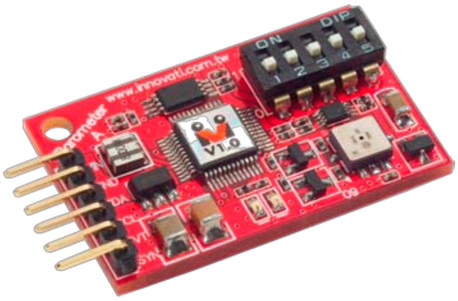
Barometer A

User's Guide

Atmospheric Pressure Sensing Module

Version: V1.0



Product Overview: Innovati's Barometer A module is a user-friendly high-precision atmospheric pressure sensing module. With the connection through the cmdBUS to the BASIC Commander, the value of the detected atmospheric pressure can be retrieved by using a simple command. Furthermore, the altitude can be derived according to the preset pressure at sea level. Meanwhile, it allows the user to configure various pressure notification events so that the user can set the required pressure values for notification according to the demands.

Application Scope:

- For general atmospheric pressure measurement, it can provide the pressure values required for meteorological observation.
- By setting the pressure at sea level, it can serve as an altimeter.
- In high-speed sensing mode, it can be used to measure a more stringent pressure variation inside an airtight space.

Product Features:

- Digitization of the value of the atmospheric pressure experienced by the sensing module.
- Atmospheric pressure measurement range: up to 300-1100 hPar.
- Resolution: up to 0.1 hPar.
- By setting the pressure value at sea level, the altitude can be obtained through calculation.
- Altitude measurement range: up to -500–9000 m.
- It can provide integer and float-point measurement readings.
- The unit conversion capability allows you to convert various atmospheric pressure and altitude values according to your demands.
- Notifications on the ranges of pressure and altitude are provided. Once the preset atmospheric pressure or altitude range is reached, it can automatically generate the notification event. Up to 8 separate ranges can be configured for both pressure and altitude.
- Alarms on the variations of pressure and altitude are provided. It allows you to set the maximum allowed variation of pressure or altitude. When the module detects an excessive variation of pressure or altitude during the start-up stage, it will automatically generate the alarm event.
- A memory storage capacity of up to 120 sets of pressure and altitude values is provided which allows the user to configure the storage interval.
- Both high-precision and high-speed modes are provided. When the user selects the high-precision measurement mode, the module can automatically average up the measurement values; when the high-speed mode is used, it can provide up to 10 measurements per second which is suitable for the sensing applications in which the atmospheric pressure changes more drastically.
- Measurement alarm events are provided. After the module is activated, when a new measurement is updated, an alarm event will be generated.
- All the setting values can be re-read for verification by using the corresponding commands.

Connection: Set the ID switch to the required number directly, and then connect the cmdBUS to the corresponding pins on the BASIC Commander so that the user can perform the required operations through the BASIC Commander.

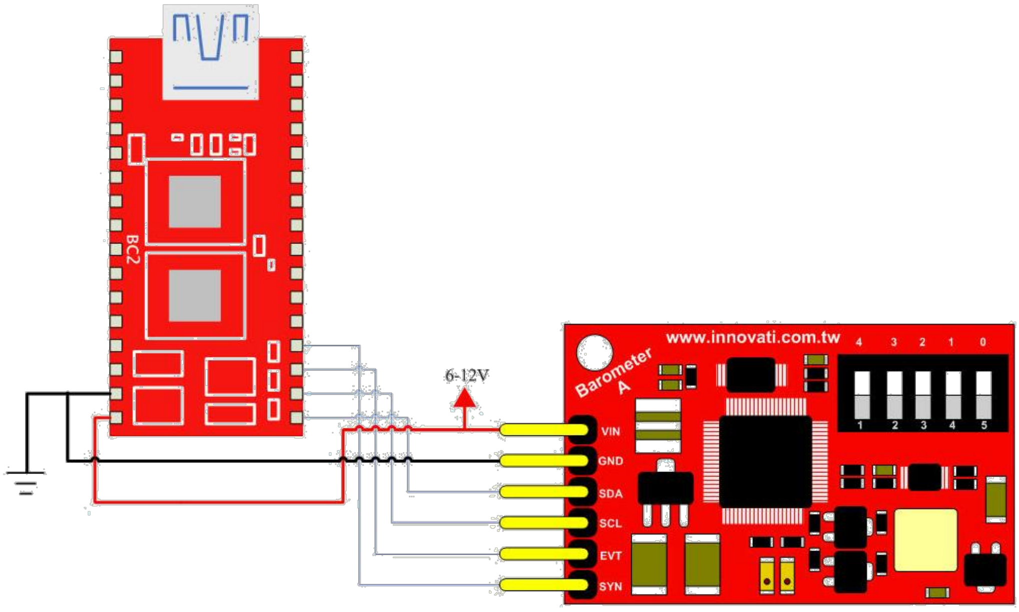
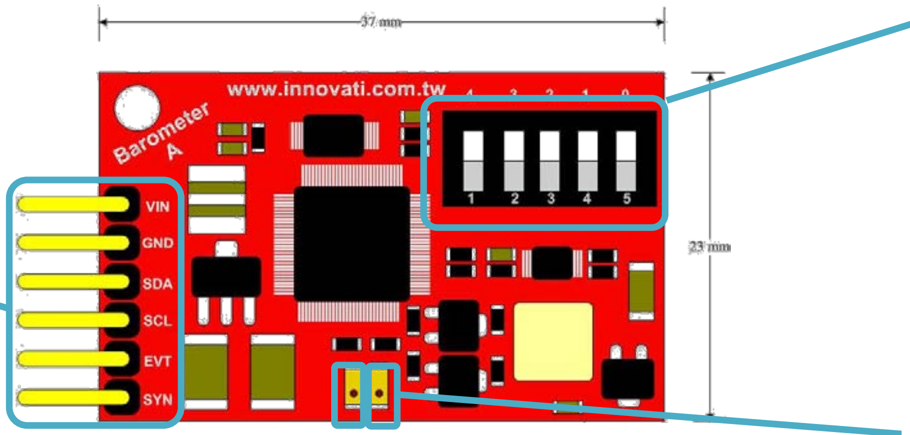


Figure 1 Connection with the BASIC Commander

Product Specifications:

Pins for cmdBUS: Connect these pins to the corresponding pins on the BASIC Commander for controlling the Barometer A module through the BASIC Commander. While connecting, please notice the pin assignment. Connect Vin to the Vin on the BASIC Commander. Incorrect pin connection may cause damage to the module.)

Module ID Setting Switch: The module ID of the Barometer A module can be configured with the binary digits from the right to the left. This ID number allows the BASIC Commander to determine the module to be controlled during the operation (Please refer to Appendix 2).



From the left to the right:
Green Event Indicator: The blinking light indicates that the module is transmitting an event.
Orange Command Indicator: The blinking light indicates the module and the SBC are receiving data.

Figure 2 Description of pins and switches on the module

Precautions for Operations:

- Do not use the module in an environment with an atmospheric pressure exceeding 5000 hPa.

Operating Temperature of the Module:	0 °C~ 65 °C
Storage Temperature of the Module:	-40 °C ~125 °C

Commands And Events:

The following list shows the commands dedicated for controlling the Barometer A module. The command name and parameters which should be input are shown in bold or bold-italic typefaces. The words in bold typeface should not be changed while being input. The words in bold-italic typefaces can be filled with parameters in the properly defined format by the user. Please note that the words in uppercase or lowercase are regarded as the same word while entering the command in the innoBASIC Workshop.

Before executing the command for Barometer A, please define the corresponding parameters and the module ID at the beginning of the program, for example:

Peripheral *ModuleName* As Barometer A @ *ModuleID*

Command Format	Command Function
Commands for Measuring the Atmospheric Pressure and Altitude	
GetPressure10mBar(*Pressure*)	Get the atmospheric pressure and store the value in ***Pressure*** in units of 0.1 mBar. The retrieved value of ***Pressure*** is an integer within the range of 3000-11000.
GetPressuremBar(*Pressure*)	Get the atmospheric pressure and store the value in ***Pressure*** in units of mBar. The retrieved value of ***Pressure*** is a floating-point value within the range of 300-1100.
GetAltitude10Feet(*Altitude*)	Get the altitude and store the value in ***Altitude*** in units of 0.1 feet. The retrieved value of ***Altitude*** is an integer within the range of -16404~295276.
GetAltitudeFeet(*Altitude*)	Get the altitude and store the value in ***Altitude*** in units of feet. The retrieved value of ***Altitude*** is a floating-point value within the range of -1640~29527.
Commands for Unit Conversion	
ConvmBar2Torr(*Num1, Num2*)	The input value of ***Num1*** in units of mBar is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of Torr. The input value of ***Num1*** should be a floating point value within the range of 300~1100, and then a corresponding floating-point value ***Num2*** will be retrieved.
ConvmBar2Atm(*Num1, Num2*)	The input value of ***Num1*** in units of mBar is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of atm. The input value of ***Num1*** should be a floating-point value within the range of 300~1100, and then a corresponding floating-point value ***Num2*** will be retrieved.
ConvmBar2kgf(*Num1, Num2*)	The input value of ***Num1*** in units of mBar is used as the initial value for the conversion.

	The converted value is stored in ***Num2*** in units of kgf/cm^2. The input value of ***Num1*** should be a floating-point value within the range of 300-1100, and then a corresponding floating-point value ***Num2*** will be retrieved.
ConvTorr2mBar(*Num1, Num2*)	The input value of ***Num1*** in units of Torr is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of mBar. The input value of ***Num1*** should be a floating-point value within the range of 225-825, and then a corresponding floating-point value ***Num2*** will be retrieved.
ConvAtm2mBar(*Num1, Num2*)	The input value of ***Num1*** in units of atm is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of mBar. The input value of ***Num1*** should be a floating-point value within the range of 0.2961-1.0857, and then a corresponding floating-point value ***Num2*** will be retrieved.
Convkgf2mBar(*Num1, Num2*)	The input value of ***Num1*** in units of kgf/cm^2 is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of mBar. The input value of ***Num1*** should be a floating-point value within the range of 0.30591-1.12167, and then a corresponding floating-point value ***Num2*** will be retrieved.
ConvMeter2Feet(*Num1, Num2*)	The input value of ***Num1*** in units of m is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of feet. The input value of ***Num1*** should be a floating-point value within the range of -500~9000, and then a corresponding floating-point value ***Num2*** will be retrieved.
ConvFeet2Meter(*Num1, Num2*)	The input value of ***Num1*** in units of feet is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of m. The input value of ***Num1*** should be a floating-point value within the range of -1640.4~29527.5, and then a corresponding floating-point value ***Num2*** will be retrieved.
Conv10mBar2Torr(*Num1, Num2*)	The input value of ***Num1*** in units of 0.1 mBar is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.1 Torr. The input value of ***Num1*** should be an integer within the range of 3000-11000, and then a corresponding integer ***Num2*** below 65535 will be retrieved.
Conv10mBar2Atm(*Num1, Num2*)	The input value of ***Num1*** in units of 0.1 mBar is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.001 atm. The input value of ***Num1***

	should be an integer within the range of 3000-11000, and then a corresponding integer ***Num2*** below 65535 will be retrieved.
Conv10mBar2kgf(*Num1, Num2*)	The input value of ***Num1*** in units of 0.1 mBar is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.001 kgf/cm^2. The input value of ***Num1*** should be an integer within the range of 3000-11000, and then a corresponding integer ***Num2*** below 65535 will be retrieved.
ConvTorr210mBar(*Num1, Num2*)	The input value of ***Num1*** in units of 0.1 Torr is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.1 mBar. The input value of ***Num1*** should be an integer within the range of 2250-8250, and then a corresponding integer ***Num2*** below 65535 will be retrieved.
ConvAtm210mBar(*Num1, Num2*)	The input value of ***Num1*** in units of 0.001 atm is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.1 mBar. The input value of ***Num1*** should be an integer within the range of 296-1085, and then a corresponding integer ***Num2*** below 65535 will be retrieved.
Convkgf210mBar(*Num1, Num2*)	The input value of ***Num1*** in units of 0.001 kgf/cm^2 is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.1 mBar. The input value of ***Num1*** should be an integer within the range of 305-1121, and then a corresponding integer ***Num2*** below 65535 will be retrieved.
Conv10Meter2Feet(*Num1, Num2*)	The input value of ***Num1*** in units of 0.1 m is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.1 feet. The input value of ***Num1*** should be an integer within the range of -5000~90000, and then a corresponding integer ***Num2*** will be retrieved.
Conv10Feet2Meter(*Num1, Num2*)	The input value of ***Num1*** in units of 0.1 feet is used as the initial value for the conversion. The converted value is stored in ***Num2*** in units of 0.1 m. The input value of ***Num1*** should be an integer within the range of -16404~295275, and then a corresponding integer ***Num2*** will be retrieved.
Commands for Automatic Record	
SetRecordPressureCnt(*Cnt*)	The input value of ***Cnt*** is used as the number of counts for recording the atmospheric pressure. If 0 is input, the module will store every count of the measurement values; while 1 is input, the module will store every other count of the

	measurement values, and so on. The input value of ***Cnt*** can be an integer within the range of 0-65535.
SetRecordAltitudeCnt(*Cnt*)	The input value of ***Cnt*** is used as the number of counts for recording the altitude. If 0 is input, the module will store every count of the measurement values; while 1 is input, the module will store every other count of the measurement values, and so on. The input value of ***Cnt*** should be an integer within the range of 0-65535.
GetRecordPressureCnt(*Cnt*)	Get the number of count for recording the measurement values of the atmospheric pressure and store it in ***Cnt***. The retrieved value of ***Cnt*** is an integer within the range of 0-65535.
GetRecordAltitudeCnt(*Cnt*)	Get the number of count for recording the measurement values of the altitude and store it in ***Cnt***. The retrieved value of ***Cnt*** is an integer within the range of 0-65535.
StartAutoPressureRecord()	Start the automatic recording operation for the atmospheric pressure. Up to 120 record items can be stored. Each time the automatic recording operation is started, it will store the data from the first item. After 120 items are recorded, the oldest item will be over-written automatically.
StartAutoAltitudeRecord()	Start the automatic recording operation for the altitude. Up to 120 record items can be stored. Each time the automatic recording operation is started, it will store the data from the first item. After 120 items are recorded, the oldest item will be over-written automatically.
StopAutoPressureRecord()	Stop the automatic recording operation for the atmospheric pressure.
StopAutoAltitudeRecord()	Stop the automatic recording operation for the altitude.
SaveCurPressure(*Num*)	Store the latest measurement value of the atmospheric pressure into the location specified by ***Num***. The input value of ***Num*** should be an integer within the range of 1-120.
SaveCurAltitude(*Num*)	Store the latest measurement value of the altitude into the location specified by ***Num***. The input value of ***Num*** should be an integer within the range of 1-120.
LoadPressure10mBar(*Num, Pressure*)	Read the value of the atmospheric pressure stored in the location specified by ***Num*** and then store the retrieved value in ***Pressure***. The input value of ***Num*** should be an integer

	within the range of 1-120. The retrieved value of ***Pressure*** will be an integer within the range of 3000-11000.
LoadPressuremBar(*Num, Pressure*)	Read the value of the atmospheric pressure stored in the location specified by ***Num*** and then store the retrieved value in ***Pressure***. The input value of ***Num*** should be an integer within the range of 1-120. The retrieved value of ***Pressure*** will be a floating-point value within the range of 300-1100.
LoadAltitude10Feet(*Num, Altitude*)	Read the value of the altitude stored in the location specified by ***Num*** and then store the retrieved value in ***Altitude***. The input value of ***Num*** should be an integer within the range of 1-120. The retrieved value of ***Altitude*** will be an integer within the range of -16404~295275.
LoadAltitudeFeet (*Num, Altitude*)	Read the value of the altitude stored in the location specified by ***Num*** and then store the retrieved value in ***Altitude***. The input value of ***Num*** should be an integer within the range of 1-120. The retrieved value of ***Altitude*** will be a floating-point value within the range of -1640.4~29527.5.
Commands for the Configuration of Alarm Event	
EnableRefreshEvent()	Enable the measurement refresh event **RefreshEvent** for notifying the update of measurement value.
DisableRefreshEvent ()	Disable the measurement refresh event **RefreshEvent** for notifying the update of measurement value.
***bStatus* = GetRefeshStatus()**	Get the status of the RefreshEvent. The retrieved value ***bStatus*** will be set as 1 after a refresh event occurs. Then, it will be reset as 0 after the status is retrieved and remains 0 till the next refresh event occurs.
SetPressure10mBarAlarmEvent(*Num, Low, High*)	Set the No. of the event to be modified by specifying the value of ***Num***. The values of ***Low*** and ***High*** should be specified to indicate the range for activating the event. The value of ***Num*** should be an integer within 0-7. The values of ***Low*** and ***High*** should be integers within the range of 3000-11000. Note that the value of ***High*** should be greater or equal to the value of ***Low***. Otherwise, the command will be regarded as an invalid command. The default values of ***Low*** and ***High*** are 10140 and 11000 respectively in units of 0.1 mBar.
SetPressuremBarAlarmEvent(*Num, Low, High*)	Set the No. of the event to be modified by specifying the value of ***Num***. The values of ***Low*** and ***High*** should be specified to indicate the range for activating the event. The value of ***Num*** should be an integer

	between 0-7. The values of ***Low*** and ***High*** should be integers within the range of 300-1100. Note that the value of ***High*** should be greater or equal to the value of ***Low***. Otherwise, the command will be regarded as an invalid command. The default values of ***Low*** and ***High*** are 1014 and 1100 respectively in units of mBar.
SetPressureChangeEvent(*Scale*)	Set the value of the allowed pressure change specified by ***Scale*** for the pressure change event. The input value of ***Scale*** should be an integer within the range of 1-800 in units of mBar. The default value is 5.
SetAltitude10FeetAlarmEvent(*Num, Low, High*)	Set the No. of the event to be modified by specifying the value of ***Num***. The values of ***Low*** and ***High*** should be specified to indicate the range for activating the event. The value of ***Num*** should be an integer within 0-7. The values of ***Low*** and ***High*** should be integers within the range of -16404-295275. Note that the value of ***High*** should be greater or equal to the value of ***Low***. Otherwise, the command will be regarded as and invalid command. The default values of ***Low*** and ***High*** are 5000 and 295270 respectively in units of 0.1 feet.
SetAltitudeFeetAlarmEvent(*Num, Low, High*)	Set the No. of the event to be modified by specifying the value of ***Num***. The values of ***Low*** and ***High*** should be specified to indicate the range for activating the event. The value of ***Num*** should be an integer within 0-7. The values of ***Low*** and ***High*** should be floating-point values within the range of -1640.4-29527.5. Note that the value of ***High*** should be greater or equal to the value of ***Low***. Otherwise, the command will be regarded as and invalid command. The default values of ***Low*** and ***High*** are 500 and 29527 respectively in units of feet.
SetAltitudeChangeEvent(*Scale*)	Set the value of the allowed change specified by ***Scale*** of the altitude for the altitude change event. The input value of ***Scale*** should be an integer within the range of 1-30000 in units of feet. The default value is 500.
GetPressure10mBarEvent(*Num, Low, High*)	Set the value of ***Num*** to specify the No. of the event to be retrieved. The retrieved values of ***Low*** and ***High*** are the range values for activating the event. The value of ***Num*** should be an integer within 0-7. The retrieved values of ***Low*** and ***High*** will be integers within the range of 3000~11000. The default values of ***Low*** and ***High*** are

	10140 and 11000 respectively in units of 0.1 mBar.
GetPressuremBarEvent(*Num, Low, High*)	Set the value of ***Num*** to specify the No. of the event to be retrieved. The retrieved values of ***Low*** and ***High*** are the range values for activating the event. The value of ***Num*** should be an integer within 0-7. The retrieved values of ***Low*** and ***High*** will be floating-point values within the range of 300-1100. The default values of ***Low*** and ***High*** are 1014 and 1100 respectively in units of mBar.
GetPressureChangeEvent(*wScale*)	The retrieved value of ***wScale*** is the value of pressure change for the pressure change event which is an integer within the range of 1-800 in units of mBar. The default value is 5.
GetAltitude10FeetEvent(*Num, Low, High*)	Set the value of ***Num*** to specify the No. of the event to be retrieved. The retrieved values of ***Low*** and ***High*** are the range values for activating the event. The value of ***Num*** should be an integer within 0-7. The retrieved values of ***Low*** and ***High*** will be floating-point values within the range of -16404~295275. The default values of ***Low*** and ***High*** are 5000 and 295270 respectively in units of 0.1 feet.
GetAltitudeFeetEvent(*Num, Low, High*)	Set the value of ***Num*** to specify the No. of the event to be retrieved. The retrieved values of ***Low*** and ***High*** are the range values for activating the event. The value of ***Num*** should be an integer within 0-7. The retrieved values of ***Low*** and ***High*** will be floating-point values within the range of -16404~295275. The default values of ***Low*** and ***High*** are 5000 and 295270 respectively in units of feet.
GetPressureChangeEvent(*wScale*)	The retrieved value of ***wScale*** is the value of pressure change for the pressure change event which is an integer within the range of 1-800 in units of feet. The default value is 500.
EnablePressureAlarmEvent(*Num*)	Enable the pressure alarm event **PressureAlarmEvent** with the event No. specified by ***Num***. The input value of ***Num*** should be an integer within the range of 0-7.
EnablePressureChangeEvent()	Enable the pressure change event **PressureChangeEvent.**
EnableAltitudeAlarmEvent(*Num*)	Enable the altitude alarm event **AltitudeAlarmEvent** with the event No. specified by ***Num***. The input value of ***Num*** should be an integer within the range of 0-7.
EnableAltitudeChangeEvent()	Enable the altitude change event **AltitudeChangeEvent.**

DisablePressureAlarmEvent(*Num*)	Disable the pressure alarm event **PressureAlarmEvent** with the event No. specified by ***Num***. The input value of ***Num*** should be an integer within the range of 0-7.
DisablePressureChangeEvent()	Disable the pressure change event **PressureChangeEvent.**
DisableAltitudeAlarmEvent(*Num*)	Disable the altitude alarm event **AltitudeAlarmEvent** with the event No. specified by ***Num***. The input value of ***Num*** should be an integer within the range of 0-7.
DisableAltitudeChangeEvent()	Disable the altitude change event **AltitudeChangeEvent.**
***bStatus* = GetPressureAlarmStatus()**	Get the status of the pressure alarm event. The retrieved value of ***bStatus*** represents the status of the event of the corresponding No. by the variation of each bit. If a pressure change event is enabled and the pressure is within the preset range for the corresponding event, the corresponding bit will be 1. If the event is not enabled or the pressure is not within the preset range, the corresponding bit will be 0. For example, if only the No. 1 event is enabled and the measured pressure value is within the preset range, the returned value of ***bStatus*** will be 2.
***bStatus* = GetAltitudeAlarmStatus()**	Get the status of the altitude alarm event. The retrieved value of ***bStatus*** represents the status of the event of the corresponding No. by the variation of each bit. If an altitude change event is enabled and the altitude is within the preset range for the corresponding event, the corresponding bit will be 1. If the event is not enabled or the altitude is not within the preset range, the corresponding bit will be 0. For example, if only the No. 1 event is enabled and the measured pressure value is within the preset range, the returned value of ***bStatus*** will be 2.
EnableRecPOverEvent()	Enable the pressure recording storage overflow event **RecPOverEvent.**
DisableRecPOverEvent ()	Disable the pressure recording storage overflow event **RecPOverEvent.**
EnableRecAOverEvent()	Enable the altitude recording storage overflow event **RecAOverEvent.**
DisableRecAOverEvent ()	Disable the altitude recording storage overflow event **RecAOverEvent.**
Commands for Various Settings	
SetMode(*Mode*)	Set the pressure measurement mode by the values of ***Mode***. The default value is 0. The input value of ***Mode*** should be 0 or 1. ***Mode*** = 0 ➔ Perform 4 measurements per second and then the average of the 4 values will be used as the final measurement value. Only one measurement value is updated per

	second. ***Mode*** = 1 ➔ Perform 10 measurements per second. The measurement value is updated every 0.1 second.
GetMode(*Mode*)	Get the pressure measurement mode. The returned value of ***Mode*** will be 0 or 1.
SetSeaLevelPressure(*Pressure*)	Set the pressure at sea level by entering the value of ***Pressure***. The input value of ***Pressure*** should be an integer within the range between 3000 and 11000 in units of 0.1 mBar. As the pressure at sea level is set, the retrieved altitude will be affected because the altitude is calculated according to both the pressure at sea level and the measured pressure value from the barometer.
GetSeaLevelPressure(*Pressure*)	Get the pressure at sea level and store the value in ***Pressure***. The returned value of ***Pressure*** is an integer within the range between 3000 and 11000 in units of 0.1 mBar.

Table 1 :Command Table

Event	Activation Condition
PressureAlarmEvent	After the command **EnablePressureAlarmEvent** (***Num***) is executed, if the measured atmospheric pressure is within the preset range for the event which is specified by ***Num***, then the corresponding event will be activated.
PressureChangeEvent	After the command **EnablePressureChangeEvent** () is executed, if the difference between the measured atmospheric pressure and the startup pressure exceeds the preset value, the corresponding event will be activated.
AltitudeAlarmEvent	After the command **EnableAltitudeAlarmEvent** (***Num***) is executed, if the measured altitude is within the preset range for the event that is specified by ***Num***, then the corresponding event will be activated.
AltitudeChangeEvent	After the command **EnableAltitudeChangeEvent** () is executed, if the difference between the measured altitude and the startup altitude exceeds the preset value, the corresponding event will be activated.
RefreshEvent	After the command **EnableRefreshEvent()** is activated, once the measurement value is updated, the corresponding event will be activated.
RecPOverEvent	While the command **EnableRecPOverEvent** is executed and the automatic recording for the pressure is enabled, if the number of record items exceeds 120, this even is activated.
RecAOverEvent	While the command **EnableRecAOverEvent** is executed and the automatic recording for the altitude is enabled, if the number of record items exceeds 120, this even is activated.

Table 2 :Event Table

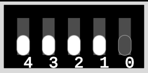
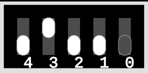
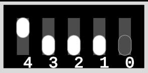
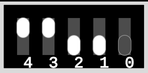
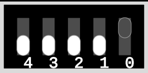
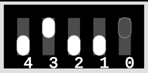
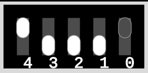
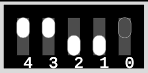
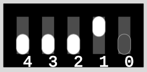
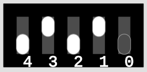
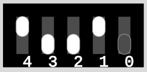
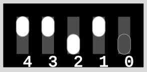
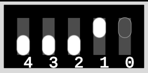
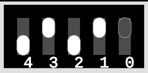
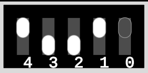
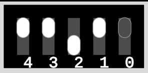
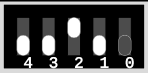
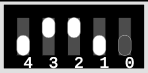
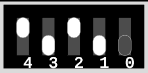
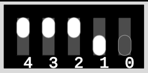
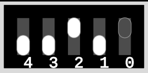
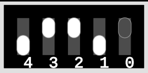
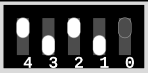
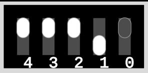
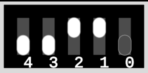
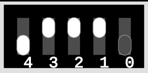
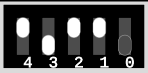
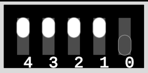
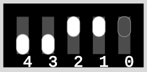
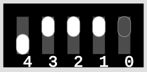
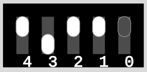
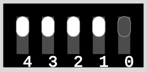
Demonstration Program:

```
Peripheral myB As BarometerA @ 0                        '   Set the module ID as 0

Dim g_wP As Word                                        '   Store the retrieved atmospheric pressure
Dim g_lA As Long                                        '   Store the retrieved altitude

Sub Main()                                              '   Main program starts here
    Dim fP As Float                                     '   Store the input pressure value at sea level

'-----------------------------------------------------------------------------------------
'   Do loop. The loop is terminated by entering a reasonable sea level.
'-----------------------------------------------------------------------------------------
    Do
        Debugin "Please enter the pressure at sea level: ", fP  '   Display the inquiry message on the terminal
    Loop Until (fP>300 And fP<1100)                     '   The loop is terminated by entering a number in the
                                                            range of 300-1100

    Debug fP                                            '   Display the last input value
    g_wP = Float2word(fP*10)                            '   Convert the input value into an integer
    myB.SetSeaLevelPressure(g_wP)                       '   Set the pressure at sea level
    myB.EnableRefreshEvent()                            '   Enable the measurement refresh event

'-----------------------------------------------------------------------------------------
'   An infinite loop, while the program stays in this loop, it will receives new measurement values continuously
'   and display the updated measurement value on the terminal window.
'-----------------------------------------------------------------------------------------
    Do
    Loop
End Sub

Event myB.RefreshEvent()                                '   The measurement refresh event
    myB.SetSeaLevelPressure(g_wP)                       '   Get the pressure value in the integer format
    myB.GetAltitude10Feet(g_lA)                         '   Get the altitude value in the integer format
    g_wP = g_wP \ 10                                    '   Convert the unit of the pressure into mBar.
    g_lA = g_lA \ 10                                    '   Convert the unit of the altitude into feet
    Debug CSRXY(1, 3), "Pressure: ", %DEC4 g_wP         '   Display the pressure value on the terminal window
    Debug CSRXY(1, 4), "Altitude: ", %DEC4 g_lA         '   Display the altitude value on the terminal window
End Event
```

Appendix

1. Known Problem:

2. List of the Configuration of the Module ID Switch:

	0		8		16		24
	1		9		17		25
	2		10		18		26
	3		11		19		27
	4		12		20		28
	5		13		21		29
	6		14		22		30
	7		15		23		31