

MicroLifeDevice SDK (WBP Office Vascular- Windows)

Table of Contents

Chapter 1 Development Environment

Chapter 2 Connection Sequence of WBP_Office_Vascular

Chapter 3 APIs of WBP_Office_Vascular

Chapter 4 Class & Object of WBP_Office_Vascular

Chapter 5 Instruction of Demo App

Chapter 6 The description for each command of Demo App

Revise history

Date	Document Version	Description
2025/10/07	1.0	First release.
2026/09/29	1.1	Updated for SDK 2.0.7. ** Revised 3-1-14 Stop remote measurement. Other API corrections.

Chapter 1 Development Environment

This user manual serves as a quick guide to MicroLifeDeviceSDK / APIs and shows how to integrate into a Windows C# Demo App. Development Environment in the following

Compatible Development Tools	Microsoft Visual Studio 2019 (recommended)
Programming language:	C#
Target framework:	.NET Standard 2.0

Importing steps are described below.

1.1. First of all, add WBP_Office_Vascular.dll & Connection.dll into a development project.

1.2. Import class as below.

```
using WBP_Office_Vascular.Class;  
using Connection;  
using Connection.Class;
```

Chapter 2 Connection Sequence of WBP_Office_Vascular

The WBP_Office_Vascular object is applied managing the USB/Bluetooth communication.

- 2.1 Initiate WBP_Office_Vascular Object with API key and set connected/disconnected/received data delegation for WBP_Office_Vascular.

```
Office_Vascular = new WBP_Office_Vascular.WBP_Office_Vascular(key: "");
Office_Vascular.OnConnected += Office_Vascular_OnConnected;
Office_Vascular.OnDisConnected += Office_Vascular_OnDisConnected;
Office_Vascular.OnReceived += Office_Vascular_OnReceived;
```

- 2.2 Call InitDeviceWatcher to initialize WBP_Office_Vascular monitoring for USB status.

```
Office_Vascular.InitDeviceWatcher();
```

2.3 Connection

- 2.3-1 For USB, the OnConnected delegate will be called automatically when the device is connected to the computer via USB.

- 2.3-2 For BLE, Call ConnectWithBLE API with device's BLE id, the OnConnected delegate will be called automatically when the device is connected successfully.

- 2.4 When the data is transferred via USB/Bluetooth, OnReceived delegate will receive the parsed data.

- 2.5 The OnDisConnected delegate will be called when the device is disconnected.

The following is the sample code.

```
private static WBP_Office_Vascular.WBP_Office_Vascular vascular;
1 reference
public Form1()
{
    InitializeComponent();
    vascular = new WBP_Office_Vascular.WBP_Office_Vascular(key: "");
    vascular.OnConnected += Vascular_OnConnected;
    vascular.OnDisConnected += Vascular_OnDisConnected;
    vascular.OnReceived += Vascular_OnReceived;
    vascular.InitDeviceWatcher();
}

1 reference
private void Vascular_OnReceived(Command Command)
{
}

1 reference
private void Vascular_OnDisConnected()
{
}

1 reference
private void Vascular_OnConnected()
{
}
```

Chapter 3 APIs of WBP_Office_Vascular

This chapter provides a detailed explanation of each API's functionality and the significance of its parameters. Since the device supports remote measurement, certain commands—such as initiating a remote measurement—require waiting for the device's response after the command is issued. The API will only reply whether the command itself is successfully sent or not.

3-1.Send Command to device

	(bool Success, Command Command) SendMessage (Message message, Func<Command, bool> predicate = null, int retry = 3, bool ResetCommand = true, int timeout = 0);
Function	Transmit message to the Device. An asynchronous version with the same parameters is also available: Task<(bool Success, Command Command)> SendMessageAsync (...).
Return object	Success: Indicates whether the command was successfully written to the device. Command: A class containing parsed data and message which sent to the device, described in 4-1-1 . ** If the Success is true, and the Command.CMD is not NACK, the device has successfully received this command.
Parameter	Message: A class containing message which sent to the device. Predicate: If the command requires additional data, it will be provided here. Retry: Retry times ResetCommand: Whether to clear the last command timeout: timeout in milliseconds (0: use the default timeout of the command, for example dataRead 15000 ms, dataClear 10000 ms, userIDWrite 5000 ms, StopRemoteMeasurement 2000 ms. The wait restarts whenever data is received.) ** The contents of the above parameters are generated by the following API.

****The following commands will receive a response immediately after they are sent.**

3-1-1. Read all history data from BPM

Command	dataReadAll
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand. dataReadAll ());
Return Command Data Type.	The object type of Command.data is DRecord class, which is a class containing parsed BP data, described in 4-1-2 .

3-1-2. Read CBP/PWV memory data by index from BPM

Command	dataRead
---------	-----------------

API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.dataRead (int index, Enum.Enum_DataFormat format));
Return Command Data Type.	The object type of Command.data is WaveData class, which is a class containing parsed wave data, described in 4-1-8.
Parameter	index: Index of the history data (MData.Index) **To retrieve the data index of waveform information, use the GetCBPWaveIndex() and GetABIWaveIndex() methods from the DRecord class for verification format: Wave data sampling frequency format, described in 4-2-5

3-1-3. Clear the all history data of the BPM

Command	dataClear
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.dataClear());
Return Command Data Type.	None. ** If the Success is true, and the Command.CMD is not NACK, the device has successfully received this command.

3-1-4. Read user ID and version data from BPM.

Command	userInfoRead
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.userInfoRead());
Return Command Data Type.	The object type of Command.data is UserInfo class, which is a class containing parsed user's information data, described in 4-1-9.

3-1-5. Write a new user ID to BPM.

Command	userIDWrite
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.userIDWrite(string ID));
Return Command Data Type.	None. ** If the Success is true, and the Command.CMD is not NACK, the device has successfully received this command.
Parameter	ID: User ID string to be written

3-1-6. Read BPM setting values from BPM.

Command	settingRead_BPM
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.settingRead_BPM());

Return Command Data Type.	The object type of Command.data is Setting_BPM class, which is a class containing parsed BPM setting data, described in 4-1-10. **If Protocol ID is less than or equal to V1.0.2 ✖RestTime of ABI off-line test, it's fixed 60 seconds. ✖IntervalTime of ABI off-line test, it's fixed 60 seconds. ✖AutoMeasureNumber of ABI off-line/on-line test, it's fixed 2 times.
---------------------------	---

3-1-7. Write BPM setting values to BPM.

Command	settingWrite_BPM
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.settingWrite_BPM(Setting_BPM setting));
Return Command Data Type.	None. ** If the Success is true, and the Command.CMD is not NACK, the device has successfully received this command.
Parameter	setting: BPM Setting to be written, described in 4-1-10 **Note: Below are illegal settings, the SDK throws WrongSettingException (namespace WBP_Office_Vascular.VascularException) and the command is not sent. (1) AutoMeasureNumber=1 or 2 and SW_AVG_no_include_first=1 (2) AutoMeasureNumber=1 and SW_AFib =1 (3) Height <120, Height>210 (4) AutoMeasureNumber <1 or >6 (5) RestTime <1 or >300 (6) IntervalTime <1 or >600 **If Protocol ID is less than or equal to V1.0.2 ✖RestTime of ABI off-line test, it's fixed 60 seconds. ✖IntervalTime of ABI off-line test, it's fixed 60 seconds. ✖AutoMeasureNumber of ABI off-line/on-line test, it's fixed 2 times.

3-1-8. Read device ID and info from BPM.

Command	deviceIDRead
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.deviceIDRead());
Return Command Data Type.	The object type of Command.data is DeviceID_and_Info class, which is a class containing parsed Device ID & info data, described in 4-1-11.

3-1-9. Read device Time from BPM.

Command	timeRead
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.timeRead ());
Return Command Data Type.	The object type of Command.data is DateTime.

3-1-10. Write device Time to BPM.

Command	timeWrite
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.timeWrite(DateTime dateTime));
Return Command Data Type.	None. ** If the Success is true, and the Command.CMD is not NACK, the device has successfully received this command.
Parameter	dateTime: DateTime to be written.

3-1-11. Read BPM function setting value from BPM.

Command	settingRead_BPM_Function
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.settingRead_BPM_Function ());
Return Command Data Type.	The object type of Command.data is Setting_BPM_Function class, which is a class containing parsed BPM function data. These values are set at the factory and are only available for reading and cannot be rewrite, described in 4-1-12 .

3-1-12. Read BTmodule name from BPM.

Command	bluetoothnameRead
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.bluetoothnameRead());
Return Command Data Type.	The object type of Command.data is string.

**** The following commands should be waited for a follow-up response from the BPM after they are sent, and the responding data will be received from OnReceived delegate.**

3-1-13. Start remote measurement

Command	StartRemoteMeasurement
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand.StartRemoteMeasurement (Enum.Enum_HistoryMode mode,

	Enum.Enum_DataFormat dataFormat));
Parameter	mode: A Enum of HistoryMode, described in 4-2-1 dataFormat: A Enum of format, described in 4-2-5. **This parameter dataFormat determines whether waveform data is automatically returned after the measurement is completed. However, it is recommended to set StartRemoteMeasurement to Enum_DataFormat.No_Raw_Data, and then retrieve the waveform data using dataRead after the measurement is finished. **Full_PWV_raw_data (1024Hz) is not supported by StartRemoteMeasurement; the SDK throws an exception.
received from OnReceived delegate.	command.CMD: StartRemoteMeasurement command.Data: The returned object is a ValueTuple (HistoryMode, DataFormat) of the input mode and data format.

3-1-14. Stop remote measurement

Command	StopRemoteMeasurement
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand. StopRemoteMeasurement ());
Parameter	none.
received from OnReceived delegate.	command.CMD: StopRemoteMeasurement command.Data: The object type of Command.data is Enum_StopResult, described in 4-2-11. MeasurementStopped: the measurement in progress was stopped. NotMeasuring: no measurement was in progress (the device replied NACK). The command is still treated as successful, and Command.CMD stays StopRemoteMeasurement.

****The following Commands are sent by the device actively, and the software does not need to send any Command.**

3-1-15. RemoteStatus

received from OnReceived delegate.	command.CMD: RemoteStatus command.Data: The object type of Command.data is Remote_Measurement_Status class, which is a class containing remote measurement status data, described in 4-1-13.
---	--

3-1-16. MeasurementResults

received from OnReceived delegate.	command.CMD: MeasurementResults command.Data: The object type of Command.data is MData_Measurement class, which is a class containing remote measurement result data, described in 4-1-14 .
---	---

****The following Commands can only be used when the Device's Protocol ID is greater than or equal to V1.0.3 . For Protocol ID, please refer to 3-1-4 and 4-1-3.**

3-1-17. Read device SN

Command	deviceSNRead
API	SendMessage(WBP_Office_Vascular.SendMessage.ToCommand. deviceSNRead ());
Return Command Data Type.	command.CMD: deviceSNRead command.Data: The object type of Command.data is String.

Chapter 4 Class & Object of WBP_Office_Vascular

4-1 Class

4-1-1.Command Class

Name:	Command
Definition	A class containing parsed data and message sent to the device.
members	byte CMD: Record the current command, if it is NACK(0x91), it means the transmission failed. byte Device: Record the device. object Data: parsed data

4-1-2. DRecord Class

Name:	DRecord
Definition	A class containing parsed BP data.
members	Enum_HistoryMode HistoryMode: A Enum of HistoryMode, described in 4-2-1. List<MData> MDatas: A list of BPData, described in 4-1-3 HistoryBPMSetting HistoryBPMSetting: A class containing parsed BPM setting, described in 4-1-5 List<int> GetCBPWaveIndex(): This API returns the data index that include CBPWave. List<int> GetABIWaveIndex(): This API returns the data index that include ABIWave.

4-1-3.MData Class

Name:	MData
Definition	A class containing parsed BP data.
members	int Index: Data Index. Data R: The BP data of right side, described in 4-1-4 Data L: The BP data of left side, described in 4-1-4 ** R or L is null when that side has no data (see "L: null" in the example of 6-1). Int CurrentMeasureNumber: Current measure number int Hr: Pulse rate (unit:beats/min) ** If Heart rate>=240 or <30, note HI(Heart rate>=240), LO(Heart rate <30) in Code. Hr = 0 is not marked as LO. double ABI: The value of ABI. int PTT: pulse transit time (unit:ms). int PWV: The value of PWV. int Height: The value of Height.

	InfoStatus InfoStatus: A class containing parsed Info status, described in 4-1-6 DateTimeOffset Time: Record time ABI_Error ABI_Error: A Enum of ABI Error, described in 4-2-2. ** If PWV is out of 800~2400 while neither side has an error, the SDK reports ABI_Error as E25. If ABI_Error is E21 or E23, the SDK sets PWV and PTT to 0. bool MData_Enable_M_PWV: true: measure with PWV detection. Dictionary<string, List<string>> Code: A Dictionary of BP Code The keys are categorized as the Status.Side of variables L and R, as well as "All". When the key is the Status.Side of variables L and R, value record Code, please refer to Section 4-3-2. When the key is "All", value records ABI Error and whether heartrate signals show LO or HI conditions. Please refer to Section 4-3-1.
--	--

4-1-4. Data Class

Name:	Data
Definition	A class containing parsed left or right side BP data.
members	Int Condition: Describe the measurement mode, described in 4-4-1 Int Systole: The value of systole. Int DiaStole: The value of diastole. ** If the Status.Error is true, these two values are 0 Int Formula_MAP: The value of Formula_MAP. ** Valid for Protocol ID version V1.0.3 or higher. Int Oscillometric_MAP: The value of Oscillometric_MAP. ** Valid for Protocol ID version V1.0.3 or higher. Int MAP: The value of MAP. ** Supported across all versions; shares the same value as Oscillometric_MAP. Status Status: A class of Status, described in 4-1-7 Int mean_CBP_data: The value of mean CBP data. Int PVR_length: The value of PVR length. bool CBP_Error: true: measure with CBP error. Int CSBP: The value of CSBP. Int CDIA: The value of CDIA. Int CPP: The value of CPP. bool LowBattery: true: occurred low battery. List<string> Code: A list of code described in 4-3-2.

4-1-5. HistoryBPMSSetting Class

Name:	HistoryBPMSetting
Definition	A class containing parsed BPM'setting data.
members	bool Enable_AFIB: true: measure with Afib detection. bool Enable_CBP: true: measure with CBP detection. bool Enable_ABI: true: measure with ABI detection. bool Enable_PWV: true: measure with PWV detection. bool Enable_ExcludeFirst: true: measure with D-1 function (calculate average exclude 1st data).

4-1-6. InfoStatus Class

Name:	InfoStatus
Definition	A class containing parsed info status data.
members	bool Left_IAD_Warming: true: IAD warming left side.. bool Right_IAD_Warming: true: IAD warming right side. bool AFIB: true: detection AFib. bool LowBattery: true: occurred low battery. bool PWV_Error: Deprecated

4-1-7. Status Class

Name:	Status
Definition	A class containing parsed status data.
members	bool CBP_Error: true: measure with CBP error. bool Error: true: error measurement. Side : A Enum of side, described in 4-2-3

4-1-8. WaveData Class

Name:	WaveData
Definition	A class containing parsed wave data.
members	Enum_WaveType WaveType: A Enum of Wave Type, described in 4-2-4 . Enum_DataFormat Format: A Enum of data format, described in 4-2-5 . List<double>[] Data: Wave data lists (array of 2): SingleCBP uses Data[0] only; ABI uses Data[0] and Data[1] for the two channels. List< int> CalCBP, CalCBP Data list

4-1-9. UserInfo Class

Name:	UserInfo
Definition	A class containing parsed user's information data.
members	string UserID: User ID, maximum 30 characters

	string FMVersion: FW version in BPM, send the ASCII code DateTimeOffset FMDate: The release date of firmware. int MaxMemory: Maximum of memory data can be saved for every user. bool PWV: true: PWV Enable bool ABI: true: ABI Enable bool CBP: true: CBP Enable. bool Afib: true: AFIB Enable float BatteryVoltage: Voltage of the device battery. string ProtocolID: Protocol ID, The format is strictly defined as VN.N.N (where N represents a digit), for example V1.0.3. Enum_arrhythmiaDisplay arrhythmiaDisplay: Display arrhythmia name in software. It may be one of NONE, IHB, PAD described in 4-2-6
--	---

4-1-10. Setting_BPM Class

Name:	Setting_BPM
Definition	A class containing parsed BPM setting data.
members	int Height: Height Value. Enum_HI_infPressure HI_infPressure: Highest inflation pressure of Auto mode Valid values: Auto (0, not setting), 160,180, 200, 220, 240, described in 4-2-12 int RestTime: Rest time of auto mode Start countdown base on rest time before 1st measurement in auto mode. Range: 1~300 seconds. **If Protocol ID is less than or equal to V1.0.2 ※RestTime of ABI off-line test, it's fixed 60 seconds. int IntervalTime: Interval time of auto mode. Start countdown base on interval time before 2nd~6th measurement in auto mode. Range: 1~600 seconds **If Protocol ID is less than or equal to V1.0.2 ※IntervalTime of ABI off-line test, it's fixed 60 seconds. int AutoMeasureNumber: It's number of measurements in auto mode. Range: 1~6 times **If Protocol ID is less than or equal to V1.0.2 ※AutoMeasureNumber of ABI off-line/on-line test, it's fixed 2 times.. bool SW_hide: true: Don't show readings during rest time in auto mode. false: Show readings during rest time in auto mode. bool SW_SEL_silent: true: Beeper disabled. false: Beeper enabled. bool SW_AVG_no_include_first: true: Average is not including first memory data.

	false: Average is including first memory data. bool SW_CBP: true: CBP measurement enabled. false: CBP measurement disabled. bool SW_AFib: true: AFib measurement enabled. false: AFib measurement disabled. Enum_AMPM SW_AMPM: A Enum of AMPM setting, described in 4-2-8. o24: only 24hr o12: select 24hr or 12hr by UI. Enum_PressureUnit SW_Kpa: A Enum of PressureUnit, described in 4-2-10. Kpa: Pressure unit: Kpa. mmHg: Pressure unit: mmHg. bool SW_SkipRest: true: skip rest time. false: keep rest time. **Note: Below are illegal settings, the SDK throws WrongSettingException (namespace WBP_Office_Vascular.VascularException) and the command is not sent. (1) AutoMeasureNumber=1 or 2 and SW_AVG_no_include_first=1 (2) AutoMeasureNumber=1 and SW_AFib =1 (3) Height <120, Height>210 (4) AutoMeasureNumber <1 or >6 (5) RestTime <1 or >300 (6) IntervalTime <1 or >600
--	--

4-1-11. DeviceID_and_Info Class

Name:	DeviceID_and_Info
Definition	A class containing parsed Device ID and information data.
members	string ID: Device ID string Enum_ConnectType ConnectType: A Enum of ConnectType which would be one of the following, described in 4-2-7 BothOfUSBAndBT: Device support USB & BT. USBOnly: Device only support USB. int R_Mea_times: Total number of right side's measurements int L_Mea_times: Total number of left side's measurements int[] ErrorCountR: The array of the total number of right side's errors is recorded in the order of error1,error2,error3,error5,errorF. int[] ErrorCountL: The array of the total number of left side's errors is recorded in the order of error1,error2,error3,error5,errorF.

4-1-12. Setting_BPM_Function Class

Name:	Setting_BPM_Function
Definition	A class containing parsed BPM function setting data.
members	bool Fun_SEL_ABI: true: Enable ABI function. false: Disable ABI function. bool Fun_SEL_PWV: true: Enable PWV function. false: Disable PWV function. bool Fun_SEL_CBP: true: Enable CBP function. false: Disable CBP function. bool Fun_SEL_AFIB: true: AFIB algorithm ON. false: AFIB algorithm OFF. bool Fun_SEL_Bluetooth: true: Bluetooth function ON. false: Bluetooth function OFF. bool Fun_SEL_UnitKpa: true: select Kpa or mmHg by UI. false: only mmHg. Enum_AMPM Fun_SEL_AMPM: A Enum of AMPM setting, described in 4-2-8. o24: only 24hr o12: select 24hr or 12hr by UI bool Fun_SEL_RS232: true: LabView data transmission ON. false: LabView data transmission OFF.

4-1-13. Remote_Measurement_Status Class

Name:	Remote_Measurement_Status
Definition	A class containing parsed remote measurement status data.
members	Enum_Status Status: A Enum of remote status which would be one of the following, Wait_countdown_for_next_measurement/ Start_BP_measurement / Manual_press_IO_to_stop_measurement/ EmptyBattery. described in 4-2-9, int Measurement_Number: Send current measurement number in auto mode. int Measurement_Total: Send total measurement number in auto mode. int Countdown: Send current countdown time in auto mode.

	int TotalMeasurementTime: Send total measurement time (seconds) in auto mode. Total measurement are count between 1st measurement to last measurement. (exclude rest time)
--	--

4-1-14. MData_Measurement Class

Name:	MData_Measurement
Definition	A class containing parsed remote measurement result data.
members	int History_Measurement_Times: The history measurement times store in memory. int Current_Measurement_Times: Send current measurement times to APP. MData MData: BP Data. From device, described in 4-1-3 . HistoryBPMSetting BPMSetting: A class containing parsed BPM setting, described in 4-1-5. Enum_HistoryMode Mode: A Enum of HistoryMode, described in 4-2-1 . bool HasCBPWave: Deprecated bool HasABIWave: Deprecated

4-2 Enum

4-2-1 Enum_HistoryMode

Enum_HistoryMode	
0	RightArm
1	LeftArm
2	RightArmAndLeftArm
3	RightArmAndLeftLeg
4	RightArmAndRightLeg
5	LeftArmAndRightLeg
6	LeftArmAndLeftLeg

4-2-2 ABI_Error

ABI_Error		
0	None	
21	E21	signals is too weak
23	E23	no pressure in the cuff
25	E25	Abnormal result(PWV)
254	N/A	Not start ABI wave detection. Because occurs error with the ankle or brachial side, device stops the ABI wave detections.

4-2-3 Side

Side	
1	LeftArm
2	RightArm
4	LeftLeg
8	RightLeg

4-2-4 Enum_WaveType

Enum_WaveType	
1	SingleCBP
3	ABI

4-2-5 Enum_DataFormat

DataFormat		Note
0	No_Raw_Data	No CBP raw data
1	Low_resolution_CBP_data	sampling rate =16Hz
2	Full_CBP_raw_data	sampling rate=256Hz, CBP full

		resolution
3	Full_PWV_raw_data	sampling rate=1024Hz, PWV full resolution (dataRead only; not supported by StartRemoteMeasurement)

4-2-6 Enum_arrhythmiaDisplay

arrhythmiaDisplay	
0	NONE
1	IHB
2	PAD

4-2-7 Enum_ConnectType

ConnectType	
0x55	USBOnly
0x42	BothOfUSBAndBT

4-2-8 Enum_AMPM

AMPM	
0	o24
1	o12

4-2-9 Enum_Status

Status		Note
0x01	Wait_countdown_for_next_measurement	Wait countdown for next measurement
0x02	Start_BP_measurement	Start BP measurement
0x04	Manual_press_IO_to_stop_measurement	Manual press IO to stop measurement
0x05	EmptyBattery	Empty Battery

4-2-10 Enum_PressureUnit

PressureUnit	
0x00	mmHg
0x01	Kpa

4-2-11 Enum_StopResult

Enum_StopResult		Note
0x00	MeasurementStopped	The measurement in progress was

		stopped (device replied ACK).
0x91	NotMeasuring	No measurement was in progress (device replied NACK). The command is still treated as successful.

4-2-12 Enum_HI_infPressure

Enum_HI_infPressure	
0	Auto (not setting)
160	_160
180	_180
200	_200
220	_220
240	_240

4-3 list

4-3-1 Code for MData Class

Value	Description
LO	Hr<30
HI	Hr>=240
ER 21	Signals is too weak
ER 23	No pressure in the cuff
ER 25	Abnormal result

4-3-2 Code for Data Class

Value	Description
4	Low Battery, occurred warming battery.
5	Empty Battery, The batteries need to be charged.
ER 1	Signals is too weak
ER 2	Error signal
ER 3	No pressure in the cuff
ER 5	Abnormal result
ER F	reach of maximum 30 min of measurements
ER 11	Signal too weak during central blood pressure measurement
ER 12	Error signal during central blood pressure measurement
ER 13	Cuff pressure errors during central blood pressure measurement
ER 15	Abnormal result of central blood pressure reading

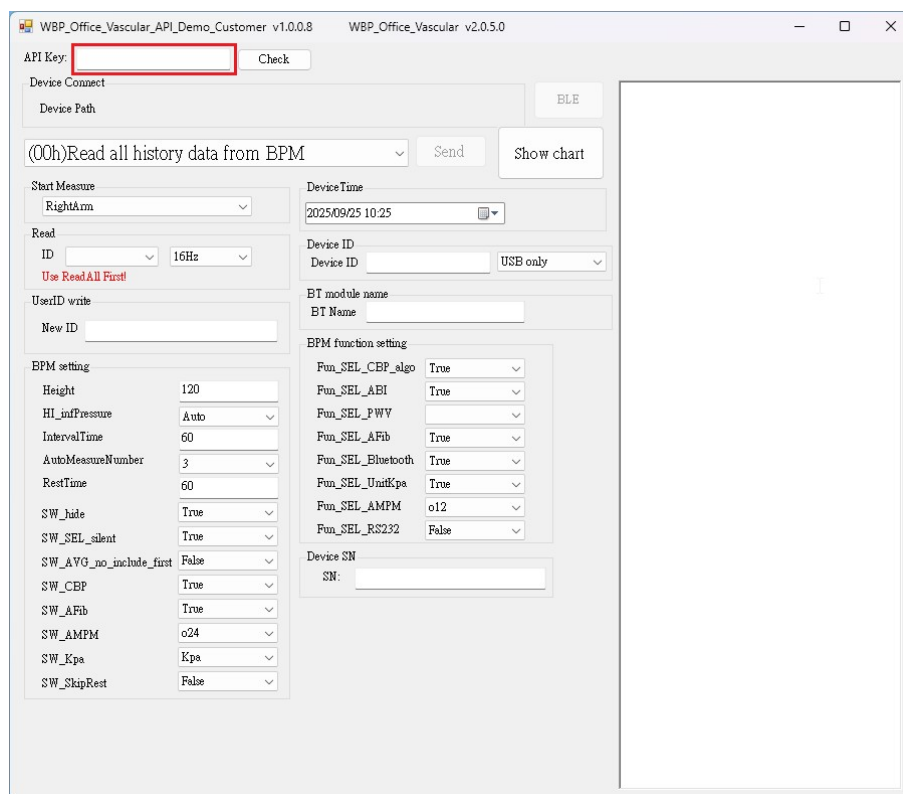
4-4 Others

4-4-1 **R_Condition 、L_Condition**

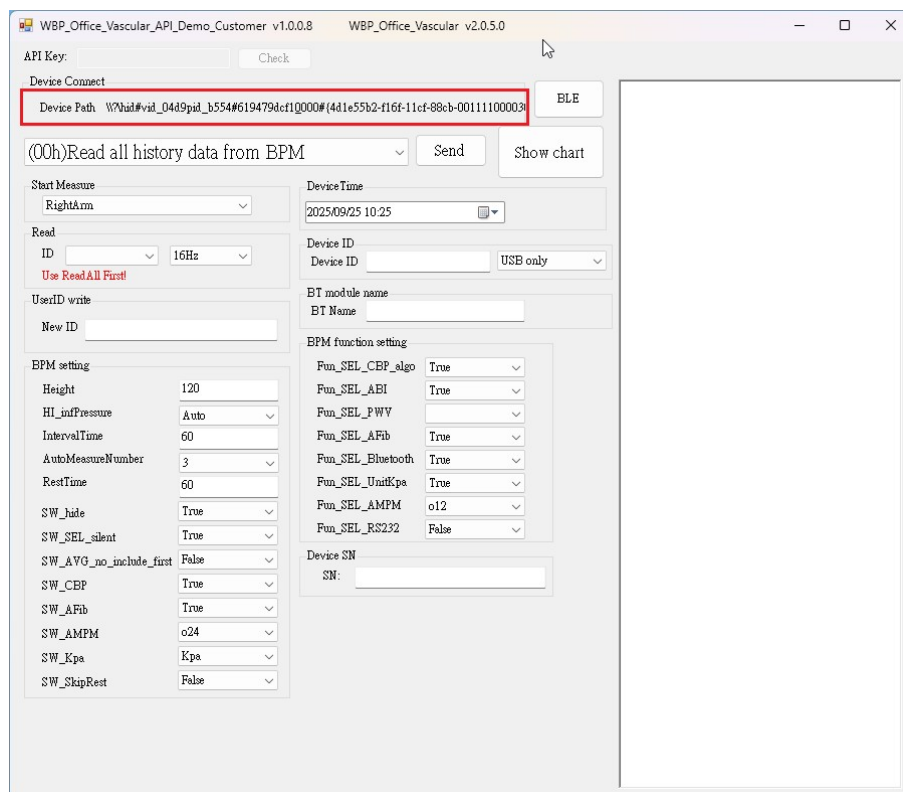
Value	Description
0	Right Arm
1	Left Arm
2	Right Arm & Left Arm
3	Right Arm & Left Leg
4	Right Arm & Right Leg
5	Left Arm & Right Leg
6	Left Arm & Left Leg
254	No data

Chapter 5 Instruction of Demo App

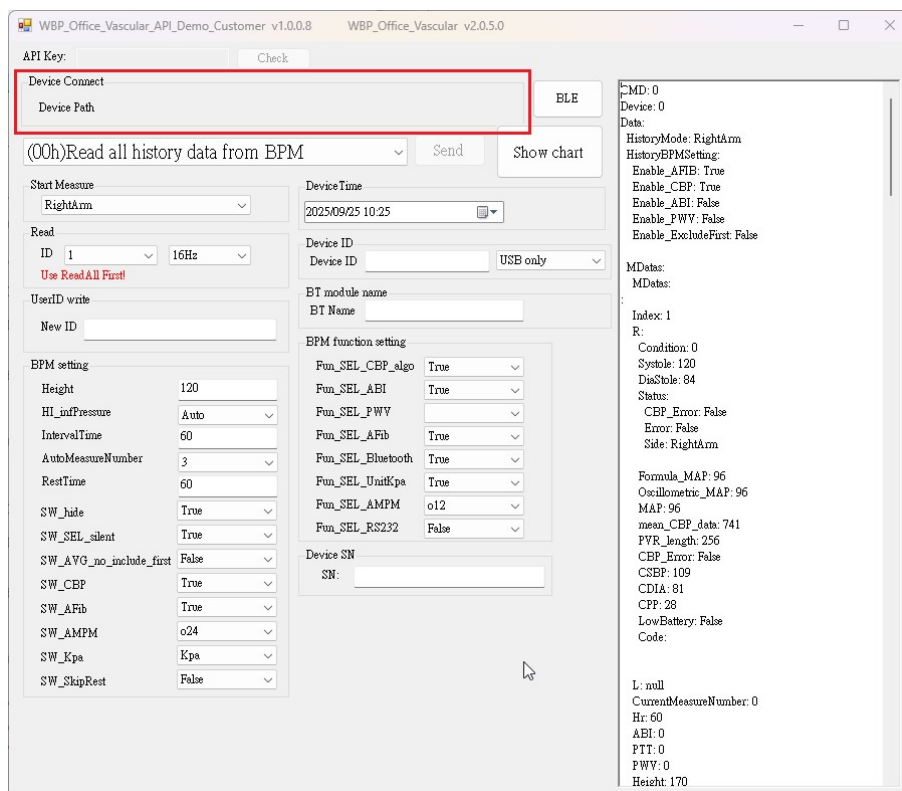
5-1. Input the API Key on the API Key textbox and click “Check” button to active the demo.



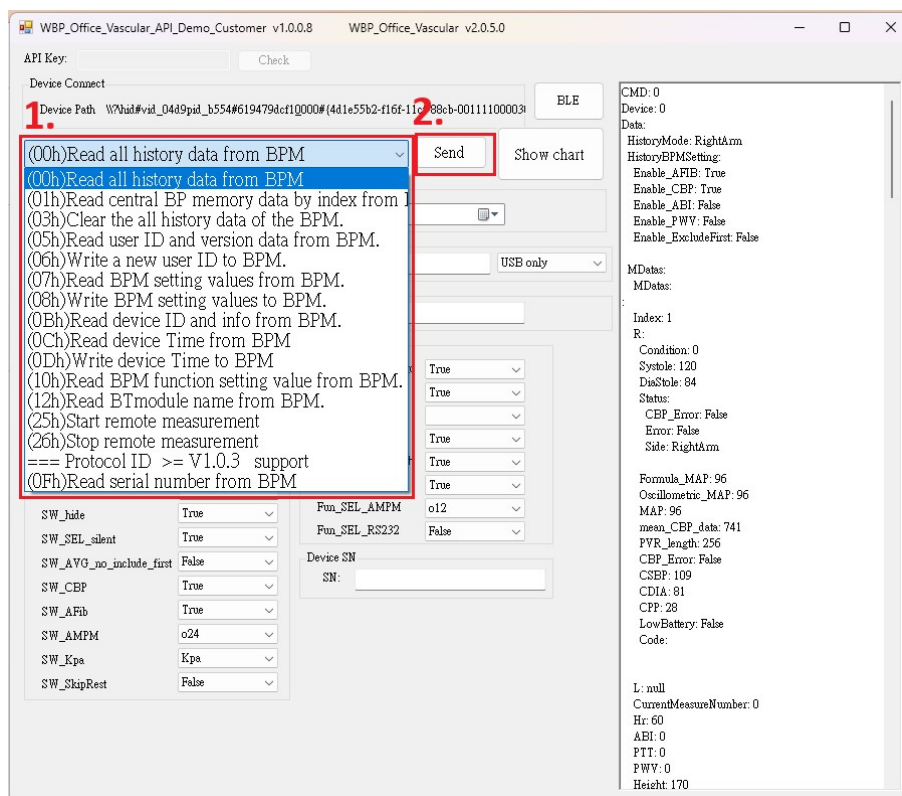
5-2. If the device is connected, Device path will be displayed.



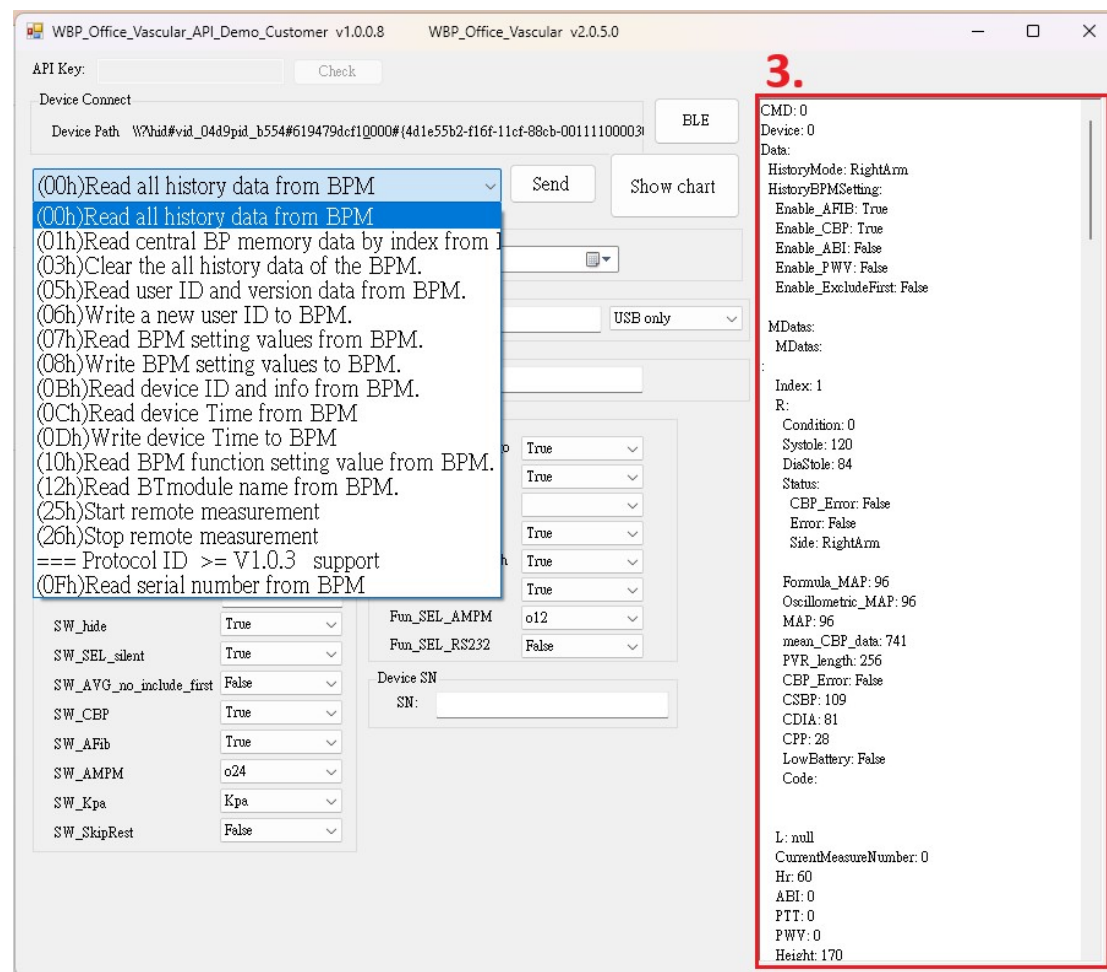
Similarly, if the device is disconnected in anytime, The device path will not be displayed.



5-3. Select the command on the combobox and click the “Send” button to send it.



The output will be displayed in the textbox on the right



Chapter 6 The description for each command of Demo App

6-1. (00h)Read all history data from BPM

The results of the data received are as follows

CMD: 0

Device: 67

Data:

HistoryMode: RightArm

HistoryBPMSetting:

Enable_AFIB: True

Enable_CBP: True

Enable_ABI: False

Enable_PWV: False

Enable_ExcludeFirst: False

MData:

MData:

:

Index: 1

R:

Condition: 0

Systole: 120

DiaStole: 84

Status:

CBP_Error: False

Error: False

Side: RightArm

Formula_MAP: 9

Oscillometric_MAP: 96

MAP: 96

mean_CBP_data: 741

PVR_length: 256

CBP_Error: False

CSBP: 109

CDIA: 81

CPP: 28

LowBattery: False

Code:

L: null
CurrentMeasureNumber: 0
Hr: 60
ABI: 0
PTT: 0
PWV: 0
Height: 170
InfoStatus:
 Left_IAD_Warming: False
 Right_IAD_Warming: False
 AFIB: False
 LowBattery: False
 PWV_Error: False

Time: 2018-07-02 13:02:00
ABI_Error: None
MData_Enable_M_PWV: False
Code:
 All:

The description is as follows

(01)CMD:

current command

(02) Device:

The device code is fixed to 0x43 for Vascular.

(03) Data:

BP data include L / R side & BPMSetting & Mode.

(04) HistoryMode:

The modes used to record measurements will be one of the following: RightArm, LeftArm, RightArmAndLeftArm, RightArmAndLeftLeg, RightArmAndRightLeg, LeftArmAndRightLeg, LeftArmAndLeftLeg

(05) HistoryBPMSetting:

BPM setting

(06) Enable_AFIB

true: measure with Afib detection.

(07) Enable_CBP

true: measure with CBP detection.

(08) [Enable_ABI](#)

true: measure with ABI detection.

(09) [Enable_PWV](#)

true: measure with PWV detection.

(10) [Enable_ExcludeFirst](#)

true: measure with D-1 function (calculate average exclude 1st data).

(11) [MData](#)

parsed BP data include L / R side.

(12) [Index](#)

Data index.

(13) [R](#)

Data from right side.

(14) [Condition](#)

For the current measurement mode, please refer to 4-4-1.

(15) [Systole](#)

The value of systole

(16) [DiaStole](#)

The value of diastole

(17) [Status](#)

Status data, please refer to 4-1-7

(18) [Formula_MAP](#)

The value of Formula MAP.

(19) [Oscillometric_MAP](#)

The value of Oscillometric MAP.

(20) [MAP](#)

The value of MAP, shares the same value as Oscillometric_MAP.

(21) [mean_CBP_data](#)

The value of mean CBP data.

(22) [PVR_length](#)

The value of PVR length.

(23) [CBP_Error](#)

true: measure with CBP error.

(24) [CSBP](#)

The value of CSBP.

(25) [CDIA](#)

The value of CDIA.

(26) [CPP](#)

The value of CPP.

(27) LowBattery

true: occurred low battery.

(28) Code (L / R side)

The results of the Code received are as follows

Value	Description
4	Low Battery, occurred warming battery.
5	Empty Battery, The batteries need to be charged.
ER 1	Signals is too weak
ER 2	Error signal
ER 3	No pressure in the cuff
ER 5	Abnormal result
ER F	reach of maximum 30 min of measurements
ER 11	Signal too weak during central blood pressure measurement
ER 12	Error signal during central blood pressure measurement
ER 13	Cuff pressure errors during central blood pressure measurement
ER 15	Abnormal result of central blood pressure reading

(29) L

Data from left side.

(30) CurrentMeasureNumber

Current measure number.

(31) Hr

Pulse rate (unit:beats/min).

(32) ABI

The value of ABI.

(33) PTT

pulse transit time (unit:ms).

(34) PWV

The value of PWV.

(35) Height

The value of Height.

(36) InfoStatus

Parsed Info status, described in 4-1-6

(37) Time

Record time.

(38) ABI_Error

The results of the ABI_Error received are as follows

ABI_Error		
-----------	--	--

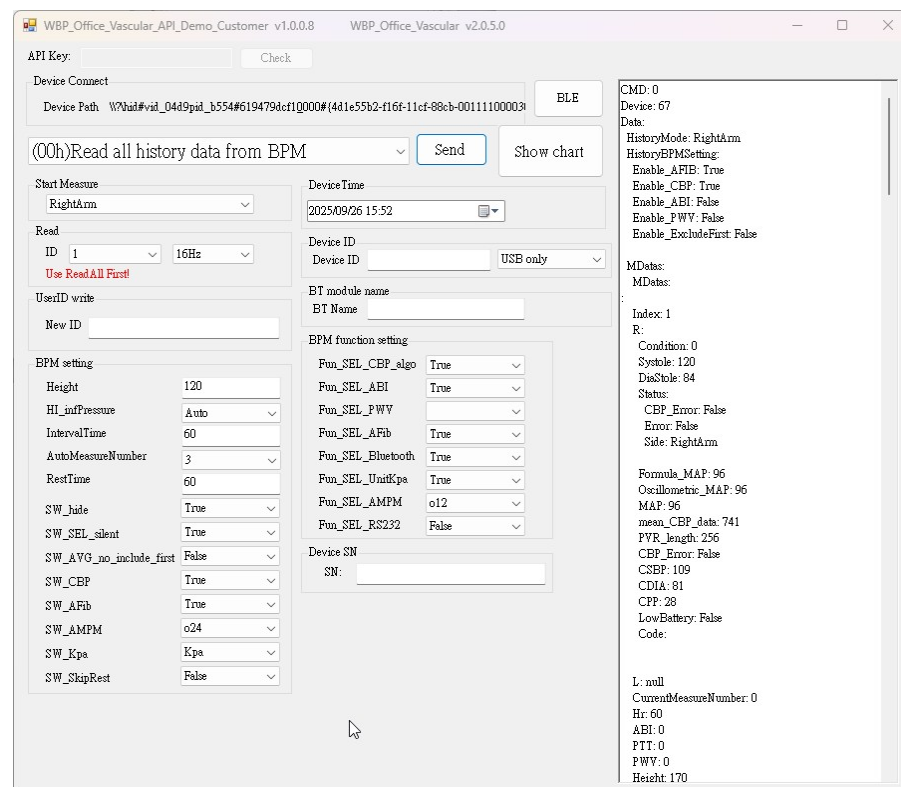
0	None	
21	E21	signals is too weak
23	E23	no pressure in the cuff
25	E25	Abnormal result(PWV)
254	N/A	Not start ABI wave detection. Because occurs error with the ankle or brachial side, device stops the ABI wave detections.

(39) MData_Enable_M_PWV

true: measure with PWV detection.

(40) Code

BP code, including the code of the left and right blood pressure data, as well as all codes



6-2. (01h)Read CBP/ABI wave data by index from BPM

The results of the data format received are as follows

(01)CMD:

current command

(02) Device:

The device code is fixed to 0x43 for Vascular.

(03) Data:

Central BP Data.

(04) WaveType:

The results of the WaveType received are as follows

Enum_WaveType	
1	SingleCBP
3	ABI

(05) Format:

The results of the Format received are as follows.

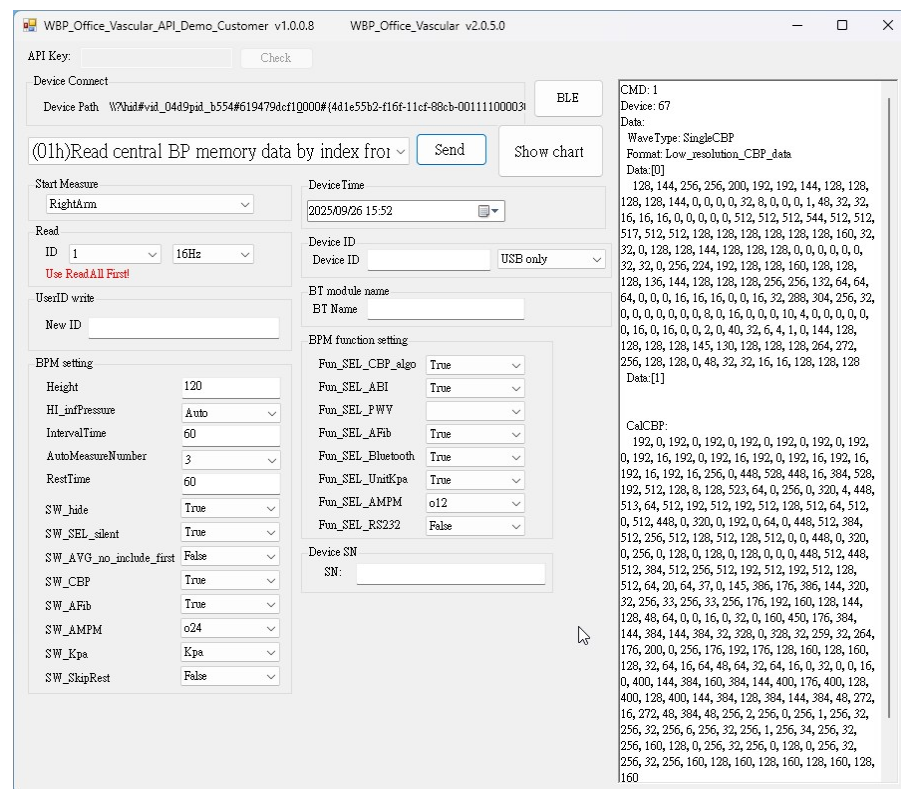
DataFormat		Note
0	No_Raw_Data	No CBP raw data
1	Low_resolution_CBP_data	sampling rate =16Hz
2	Full_CBP_raw_data	sampling rate=256Hz, CBP full resolution
3	Full_PWV_raw_data	sampling rate=1024Hz, PWV full resolution

(06) Data:

Wave Data list.

(07) CalCBP:

CalCBP Data list.

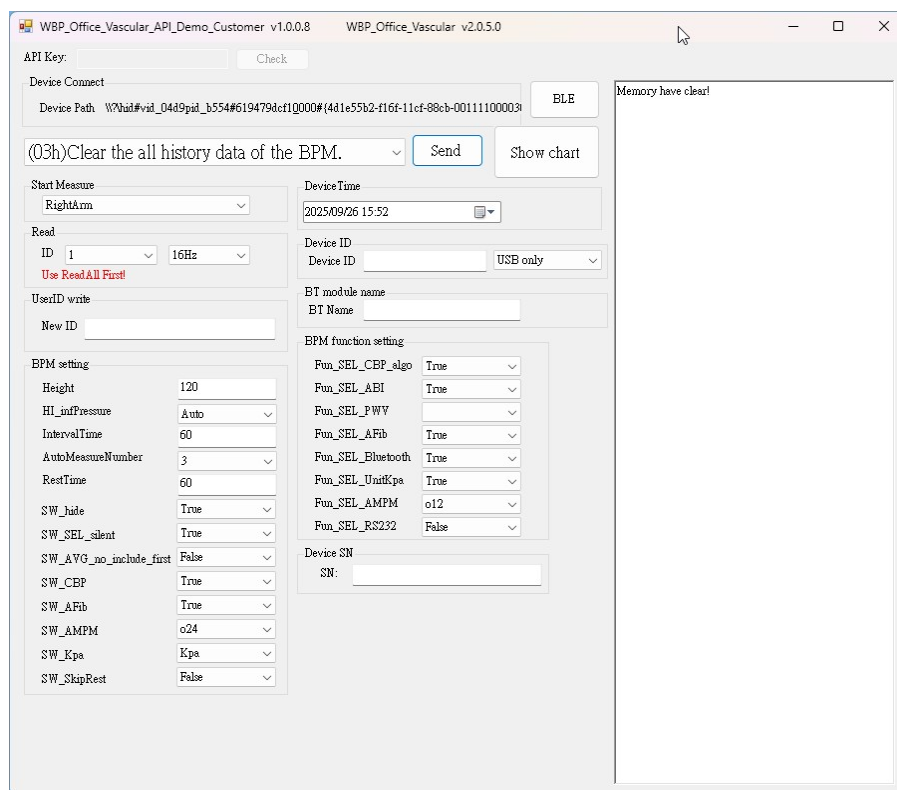
**6-3. (03h)Clear the all history data of the BPM.**

The result could be one of the following.

NACK

No call back, Please try again

Memory have clear



6-4. (05h)Read user ID and version data from BPM.

The results of the data received are as follows

CMD: 5

Device: 67

Data:

UserID: ABCDEFGHIJKLMNOabcdefghijklmno

FMVersion: RP3

FMDate: 2020-09-29 00:00:00

MaxMemory: 6

PWV: True

ABI: True

CBP: True

Afib: True

BatteryVoltage: 4.9

ProtocolID: V1.0.2

arrhythmiaDisplay: NONE

The description is as follows

(01)CMD:

current command

(02) Device:

The device code is fixed to 0x43 for Vascular.

(03) Data:

User ID and version data.

(04) UserID:

User ID, maximum 30 characters

(05) FMVersion:

FW version in BPM, send the ASCII code

(06) FMDate:

The release date of firmware.

(07) MaxMemory

Maximum of memory data can be saved for every user.

(08) PWV:

true: PWV Enable

(09) ABI:

true: ABI Enable

(10) CBP:

true:CBP Enable.

(11) Afib:

true: AFIB Enable

(12) BatteryVoltage:

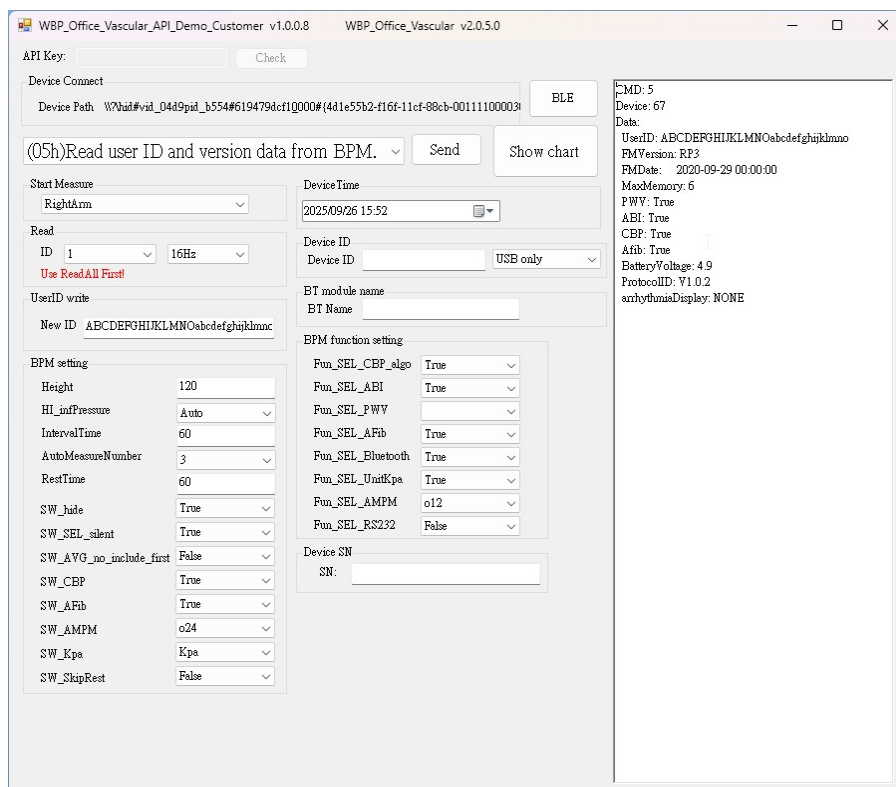
Voltage of the device battery.

(13) ProtocolID:

Protocol ID

(14) arrhythmiaDisplay:

Display arrhythmia name in software. It may be one of NONE, IHB, PAD.



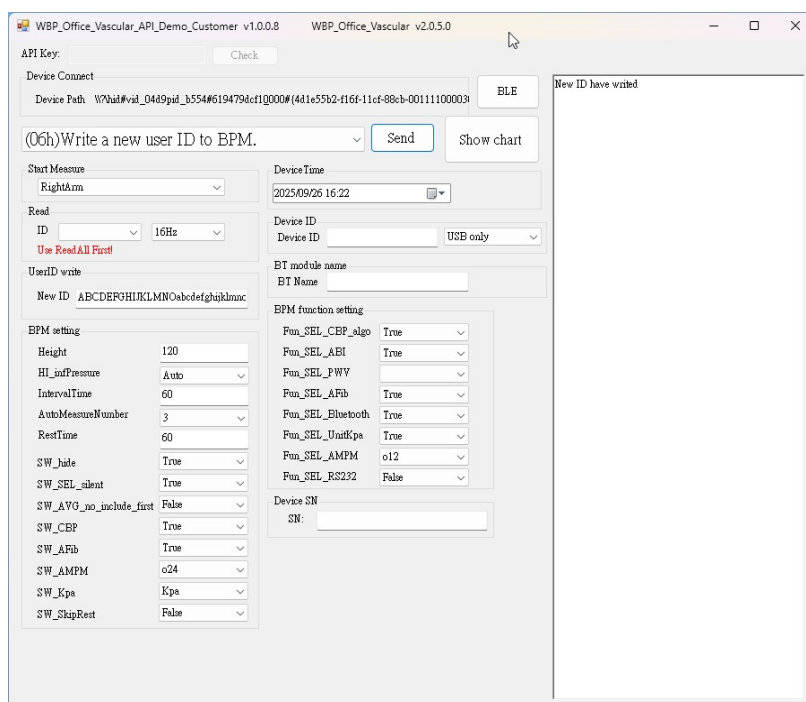
6-5. (06h)Write a new user ID to BPM.

The result could be one of the following.

ACK

No call back,Please try again

New ID have writed



6-6. (07h)Read BPM setting values from BPM.

The results of the data received are as follows

CMD: 7

Device: 67

Data:

Height: 170

HI_infPressure: _180

RestTime: 30

IntervalTime: 15

AutoMeasureNumber: 2

SW_hide: True

SW_SEL_silent: True

SW_AVG_no_include_first: False

SW_CBP: False

SW_AFib: True

SW_AMPM: o24

SW_Kpa: Kpa

SW_SkipRest: False

The description is as follows

(01)CMD:

current command

(02) Device:

The device code is fixed to 0x43 for Vascular.

(03) Data:

BPM setting data.

(04) Height:

Height Value.

(05) HI_infPressure:

Highest inflation pressure of Auto mode

Valid parameter: 0(not setting), 160, 180, 200, 220, 240

(06) RestTime:

Rest time of auto mode

Start countdown base on rest time before 1st measurement in auto mode. Range: 1~300 seconds.

(07) IntervalTime:

Interval time of auto mode

Start countdown base on interval time before 2nd~6th measurement in auto mode.

(08) **AutoMeasureNumber:**

It's number of measurements in auto mode. Range: 1~6 times

(09) **SW_hide:**

true: Don't show readings during rest time in auto mode.

false: Show readings during rest time in auto mode.

(10) **SW_SEL_silent:**

true: Beeper disabled

false: Beeper enabled

(11) **SW_AVG_no_include_first:**

true: Average excludes first memory data.

false: Average includes first memory data.

(12) **SW_CBP:**

true: CBP measurement enabled

false: CBP measurement disabled

(13) **SW_AFib:**

true: AFib measurement enabled

false: AFib measurement disabled

(14) **SW_AMPM:**

o24: 24-hour clock

o12: select 24hr or 12hr by UI

(15) **SW_Kpa:**

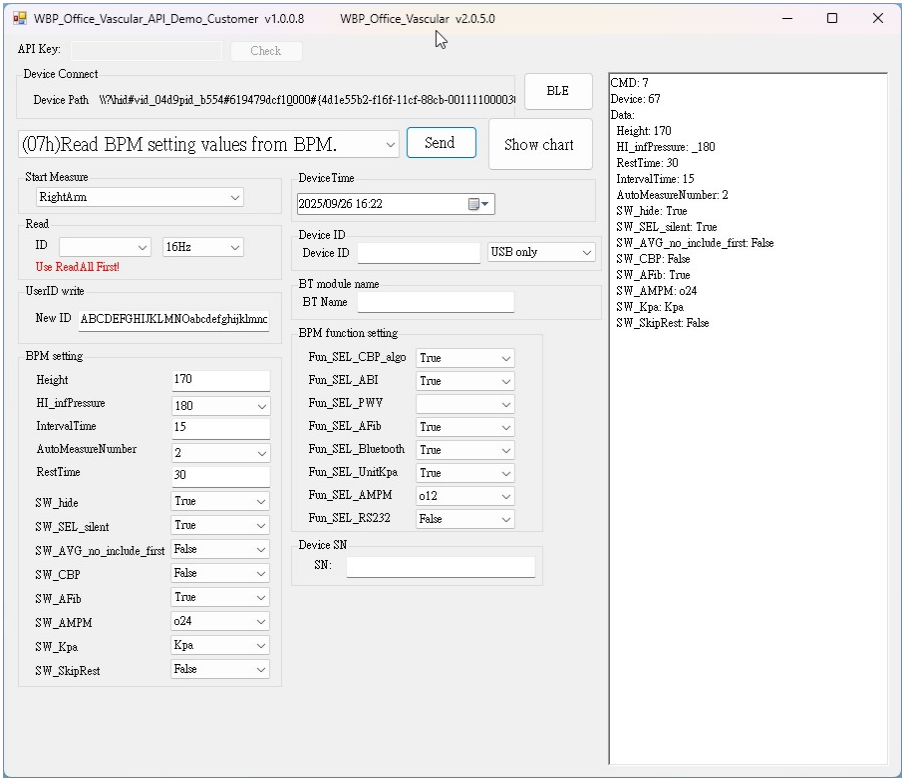
mmHg,

Kpa

(16) **SW_SkipRest:**

true: skip rest time.

false: keep rest time.



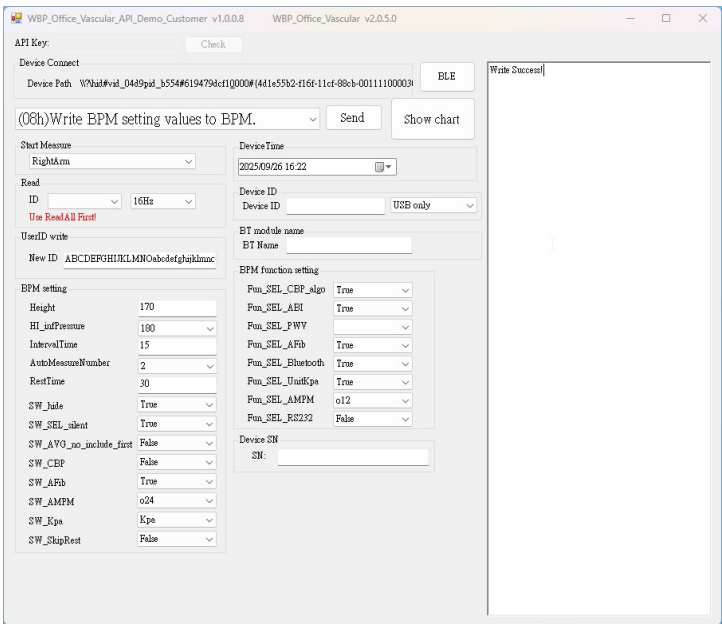
6-7. (08h)Write BPM setting values to BPM.

The result could be one of the following.

NACK

No call back,Please try again

Write Success!



6-8. (0Bh)Read device ID and info from BPM.

The results of the data received are as follows

CMD: 11

Device: 67

Data:

ID: D1FD9DA2E4A3

ConnectType: BothOfUSBAndBT

R_Mea_times: 413

L_Mea_times: 472

ErrorCountR:

12, 5, 86, 9, 0

ErrorCountL:

3, 16, 94, 5, 0

The description is as follows

(01)CMD:

current command

(02) Device:

The device code is fixed to 0x43 for Vascular.

(03) Data:

Device ID and info data.

(04) ID:

Device ID string

(05) ConnectType:

BothOfUSBAndBT

USBOnly

(06) R_Mea_times:

Right side measurement times.

(07) L_Mea_times:

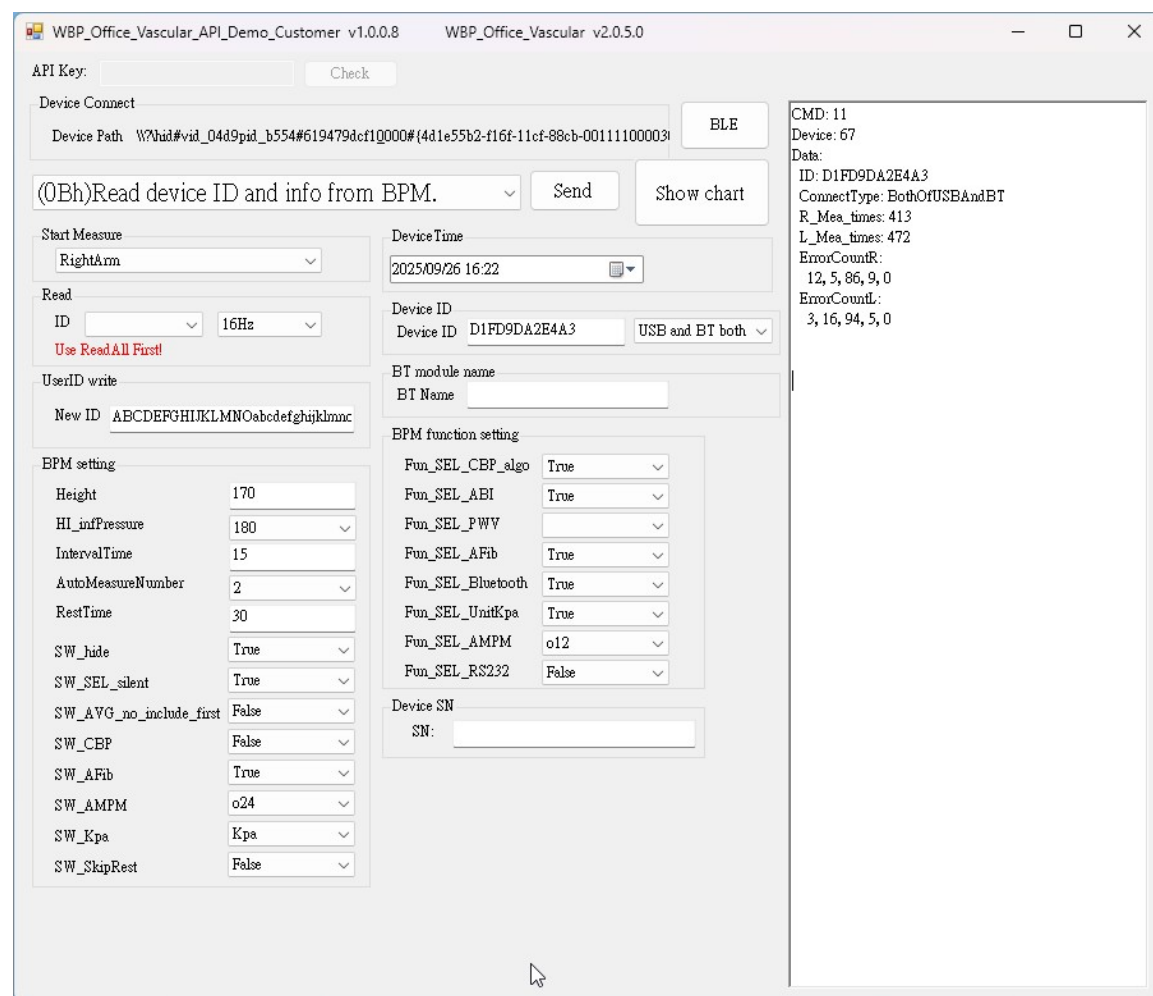
Left side measurement times.

(08) ErrorCountR:

Right side total number of occurrences of error 1, error 2, error 3, error 5,error F.

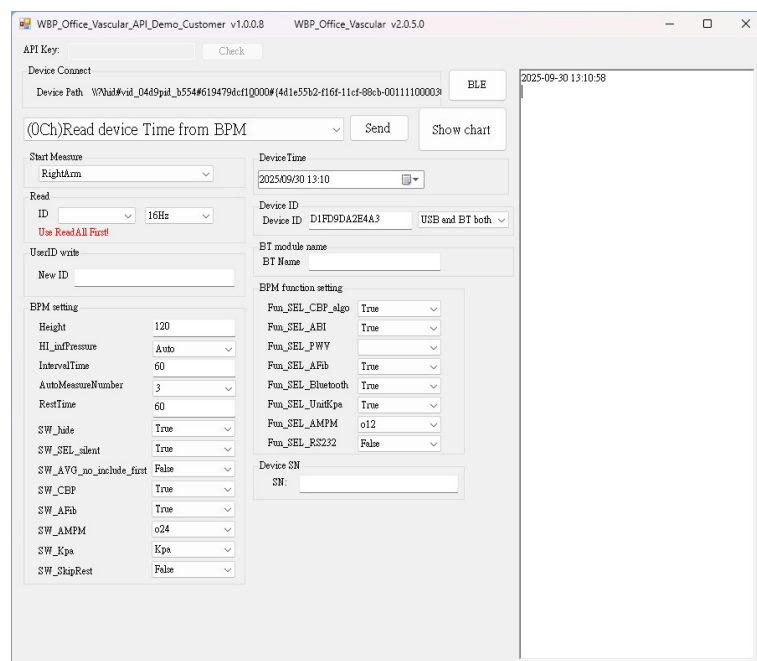
(09) ErrorCountL:

Left side total number of occurrences of error 1, error 2, error 3, error 5,error F.



6-9. (0Ch)Read device Time from BPM

The time of device will be sent back, if no time is set, NACK will be sent back.



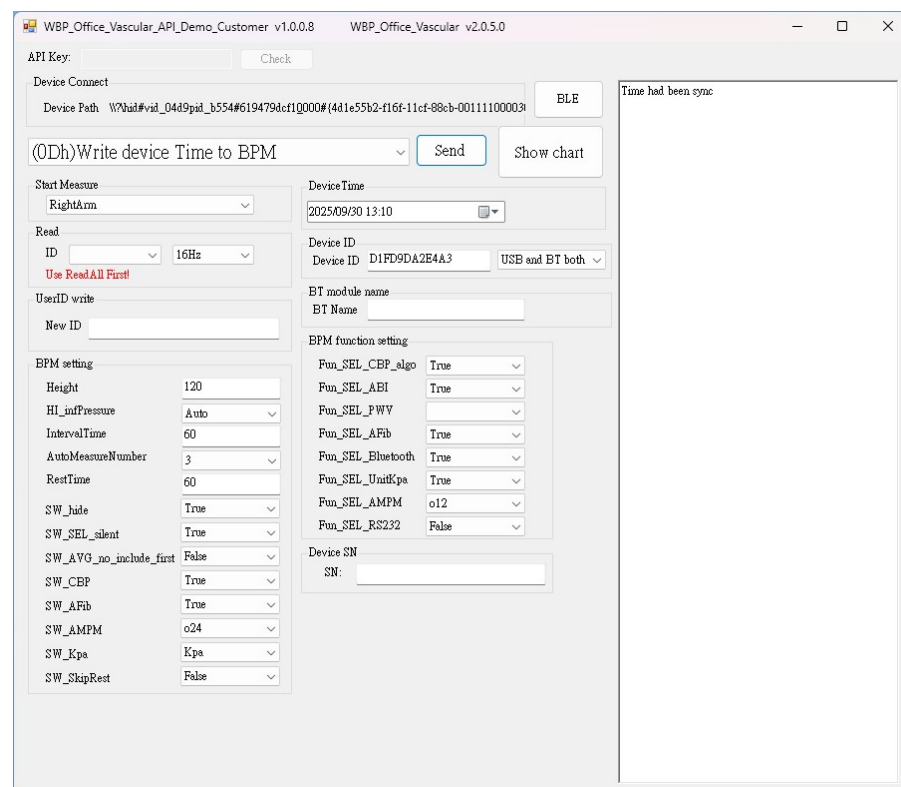
6-10. (0Dh)Write device Time to BPM

The result could be one of the following.

NACK

No call back,Please try again

Time had been sync



6-11. (10h)Read BPM function setting value from BPM.

The results of the data received are as follows

CMD: 16

Device: 67

Data:

Fun_SEL_ABI: True

Fun_SEL_PWV: True

Fun_SEL_CBP: True

Fun_SEL_AFIB: True

Fun_SEL_Bluetooth: True

Fun_SEL_UnitKpa: False

Fun_SEL_AMPM: o24

Fun_SEL_RS232: False

The description is as follows

(01)CMD:

current command

(02) Device:

The device code is fixed to 0x43 for Vascular.

(03) Data:

BPM function setting data.

(04) Fun_SEL_ABI:

true: Enable ABI function.

false: Disable ABI function.

(05) Fun_SEL_PWV:

true: Enable PWV function.

false: Disable PWV function.

(06) Fun_SEL_CBP:

true: Enable CBP function.

false: Disable CBP function.

(07) Fun_SEL_AFIB:

true: AFIB algorithm ON.

false: AFIB algorithm OFF

(08) Fun_SEL_Bluetooth:

true: Bluetooth function ON.

false: Bluetooth function OFF.

(09) Fun_SEL_UnitKpa:

true: select Kpa or mmHg by UI.

false: only mmHg.

(10) Fun_SEL_AMPM :

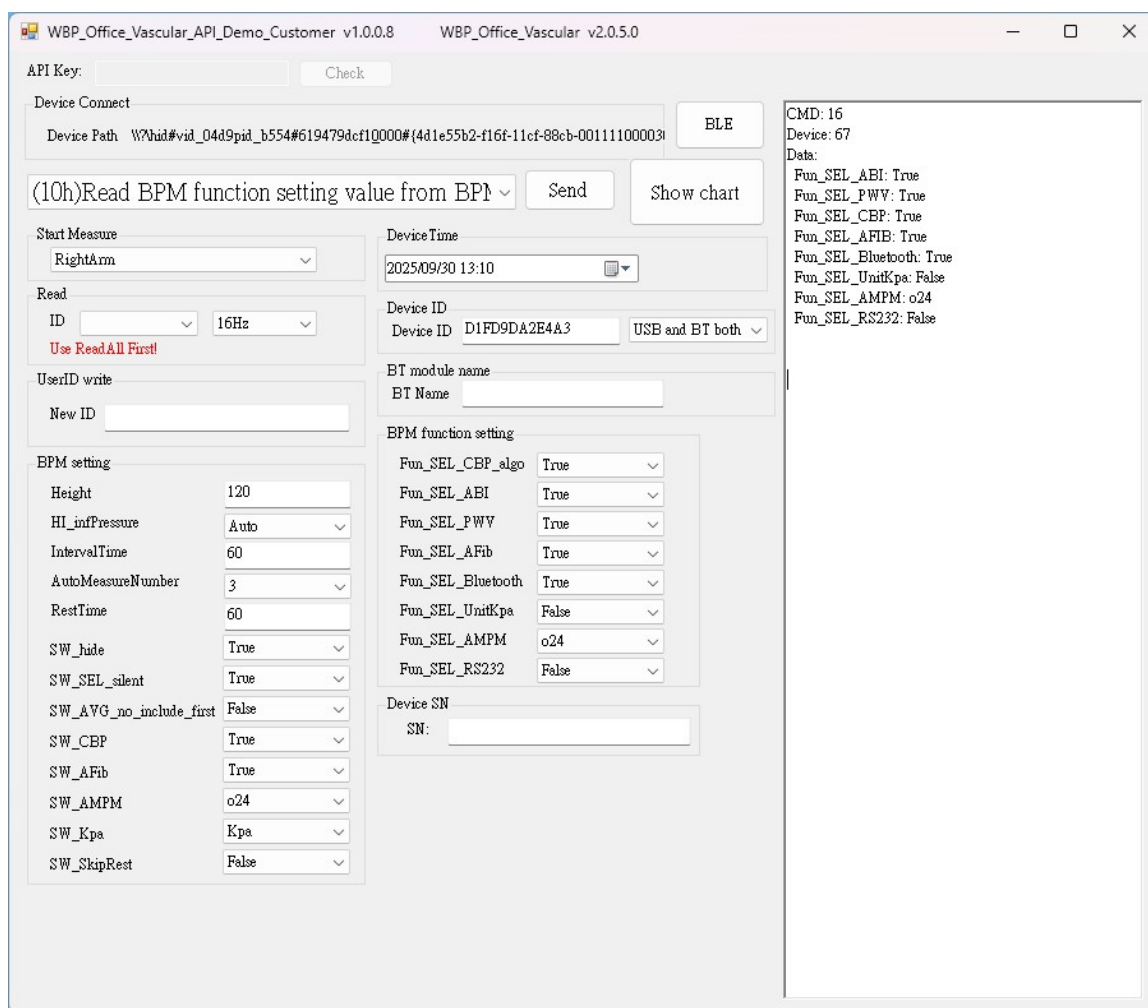
o24: only 24hr

o12: select 24hr or 12hr by UI.

(11) Fun_SEL_RS232:

true: LabView data transmission ON.

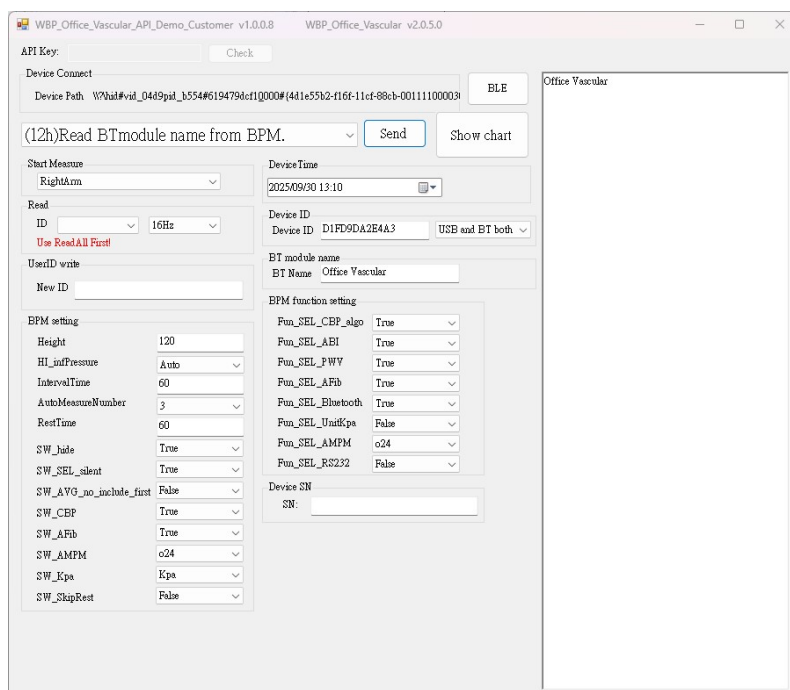
false: LabView data transmission OFF.



6-12. (13h)Read BTmodule name from BPM.

The results of the data received are as follows

The name of the BT module will be displayed in the BT Name text box.



6-13. (25h)Start remote measurement.

The results of the data received are as follows

Start Remote Measurement,DataFormat is (RightArm, No_Raw_Data)

The SDK will receive an update of the current status every five seconds after the start of the measurement.

Status: Wait_countdown_for_next_measurement

Measurement_Number: 1

Measurement_Total: 3

Countdown: 60

TotalMeasurementTime: 0

The description is as follows

(a) Status :

There are four possible status

Wait_countdown_for_next_measurement

Start_BP_measurement

Manual_press_IO_to_stop_measurement

EmptyBattery

(b) Measurement_Number :

Send current measurement number in auto mode.

(c) Measurement_Total :

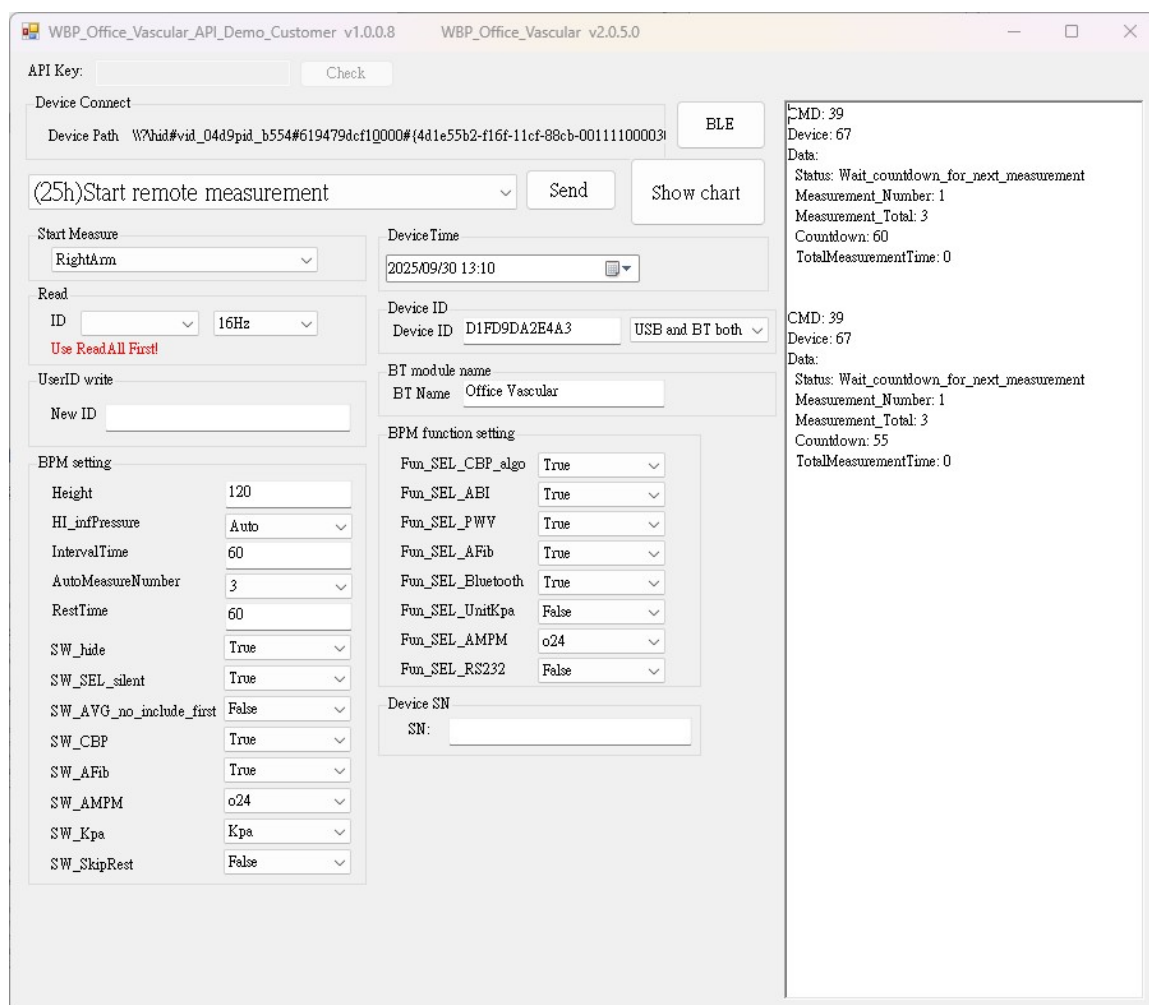
Send total measurement number in auto mode.

(d) Countdown :

Send current countdown time in auto mode.

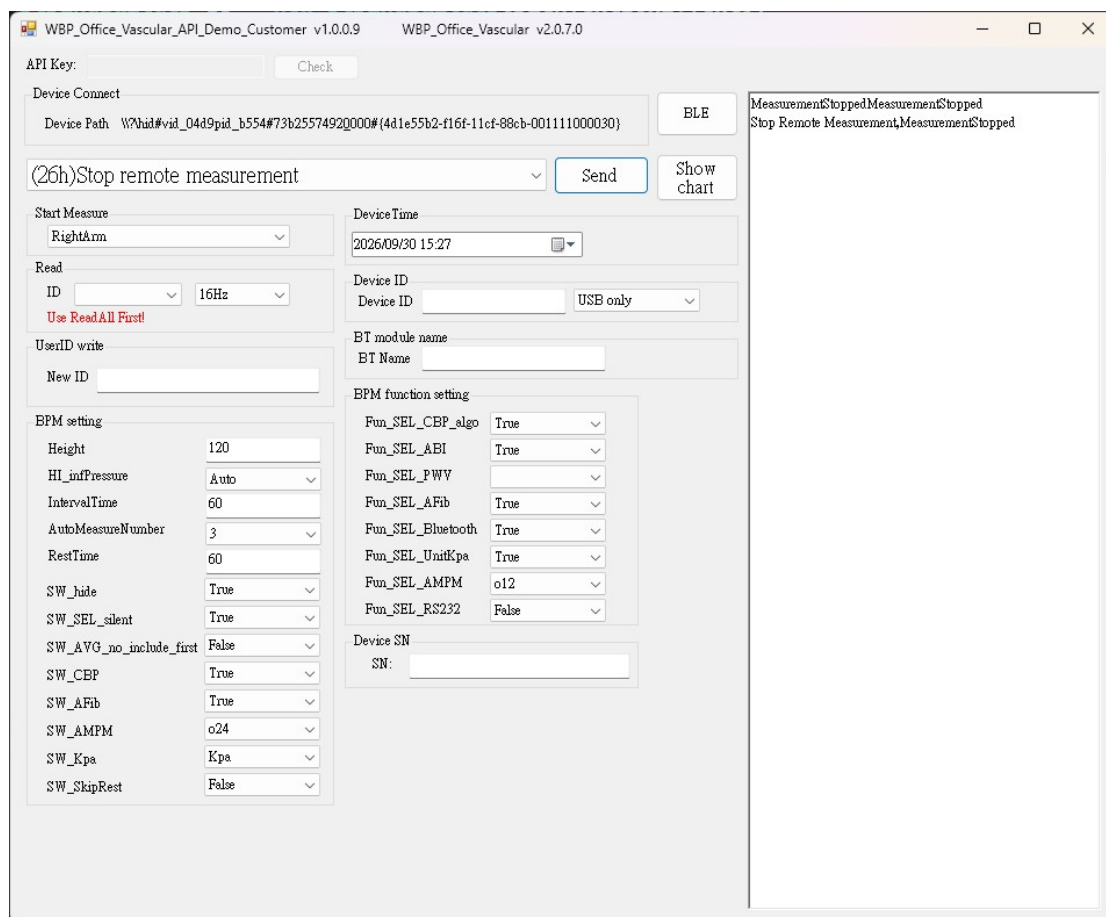
(e) TotalMeasurementTime :

Send total measurement time (seconds) in auto mode. Total measurement are count between 1st measurement to last measurement. (exclude rest time).



6-14. (26h)Stop remote measurement.

After successfully stopping the measurement, the SDK will receive “Stop Remote Measurement” event. Command.Data is Enum_StopResult (MeasurementStopped or NotMeasuring), described in 4-2-11.



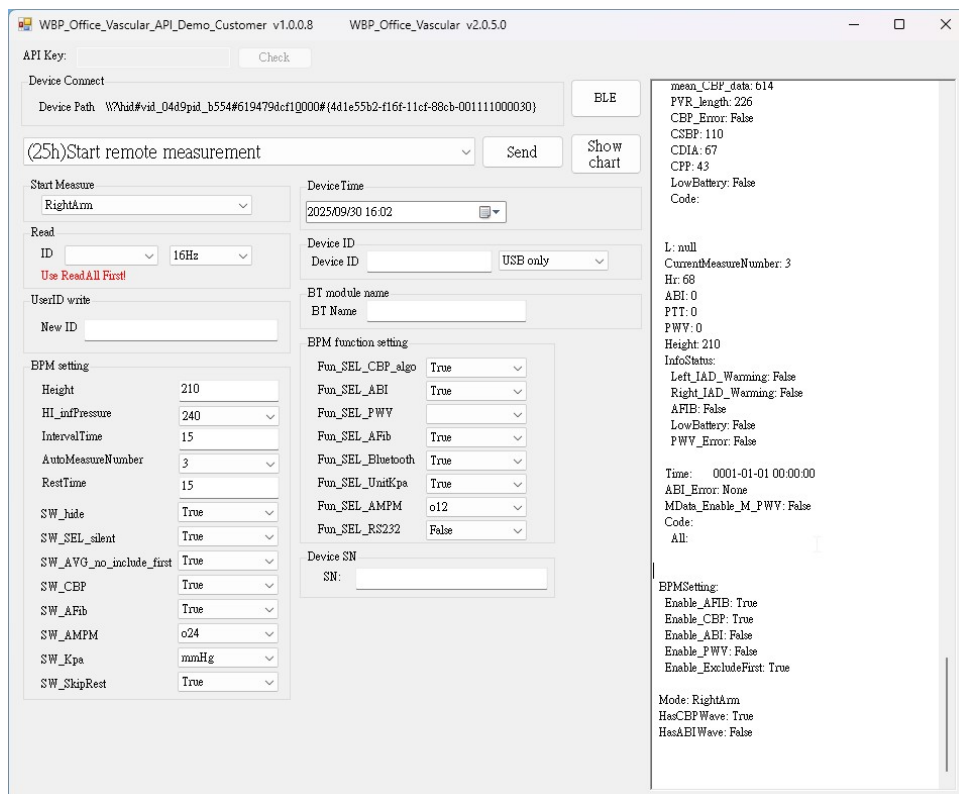
6-15. (28h) Send measurement results for each measurement

After a successful measurement, the SDK will receive the blood pressure data in the same format as the “(00h)Read all history data from BPM”, but with one new parameters.

The new parameters descriptions are as follows

(01) Mode:

A Enum of HistoryMode, described in 4-2-1.



6-16. (0Fh-00) Read serial number from BPM

The results of the data received are as follows

The serial number from BPM will be returned in string format.

