Sonar A

Ultra-sonic range finder

User's Guide

Version: 1.0

Innovati's Sonar A module can be directly controlled by Innovati's BASIC Commander® via simple connections. Through simple software functions, Sonar A module receives user instructions and measures distances or measurement counts according to user demands under proper environmental conditions. Please use "SonarA" as the module object name in program.

Trademark

Due to continual product improvements, Innovati reserves the right to make modifications to its products without prior notice. Innovati does not recommend the use of its products for application that may present a risk to human life due to malfunction or otherwise.

Disclaimer

Full responsibility for any applications using Innovati products rests firmly with the user and as such Innovati will not be held responsible for any damages that may occur when using Innovati products. This includes damage to equipment or property, personal damage to life or health, damage caused by loss of profits, goodwill or otherwise. Innovati products should not be used for any life saving applications as Innovati's products are designed for experimental or prototyping purposes only. Innovati is not responsible for any safety, communication or other related regulations. It is advised that children under the age of 14 should only conduct experiments under parental or adult supervision.

Errata

We hope that our users will find this user's guide a useful, easy to use and interesting publication, as our efforts to do this have been considerable. Additionally, a substantial amount of effort has been put into this user's guide to ensure accuracy and complete and error free content, however it is almost inevitable that certain errors may have remained undetected. As Innovati will continue to improve the accuracy of its user's guide, any detected errors will be published on its website. If you find any errors in the user's guide please contact us via email service@innovati.com.tw. For the most up-to-date information, please visit our web site at http://www.innovati.com.tw.

Table Of Content

Product Overview

Innovati's Sonar A module can be directly controlled by Innovati's BASIC Commander® via simple connections. Through simple software functions, Sonar A module receives user instructions and measures distances or measurement counts according to user demands under proper environmental conditions. Please use "SonarA" as the module object name in program.

Application

- Range-finding tools
- Obstacle avoidance to protect moving devices, such as robots, auto-piloted vehicles, etc.

Product Features

- Easy usage with functions such as Ranging and RepeatRanging.
- System events: combined with RepeatRanging, system events allow real-time measurements.
- Customized formats of returned measurements. (us, cm, and inch)

Product Specifications

- Supply voltage：6~12 VDC
- Size：23.4mm H × 48.3mm W × 18mm D
- Weight：11g (0.39oz)
- Pin assignments:

Pins	VIN	GND	SDA	SCL	EVT	SYN
Description	External Power supply	Ground	Data signal	Clock signal	Event Signal	Synchronization signal

Table 1: Pin Assignments

- Detection Limits:

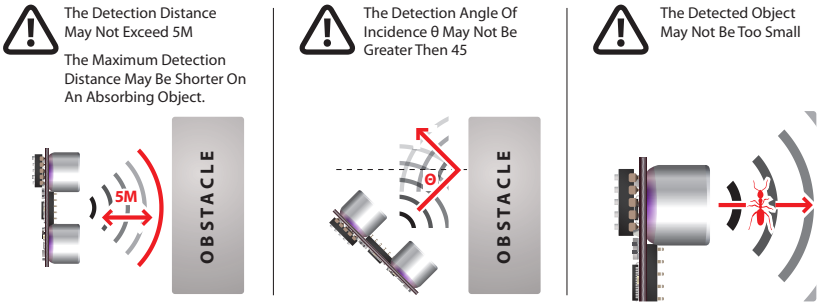


Figure 1: Detection Limits

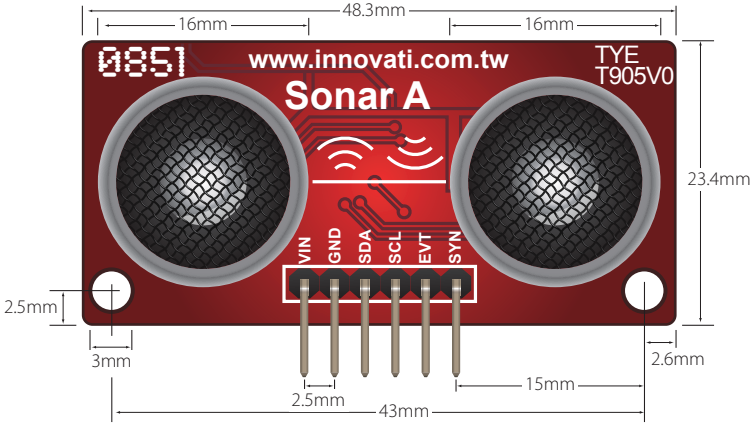
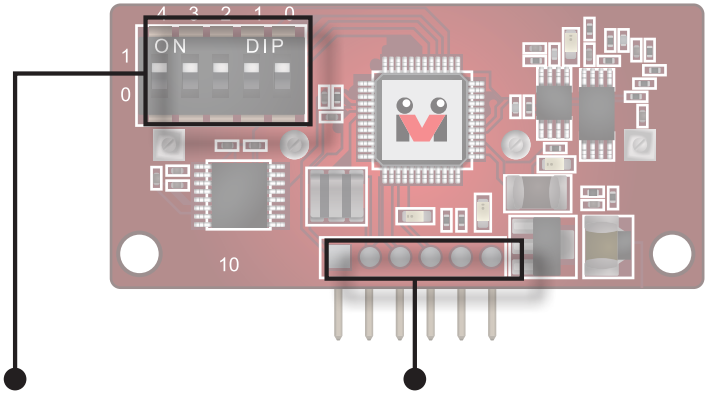


Figure 2: Front View



Module ID Switches:
These switches determines Sonar A's module ID in binary format, from right to left. Module IDs enable the BASIC Commander® to distinguish those modules under its control. (Please refer to Appendix 1.) Commander®. Incorrect pin connection may damage both modules.)

cmdBUS™ Pins:
To access Sonar A module through the BASIC Commander®, connect these pins to the corresponding pins on the BASIC Commander®. (When connecting Sonar A, please pay attention to pin assignments. For example, the Vin on the Sonar A module should be connected to the Vin on the BASIC Commander®. Incorrect pin connection may damage both modules.)

Figure 3: Back View

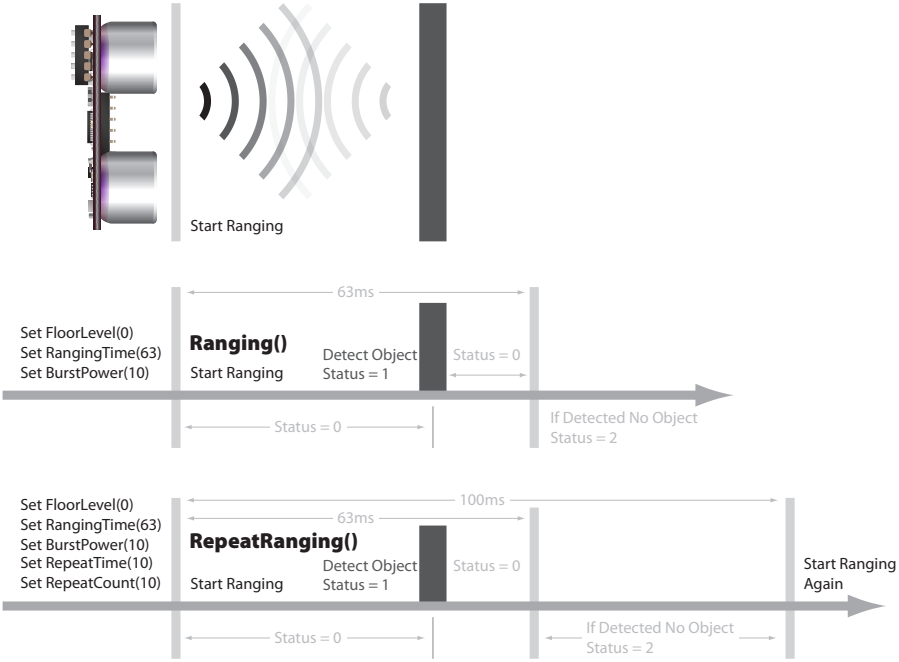


Figure 4: Illustration of a ranging operation

Connection

To access Sonar A through the BASIC Commander®, set the ID switches to the desired ID settings, and connect the cmdBUS™ to the proper pins on the BASIC Commander®.

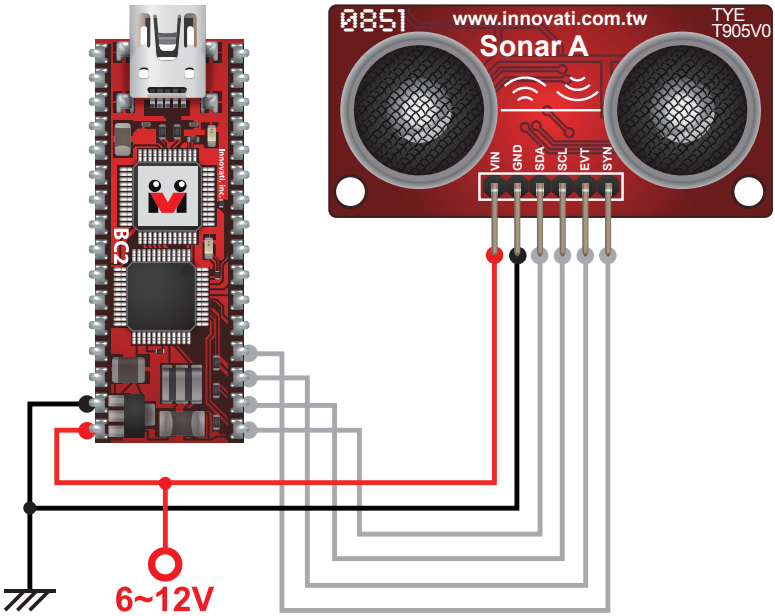


Figure 5: Connection

Absolute Maximum Ratings:

Operating temperature 0℃~70℃

Storage temperature -30℃~80℃

The ranging accuracy depends on the surface smoothness and sound absorption capability of the detected object.

Commands and Events

The following table lists all the unique commands provided with the Sonar A Module. Note that essential words in the commands will be written in **bold** type and ***italics*** in bold type. The bold type word must be written exactly as shown, whereas the italic bold type words must be replaced with the user values. Note that the innoBASIC™ language is case-insensitive.

Command Format	Description
Commands for ultra-sonic ranging setup	
SetFloorLevel(*FloorLevel*)	Set the distance of Sonar A module to the floor with the argument, ***FloorLevel***. (0: distance to the floor > 5cm, 1: distance to the floor > 20cm, 2: distance to the floor > 50cm, 3: close or small object detection. The default value is 0.) See Note 1.
GetFloorLevel(*FloorLevel*)	Read the current floor level settings and saves it in ***FloorLevel***.
SetRangingTime(*RangingTime*)	Set the wait time for Sonar A to receive ultrasonic wave after transmission. The allowed ranges is 0~63. (Unit: ms. The default value is 63 ms. The wait time is 1ms when ***RangingTime*** is zero.)
GetRangingTime(*RangingTime*)	Read the current wait time setting for Sonar A to receive ultrasonic wave after transmission and saves it in ***RangingTime***.
Ranging( )	Transmit and receives ultrasonic waves
RepeatRanging( )	Repeatedly (periodically) perform ultrasonic ranging.
SetRepeatTime(*RepeatTime*)	Set the interval of repeated ranging. The allowed range is 0~255. (Unit: Zero ***RepeatTime*** is defined as 10ms.)
GetRepeatTime (*RepeatTime*)	Read the interval of repeated ranging and saves it in ***RepeatTime***.

Command Format	Description
SetRepeatCount(*RepeatCount*)	Set the number of repeated ranging measurements. Allowed ranges is 1 ~ 255 and 0 = infinity.
GetRepeatCount (*RepeatCount*)	Read the number of repeated ranging measurements and saves it in ***RepeatCount***.
SetBurstPower (*BurstPower*)	Set the power of the transmitted ultrasonic wave. The allowed range is 0~10. (The power increases from 0 to 10. The default value is 10.)
GetBurstPower (*BurstPower*)	Read the power of the transmitted ultrasonic wave and saves it in ***BurstPower***.
StopRanging ( )	Stop transmitting ultrasonic waves.
Status＝GetDistance (*Type*, *Distance*)	Save the reception status in **Status**, specifies the result format with ***Type***, and saves the result in ***Distance***. (**Status**：0=not ready，1=ready，2=time out； ***Type***：0 = us，1 = cm，2 = inch。)
EnableRangingFinishEvent( )	Enable ***RangingFinishEvent*** notification.
DisableRangingFinishEvent( )	Disable ***RangingFinishEvent*** notification.

Table 2: Command Table

Event Name	Description
RangingFinishEvent( )	After **EnableRangingFinishEvent**( ) is executed, this event will be activated when Sonar A module finishes a ranging measurement. See Note 2.

Table 3: Event Provided By The Module

Note 1:The floor level depends on the floor material.

Note 2:When the event is enabled and triggered, it is necessary to read the value of Status. Otherwise, this event will be triggered repeatedly.

Example Program

Example A: Single Ranging

```
Peripheral mySonar As SonarA @ 0                'Set the module ID as 0

Dim Distance As Word        'Save the acquired ranging result.

Sub MAIN()

mySonar.SetFloorLevel(0)    'Set the floor level to be larger than 5cm.
mySonar.SetRangingTime(63)  'Set the ranging time as 63 ms.
mySonar.SetBurstPower(10)   'Set the bust power as 10.
Do
    mySonar.Ranging()                           'Perform repeated ranging.
    Pause 100      'Wait 100ms
    mySonar.GetDistance(1,Distance)             'Read the ranging result.
Loop
    Debug "Distance=",Distance,"cm",CR          'Display the ranging result.

End Sub
```

Example B: Repeated Ranging

```
Peripheral mySonar As SonarA @ 0      'Set the module ID as 0

Dim Distance As Word        'Save the acquired ranging result.

Sub MAIN()
mySonar.SetFloorLevel(0)    'Set the floor level to be larger than 5cm.
mySonar.SetRangingTime(63)  'Set the ranging time as 63 ms.
mySonar.SetRepeatTime(20)   'Set the repeating interval as 200ms.
mySonar.SetRepeatCount(0)   'Set the repeating count as 0 (infinity)
mySonar.SetBurstPower(10)   'Set the bust power as 10.

Debug "Distance="
    mySonar.RepeatRanging() 'Perform repeated ranging
    mySonar.EnableRangingFinishEvent()          'Enable the RangingFinishEvent
    Do                                          'Infinite loop
    Loop

End Sub

Event mySonar.RangingFinishEvent()
    mySonar.GetDistance(1,Distance)             'Read the ranging result
    Debug CSRXY(10,1), CLREOS, Distance, " CM", CR       'Display the result.
End Event
```

Appendix

Module ID Setting Table

DIP Switch	ID	DIP Switch	ID	DIP Switch	ID	DIP Switch	ID
	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

Table 4: Module ID Setting Table