to get Channel 1 PD value.



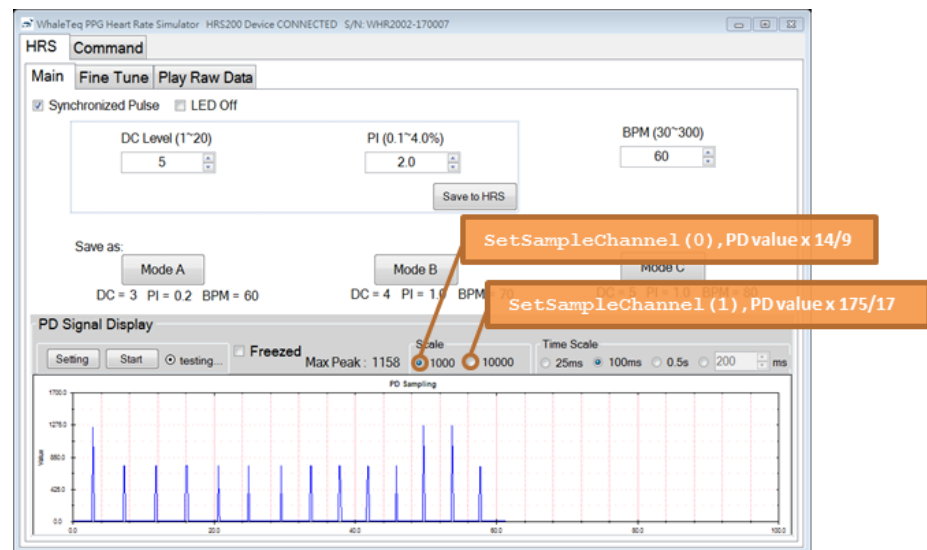
- 1: Set to get Channel 2 PD value. This channel could receive higher LED brightness. (Wider PD range.)

## Description

Call this API to select which channel of PD sampling is set.

## Return

None



## 2.2.6. SendLedOff

### Prototype

```
WHALETEQPPG_SDK_API bool SendLedOff();
```

### Parameter

None.

### Description

When calling this API, HRS will turn off LED (testing signal sending) functionality.

### Return

**TRUE:** Success.

**FALSE:** Fail.



## 2.3. Main Mode

### 2.3.1. SendCommand

#### Prototype

```
WHALETEQPPG_SDK_API bool SendCommand(int DCLevel, float PI,  
int HR, bool SyncPD);
```

#### Parameter

**DCLevel:** Setting DC Level. Range for this parameter is 1~20.

**PI:** Setting PI value. Range for PI value changes corresponding to DCLevel.

- Maximum PI value =  $20 / \text{DCLevel}$ .
- Minimum PI value =  $3 / (5 * \text{DCLevel})$

**HR:** Setting simulated Heart Rate value. Range for this parameter is 30~300 BPM (Beat Per Minutes).

**SyncPD:**

**TRUE:** Testing signal sending synchronized with PD receiving signal.

**FALSE:** HRS sends continuous testing signal.

#### Description

This API sends testing signal with specified DC Level, PI Value and Heart Rate(BPM).

#### Return

**TRUE:** Success.

**FALSE:** Fail.

## 2.4. Fine Tune Mode (Only supported on HRS200)

### 2.4.1. RegisterTimeAmpCB (HRS200 Only)

#### Prototype

```
WHALETEQPPG_SDK_API void RegisterTimeAmpCB(TimeAmplitudeCB cb);
```

#### Parameter

**TimeAmplitudeCB:**

Callback function for real time output signal amplitude.

The prototype for this function is:



*"typedef void (\_\_stdcall \*TimeAmplitudeCB)(double time, double amp)"*

### Description

Register callback to get real time output signal data. About 0.006 second (6ms) interval between each callback be triggered.

### Return

None.

## 2.4.2. OutputFunction (HRS200 Only)

### Prototype

```
enum OutputFunction_E {  
    Output_Off = 0,  
    Output_Sine = 1,  
    Output_Triangle = 2,  
    Output_Square = 3,  
    Output_PPG = 15  
    Output_InvPPG = 16  
};
```

### Description

Enumeration for testing signal waveform type. This enum is used in SendFineTune function.

## 2.4.3. SendFineTune (HRS200 Only)

### Prototype

```
WHALETEQPPG_SDK_API bool SendFineTune(OutputFunction_E  
Waveform, int NoiseHz, int DC, int AC, int BPM, bool SyncPD);
```

### Parameter

**Waveform:** Testing signal waveform type. For details please reference OutputFunction.

**NoiseHz:** This parameter sets the frequency of noise. Noise value can be 0 (without Noise), 50, 60 or 1k Hz.

**DC:** This parameter sets the DC value of simulated light. Range from 100 ~2500 uV.



**AC:** This parameter sets the AC value of simulated light. Range from 75 ~2500, which means 0.75~25uV.

**BPM:** This parameter sets the simulated BPM value. Range from 30 ~300 BPM.

**SyncPD:**

**TRUE:** Testing signal sending synchronized with PD receiving signal.

**FALSE:** HRS sends continuous testing signal.

## Description

This API sends testing signal with specified Waveform Type, Noise, DC Value, AC Value and Heart Rate (BPM).

## Return

**TRUE:** Success.

**FALSE:** HRS device not connected or Module HRS100 connected.

## 2.4.4. SendFineTuneNoiseAmp (HRS200 Only)

### Prototype

```
WHALETEQPPG_SDK_API void SendFineTuneNoiseAmp(double Amp);
```

### Parameter

**Amp:**

Setting the amplitude of noise in "Fine Tune" mode. Default value of this parameter is 0.1uV. Range from 0.01~0.2uV.

### Description

This API sets the noise amplitude in "Fine Tune" mode.

### Return

None.

## 2.5. Play Raw Data (Only supported on HRS200)

### 2.5.1. LoadDynamic (HRS200 Only)

#### Prototype

```
WHALETEQPPG_SDK_API int LoadDynamic(wchar_t* File, double* SampleRate, double* Max, double* Min, double* StopTime);
```



## Parameter

**File:** Full path including file name of "Raw Data File".

**SampleRate:** The Sample Rate in "Raw Data File".

**Max:** The maximum raw data value in "Raw Data File".

**Min:** The minimum raw data value in "Raw Data File".

**StopTime:** The total sampling seconds in "Raw Data File".

## Description

This API load "Raw Data File" for dynamic playing. When succeed, it would return the sample rate, maximum value, minimum value and total time with pointer parameters.

## Return

>0: succeed, the number of sample data loaded.

-1: Open file failed.

-2: Can't solve sample rate in Line 1.

-3: Can't solve sample number in Line 2.

-4: Raw data file is too large to fit the memory.

-5: Unsupported API for HRS100 Module

## 2.5.2. GetDynamicArray (HRS200 Only)

### Prototype

```
WHALETEQPPG_SDK_API double* GetDynamicArray(int* pNum);
```

### Parameter

**pNum:** The number of raw data values in ValueArray

### Description

This API get loaded Raw Data value array.

### Return

The raw data values array in double type.

## 2.5.3. PlayDynamic (HRS200 Only)

### Prototype

```
WHALETEQPPG_SDK_API bool PlayDynamic(double Gain, int Offset,  
PlayStopCB stopCB, PlayDynamicCB playCB, bool SyncPD);
```



## Parameter

**Gain:** Default value is 4095/(Max-Min).

**offset:** Default value is "-Min" to normalize the raw data value range.

**stopCB:** SDK supports "Loop" playing as default mode with stopCB as "(IntPtr)0". Specified "informing finish callback" to implement "non-Loop" playing mode.

The prototype for this function is: *"typedef void (\_\_stdcall \*PlayStopCB)()"*

**playCB:** Specified real time playing data callback.

The prototype for this function is: *"typedef void (\_\_stdcall*

*\*PlayDynamicCB) (double time, unsigned short DC, unsigned short AC)"*

**SyncPD:**

**TRUE:** Testing signal sending synchronized with PD receiving signal.

**FALSE:** HRS sends continuous testing signal.

## Description

This API start playing the loaded Raw Data.

## Return

**TRUE:** Success.

**FALSE:** Fail.

## 2.5.4. ResampleDynamic (HRS200 Only)

### Prototype

```
WHALETEQPPG_SDK_API void ResampleDynamic(double newSampleRate,  
double* stopTime);
```

### Parameter

**newSampleRate:** The sample rate value of resample.

**stopTime:** Updated totally playing time for Sample Rate changed.

### Description

This API changes the sample rate of loaded Raw Data for playing.

### Return

None.



## 3. API functions for C#

For C# developer, reference `C#_interface\WhaleTeqPPG_SDK_release.cs` for the APIs.

### 3.1. Fundamental Functions

This part contents the fundamental functions for communicate with HRS.

#### 3.1.1. InitPPG

##### Prototype

```
public static extern bool InitPPG(IntPtr connCB);
```

##### Parameter

**connCB:**

Callback support PPG device "Hot Plug" connecting and un-connecting.

##### Description

This API initials HRS, and must be called first. "Hot Plug" support with "connCB" parameter informing device connecting and un-connecting.

Before using this function, should call "delegate void ConnectCBDelegate (bool connected)" and use function pointer of delegate from "Marshal.GetFunctionPointerForDelegate(DelegateName)".

##### Return

**TRUE:** The device is connected.

**FALSE:** connected device not found.

#### 3.1.2. StartPPG

##### Prototype

```
public static extern bool StartPPG(IntPtr connCB);
```

##### Parameter

None.

##### Description





The first API for "Hot Plug" unsupported environment. (e.g. LabView) Same as InitPPG() with connCB as NULL.

### Return

**TRUE:** Success.

**FALSE:** Fail.

## 3.1.3. IsConnected

### Prototype

```
public static extern bool IsConnected();
```

### Parameter

None.

### Description

This API checks whether HRS device is connected.

### Return

**TRUE:** Success.

**FALSE:** Fail.

## 3.1.4. ClosePPG

### Prototype

```
public static extern void ClosePPG();
```

### Parameter

None.

### Description

Last calling API when system exiting for releasing resource.

### Return

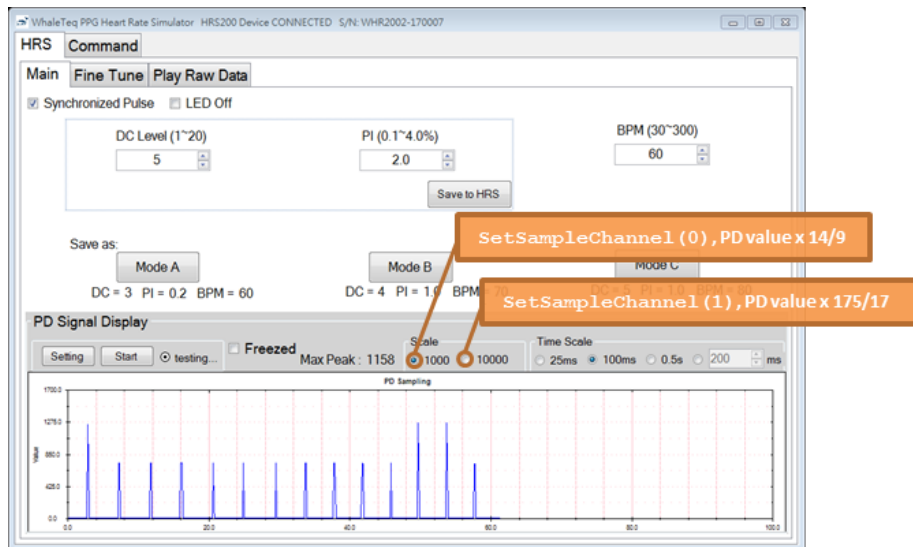
None.

## 3.2. PD Sampling Related Functions

The PD range of HRS is enhanced in July 2017. If there are different version of HRSs in hand, or would like to get the same PD value with HRS AP, please

follow below implementation detail:

- Normal PD range (**IsWiderPDRange = FALSE**)
  - Get PD value without extra adjustment.
- Wider PD range (**IsWiderPDRange = TRUE**)
  - For scale 1000: call **SetSampleChannel(0)**, PD value x 14/9
  - For scale 10000: call **SetSampleChannel(1)**, PD value x 175/17



### 3.2.1. IsWiderPDRange

#### Prototype

```
public static extern bool IsWiderPDRange();
```

#### Parameter

None.

#### Description

The API checks whether the wider PD range is supported on connected HRS device.

#### Return

**TRUE**: "Wider PD range" is **supported** on the connected HRS device.

**FALSE**: "Wider PD range" is **unsupported** on the connected HRS device.



### 3.2.2. RegisterPDSamplingCB

#### Prototype

```
public static extern void RegisterPDSamplingCB(IntPtr cb);
```

#### Parameter

**cb:**

Callback function for real time PD Sampling.

#### Description

This API register callback to get real time PD sampling data. About 0.00003 second (30us) interval between each 2 sampling data.

Before using this function, should call "delegate void PDSamplingCBDelegate (int ValueNum, IntPtr ValueArray)" and use function pointer of delegate from "Marshal.GetFunctionPointerForDelegate (DelegateName)".

#### Return

None.

### 3.2.3. RegisterPDSamplingPeakCB

#### Prototype

```
public static extern void RegisterPDSamplingPeakCB(IntPtr cb,  
double cycleSecond);
```

#### Parameter

**cb:**

Callback function for real time PD Sampling Peak value.

**cycleSecond:**

The duration of each cycle Maximum Peak detection.

#### Description

This API register callback to get real time PD sampling Maximum Peak value in cycle seconds.

Before using this function, should call "delegate void PDSamplingPeakCBDelegate (int PeakValue)" and use function pointer of delegate from "Marshal.GetFunctionPointerForDelegate(DelegateName)".

#### Return

None.



### 3.2.4. SetSampleSetting

#### Prototype

```
public static extern bool SetSampleSetting(bool Enable, bool HighResolution);
```

#### Parameter

##### Enable:

**TRUE:** Enable HRS PD Sampling. HRS receives real time lightness change from DUT (e.g. PPG module, smart watch) in this setting.

**FALSE:** Disable HRS PD Sampling. HRS will not receive lightness change from DUT (e.g. PPG module, smart watch) in the setting.

##### HighResolution:

**TRUE:** Set HRS as "HighResolution" mode. It only can be set when LED Off (by "SendLedOff()"). In "HighResolution" mode, HRS will return high resolution PD Sampling value.

**FALSE:** Set HRS as normal resolution mode.

#### Description

This API control PD Sampling functionality.

#### Return

**TRUE:** Success.

**FALSE:** Fail.

### 3.2.5. SetSampleChannel

#### Prototype

```
public static extern void SetSampleChannel(int Channel);
```

#### Parameter

**0:** Set to get Channel 1 PD value.

**1:** Set to get Channel 2 PD value. This channel could receive higher LED brightness. (Wider PD range.)

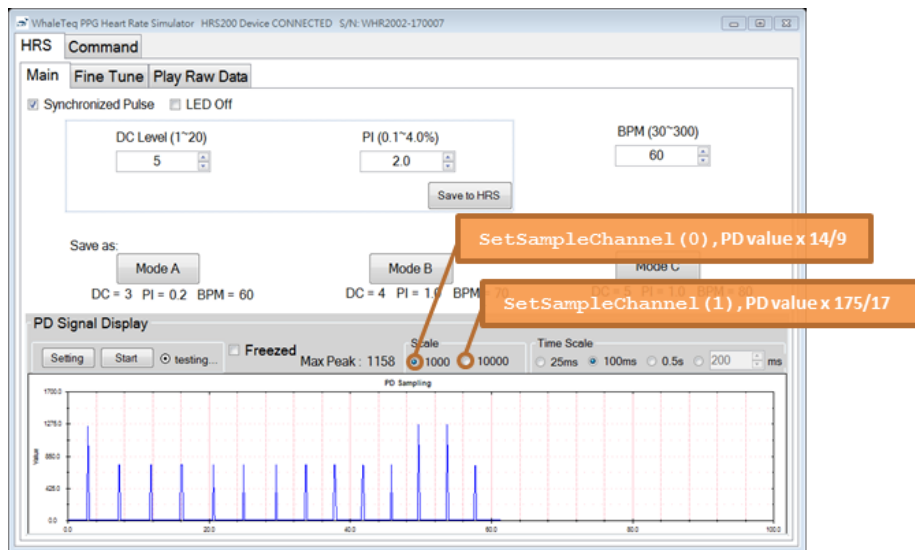
#### Description

Call this API to select which channel of PD sampling is set.



## Return

None



### 3.2.6. SendLedOff

#### Prototype

```
public static extern bool SendLedOff();
```

#### Parameter

None.

#### Description

When calling this API, HRS will turn off LED (testing signal sending) functionality.

#### Return

**TRUE:** Success.

**FALSE:** Fail.

## 3.3. Main Mode

### 3.3.1. SendCommand

#### Prototype

```
public static extern bool SendCommand(int DCLevel, float PI,  
int HR, bool SyncPD);
```



## Parameter

**DCLevel**: Setting DC Level. Range for this parameter is 1~20.

**PI**: Setting PI value. Range for PI value changes corresponding to DCLevel.

- Maximum PI value =  $20 / \text{DCLevel}$ .
- Minimum PI value =  $3 / (5 * \text{DCLevel})$

**HR**: Setting simulated Heart Rate value. Range for this parameter is 30~300 BPM (Beat Per Minutes).

**SyncPD**:

**TRUE**: Testing signal sending synchronized with PD receiving signal.

**FALSE**: HRS sends continuous testing signal.

## Description

This API sends testing signal with specified DC Level, PI Value and Heart Rate (BPM).

## Return

**TRUE**: Success.

**FALSE**: Fail.

## 3.4. Fine Tune Mode (Only supported on HRS200)

### 3.4.1. OutputFunction (HRS200 Only)

#### Prototype

```
public enum OutputFunction {  
  
    Output_Off = 0,  
  
    Output_Sine = 1,  
  
    Output_Triangle = 2,  
  
    Output_Square = 3,  
  
    Output_PPG = 15  
    Output_InvPPG = 16  
  
};
```



## Description

Enumeration for testing signal waveform type. This enum is used in SendFineTune function.

### 3.4.2. RegisterTimeAmpCB (HRS200 Only)

#### Prototype

```
public static extern void RegisterTimeAmpCB(IntPtr cb);
```

#### Parameter

**cb:**

Callback function for real time output signal amplitude.

#### Description

Register callback to get real time output signal data. About 0.006 second (6ms) interval between each callback be triggered.

Before using this function, should call "delegate void TimeAmplitudeCBDelegate (double time, double amp)" and use function pointer of delegate from "Marshal.GetFunctionPointerForDelegate (DelegateName)".

#### Return

None.

### 3.4.3. SendFineTune (HRS200 Only)

#### Prototype

```
public static extern bool SendFineTune(OutputFunction Waveform,  
int NoiseHz, int DC, int AC, int BPM, bool SyncPD);
```

#### Parameter

**Waveform:** Testing signal waveform type. For details please reference OutputFunction.

**NoiseHz:** This parameter sets the frequency of noise. Noise value can be 0 (without Noise), 50, 60 or 1k Hz.

**DC:** This parameter sets the DC value of simulated light. Range from 100 ~2500 uV.

**AC:** This parameter sets the AC value of simulated light. Range from 75



~2500, which means 0.75~25uV.

**BPM:** This parameter sets the simulated BPM value. Range from 30 ~300 BPM.

**SyncPD:**

**TRUE:** Testing signal sending synchronized with PD receiving signal.

**FALSE:** HRS sends continuous testing signal.

### Description

This API sends testing signal with specified Waveform Type, Noise, DC Value, AC Value and Heart Rate (BPM).

### Return

**TRUE:** Success.

**FALSE:** HRS device not connected or Module HRS100 connected.

## 3.4.4. SendFineTuneNoiseAmp (HRS200 Only)

### Prototype

```
public static extern void SendFineTuneNoiseAmp(double Amp);
```

### Parameter

**Amp:**

Setting the amplitude of noise in "Fine Tune" mode. Default value of this parameter is 0.1uV. Range from 0.01~0.2uV.

### Description

This API sets the noise amplitude in "Fine Tune" mode.

### Return

None.

## 3.5. Play Raw Data (Only supported on HRS200)

### 3.5.1. LoadDynamic (HRS200 Only)

### Prototype

```
public static extern int LoadDynamic(string File, ref double  
SampleRate, ref double Max, ref double Min, ref double StopTime);
```





## Parameter

**File:** Full path including file name of "Raw Data File".

**SampleRate:** The Sample Rate in "Raw Data File".

**Max:** The maximum raw data value in "Raw Data File".

**Min:** The minimum raw data value in "Raw Data File".

**StopTime:** The total sampling seconds in "Raw Data File".

## Description

This API load "Raw Data File" for dynamic playing. When succeed, it would return the sample rate, maximum value, minimum value and total time with pointer parameters.

## Return

>0: succeed, the number of sample data loaded.

-1: Open file failed.

-2: Can't solve sample rate in Line 1.

-3: Can't solve sample number in Line 2.

-4: Raw data file is too large to fit the memory.

-5: Unsupported API for HRS100 Module

### 3.5.2. GetDynamicArray (HRS200 Only)

#### Prototype

```
public static extern IntPtr GetDynamicArray(ref int num);
```

#### Parameter

**num:** The number of raw data values in ValueArray

#### Description

This API get loaded Raw Data value array.

#### Return

The raw data values array in double type.

### 3.5.3. PlayDynamic (HRS200 Only)

#### Prototype

```
public static extern bool PlayDynamic(double F, int Offset,  
IntPtr stopCB, IntPtr playCB, bool SyncPD);
```



## Parameter

**F:** The same as Gain. Default value is 4095/(Max-Min).

**offset:** Default value is "-Min" to normalize the raw data value range.

**stopCB:** SDK supports "Loop" playing as default mode with stopCB as "(IntPtr)0". Specified "informing finish callback" to implement "non-Loop" playing mode.

**playCB:** Specified real time playing data callback.

**SyncPD:**

**TRUE:** Testing signal sending synchronized with PD receiving signal.

**FALSE:** HRS sends continuous testing signal.

## Description

This API start playing the loaded Raw Data.

Before using this function, should call "delegate void PlayStopCBDelegate(void)", "PlayDynamicCBDelegate(double time, unsigned short DC, unsigned short AC)" and use function pointer of delegate from "Marshal.GetFunctionPointerForDelegate (DelegateName)".

## Return

**TRUE:** Success.

**FALSE:** Fail.

### 3.5.4. ResampleDynamic (HRS200 Only)

## Prototype

```
public static extern void ResampleDynamic(double newSampleRate,  
ref double stopTime);
```

## Parameter

**newSampleRate:** The sample rate value of resample.

**stopTime:** Updated totally playing time for Sample Rate changed.

## Description

This API changes the sample rate of loaded Raw Data for playing.

## Return

None.



## 4. Contact WhaleTeq

WHALETEQ Co., LTD

[service@whaleteq.com](mailto:service@whaleteq.com) | (O)+886 2 2550 1239

9F-1, No.104, Minguang W. Rd., Taipei City 10361, Taiwan