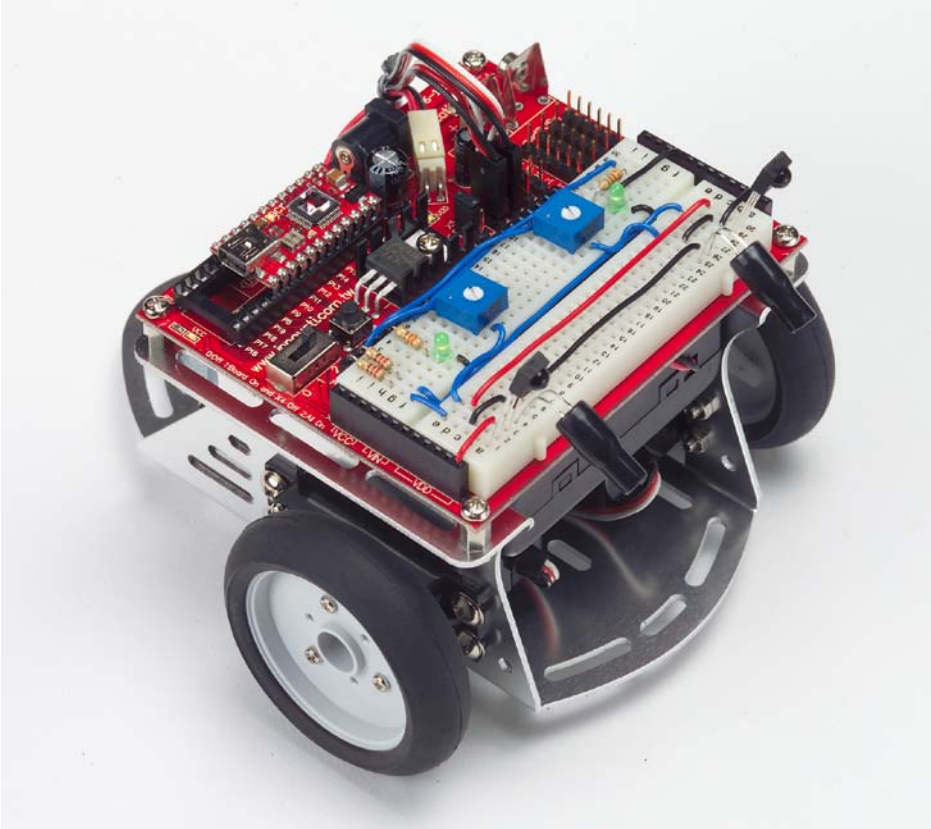
Innobot
Instruction Manual

Version 1.03



Trademark

Innovati®,  , and BASIC Commander®, are registered trademarks of Innovati Inc. InnoBASIC™ and cmdBUS™ are trademarks of Innovati Inc.

In view of unceasing improvement of products, this document and the product mentioned in this document are subject to be changed by the company without notice. It is forbidden to reproduce and distribute any part or all of the contents of the product without the written approval or authorization by the company.

Disclaimer

The user shall undertake all the risks in the applications where this product is used. The company shall not be liable for any direct, indirect or consequential damages due to the use of the product including but not limited to the loss of equipment, the loss of human safety and health and the loss of profit and reputation. The product of the company shall not be used in life saving or any related instrument and equipment. Children under 14 shall not use this product for any related experiment without being accompanied by adults.

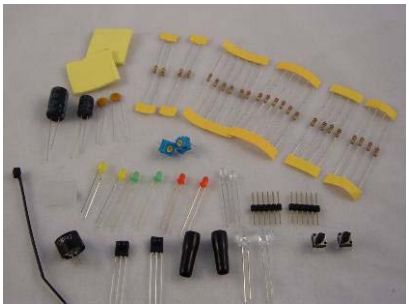
Errata

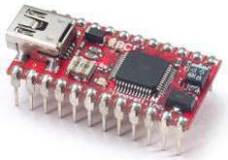
We hope the users may regard this document as a lively and practical instruction manual. We have put tremendous efforts in making this instruction manual complete and correct; however, there may be unavoidable missing parts or errors. With a view to providing the user updated and complete information in the instruction manual, we keep improving and supplement the contents of this instruction manual. If you find any error in this manual, please contact us via the e-mail service@innovati.com.tw. Any related update information will be disclosed on our website. Please visit our website http://www.innovati.com.tw for more updated information.

✧ **Assembly:**

1. Part List:

Item	Illustration	Qty.	Specifications and instructions
Assembly Kit Parts			
Innobot Aluminum Chassis		1	Thus serves as the platform for installing the auxiliary wheels, servos, and Education Board.
360° Servo		2	The servo provides the motions of 360° clockwise or counterclockwise rotation at different speeds. It is operated after being connected with the signal line, power line and ground line. Please pay attention to the polarity of the wires. Dimensions (L x W x H): 42x20.5x39.5 (mm) Weight: 44 g Speed: 60.0(R/M) at 4.8V 70.0(R/M) at 6.0V Torque: 3.3 (kg/cm) at 4.8V 4.8 (kg/cm) at 6.0V
Accessory Kit for Servos		2	Allows the user to install the robot with different swing arms.
Dual Wheel Set		1	Complies with the servo.

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 Box		1	Four AA size alkaline batteries recommended
Jumper Wires		1	Wires for the connections of experimental circuits
Material Kit		1	Includes: 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

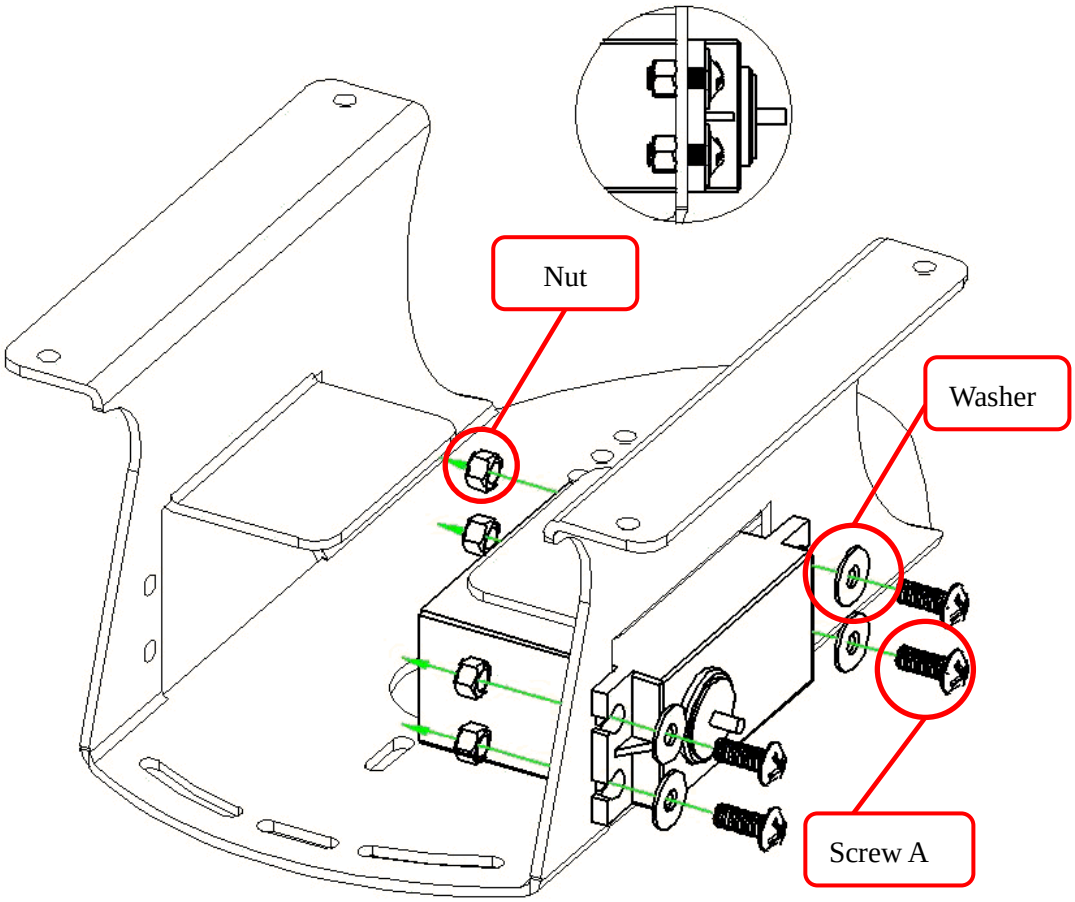
			variable resistor*2, phototransistor*2, IR receiver*2, IR transmitter*2, insulation plastic sleeve*2, cable strap, foam*2, piezoelectric buzzer, small switch*2, pin header*6*2, etc.
Electrical Tape (black)		1	Used for IR sensing applications.
Module Kits			
BC1		1	Innovati BASIC Commander, capable of storing programs and controlling operations of modules.
Education Board		1	Used for installing BC1. It has reserved cmdBUS to allow the user to connect directly to various Innovati modules. The auxiliary Breadboard allows the user to install other expansion modules and components.
USB Cable		1	Links BC1 with PC, allowing downloading of PC program to BC1, or performing communication in Debug Mode.

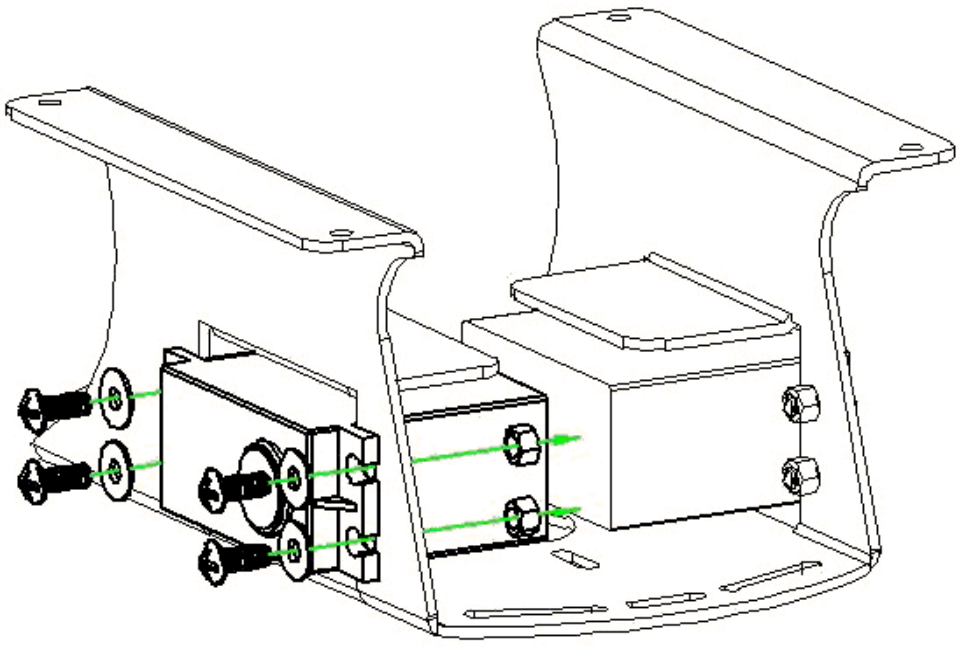
2. Tools:

- Phillips Screwdriver
- Long Nose Pliers

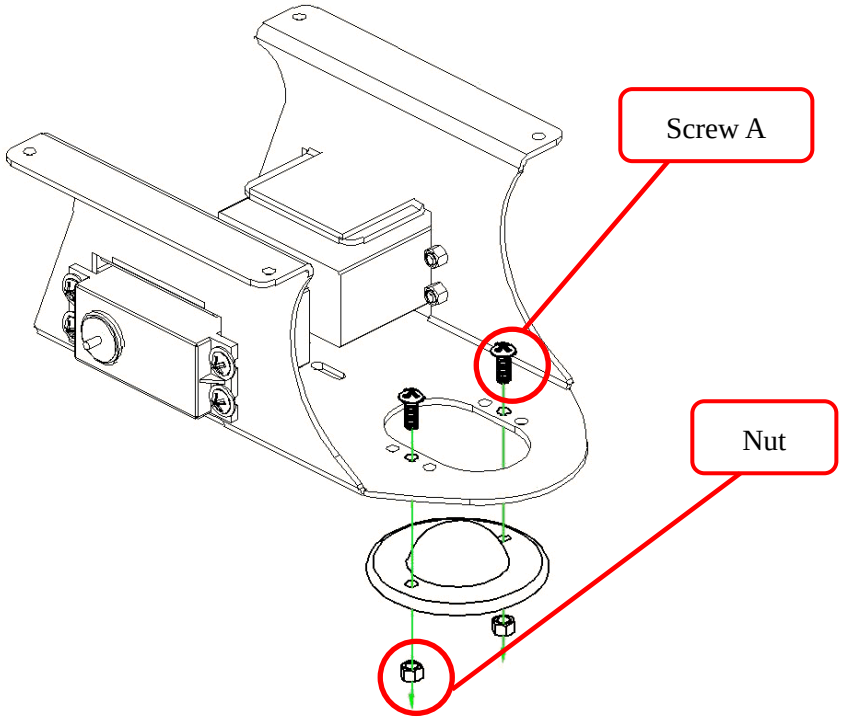
3. Assembly Procedure:

Step 1. Installing Servos

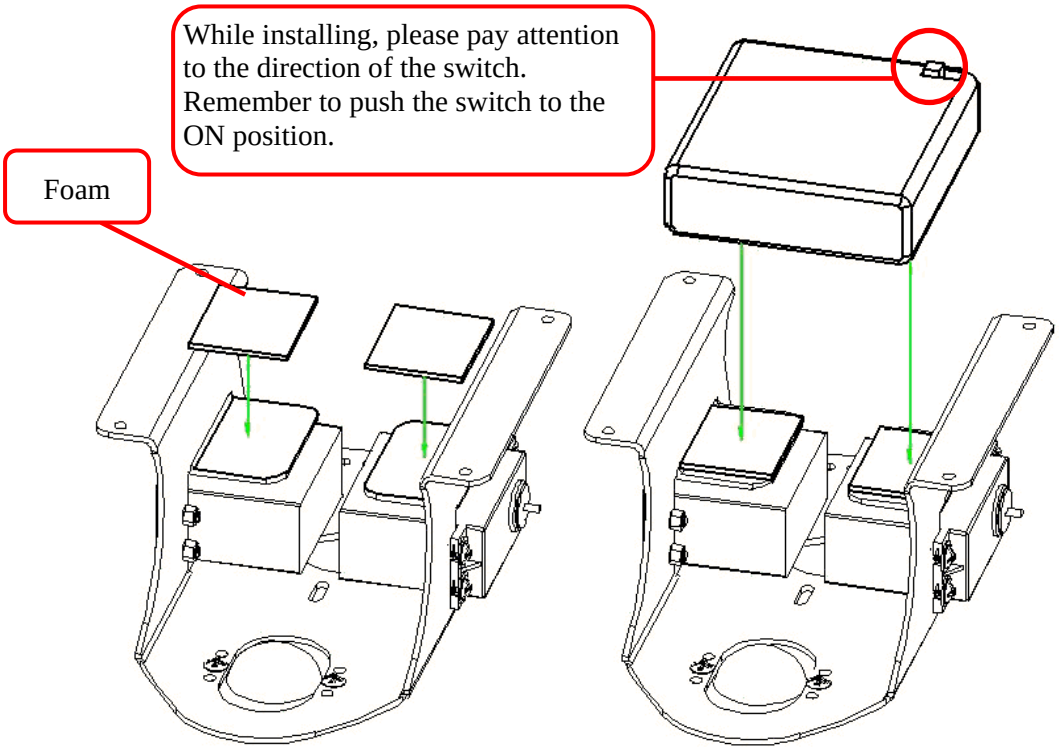




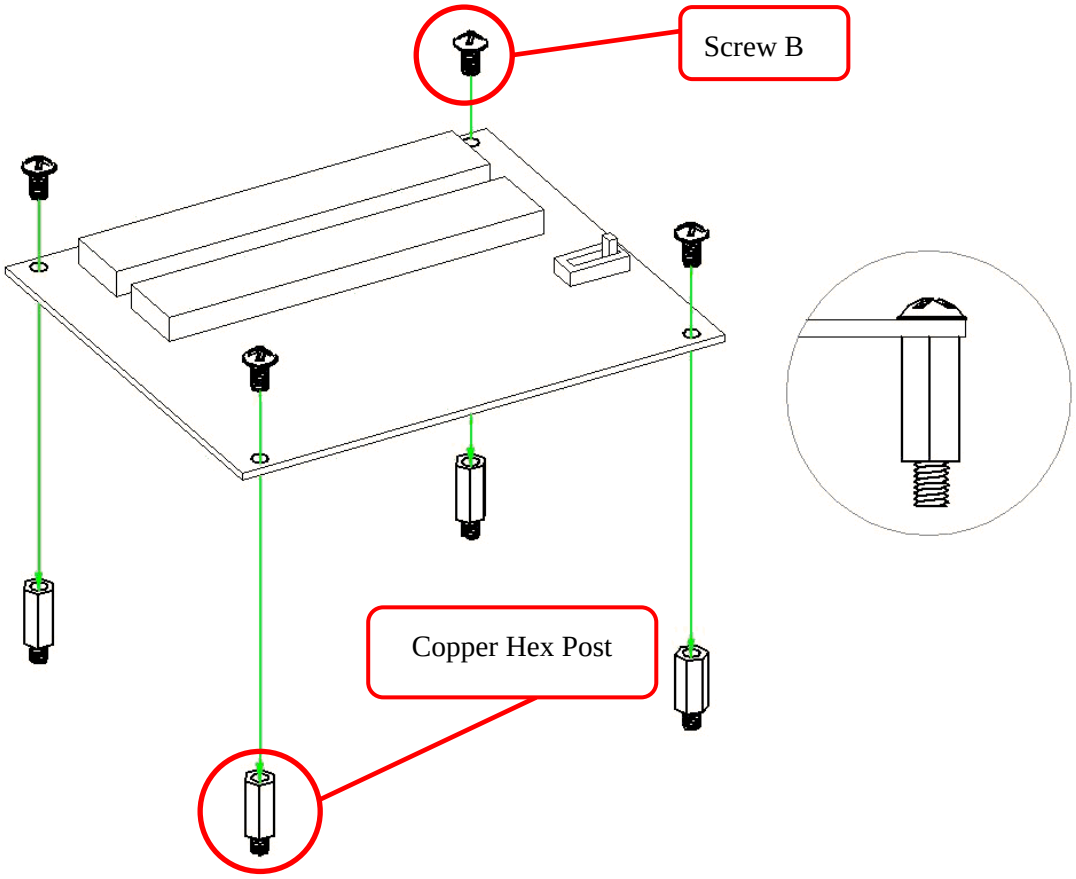
Step 2: Installing Ball Transfer Unit

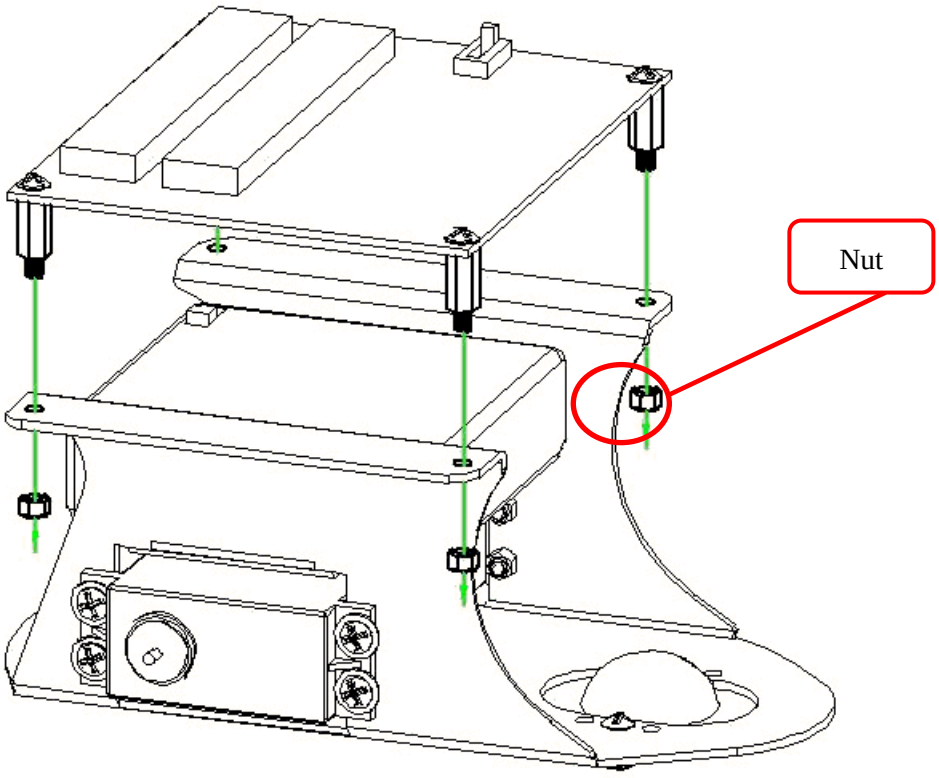


Step 3: Installing the Battery Box

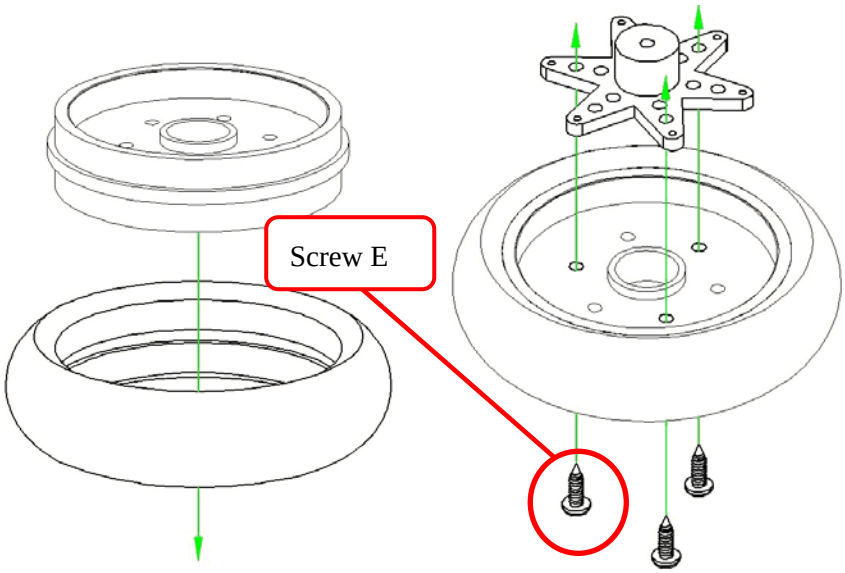


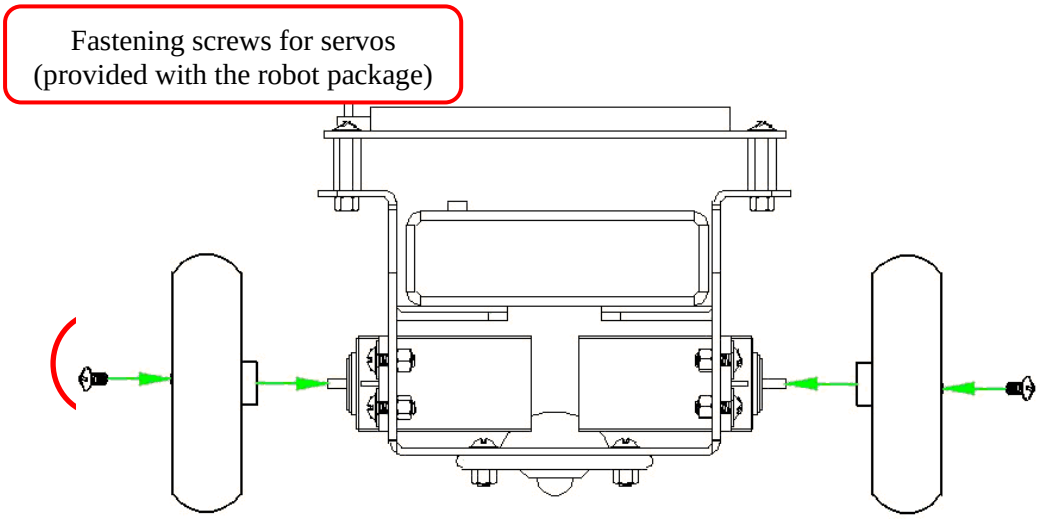
Step 4: Installing the Education Board





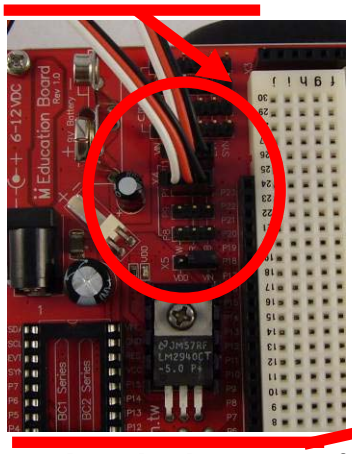
Step 5: Installing the dual wheels and connecting the signal lines of the servos.

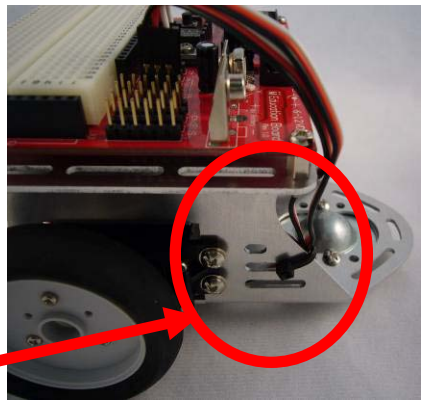




Connect the servos of the Innobot's wheels to the PORT reserved for the servos on the Education Board.

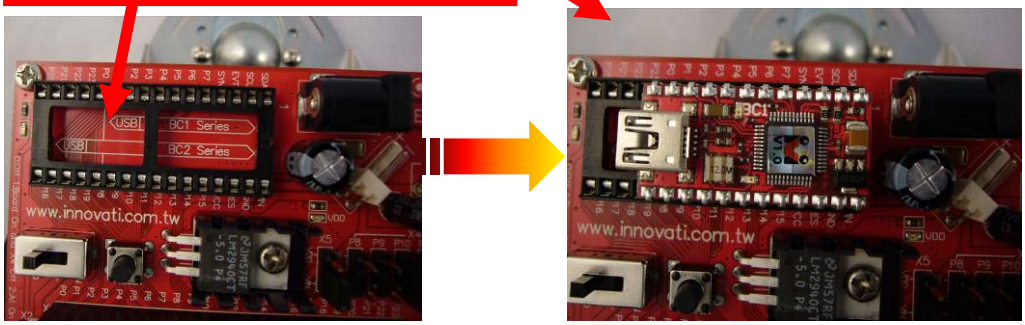
Program default as an example: Left wheel is connected to P10, and the right wheel is connected to P11. The user can modify the settings according to the demand.

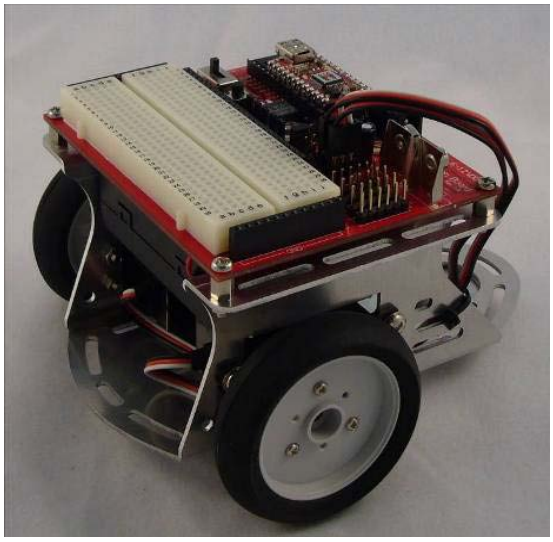




Fix the redundant wires of the 360° servo on the Innobot aluminum frame with a cable strap.

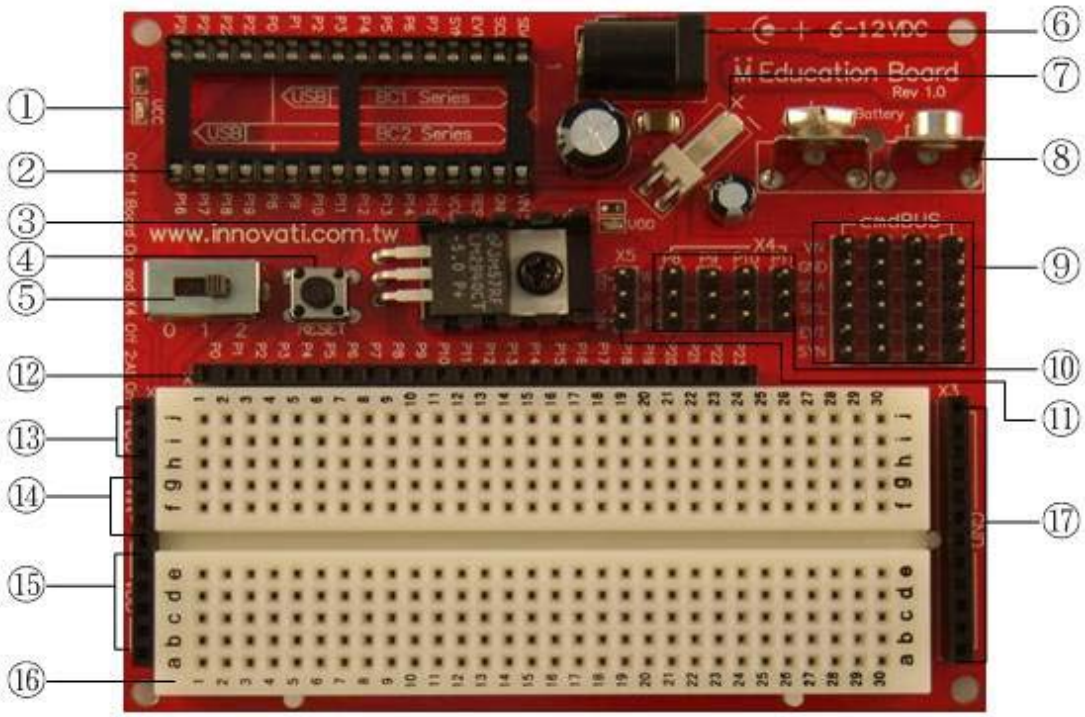
Plug the BC1 onto the Education Board and the basic installation is completed. (PS: While plugging, please pay attention to the orientation shown in the above figure. The USB port is placed outward.)





Picture of Innobot after the installation is complete

✧ Introduction to the Education Board



①-Power indicator

②-32-Pin Socket for 24/32-pin BASIC Commander:

Either the 24-pin or the 32-pin BASIC Commander® can be directly plugged into the socket. Special attention should be paid that the BASIC Commander® should be plugged in carefully according to the orientation indication printed on the board to avoid damage to the BASIC Commander® due to incorrect orientation. It is recommended to turn off the power before plugging the BASIC Commander®. Before turning on the power, please confirm if the orientation is correct.

③-VDD Power Indicator

④-Reset button: Used for resetting the BASIC Commander® to re-run the program.

⑤-Power selection: The sliding switch is used for switching the power supply. If it is switched to the “0” position, VDD is turned off; if it is switched to the “1” position, VDD is turned on but the VIN on X4 is turned off; if it is switched to the “2” position, the VINs on both VDD and X4 are turned on. As the VDD is turned on, the VDD indicator will be on. Please refer to the following table:

Switch Position	VDD/VIN/VCC (Female Header)	VIN or VDD (X4 header)
0	OFF	OFF
1	ON	OFF
2	ON	ON

⑥-Power adapter input: The external power supply VIN can be connected with a 6V-12V DC power supply.

⑦-DC power input: If an external power supply VIN is used, the user can select to connect a 6-12V DC power supply directly to the pin header for power supply. EX: The power line of the battery box for the Innobot is connected to this position.

⑧-Battery input: The user can connect the external power supply VIN with a 9-V button battery.

Note: The user only needs to select one port from ⑥, ⑦, and ⑧. The voltage from the power supply will be reduced and regulated to the 5-V VDD by the Education Board with a maximum allowed current of 1 A. When the VIN is used as the power supply, the LED at the center of the Education Board will be on.

⑨-4 sets of cmdBUS pin headers: They can be connected to four 6-wire cmdBUS™ ribbon cables for connecting with Innovati's peripheral modules.

The VIN provides an un-regulated external power supply with a voltage ranging from 6 to 12 volts DC. This power supply can be further regulated to be 5 volts for supplying the peripheral modules. Special attention should be paid to the orientation of the cmdBUS™ while plugging in. Reversed plugging orientation may lead to damage of the components. If the ID number of the module declared in the program is the same as the ID number set by the hardware DIP switch, the corresponding module may be used immediately. Of course, each module has its own commands and functions. Please refer to the operation manual of each module.

⑩-4 servo pin headers: There are 4 servo pin headers at the position marked with

X4 which are marked as P8, P9, P10 and P11 for the user to connect 4 servos. (W: Signal, R: Power, B: GND)

⑪-Servo power selection: It is located at the X5 position for selecting the VDD or VIN by the jumper cap. While using the servo, in order to reduce power consumption of the 9-V battery, the user can push the sliding switch from “2” to “1” to turn off the power supply to the servos.

⑫-I/O pins

⑬-VIN Pins (Female Header)

VIN is an unregulated external power supply with a voltage ranging from 6 to 12 volts DC.

⑭-VDD Pins (Female Header)

Its voltage is regulated to 5 volts by the Education Board to provide a maximum current of 1 A.

⑮-VCC Pins (Female Header)

Its voltage is regulated to 5 volts by the Basic Command 1 to provide a maximum current of 250mA.

⑯-Bread-Board

⑰-GND

Common ground.

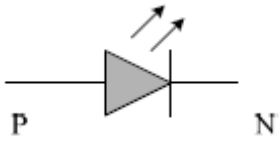
✧ Quick Start

I. Control the LED in different ways

An LED (Light Emitting Diode) is a device which converts electrical energy into light emission by means of the recombination of the electrons in the p-type semiconductor and the holes in the n-type semiconductor within the active layer; different compound materials of the active layer, such as InGaN, AlInGaP, GaAs, etc., allow the LED to emit the electromagnetic wave at different wavelengths. LED has the advantage of emitting of light of various colors, is power-saving, compact, long lifespan, fast response, low polluting, high reliability, high module flexibility, etc.

In fact, in order to allow the user to distinguish the P and N electrodes, the leads are often provided at different lengths; 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.

Physical picture



Circuit symbol

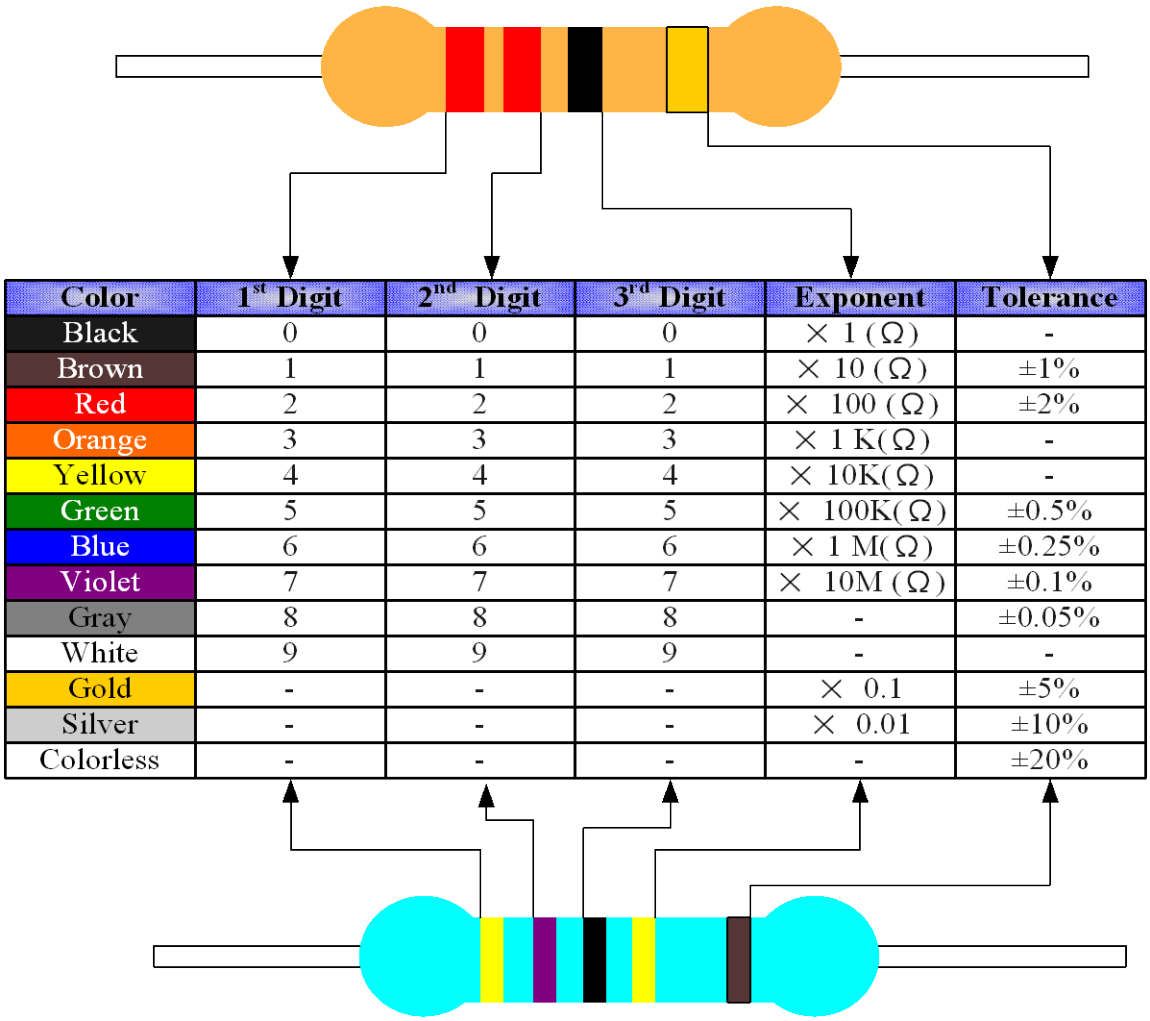
The driving voltage of the LED is typically within the range of 2.2-3.4V. However, the output voltage of the I/O pins on the BASIC Commander is about 5V with a supplied current of approx. 10mA. If such a voltage is directly applied on the LED, it may exceed the rated voltage of the LED; in addition, in order to protect the Pin4 on the Basic-Commander from being damaged due to over-voltage. Therefore, it should be connected with the resistor R1 to limit the current.

A resistor is a common component in electronic circuits. The word “resistor” implies that it can provide a resistant force to 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 with the four color rings. For example, find the color ring representing the precision tolerance from the bottom of the resistor; gold represents 5% and silver represents 10%. From the other end of the resistor, find the first and second color rings and read their corresponding numeric

values. For example, the first two color rings of the resistor shown on the top of the following figure “Resistor Color Code Table” are red, so its corresponding numeric value is 2,2 (Red, Red). Now read the third color ring representing the exponent, which is Black (1) in our example. Thus, the obtained resistance value is 22Ω. If the third ring representing the exponent is gold, move the decimal point one digit to the left. If the third ring representing the exponent is silver, move the decimal point two digits to the left.

According to Ohm's law, if 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 (10mA), the resistance of R1 should satisfy the relation $\frac{3-2.2}{R1} \leq 10mA$; of course, if the resistance of R1 is too large, the current passing through the LED is too small so that the LED will be dim or even not lit. Therefore, it is suitable to choose a resistance of 330Ω to maintain the current at approx. 8.4 mA.



Color	1st Digit	2nd Digit	3rd Digit	Exponent	Tolerance
Black	0	0	0	× 1 (Ω)	-
Brown	1	1	1	× 10 (Ω)	±1%
Red	2	2	2	× 100 (Ω)	±2%
Orange	3	3	3	× 1 K(Ω)	-
Yellow	4	4	4	× 10K(Ω)	-
Green	5	5	5	× 100K(Ω)	±0.5%
Blue	6	6	6	× 1 M(Ω)	±0.25%
Violet	7	7	7	× 10M (Ω)	±0.1%
Gray	8	8	8	-	±0.05%
White	9	9	9	-	-
Gold	-	-	-	× 0.1	±5%
Silver	-	-	-	× 0.01	±10%
Colorless	-	-	-	-	±20%

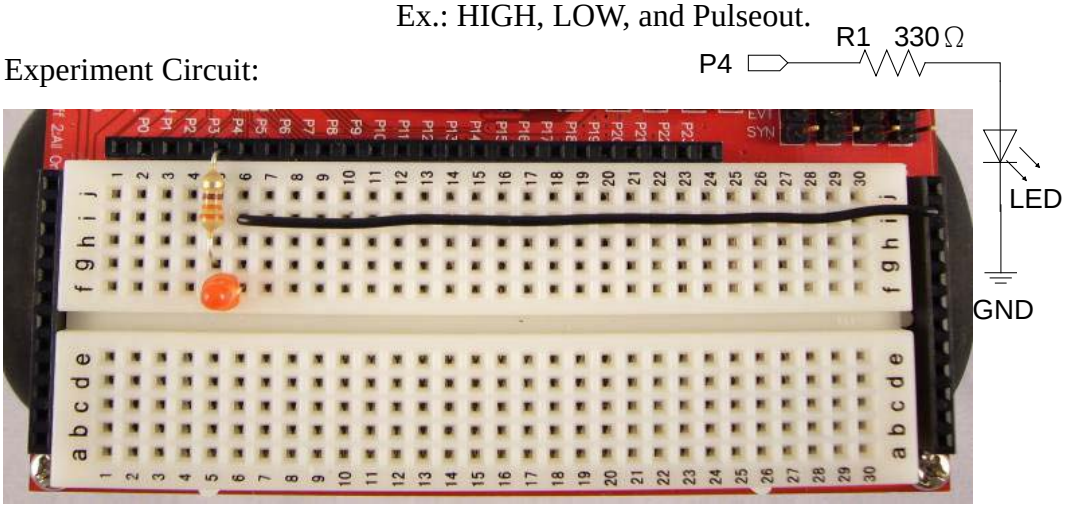
Resistor Color Code Table

According to the above explanation, the resistance of R1 is 330Ω means the

required resistor has the color code of “Orange Orange Brown Gold” according to the resistor color code table.

Objective of the Experiment: Understand the basic operation of the Education Board for downloading a program to BC1.
Ex.: HIGH, LOW, and Pulseout.

Experiment Circuit:



Program Code:

```
0   Dim X As Word          'Generate a 2-byte value of 0-65535 in the variable X
1   Sub MAIN()
2   'Control by HIGH, LOW
3   High 4                 'Set the Pin 4 to be the high level of about 5 volts
4   Pause 3000             'Force the program to wait 3000 ms (3 sec)
5   Low  4                 'Set the Pin 4 to be the low level of about 0 volt
6   Pause 1000             'Force the program to wait 1000 ms (1 sec)
7   'Control by Pulseout
8   For X=0 To 2000        ' Declare a loop of 2001 times
9   Pulseout 4,X           'Generate a pulse with a width of X on the Pin 4
10  Pause 10               'Force the program to wait 10 ms (0.01 sec)
11  Next
12  End Sub
```

Note to the program: Please refer to the operation manuals of BASIC Commander and innoBASIC Workshop

- For the declaration types, please refer to page 34.
- For the details about the FOR…NEXT command, please refer to page 113.
- For the details about the Pulseout command, please refer to page 156.

Experiment result: This program will send a signal thorough Pin 4 as the output pin. The result shows that the LED is on by the HIGH level signal for 3 seconds and then is off by the LOW level for 1 second.

Then, the FOR loop and Pulseout command are used to increase the light intensity of the LED gradually from the off state and finally the LED is off.

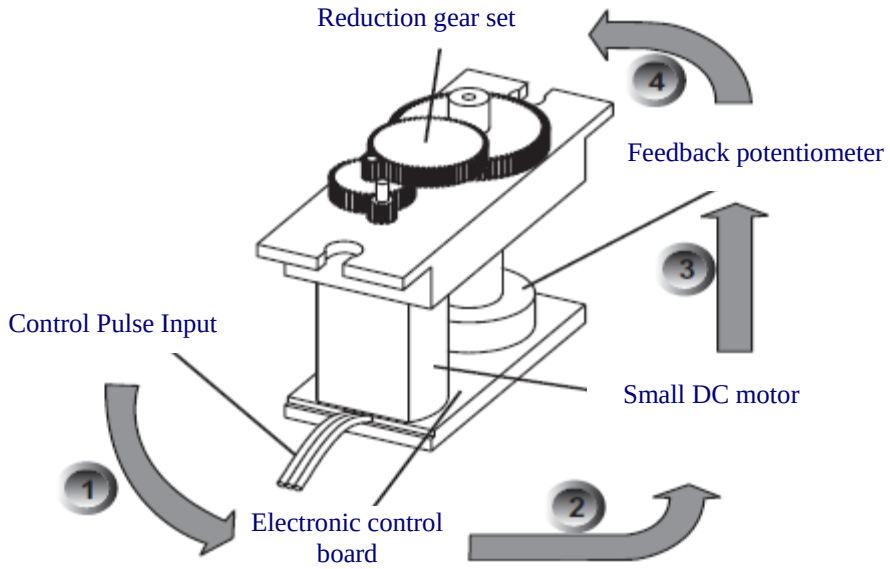
II. Control of the 360° Servo

(1) Internal Construction of the Servo

A servo 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.

(2) Operation Principle of the Servo

The reduction gear set is driven by the motor. Its output end drives a linearly proportional potentiometer for the position detection. The potentiometer will convert the rotational coordinate into a proportional voltage feedback to the control circuit board. The control circuit board will compare it with the input control pulse signal. Then it generates a correction pulse to drive the motor to rotate clockwise or counterclockwise so as to make the output position of the gear set to be the same as the expected value. As a result, the correction pulse will gradually tend into zero so as to achieve the precise positioning of the servo.



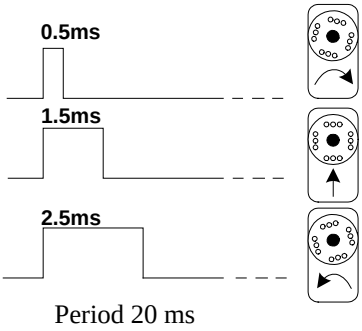
Schematic diagram of the servo structure and operation principle

(3) How to Control the Servo

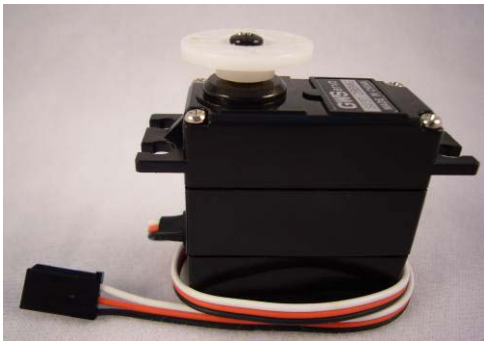
The 360° servo is controlled by the pulse signal of 0.5-2.5ms 50Hz HIGH PULSE; with the 1.5ms HIGH PULSE, it is in the stationary state; with the pulses less than 1.5ms, it rotates clockwise; the smaller pulse width, the faster it rotates; with the pulses larger

than 1.5ms, it rotates counterclockwise.

PS. Each servo may have some difference in the control pulse width, the range 0.5-2.5ms is a reference value. It actually has some difference.



Period 20 ms



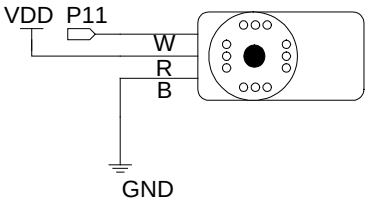
The servo has three power wires as shown in the left figure. Among the three wires of the servo, the white one is the control wire that is connected to the control chip. The middle red wire is the power line for the operation of the servo, which is typically 5 volts. The third wire is the ground wire.

Note: In the circuit diagram, these wires are referred to as W (white), R (red), and B (black).

1. Control the 360° servo for clockwise rotation, counterclockwise rotation, and stopping.

Objective of the Experiment: Strengthen the practice of the Pulseout command and understand the control method of the 360° servo.

Experiment circuit:



While connecting the circuit, use the servo pin header at the X4 position on the Education Board. Pay attention to the color of the wires during the connection, W(white), R(red), and B(black).

Program Code:

```
0   Dim X As Word
1   Sub MAIN( )
2   Low 11
3   Do                          'Generate an infinite loop
4   'Fast clockwise rotation
5   For X=0 To 100              'Generate a loop of 101 times
6       Pulseout 11,100         'Generate a pulse of 0.5ms in width on Pin 11
7       Pause 16                'Force the program to wait 16 ms
8   Next
```

```
'Slow clockwise rotation
For X=0 To 100
    Pulseout 11,280     'Generate a pulse of 1.4ms in width on Pin 11
    Pause 16
Next
'Stop
For X=0 To 100
    Pulseout 11,300     'Generate a pulse of 1.5ms in width on Pin 11
    Pause 16
Next
'Slow counterclockwise rotation
For X=0 To 100
    Pulseout 11,320     'Generate a pulse of 1.6ms in width on Pin 11
    Pause 16
Next
'Fast counterclockwise rotation
For X=0 To 100
    Pulseout 11,500     'Generate a pulse of 2.5ms in width on Pin 11
    Pause 16
Next
Loop
End Sub
```

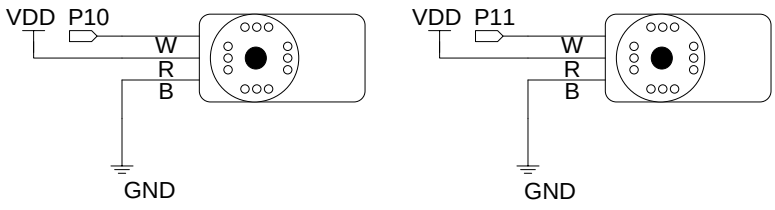
Experiment result: The servo will make a fast clockwise rotation, a slow clockwise rotation, stop, a slow counterclockwise a fast counterclockwise rotation, rotation in sequence.

PS: Each servo has a slightly different pulse width threshold. It should be adjusted according to actual conditions.

2. Control the Innobot for Simple Movement

Objective of the Experiment: Apply the Experiment 2 to the control of Innobot to understand the motion of Innobot.

Experiment circuit:



Program code:

```
0   Dim X As Word
1   Sub MAIN( )
2   Low  10
3   Low  11
4   'PIN 10 Left servo, PIN 11 Right servo
5   'Fast forward
6   For X=0 To 50
7       Pulseout 10,500
8       Pulseout 11,100
9       Pause 16
10  Next
11  'Fast backward
12  For X=0 To 50
13      Pulseout 10,100
14      Pulseout 11,500
15      Pause 16
16  Next
17  'Stop
18  For X=0 To 100
19      Pulseout 10,300
20      Pulseout 11,300
21      Pause 16
22  Next
23  'Turn right at the current position
24  For X=0 To 100
25      Pulseout 10,320
26      Pulseout 11,320
27      Pause 16
28  Next
29  'Turn left at the current position
30  For X=0 To 100
31      Pulseout 10,280
32      Pulseout 11,280
33      Pause 16
34  Next
35  Pulseout 10,300
```

```
36  Pulseout 11,300
37  Pause 16
38  End Sub
```

III. Simple Music Score Key-in Output Mode

A speaker (loudspeaker) or buzzer is a kind of transducer that converts electrical signals into sound, and is the commonly used as sound making component in electronic products. Although there are various kinds of speakers, the basic operation principle is approximately the same: when a current signal passes through a coil, the coil will generate a magnetic field repelling the magnetic field of the magnet so as to force the cone diaphragm to vibrate. The sound wave generated from such a vibration is transmitted through the air to the ear drum to become the sound being perceived; the magnitude of the current and the frequency of the vibration determine the loudness and the pitch of the sound respectively! In fact, the generation of sound waves of a speaker by pushing the air is quite a complicated process.

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:

For example:

B is a whole tone higher than A (two semitones equal a whole tone), so the frequency of the note B:

B ＝ 440Hz×1.059×1.059＝493.9Hz

C ＝ 440Hz×1.059×1.059×1.059＝493.9Hz×1.059＝523Hz

Note: C& is a semitone higher than B.

Therefore, according to the above principle, the frequency of each note in the scale between the “low C” (Do) and the “high C” (Do) can be calculated.

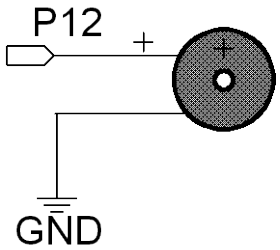
The following figure is a Frequency Look-up Table:

Note	Frequency (Hz)	Period (ms)	Half Period (ms)
Do	523	1.91	0.96
Do#	554	1.8	0.9
Re	587	1.7	0.85
Re#	622	1.6	0.8
Mi	659	1.52	0.76
Fa	698	1.43	0.72
Fa#	740	1.35	0.68
Sol	785	1.27	0.64
Sol#	831	1.2	0.6
La	880	1.14	0.57
La#	932	1.07	0.54
Ti	988	1.00	0.50
$\overline{Do}$	1047	0.96	0.48

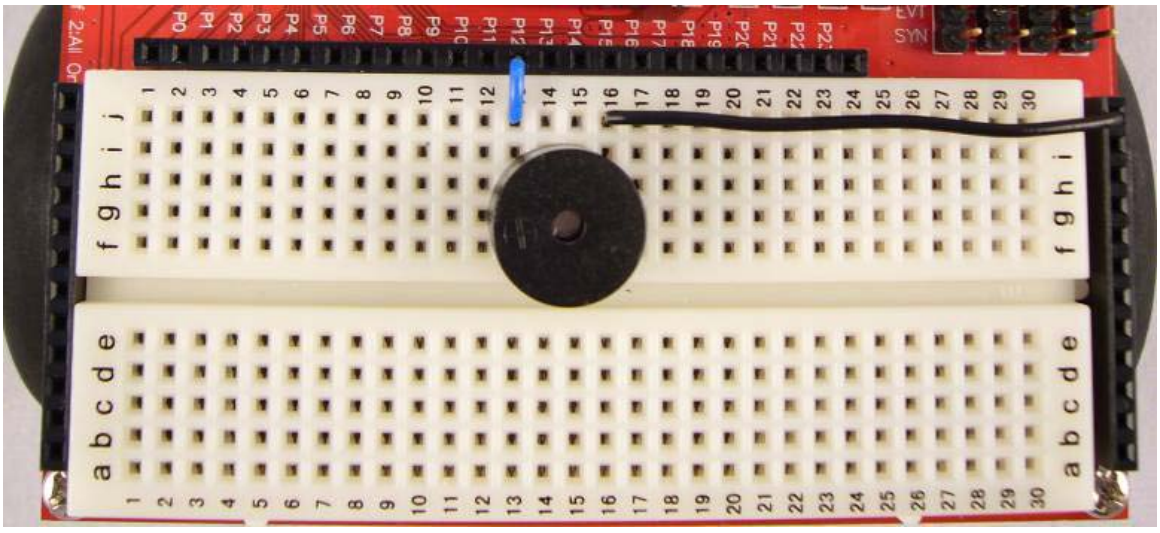
Objective of the Experiment: Understand the relation between the sound generated from the piezo buzzer and the frequency, and the use of the SOUND command.

Use the DEBUG, DEBUGIN and SOUND commands to complete the simple interactions.

Experiment Circuit: 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:

```
0   Dim X As Byte
1   Dim Y As Byte
2   Dim CODEIN As Byte
3   Dim CODE(99) As Byte 'Generate an array of 100 elements. Each element is
    one byte.
4   Sub MAIN( )
5   Do
6   Debug CR,"KEY=10 TO SET;KEY=!10 TO RUN",CR
7   Debugin CODEIN
8   If CODEIN = 10 Then
9       X = 0
10      Debug CLS,"Do-Si = 1-7 High Do=8 Rest=9",CR
11      Do
12          Debugin CODEIN
13          If 0<CODEIN And CODEIN<10 Then
14              Debug CODEIN
15              X+=1
16              CODE(X)=CODEIN
17          End If
18      Loop Until CODEIN = 0
19  End If
20  Debug CR
21  For Y=1 To X Step 1
22      Debug CODE(Y)
23      Select CODE(Y)
24      Case 1
25      SOUND 12,250,523        'Do      1   C
26      Case 2
27      SOUND 12,250,587        'Re      2   D
28      Case 3
29      SOUND 12,250,659        'Mi      3   E
30      Case 4
31      SOUND 12,250,698        'Fa      4   F
32      Case 5
33      SOUND 12,250,785        'Sol     5   G
34      Case 6
35      SOUND 12,250,880        'La      6   A
```

```
36          Case 7
37          SOUND 12,250,988        'Si       7    B
38          Case 8
39          SOUND 12,250,1047       'Do    High C
40          Case  9                 'Rest
41          Pause 250
42          End Select
43      Next
44      Loop
45      End Sub
```

Description of the program:

6 Use the DEBUG command to display KEY=10 TO SET;KEY=!10 TO RUN. CR means carriage return.

7 Use the DEBUGIN command to input the data from the terminal and store it into CODEIN.

8 Use the IF…ELSE command to determine if the input value is 10.

11-18 Use the DO…LOOP UNTIL command to control the loop. If the input value is 0, exit the loop. Otherwise, the loop continues.

21-43 Use the FOR…NEXT and SELECT…CASE commands, the number of loops is determined by X. Read the input value orderly, display it on the terminal and generate sound according to the contents in the CASE statement.
The SOUND command generates a square wave with the parameters as follows:
SOUND Pin#, duration (ms), width of the square wave (Hz)

Note to the program: Please refer to the operation manuals of BASIC Commander and innoBASIC Workshop.

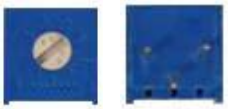
- For more details about the DEBUG command, please refer to page 80.
- For more details about the DEBUGIN command, please refer to page 84.
- For more details about the IF…ELSE command, please refer to page 80.
- For more details about the DO…LOOP UNTIL command, please refer to page 88.
- For more details about the SELECT…CASE command, please

refer to page 174.

Experiment result: After the user executes the program, if the input value is 10, the user can set the music score. During the setting of the music score, if the input value is 0, it will play the music score entered so far.

IV IR Receiving and Transmission Test

Variable resistor:



A 5KΩ variable resistor is used. The variable resistor has three leads. Choose any one of the left and right leads to work together with the middle lead. The adjustment of the resistance can be performed by rotating the screw the on the top.

IR Receiver:

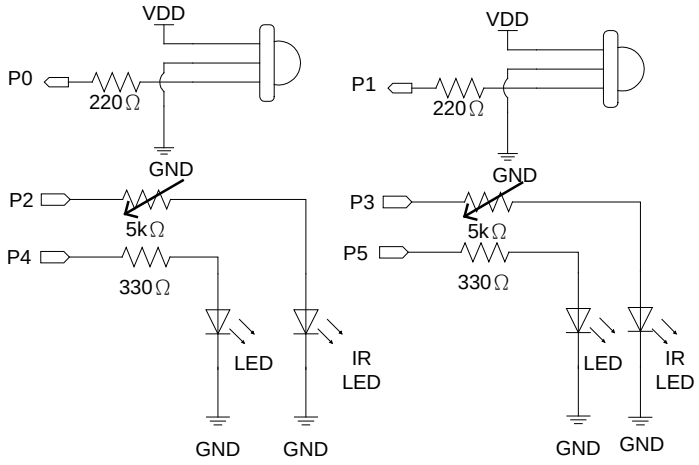
Voltage range of the power supply: 2.7V -6.0V

Receiving frequency: 37.9KHz

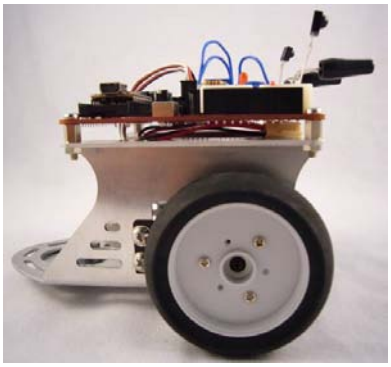
IR Transmitter: The long lead is the positive electrode. Because the IR transmitter has a wide emission angle, the insulation sleeve is used to limit the emission range so as to facilitate the experiment. While attaching the insulation sleeve, slip the sleeve into the IR transmitter from the larger opening end as shown in the following figure.

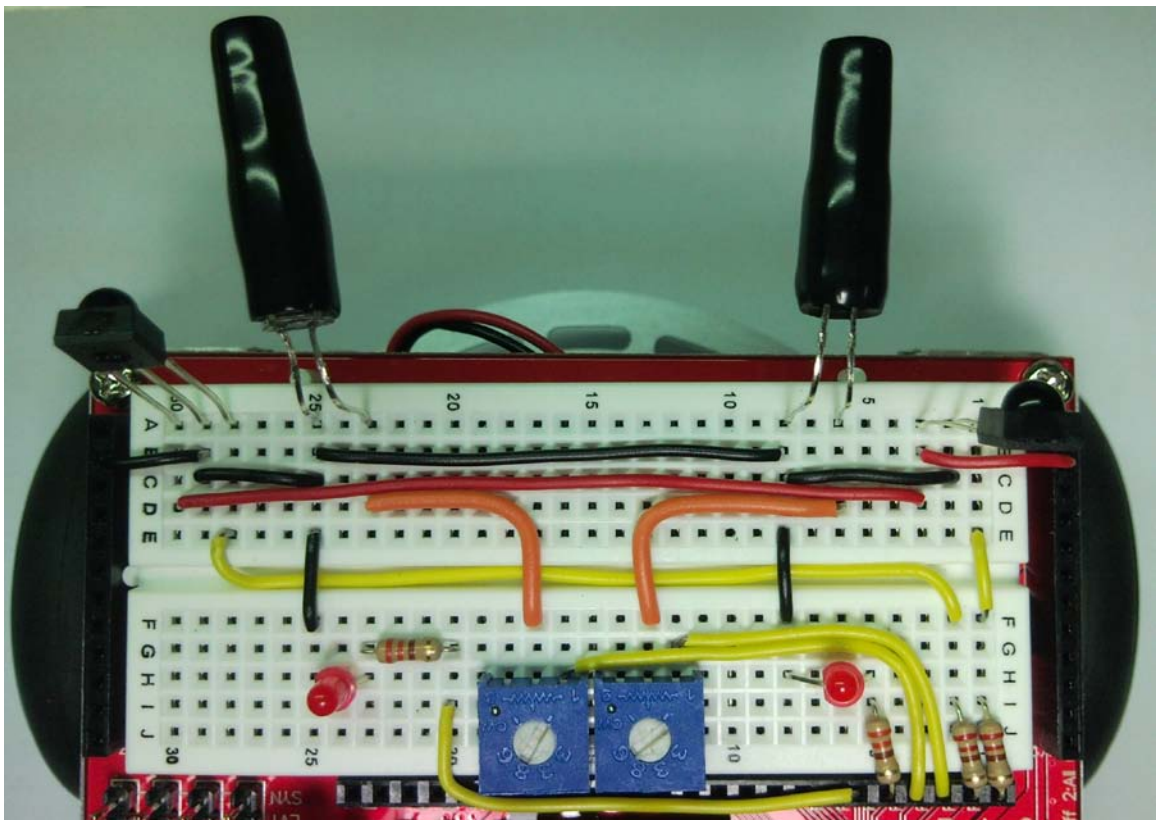
Objective of the Experiment: With the exercises in Experiments (1) and (3), the control of the light intensity of the LED and the control of the variable resistor can be used to adjust the sensitivity for the IR receiving and transmission.

Experiment circuit:



Place the IR transmitter and receiver facing the same direction.





Program code:

```
0   Dim R_IR As Byte
1   Dim L_IR As Byte
2   Sub MAIN( )
3   Do
4   SOUND 2,5,38500 'Generate a square wave with the duration of 5ms at
    38.5kHz on Pin 2.
5   R_IR = In(0)        'Assign the logic input value of Pin 0 into R_IR.
6   If R_IR = 0 Then    'Determine if R_IR is 0. If it is 0, it implies that there is
```

```
        reflection.
7           High 4
8       Else
9           Low 4
10      End If
11
12      SOUND 3,5,38500 'Generate a square wave with the duration of 5ms at
        38.5kHz on Pin 3.
13      L_IR = In(1)        'Assign the logic input value of Pin 1 into L_IR.
14      If L_IR = 0 Then
15          High 5
16      Else
17          Low 5
18      End If
19      Pause 16
20      Loop
21      End Sub
```

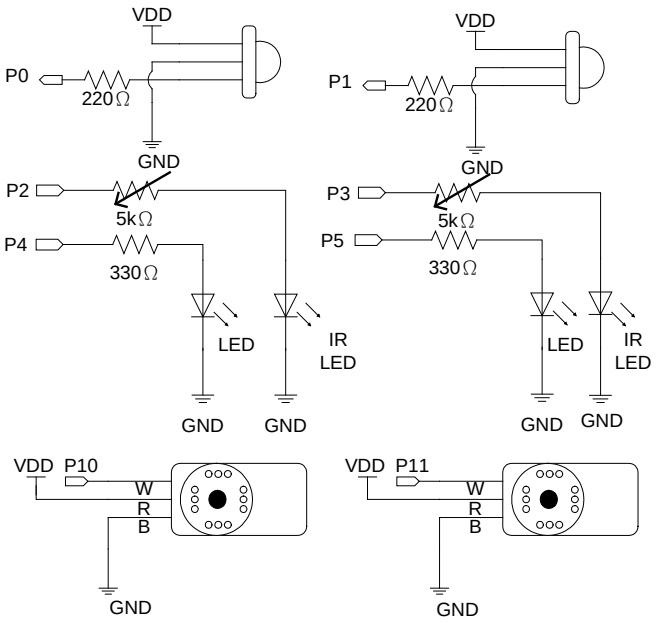
Note to the program: Please refer to the operation manuals of BASIC Commander and innoBASIC Workshop

- For more details about the IN command, please refer to page 125.

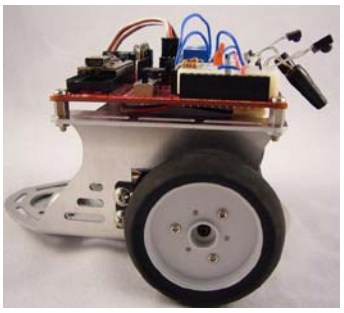
V. IR transmission and receiving, black line movement + LED indication

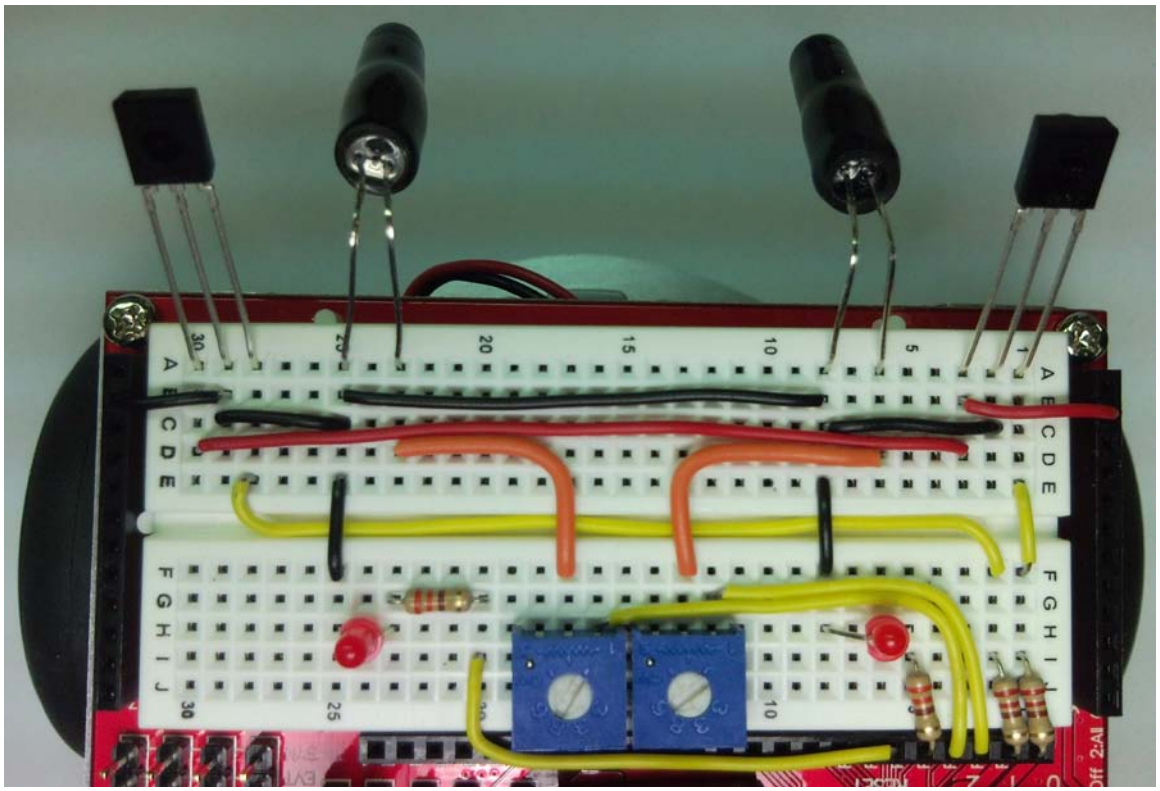
Objective of the Experiment: Make an application that integrates the IR, LED and the 360° servo. Adjust the required sensitivity from Experiment (4) to complete the setting of the movement.

Experiment circuit:



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





Program code:

```
0   Dim R_IR As Byte
1   Dim L_IR As Byte
2   Sub MAIN()
3   Low 10
4   Low 11
5   Do
6   SOUND 2,5,38500
7   R_IR = In(0)
8   If R_IR = 0 Then
9       High 4
10  Else
11      Low 4
12  End If
13
14  SOUND 3,5,38500
15  L_IR = In(1)
16  If L_IR = 0 Then
17      High 5
18  Else
19      Low 5
20  End If
21  If R_IR + L_IR = 2 Then
22      Pulseout 10,350
23      Pulseout 11,250
24      Pause 16
25  Elseif R_IR + L_IR = 0 Then
26      Pulseout 10,350
27      Pulseout 11,250
28      Pause 16
29  Elseif R_IR = 0 Then
30      Pulseout 10,250
31      Pulseout 11,250
32      Pause 16
33  Elseif L_IR = 0 Then
34      Pulseout 10,350
35      Pulseout 11,350
36      Pause 16
37  End If
```

```
38  Loop
39  End Sub
```

Description of the program:

4-18 Check if the left and right IR sensors sense any signal and display the situation on the LED.

19-35 Compare the received sensor data and select the movement path after determination.

Experiment result: It can move along the path of the black line.

PS. The IR receiver may be influenced by other IR emission sources. During the experiment, please turn off or keep away from other IR sources. (Ex.: computer monitor)

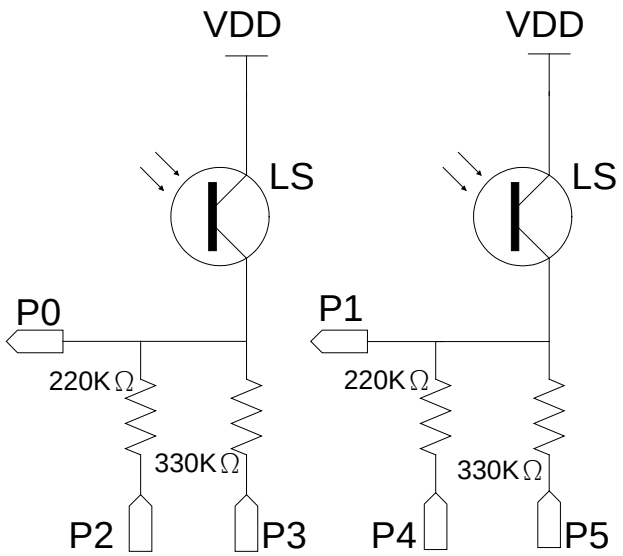
VI. Phototransistor applications

Objective of the Experiment: Understand the control of phototransistors. Use the I/O pins and commands to serve as switching circuit.

Experiment circuit:

LS3200

The range of the used resistor: 10KΩ-470KΩ. If the resistance is higher, its sensitivity is higher.



Program code:

```
0   Dim LS_R As Byte
1   Dim LS_L As Byte
2   Dim DATA_R As Byte
3   Dim DATA_L As Byte
4   Sub MAIN()
5   Low 10
6   Low 11
7   Do
8   Low 2               '220KΩ lead is set as LOW (as short circuit)
9   Input 3             '330KΩ lead is set as INPUT (as open circuit)
10  LS_R = In(0)        'Obtain the data from Pin 0 and store it into LS_R
11  If LS_R = 0 Then    'Determine if LS_R is 0. If it is 1, DATA_R
12                      'Is set as 2 to imply that the light source is nearer.
13  Low 3               '330KΩ lead is set as LOW (as short circuit)
14  Input 2             '220KΩ lead is set as INPUT (as open circuit)
15  LS_R = In(0)        'Obtain the data from Pin 0 and store it into LS_R
16  If LS_R = 0 Then    'Determine if LS_R is 0. If it is 1, DATA_R
17                      'Is set as 1 to imply that a light source exists.
18      DATA_R = 0
```

```
19          Else
20          DATA_R = 1
21          End If
22    Else
23          DATA_R = 2
24    End If

25    Low 4                 '220KΩ lead is set as LOW (as short circuit)
26    Input 5               '330KΩ lead is set as INPUT (as open circuit)
27    LS_L = In(1)          'Obtain the data from Pin 1 and store it into LS_L
28    If LS_L = 0 Then      'Determine if LS_L is 0. If it is 1,
29                          'DATA_L is set as 2 to imply that the light source is nearer.
30    Low 5                 '330KΩ lead is set as LOW (as short circuit)
31    Input 4               '220KΩ lead is set as INPUT (as open circuit)
32    LS_L = In(0)          'Obtain the data from Pin 1 and store it into LS_L
33    If LS_L= 0 Then       'Determine if LS_L is 0. If it is 1,
34                          'DATA_L is set as 1 to imply that a light source exists
35    DATA_L = 0
36    Else
37    DATA_L = 1
38    End If
39    Else
40    DATA_L = 2
41    End If
42    'If the two sensors at both the left and the right sides apparently receive the same
      light intensity, move forward.
43    If DATA_R=2 And DATA_L= 2 Then
44              Pulseout 10,350
45              Pulseout 11,250
46              Pause 16
47    'If the light intensity on the right is higher than that on the left, turn right.
48    Elseif DATA_L < DATA_R Then
49              Pulseout 10,350
50              Pulseout 11,300
51              Pause 16
52    'If the light intensity on the left is higher than that on the right, turn left.
53    Elseif DATA_L > DATA_R Then
54              Pulseout 10,300
```

```
55              Pulseout 11,250
56              Pause 16
57  'If there is no apparent light source, stop the movement.
58  Else
59              Pulseout 10,300
60              Pulseout 11,300
61              Pause 16
62  End If
63  Loop
64  End Sub
```

SOUND

Syntax

SOUND Pin, Duration, Width

Operation

Generate a square wave on the specified pin.

- Pin - a constant or variable value (0-23) to specify the pin that generates the square wave signal. For a 24-pin BASIC Commander, the range of the pin value is 0-15.
- Duration - a constant or variable value (0-65535) to specify the duration of the signal generation in ms.
- Width - a constant or variable value (0-65535) to specify the width of the square wave in Hz.

Example

The following program is an example that uses the SOUND command to generate a sound of the note “low C” (Do) on the piezo buzzer through Pin 0 for 5 seconds. (The required frequency of the note “C” is 523Hz.)

```
Sub main( )
SOUND 0,5000,523
End Sub
```