Innobot™

User's Manual

Document Rev. 2.0

Apr. 15, 2014

Passion for innovation

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.

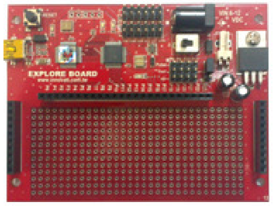
Table of Contents

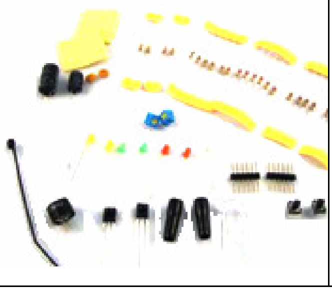
Product Overview

Innovati's Innobot^{TM} kit includes the Explore Board, aluminum chassis, servos and wheels. The Explore Board has a built-in BASIC Commander® module which supports 24 general purpose I/Os. Four of the I/Os have also been arranged with servo connectors for servo connection. Four cmdBUS^{TM} connectors are available on the board, users may plug in Innovati's smart modules for more complex applications. A bread board is included in this kit, users may put it on the Explore Board with its adhesive foam and make simple electronics and robotics experiments.

The Innobot^{TM} kit is specially designed for basic autonomous line follower. It is also an excellent platform for other robotics exploration.

Part List

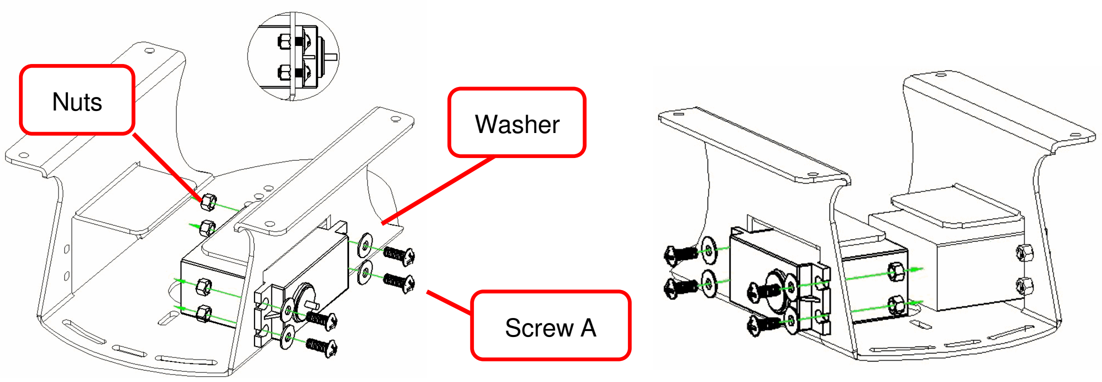
Item	Illustration	Qty.	Description
Explore Board		1	The Explore Board is the main control board of the Innobot kit. It has cmdBUS connectors for connecting Innovati's smart modules. The auxiliary Bread board allows users to add additional electronic components.
Aluminum Chassis		1	Chassis for installing the servos, wheels and Explore Board.
Continuous rotation servo		2	Continuous rotation servos. Pay attention to the polarity of the cable while connecting to the board. 42 x 20.5 x 39.5 (mm), weight: 44 g.
Servo Accessories		1	Servo horn screw and other accessories.

Dual Wheel Set		1	Two plastic wheels with rubber tires.
Ball Transfer Unit		1	Assists the dual wheels for the movement of the robot body.
Screw A		10	ISOT 3 x 8 mm
Screw B		4	ISOP 3 x 6 mm
Screw E		6	TP1P 2 x 5 mm
Copper Hex Post		4	3 x 6 mm
Nut		14	3 x 5 mm
Washer		8	3 x 0.4 x 8 mm
Battery Holder		1	Four AA size alkaline batteries recommended
Jumper Wires		1	Wires for the connections of experimental circuits
Electronic Components		1	LED (Red*2, Yellow*2, Green*2), capacitor (0.1μF*2, 330μF*1, 1000μF*1), resistor (220Ω*6, 330Ω*6, 1KΩ*4, 2.2KΩ*4, 10KΩ*4, 100KΩ*2, 220KΩ*2, 330KΩ*2), 5KΩ variable resistor*2, phototransistor*2, IR receiver*2, IR transmitter*2, insulation

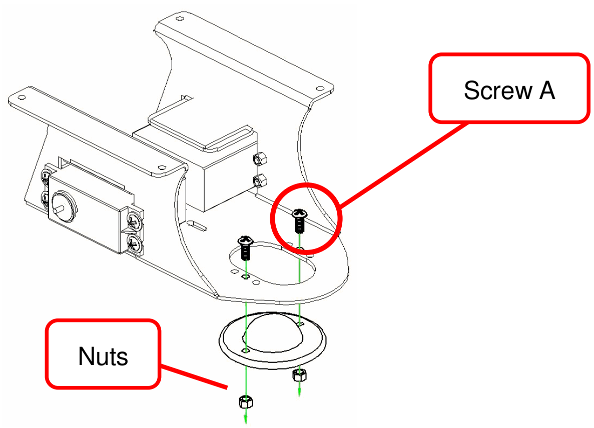
			tube*2, cable ties, foam*2, buzzer, button*2, 6-pin header*2, etc.
Electrical Tape (black)		1	Used as route for IR sensing.
USB Cable		1	Connecting Explore Board and PC, for program downloading, or human-machine interface for debugging.

Assembly Procedure

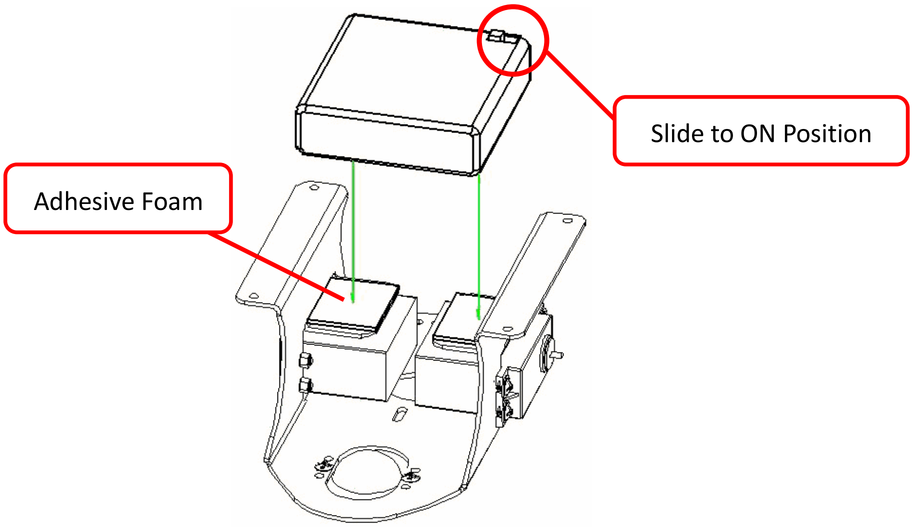
Step 1. Installing Servos



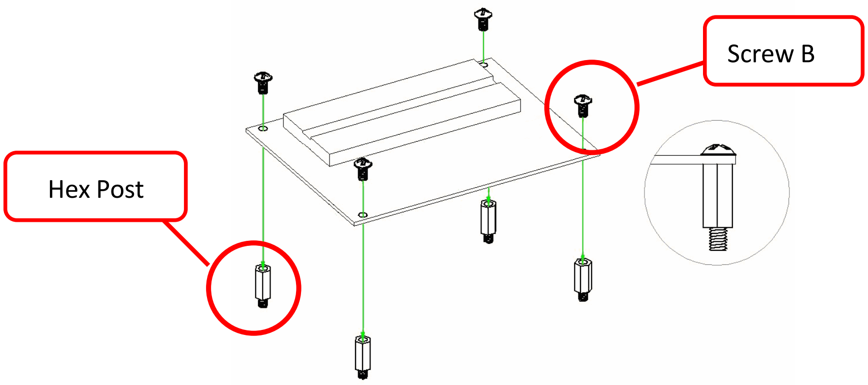
Step 2: Installing Ball Transfer Unit

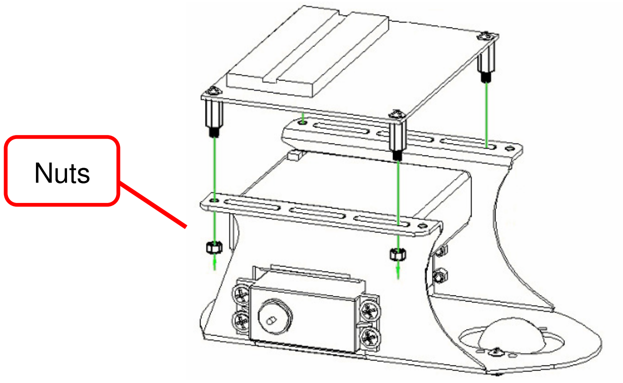


Step 3: Installing the Battery Holder

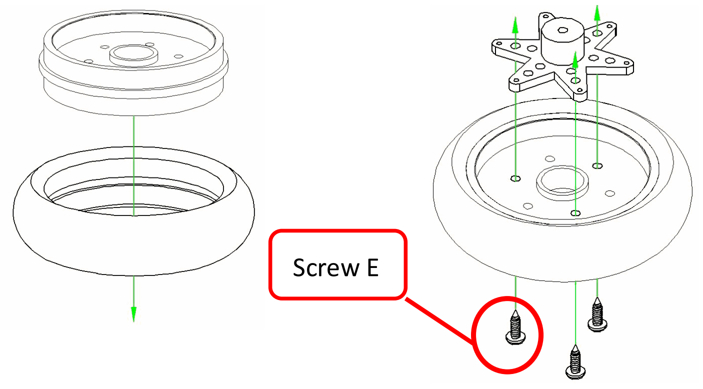


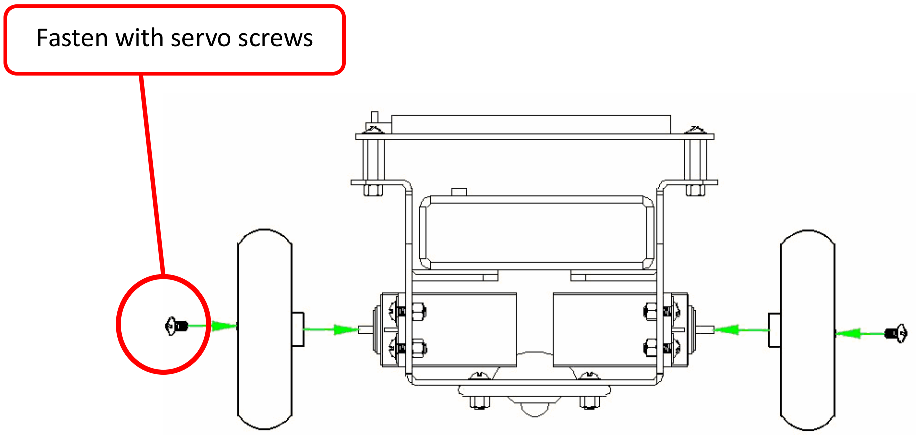
Step 4: Installing the Explore Board





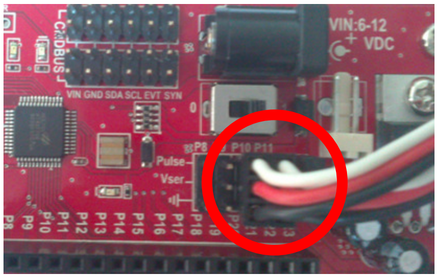
Step 5: Installing the wheels

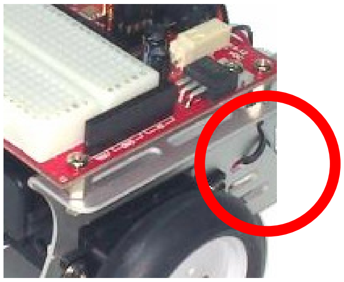




Step 6: Connecting the servo cables

To be compliant with the tutorial program, connect the left wheel servo to P10, and the right wheel servo to P11. Fix the servo cables on the Innobot aluminum chassis with cable ties

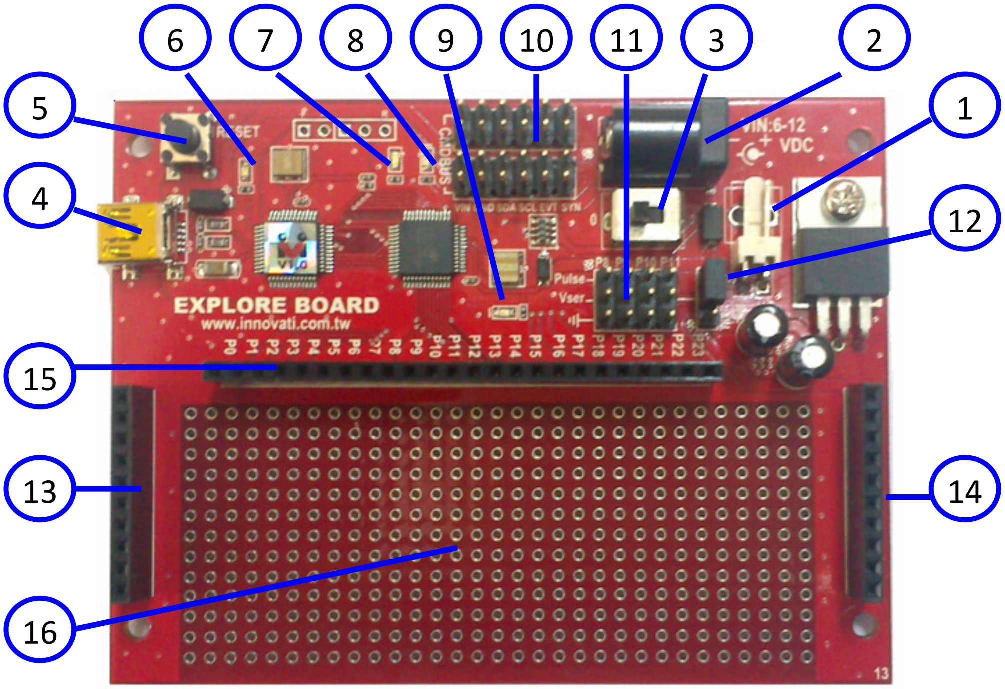




.

Innobot after the installation is complete

Explore Board


6
7
8
9
10
11
3
2
5
1
4
12
15
13
14
16
RESET
VIN:6-12 VDC
EXPLORE BOARD
www.innovati.com.tw
P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22 P23
Pulse
Vser

Item	Description
1	External power supply connector. Input voltage ranges DC 6V~12V denoted as VIN. Innobot battery holder is connected to this position
2	Optional external power adaptor connector. Input voltage ranges DC 6V~12V denoted as VIN.
3	Power on/off switch of external power (VIN).

4	Mini USB connector. Connecting to computer via USB cable for program downloading and debugging.
5	RESET Button. To restart the program while the program is in execution. Do not press the RESET button during downloading, which will result in download failure.
6	Green color Status LED will blink when Explore Board is communicating with PC.
7	Yellow color Event LED will blink when Explore Board is communicating with peripheral modules.
8	Red color VDD power indicator. The 5V VDD is regulated from the external VIN power supply.
9	Red color VCC power indicator, which may come from either the USB port or VDD.
10	Four cmdBUS™ connectors. Using 6-wired cmdBUS™ cables for connecting with Innovati's peripheral modules.
11	Four Servo Connectors. Each of them is arranged with VDD (or VIN) and GND power supply and control I/O as P8, P9, P10 and P11. The white servo cable is for control signal, red for power and black for ground. They are denoted as W, R and B next to the servo connectors.
12	Servo power source selection jumpers. Short jumpers at the upper position, the 5V DVV power will be used; Short jumpers at the lower position the external VDD power will be used.
13	VIN, VDD and VCC power supply is available on this female connector. The maximum VDD supply current is 1 Amp.
14	GND of power supply is available on this female connector.
15	24 general-purpose digital I/Os with labeled pin numbers P0~P23 on the board. Through the built-in software commands, they can be used as I2C or UART pins.
16	Prototype PCB area for soldering user's own electronic circuitry. User can attach the provided Bread Board on this area for his/her own circuitry.

Appendix --- Tutorial Programs

To help you be familiar with the Innobot Kit, some tutorial programs with brief introduction are provided in this section.

To maintain the tutorial programs free of error and up-to-date, they are subject to change without notice. For new users, who are not familiar with the BASIC Commander®, please refer to the "BASIC Commander® and innoBASIC™ Workshop User's Manual" for more detailed information.

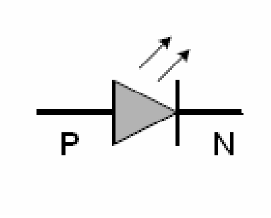
Ex. 1 --- Control LED in different ways

LED and Resistor Basics

An LED (Light Emitting Diode) is a device which converts electrical energy into light emission. Different compound materials, such as InGaN, AlInGaP, GaAs, etc., allow the LEDs to emit the light of different colors. LED has the advantages of power-saving, compact, long lifespan, fast response time, low polluting, high reliability, high module flexibility, etc.

LED has polarity, the P electrode has a long lead that should be supplied with the positive voltage and the N electrode has a short lead that should be supplied with the negative voltage.

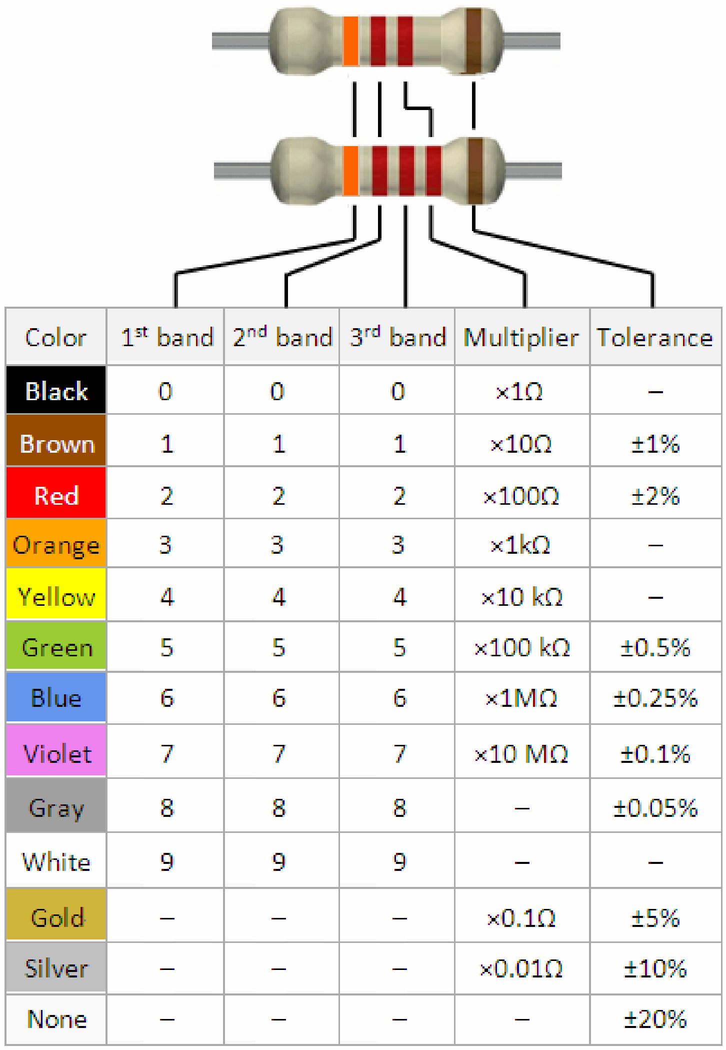
A Red LED



Circuit symbol

The driving voltage of the LED is typically within the range of 2.2V ~3.4V. However, the output voltage of the I/O pins is about 5V with a supplied current of approx. 10mA. If such a voltage is directly applied on the LED, it may damage the LEDs permanently. Therefore, it should be connected with the resistor to limit the current.

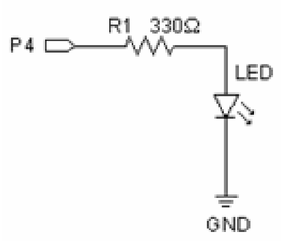
A resistor is a common component in electronic circuits which limit the current flowing in the electric wire and is usually represented as “R” in Ω (Ohm). Resistors have different packages and specifications. The resistance of a resistor can be identified by the color bands. See the resistor color code table below.

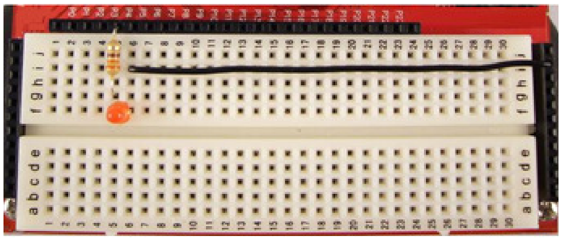


Color	1st band	2nd band	3rd band	Multiplier	Tolerance
Black	0	0	0	×1Ω	–
Brown	1	1	1	×10Ω	±1%
Red	2	2	2	×100Ω	±2%
Orange	3	3	3	×1kΩ	–
Yellow	4	4	4	×10 kΩ	–
Green	5	5	5	×100 kΩ	±0.5%
Blue	6	6	6	×1MΩ	±0.25%
Violet	7	7	7	×10 MΩ	±0.1%
Gray	8	8	8	–	±0.05%
White	9	9	9	–	–
Gold	–	–	–	×0.1Ω	±5%
Silver	–	–	–	×0.01Ω	±10%
None	–	–	–	–	±20%

According to Ohm's law, assuming the voltage across the operating LED is 2.2V, in order to limit the current passing through it within the allowed range of the BASIC Commander I/Os (10mA), the resistance of R1 should satisfy the relation (5-2.2)/R1 < 10mA course. It is suitable to choose a resistance of 330Ω to maintain the current at approximate 8.4 mA. The color code for 330Ω resistor will be “Orange Orange Brown Gold”.

Circuit Diagram





Program Code

The objective of this exercise is to understand the basic operation of lighting an LED in different way using different commands. First, the LED will be turned on by HIGH command for 3 seconds and then off by LOW command for 1 second. Then use the FOR LOOP and Pulseout commands to increase the light on the LED gradually from the off state and finally the LED is at its full brightness.

```
Sub Main()
Dim X As Word             'declare X as a 2-byte unsigned variable ranging 0~65535
'using HIGH, LOW command to make the LED blinking
    High 4                'set the Pin 4 to high level (5V)
    Pause 3000            'wait 3000 ms (3 sec.)
    Low 4                 'set the Pin 4 to low level (0V)
    Pause 1000            'wait 1000 ms (1 sec.)

'using Pulseout command to make the LED dimming
    For X=0 To 2000       ' Declare a loop of 2001 times
        Pulseout 4,X      'Generate a pulse with a width of X on the Pin 4
        Pause 10          'Force the program to wait 10 ms (0.01 sec.)
    Next
End Sub
```

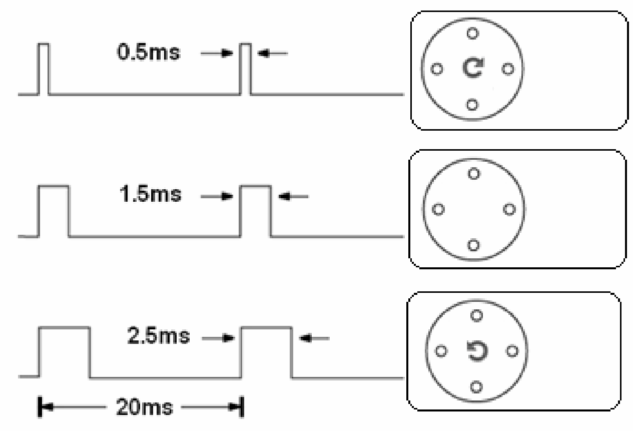
Ex. 2 --- Control Continuous Rotation Servo

Continuous Rotation Servo Basics

A continuous rotation is similar to the conventional RC servos, which internally comprises a small DC motor, a reduction gear set, a feedback potentiometer and an electronic control board. The initial power is provided by the DC motor rotating at a high speed and transmitted to the reduction gear set to generate a high-torque output. The larger the gear ratio of the reduction gear set is, the larger the output torque of the servo will be. Thus, it can withstand a larger force but the rotation speed will be decreased accordingly.

For an RC servo, its output shaft drives a proportional potentiometer for the position detection. The potentiometer will convert the rotational coordinate into a proportional voltage feedback to the control circuit board, so the control circuit board will compare and generate a correction pulse to drive the motor to rotate clockwise or counterclockwise so as to keep at the desired position.

For a continuous rotation servo, control pulse signal is the same, but it is used to control the rotation direction and speed instead the position of the shaft. For example, a train of 1.5ms high pulse signal every 20 ms, the servo shaft will be in the stationary state; with the pulses width less than 1.5ms, it rotates clockwise; the smaller the pulse width is, the faster it rotates; with the pulses larger than 1.5ms, it rotates counterclockwise, the wider the pulse width is, the faster it rotates. Servos vary in the control pulse width, the range 0.8ms ~2.2ms is a reference value. Check datasheets of the servo before using it it. Out of range control signal may damage the servo permanently.

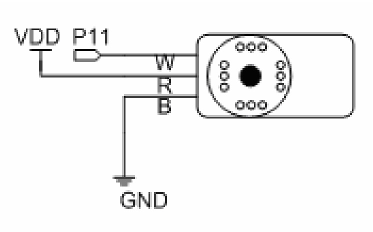


The servo has three power wires as shown in figure below. The white wire is the control wire that is connected to the control chip. The middle red wire is the power line for the operation of the servo, which is typically 5 volts. The third wire is the ground wire.

The objective of this exercise is to control a continuous rotation servo by using the Pulseout command. The servo will make a fast clockwise rotation, a slow clockwise rotation, stop, a slow counterclockwise a fast counterclockwise rotation. Note that each servo has a slightly different pulse width threshold. It should be adjusted according to actual conditions.

Circuit Diagram

While connecting the circuit, use the servo pin header of pin 11 on the Explore Board. Pay attention to the color of the wires W(white), R(red) and B(black) when inserting.



Program Code

The objective of this exercise is to control a continuous rotation servo by using the Pulseout command. The servo will make a fast clockwise rotation, a slow clockwise rotation, stop, a slow counterclockwise a fast counterclockwise rotation. Note that each servo has a slightly different pulse width threshold. It should be adjusted according to actual conditions.

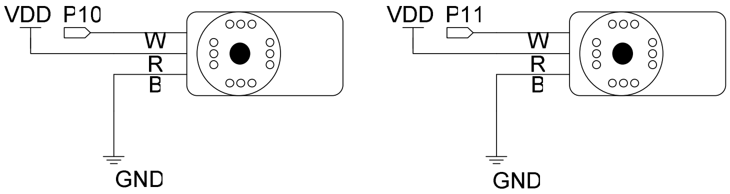
```
Sub Main()
Dim X As Byte        'declare X as a 1-byte unsigned index ranging 0~255
    Low 11
    Do   'infinite loop
        'Fast clockwise rotation
        For X=0 To 99                   'sending 100 pulses
            Pulseout 11,100             'a pulse of 0.5ms in width on Pin 11
            Pause 19                    'wait for 19 ms to make a 20ms period
        Next

        'Slow clockwise rotation
        For X=0 To 99                   'sending 100 pulses
            Pulseout 11,280             'a pulse of 1.4ms in width on Pin 11
            Pause 19                    'wait for 19 ms to make a 20ms period
        Next

        'Stop rotation
        For X=0 To 99                   'sending 100 pulses
            Pulseout 11,300             'a pulse of 1.5ms in width on Pin 11
            Pause 18                    'wait for 18 ms to make a 20ms period
        Next

        'Slow counterclockwise rotation
        For X=0 To 99                   'sending 100 pulses
            Pulseout 11,320             'a pulse of 1.6ms in width on Pin 11
            Pause 18                    'wait for 18 ms to make a 20ms period
        Next

        'Fast counterclockwise rotation
        For X=0 To 99                   'sending 100 pulses
            Pulseout 11,500             'a pulse of 2.5ms in width on Pin 11
            Pause 17                    'wait for 17 ms to make an approx. 20ms period
        Next
    Loop
End Sub
```

Ex. 3 --- Control the Innobot to Move

The objective of this exercise is to extend what we learned to control two continuous rotation servos to make the Innobot move around.

Circuit Diagram

While connecting the circuit, use the servo pin header of pin 10 and 11 on the Explore Board. Pay attention to the color of the wires W(white), R(red) and B(black) when inserting.



Program Code

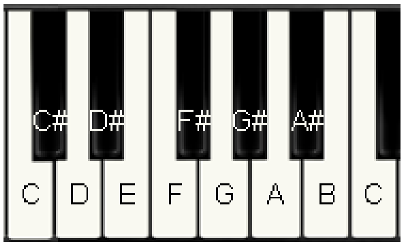
```
Sub Main()
Dim X As Byte       'declare X as a 1-byte unsigned index ranging 0~255
    Low 10          'left servo
    Low 11          'right servo
    Do  'infinite loop
        'Fast forward
        For X=0 To 49                   'sending 50 pulses
            Pulseout 10,500             'a pulse of 2.5ms in width on Pin 10
            Pulseout 11,100             'a pulse of 0.5ms in width on Pin 11
            Pause 17                    'wait for 17 ms to make an approx. 20ms period
        Next

        'Fast backward
        For X=0 To 49                   'sending 50 pulses
            Pulseout 10,100             ' pulse of 0.5ms in width on Pin 10
            Pulseout 11,500             'a pulse of 2.5ms in width on Pin 11
            Pause 17                    'wait for 17 ms to make an approx. 20ms period
        Next

        'Stop moving
        For X=0 To 99                   'sending 100 pulses
            Pulseout 10,300             'a pulse of 1.5ms in width on Pin 10
            Pulseout 11,300             'a pulse of 1.5ms in width on Pin 11
            Pause 17                    'wait for 17 ms to make an approx. 20ms period
        Next
        'Turn right
        For X=0 To 99               'sending 100 pulses
            Pulseout 10,320         'a pulse of 1.6ms in width on Pin 10
            Pulseout 11,320         'a pulse of 1.6ms in width on Pin 11
            Pause 17                'wait for 17 ms to make an approx. 20ms period
        Next

        'Turn left
        For X=0 To 99               'sending 100 pulses
            Pulseout 10,280         'a pulse of 1.4ms in width on Pin 10
            Pulseout 11,280         'a pulse of 1.4ms in width on Pin 11
            Pause 17                'wait for 17 ms to make an approx. 20ms period
        Next

        'Stop moving
        For X=0 To 99               'sending 100 pulses
            Pulseout 10,300         'a pulse of 1.5ms in width on Pin 10
            Pulseout 11,300         'a pulse of 1.5ms in width on Pin 11
            Pause 17                'wait for 17 ms to make an approx. 20ms period
        Next
    Loop
End Sub
```

Ex. 4 --- Input Music Notes and Play Back

Music Notes Basics

A buzzer is a kind of transducer that converts electrical signals into sound, and is commonly used as sound making component in electronic products. Buzzers can be categorized into piezo and magnetic type according to the operation principle. The piezoelectric buzzer (piezo buzzer) generates sound waves by driving a metal diaphragm based on the piezo-electric effect of the piezoelectric ceramics; in a magnetic buzzer, the metal diaphragm will be pulled down while being energized and rebound back while not being energized based on the electro-magnetic effect. In this experiment, the used buzzer has an operating of 3V~7.5V.

The method for calculating the frequency of each note in the music scale: First, remember that the note “low A” has the frequency of 440Hz. The frequency of every adjacent semitone is 1.059 times of the frequency of the previous note. For the convenience of the user to memorize, the following figure shows the relation between the piano keyboard and the corresponding notes:

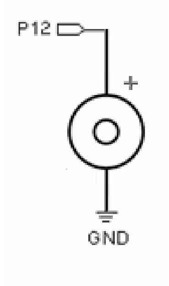


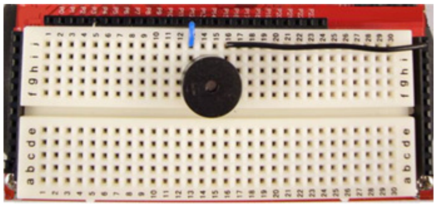
The following is a Frequency Look-up Table:

Note	Frequency (Hz)	Period (ms)
Do	523	1.91
Do#	554	1.8
Re	587	1.7
Re#	622	1.6
Mi	659	1.52
Fa	698	1.43
Fa#	740	1.35
Sol	784	1.27
Sol#	831	1.2
La	880	1.14
La#	932	1.07
Ti	988	1.0
Do	1047	0.96

Circuit Diagram

The “+” symbol marked on the top of the buzzer indicates the positive electrode, which should be connected to Pin 12. If the used buzzer has no indication mark for the positive electrode, connect the longer lead as the positive electrode.





Program Code

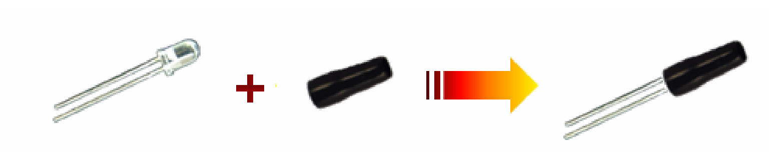
The objective of this exercise is to use the DEBUG, DEBUGIN and SOUND commands to compose simple melody and play it back. The SOUND command generates a square wave at specified I/O pin with the duration and frequency parameters.

After the program is executed, if the input value is 10, you can start to compose the music notes. If the input value is 0, it ends the recording processing and play the music notes entered so far.

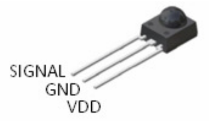
```
Sub Main()
Dim X As Byte           'declare as composition array index
Dim Y As Byte           'declare as playback array index
Dim CODEIN As Byte
Dim CODE(99) As Byte    'an array of 100 elements to store music notes.
    Do
        Debug CR,"KEY=10 TO COMPOSE; KEY=0 TO PLAY",CR
        Debugin CODEIN                  'get data and store in CODEIN.
        If CODEIN = 10 Then
            X = 0
            Debug CLS,"Do-Ti = 1-7 High Do=8 Rest=9",CR
            Do
                Debugin CODEIN
                If 0<CODEIN And CODEIN<10 Then
                    Debug CODEIN
                    X+=1
                    CODE(X)=CODEIN              'store notes into array
                End If
            Loop Until CODEIN = 0               'end of composition
        End If
        Debug CR
        For Y=1 To X Step 1
            Debug CODE(Y)
            Select CODE(Y)               'look up and play notes
            Case 1
            SOUND 12, 250, 523           'Do    1   C
            Case 2
            SOUND 12, 250, 587           'Re    2   D
            Case 3
            SOUND 12, 250, 659           'Mi    3   E
            Case 4
            SOUND 12, 250, 698           'Fa    4   F
            Case 5
            SOUND 12, 250, 784           'Sol   5   G
            Case 6
            SOUND 12, 250, 880           'La    6   A
            Case 7
            SOUND 12, 250, 988           'Ti    7   B
            Case 8
            SOUND 12, 250, 1047          'Do   High C
            Case 9                        'Rest Note
            Pause 250                    '0.25ms silence interval between notes
            End Select
        Next
    Loop
End Sub
```

Ex. 5 --- IR Transmitting and Receiving

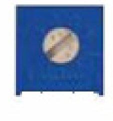
The Infrared LED is the same as color LED. It has two leads and the long lead is the positive electrode. An insulation sleeve is used to limit the emission range so as to facilitate the experiment. While attaching the insulation sleeve, slip the sleeve into the IR transmitter from the larger opening end as shown in the following figure.



The Infrared Receiver has three leads, they are denoted as shown in the figure below. The operating voltage range of the Infrared Receiver is 2.7V ~6.0V, and its high efficient receiving frequency is around 37.9KHz.

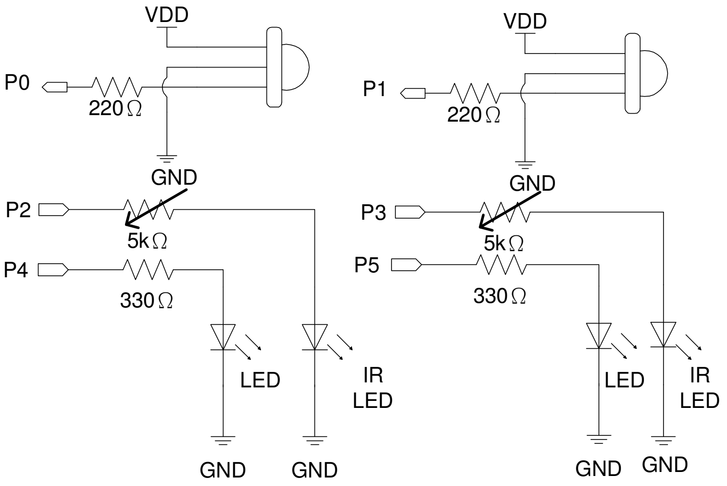


A 5KΩ variable resistor is used which has three leads. Choose either left or right lead and the middle lead and the center lead as the two leads of the resistor. Rotate the screw on the top gently with a screw driver to adjust the resistance.



Circuit Diagram

Place the IR transmitter and receiver facing the same direction toward the target.



Program Code

This exercise is to show how to turn on/off LEDs to indicate current detection status by adjusting the Variable Resistor (VR) to control the infrared light intensity so the receivers can detect the reflection intensity.

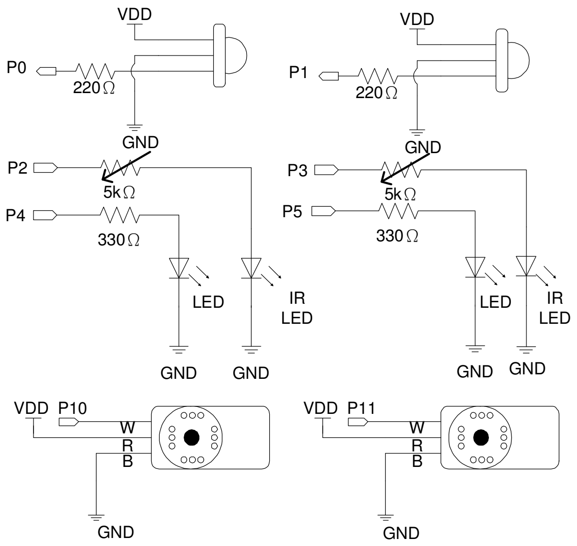
```
Sub Main( )
Dim L_IR As Byte
Do
    SOUND 2,5,38500          'generate 38.5kHz square wave for 5ms at pin 2
    R_IR = In(0)             'input pin 0 status and store in R_IR
    If R_IR = 0 Then
```

```
          High 4                ‘reflection detected, turn on LED
     Else
          Low 4                 ‘reflection not detected, turn off LED
     End If

     SOUND 3,5,38500            'generate 38.5kHz square wave for 5ms at pin 3
     L_IR = In(1)               ''input pin 1 status and store in L_IR.
     If L_IR = 0 Then
          High 5                ‘reflection detected, turn on LED
     Else
          Low 5                 ‘reflection not detected, turn off LED
     End If
     Pause 16
Loop
End Sub
```

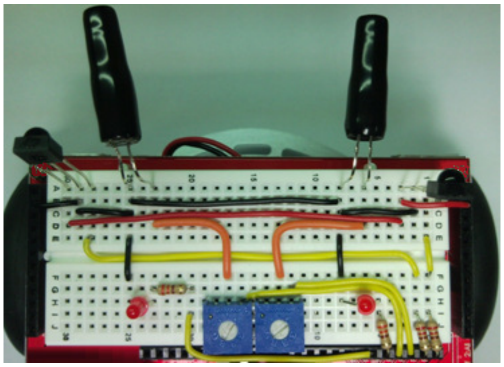
Ex. 6 --- Making a Line Following Bot

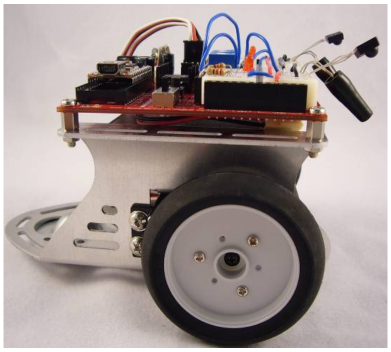
To make a line following bot, the required sensitivity of IR sensors to detect the black tape and control two continuous rotation servo wheels accordingly. Note that the IR receiver may be influenced by other IR emission sources. Keep away from other IR sources such as a computer monitor.

Circuit Diagram



Place the IR transmitter and receiver toward the ground to detect the black line.





Program Code

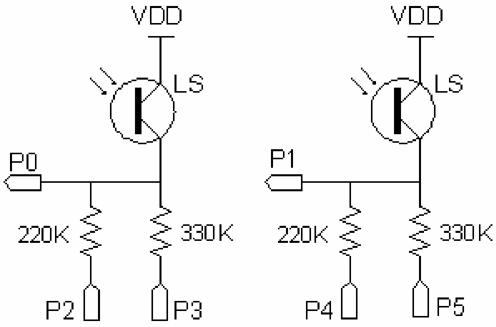
```
Sub Main()
Dim R_IR As Byte                'right IR sensor
Dim L_IR As Byte                'left IR sensor
Low 10
Low 11
Do
    SOUND 2,5,38500
    R_IR = In(0)
    If R_IR = 0 Then
        High 4
    Else
        Low 4
    End If

    SOUND 3,5,38500
    L_IR = In(1)
    If L_IR = 0 Then
        High 5
    Else
        Low 5
    End If

    If R_IR + L_IR = 2 Then
        Pulseout 10,350
        Pulseout 11,250
        Pause 17                'wait for 17 ms to make an approx. 20ms period
    Elseif R_IR + L_IR = 0 Then
        Pulseout 10,350
        Pulseout 11,250
        Pause 17                'wait for 17 ms to make an approx. 20ms period
    Elseif R_IR = 0 Then
        Pulseout 10,250
        Pulseout 11,250
        Pause 17                'wait for 17 ms to make an approx. 20ms period
    Elseif L_IR = 0 Then
        Pulseout 10,350
        Pulseout 11,350
        Pause 17                'wait for 17 ms to make an approx. 20ms period
    End If
    Loop
End Sub
```

Ex. 7 --- Making a Light Following Bot

The objective of this exercise is to understand the control of phototransistors. Use the I/O pins and commands to switch the pull download resistors and distinguish the light intensity on both light sensors and move the bot accordingly. Now you have a torch light following bot. Once it is programmed, use a flashlight and point to the innobot to make it follow the light.

Circuit Diagram

The recommended resistance range for LS3200 light sensor is 10K~470K. If the resistance is higher, the sensitivity is higher.



Program Code

```
Sub Main()
Dim LS_R As Byte                'right light sensor
Dim LS_L As Byte                'left light sensor
Dim DATA_R As Byte              'right light intensity
Dim DATA_L As Byte              'left light intensity

Low 10
Low 11
Do
    Low 2                       'configure 220K as pull down resistor
    Input 3                     '330K resistor is open (not activated)
    LS_R = In(0)                'read pin 0 status and store in LS_R
    If LS_R = 0 Then
        Low 3                   'configure 330K as pull down resistor
        Input 2                 '220K resistor is open (not activated)
        LS_R = In(0)            'read pin 0 status and store in LS_R
        If LS_R = 0 Then
            DATA_R = 0          'least light intensity detected
        Else
            DATA_R = 1          'higher light intensity detected

        End If
    Else
    DATA_R = 2                  'highest light intensity detected
    End If

    Low 4                       'configure 220K as pull down resistor
    Input 5                     '330K resistor is open (not activated)
    LS_L = In(1)                'read pin 0 status and store in LS_R
    If LS_L = 0 Then
        Low 5                   'configure 330K as pull down resistor
        Input 4                 '220K resistor is open (not activated)
        LS_L = In(1)            'read pin 0 status and store in LS_R
        If LS_L = 0 Then
            DATA_L = 0          'least light intensity detected
        Else
```

```
            DATA_L = 1     'higher light intensity detected

        End If
    Else
        DATA_L = 2         'highest light intensity detected
    End If

    If DATA_R=2 And DATA_L= 2 Then       'same high light intensity, move forward
            Pulseout 10,350
            Pulseout 11,250
            Pause 16

    Elseif DATA_L < DATA_R Then           'right side is higher, turn right
            Pulseout 10,350
            Pulseout 11,300
            Pause 16

    Elseif DATA_L > DATA_R Then           'left side is higher, turn left
            Pulseout 10,300
            Pulseout 11,250
            Pause 16

    Else                                  'no apparent light, stop
            Pulseout 10,300
            Pulseout 11,300
            Pause 16
    End If
    Loop
End Sub
```