

MicroLifeDevice SDK (WBP_Home A BT - Windows)

Table of Contents

Chapter 1 Development Environment

Chapter 2 Connection Sequence of WBP_Home A BT

Chapter 3 APIs of WBP_Home A BT

Chapter 4 Class & Object of WBP_Home A BT

Chapter 5 Instruction of Demo App

Chapter 6 The description for each command of Demo App

Revise history

Date	Document Version	Description
2024/10/24	1.0	First release.

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 2022(recommended)
Programming language:	C#
Target framework:	.NET Standard 2.0

Importing steps are described below.

- 1.1. First of all, add WBP_Home_A.dll, Connection.dll, Bluetooth.dll, UsbHid.dll into a development project.
- 1.2. Import class as bellows.
using WBP_Home_A_BT.Class;
using Connection.Class;

Chapter 2 Connection Sequence of WBP_Home A BT

The WBP_Home_A_BT object is applied managing the Bluetooth communication.

2.1 Initiate WBP_Home_A_BT Object with API key and set connected/disconnected delegation for WBP_Homa_A_BT.

```
home = new WBP_Home_A_BT.WBP_Home_A_BT(key: "");
home.OnConnected += Home_OnConnected;
home.OnDisConnected += Home_OnDisConnected;
```

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

```
var connectionResult = await home.ConnectWithBLE(form_BLE_Discover.BLE_ID);
```

2.3 When the data is transferred via Bluetooth. The parsed data will be returned through the command object, which will be explained in section 3-3.

```
(bool Success, Command Command) callback;
callback = home.SendMessage(WBP_Home_A_BT.SendMessage.ToCommand.timeRead());
if (callback.Command.CMD == (byte)WBP_Home_A_BT.Enum.Command.NACK) {
    Console.WriteLine(value: "NACK");
    return;
} else if (!callback.Success) {
    Console.WriteLine(value: "No call back,Please try again");
    return;
}
DateTime dateTime = (DateTime)callback.Command.Data;
```

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

Chapter 3 APIs of WBP_Home A BT

This chapter will explain the application of each API and the meaning of its parameters.

3-1.WBP_Home_A_BT Initialize

API	WBP_Home_A_BT (string key);
Function	Initialize WBP_Home_A_BTObject
Return object	WBP_Home_A_BT: After the initialization is successful, a WBP_Home_A_BT object will be created.
Parameter	key: API Key is required to use subsequent APIs, if you do not have it, please contact Microlife ** If the API key is incorrect, you will receive a "Key Fail!" exception..

3-2 Connect with device by BLE

API	Task< ConnectionResult > ConnectWithBLE(string deviceID);
Function	Use this API to connect with device by BLE..
Return object	ConnectionResult: A class containing connection result, described in 4-1-3
Parameter	DevuceID: Bluetooth ID of the device to be connected.

3-3.Send Command to device

API	(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.
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 ** The contents of the above parameters are generated by the following API.

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

Command	userInfoRead
API	SendMessage(WBP_Home_A_BT.SendMessage.ToCommand. userInfoRead ());

Return Command Data Type.	The object type of Command.data is UserInfo class, which is a class containing parsed BP data, described in 4-1-2 .
---------------------------	---

3-3-2. Write device Time to BPM.

Command	timeWrite
API	SendMessage(WBP_Home_A_BT.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.

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

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

3-3-4. Read usual mode data from BPM

Command	dataReadUsual
API	SendMessage(WBP_Home_A_BT.SendMessage.ToCommand. dataReadUsual ());
Return Command Data Type.	The object type of Command.data is Data class, which is a class containing parsed BP data, described in 4-1-3

3-3-5. Read diagnostic mode data from BPM

Command	dataReadDiagnostic
API	SendMessage(WBP_Home_A_BT.SendMessage.ToCommand. dataReadDiagnostic ());
Return Command Data Type.	The object type of Command.data is Data class, which is a class containing parsed BP data, described in 4-1-3

3-3-6. Clear select mode history data of the BPM

Command	dataClearSelectMode
---------	----------------------------

API	SendMessage(WBP_O3.SendMessage.ToCommand. dataClearSelectMode (bool diagnostic_Mode, bool usual_Mode));
Parameter	diagnostic_Mode : If true, diagnostic mode data will be cleared usual_Mode : If true, usual mode data will be cleared
Return Command Data Type.	None. ** If the Success is true, and the Command.CMD is not NACK, the device has successfully received this command.

3-3-7. Clear current mode history data of the BPM

Command	dataClearCurrentMode
API	SendMessage(WBP_O3.SendMessage.ToCommand. dataClearCurrentMode())
Parameter	diagnostic_Mode : If true, diagnostic mode data will be cleared usual_Mode : If true, usual mode data will be cleared
Return Command Data Type.	None. ** If the Success is true, and the Command.CMD is not NACK, the device has successfully received this command.

3-3-8. Write a new user ID to BPM.

Command	userIDWrite
API	SendMessage(WBP_O3.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, maximum 11 characters

3-3-9. Read serial number from BPM

Command	deviceSNRead
API	SendMessage(WBP_O3.SendMessage.ToCommand.deviceSNRead());
Return Command Data Type.	The object type of Command.data is string.

3-4 Disconnect the Bluetooth with BPM

API	DisconnectWithBLE
Function	Disconnect the Bluetooth connection from the blood pressure monitor
Return object	BOOL: If the connection is successfully disconnected, true will be returned, otherwise false will be returned..

Chapter 4 Class & Object of WBP_O3 2G

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, it means the transmission failed. byte Device: Record the device. object Data: parsed data

4-1-2. UserInfo Class

Name:	UserInfo
Definition	A class containing parsed user's information data.
members	string UserID: User ID, maximum 11 characters string ProtocolID: Protocol ID int MaxMemory: Maximum of memory data can be save for every user=250 int MaxUser: Maximum of user number in BPM=1. bool Diagnostic_Afib: true: Diagnostic mode Afib enable bool Usual_Afib: true: Usual mode Afib enable bool Nocturnal_Mode: Deprecated string FMVersion: FW version in BPM, send the ASCII code DateTimeOffset FMDate: The release date of firmware. float BatteryVoltage: Voltage of the device battery. BPTYPE BPTYPE: Display BP type. It may be one of WBPHome, WBPHomeA described in 4-2-1 Current_Mode Current_Mode: Displays the currently set BP measurement mode. It may be one of UsualMode, DiagnosticMode, Others described in 4-2-2 int OffsetYear: reserved

4-1-3.Data Class

Name:	Data
Definition	A class containing parsed BP data.
members	int ID: Deprecated DateTimeOffset MeasureDateTime: Record time int Systole: The value of systole int Diastole: The value of diastole

	int Pulse: heart rate int Mam: Deprecated int Comments: Deprecated ArrhythmiaEnum Arr: A Enum of ArrhythmiaEnum, described in 4-2-4. AMPM Ampm: A Enum of AMPM, described in 4-2-5. DayNightEnum DayNight: Deprecated DataTypeEnum DataType: A Enum of DataTypeEnum, described in 4-2-6. ErrMsgEnum ErrMsg: Deprecated int MAP: Deprecated string Err: Deprecated
--	--

4-1-4. ConnectionResult Class

Name:	ConnectionResult
Definition	A class containing Bluetooth connection status
members	string DeviceId: Device ID string string Name: Device's BLE module name bool IsConnected: true: Device is connected. bool HasError: true: Device has error. string ErrorMessage: Error message string

4-2 Enum

4-2-1 BPTYPE

value	
0	WBPHome
1	WBPHomeA

4-2-2 CurrentMode

value		Note
0x00	UsualMode	
0x01	DiagnosticMode	
0xff	Others	reserved

4-2-3 DateTypeEnum

value		Note
0	normal	Default
1	Average	average of all morning diag bp records
2	Morning	average of all morning diag bp records
3	Evening	average of all evening diag bp records

4-2-4 ArrhythmiaEnum

value	
0	NONE
1	PAD
2	AFIB

4-2-5 AMPM

value		Not
0	none	
1	am	The data is measured in night with diagnostic mode
2	pm	The data is measured in morning with diagnostic mode

4-2-6 DataTypeEnum

value		Note
0	none	Deprecated
1	all	Deprecated

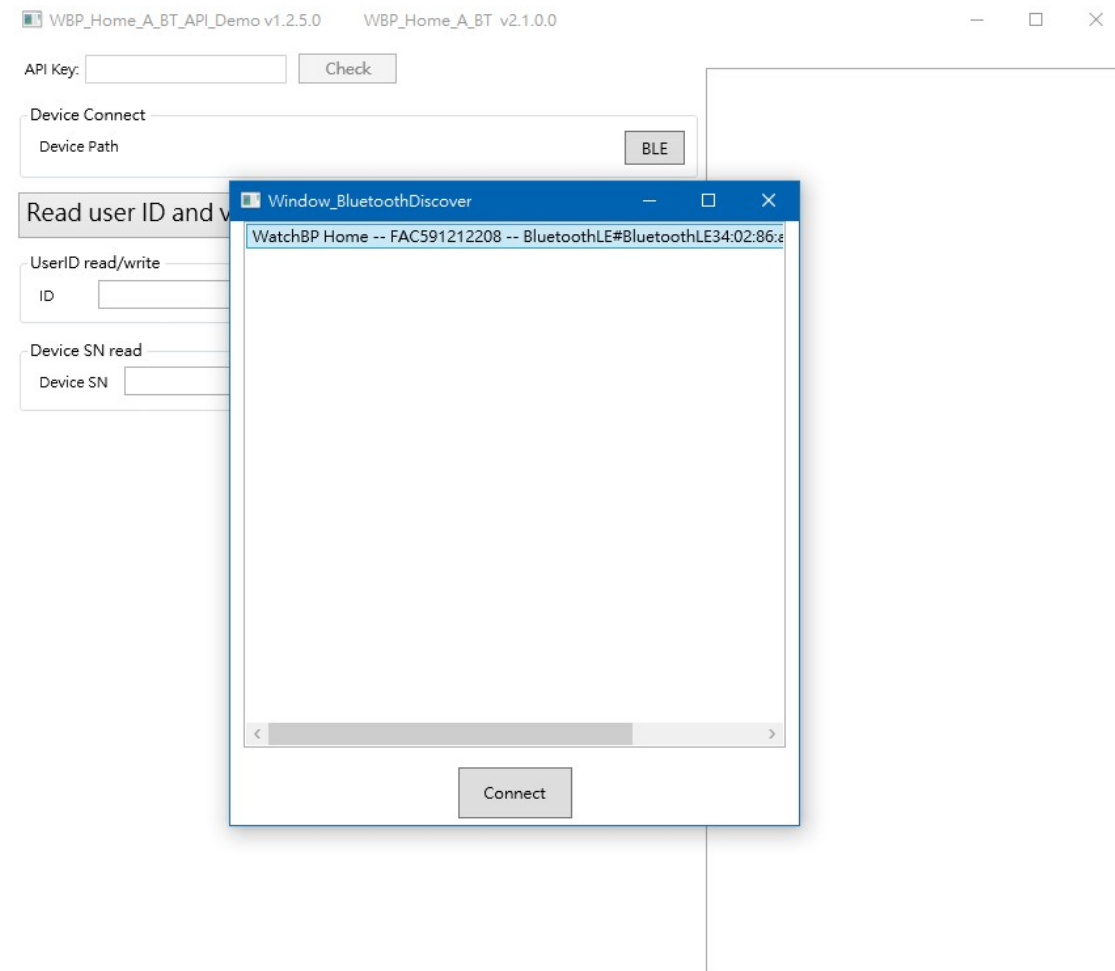
MicroLifeDeviceSDK Windows

2	usual	
3	diagnostic	
4	ABP	Deprecated
5	memo	Deprecated

Chapter 5 Instruction of Demo App

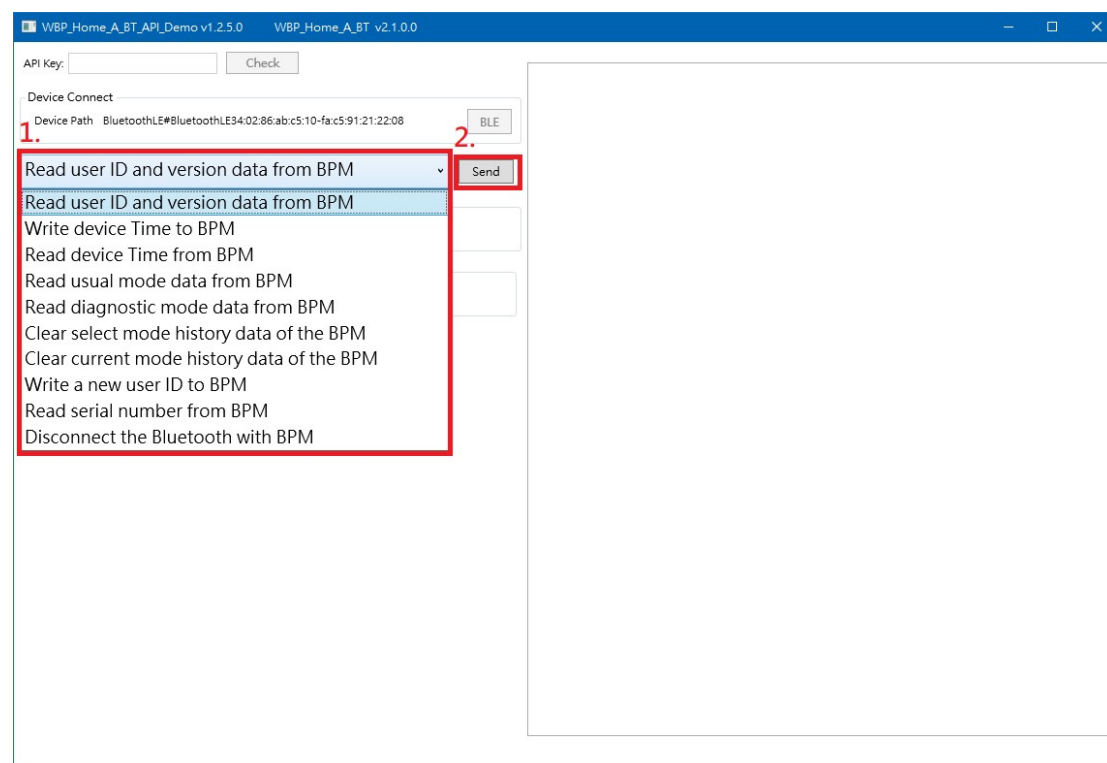
5-1. Input the API Key on the API Key textbox and click “Check” button to active the demo.

5-2. Turn on the Bluetooth of the device and click the BLE button. In the pop-up window, select WatchBP Home device and click Connect to connect.

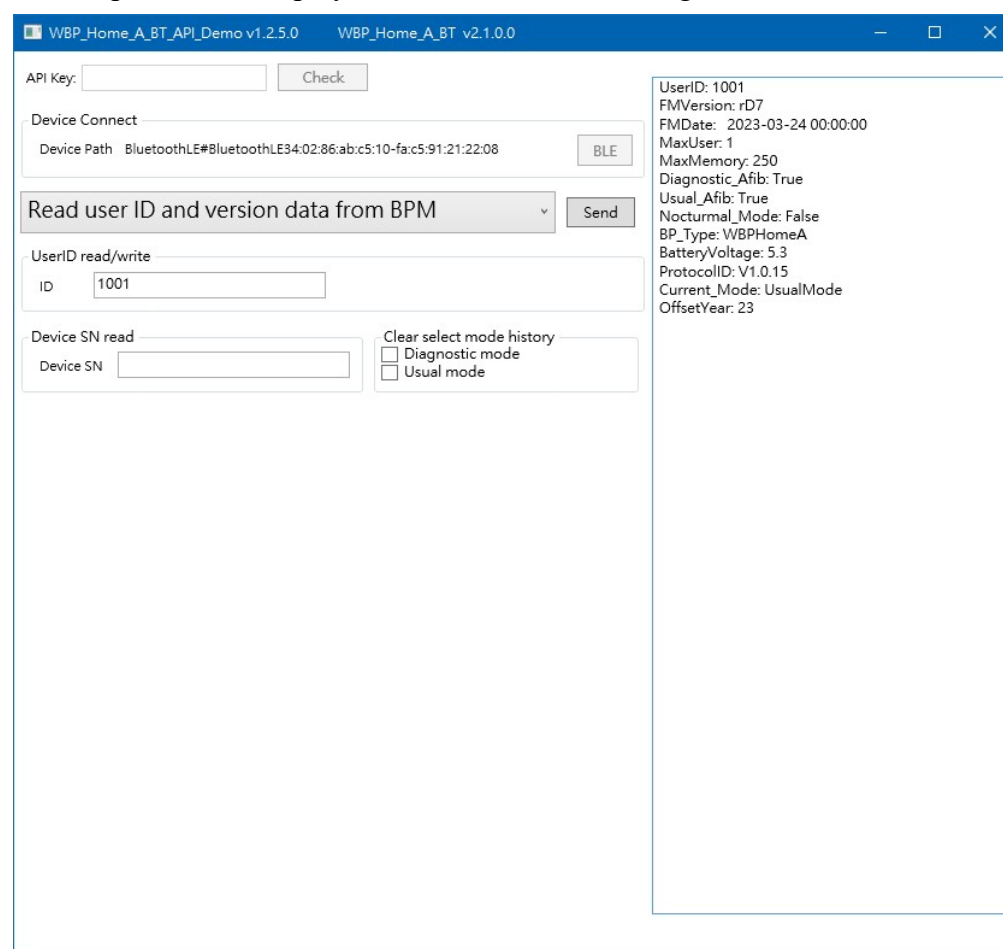


Similarly, if the device is disconnected in anytime, the Send button will be disabled.

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. Read user ID and version data from BPM

The results of the data received are as follows

UserID: 1001
 FMVersion: rD7
 FMDate: 2023-03-24 00:00:00
 MaxUser: 1
 MaxMemory: 250
 Diagnostic_Afib: True
 Usual_Afib: True
 Nocturnal_Mode: False
 BP_Type: WBPHomeA
 BatteryVoltage: 5.3
 ProtocolID: V1.0.15
 Current_Mode: UsualMode
 OffsetYear: 23

The description is as follows

(a) UserID :

User ID, maximum 11 characters

(b) FMVersion:

FW version in BPM, send the ASCII code

(c) FMDate

The release date of firmware.

(c) MaxUser:

Maximum of user number in BPM=1.

(d) MaxMemory:

Maximum of memory data can be save for every user=250

(e) Diagnostic_Afib:

true: Diagnostic mode Afib enable

(f) Usual_Afib

true: Usual mode Afib enable

(g) Nocturnal_Mode

Deprecated

(h) BP_Type

The results of the BP_Type received are as follows

value	
0	WBPHome
1	WBPHomeA

(i) **BatteryVoltage**

Voltage of the device battery.

(j) **ProtocolID**

Protocol ID

(k) **Current_Mode**

The results of the BP_Type received are as follows

value		Note
0x00	UsualMode	
0x01	DiagnosticMode	
0xff	Others	reserved

(l) **OffsetYear**

reserved

6-2. Write device Time to BPM.

The result could be one of the following.

ACK

No call back,Please try again

Time had been sync

6-3. Read device Time from BPM

The time of device will be sent back

6-4. Read usual mode data from BPM.

The results of the data received are as follows

	Sys	Dia	Hr	Arr
0	2023-01-01	01:01:00	120	80 60 noArr
1	2023-01-02	02:02:00	120	80 60 noArr
2	2023-01-03	03:03:00	120	80 60 noArr

The description is as follows

(a) **MeasureDateTime:**

Record time

(b) **Systole:**

The value of systole

(c) **Diastole:**

The value of diastole.

(d) **Pulse:**

heart rate

(e) Arr:

The results of the Arr received are as follows

value	
0	NONE
1	PAD
2	AFIB

6-5. Read diagnostic mode data from BPM.

The results of the data received are as follows

			Sys	Dia	Hr	Arr
0	2023-07-02	07:00:00	151	96	61	AFIB
1	2023-07-02	07:03:00	150	95	60	AFIB
4	2023-07-02	19:00:00	151	96	61	AFIB
5	2023-07-02	19:03:00	150	95	60	AFIB
8	2023-07-03	07:00:00	122	72	62	AFIB
9	2023-07-03	07:03:00	120	70	60	noArr
16	2023-07-04	07:00:00	121	71	61	AFIB
average	Average		122	72	62	
average	Morning		122	72	62	
average	Evening		123	73	63	

The description is as follows

(a) MeasureDateTime:

Record time

(b) Systole:

The value of systole

(c) Diastole:

The value of diastole.

(d) Pulse:

heart rate

(e) Arr:

The results of the Arr received are as follows

value	
0	NONE
1	PAD
2	AFIB

(f) DateType:

The results of the DateType received are as follows

value		Note
0	normal	Default
1	Average	average of all morining diag bp records
2	Morning	average of all morining diag bp records
3	Evening	average of all evening diag bp records

6-6. Clear select mode history data of the BPM

The result could be one of the following.

NACK

No call back, Please try again

The selected mode memory has been cleared

6-7. Clear current mode history data of the BPM

The result could be one of the following.

NACK

No call back, Please try again

The current mode memory has been cleared

6-8. Write a new user ID to BPM

The result could be one of the following.

NACK

No call back, Please try again

New ID have writed

6-9. Read serial number from BPM

The SN of device will be sent back

6-10. Disconnect the Bluetooth

The result could be one of the following.

Bluetooth disconnection successful.

Bluetooth disconnection failed