Thanks for your support on our products, we will continue to provide you better quality and service!

Content

About keyestudio

1. Keyestudio is a best-selling brand owned by KEYES Corporation, our product lines range from Arduino boards, shields, sensor modules, Raspberry Pi, micro:bit extension boards and smart car to complete starter kits designed for customers of any level to learn Arduino knowledge.
2. All of our products comply with international quality standards and are greatly appreciated in a variety of different markets throughout the world.
3. Welcome check more contents from our official website:
4. http://www.keyestudio.com

*References and After-sales Service

5. Download Profile : https://fs.keyestudio.com/KS0470
6. Feel free to contact us please, if there is missing part or you encounter some troubles. Welcome to send email to us : service@keyestudio.com. We will update projects and products continuously based on your sincere advice.

*Warning

1. This product contains tiny parts(screws, copper pillars), keep it out of reach of children under 7 years old please.

2. This product contains conductive parts (control board and electronic module). Please operate according to the requirements of this tutorial. Improper operation may cause overheating and damage parts. Do not touch and immediately disconnect the circuit power.

*Copyright

4WD BT Multi-purpose Smart Car V2.0 Kit

Mixly tutorial

1. Introduction

Nowadays, technological education such as VR, kids programming, and artificial intelligence, has become mainstream in educational industry. Thereby, people attach importance to STEAM education.

The 4WD multi-purpose robot car, newly upgraded by the Keyes team, is one of the most favoured programming robots.

It is not only beautiful in appearance, but also powerful in function. In addition to the common function like line tracking, obstacle avoidance and remote control, etc.

15 learning projects, from simple to complex, will guide you how to make a smart 4wd robot on you own and introduce the detailed knowledge

about sensors and modules.

Simultaneously, it is the best choice if you intend to obtain a DIY robot for learning programming, entertainment and competition requirement.

Note: The experiment you did should be in line with wiring diagram, including about components and wiring method. For example, we supply power with external power in the hook-up diagram, so you also have to use external power rather than USB cable .

2. Features

1. Multi-purpose function: Obstacle avoidance, follow, IR remote control, Bluetooth control, ultrasonic follow and displayed face emoticons.
2. Simple assembly: No soldering circuit required, complete assembly easily.
3. High Tenacity: Aluminum alloy bracket, metal motors, high quality wheels and tracks
4. High extension: expand other sensors and modules through motor driver shield and sensor shield
5. Multiple controls: IR remote control, App control(IOS and Android

system)

6. Basic programming : Mixly programming.

3. Specification

Working voltage: 5v

Input voltage: 7-12V

Maximum output current: 2A

Maximum power dissipation: 25W (T=75℃)

Motor speed: 5V 200 rpm/min

Motor drive mode: dual H bridge drive

Ultrasonic induction angle: <15 degrees

Ultrasonic detection distance: 2cm-400cm

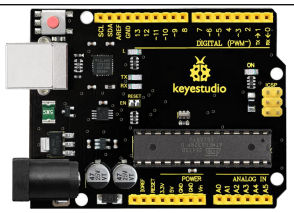
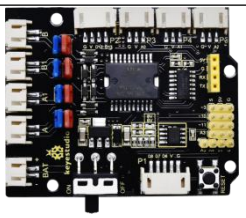
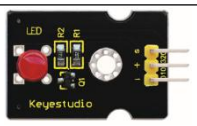
Infrared remote control distance: 10 meters (measured)

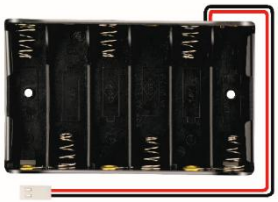
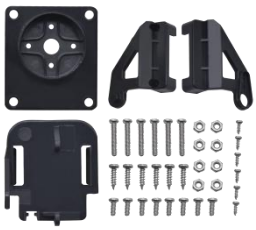
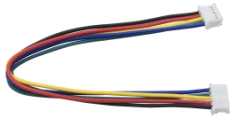
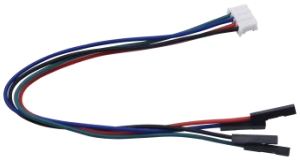
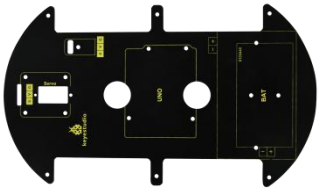
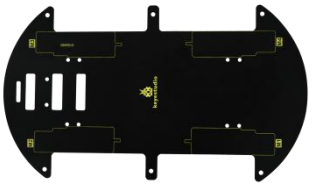
Bluetooth remote control distance: 50 meters (measured)

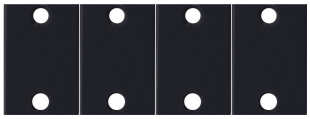
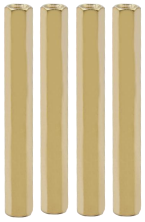
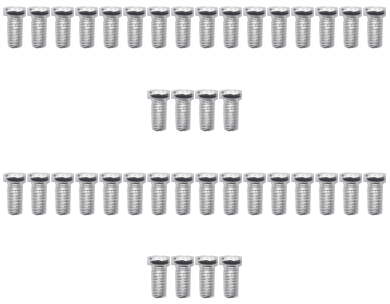
Bluetooth control: support Android and iOS system

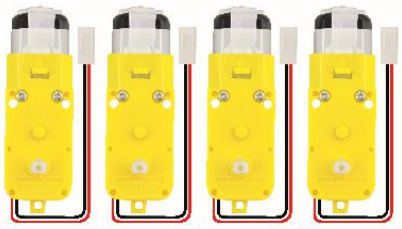
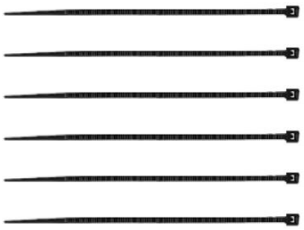
4. Product List

No	Name	QTY	Picture

1	Keyestudio V4.0 Board	1	
2	Keyestudio Motor Driver Shield	1	
3	Keyestudio HM-10 Bluetooth-4.0	1	
4	Red LED Module	1	
5	HC-SR04 Ultrasonic Sensor	1	
6	Keyestudio Line Tracking Sensor	1	
7	Keyestudio IR Receiver Sensor	1	
8	Keyestudio 8*16 LED Dot Matrix	1	
	4pinDupont Line	1	
9	Keyestudio 9G Servo	1	
10	Keyestudio Remote Control	1	

11	USB Cable	1	
12	18650 Battery Holder	1	
13	6 AA Battery Holder	1	
14	Servo Platform	1	
15	15CM F-F 5P 24AWG Dupont Wire	1	
16	8cm F-F 3P 24AWG Dupont Wire	1	
17	4P to 1P Female DuPont Wire	1	
18	Acrylic Board	1	
19	Keyestudio 4WD Smart Car V2.0 Top Board	1	
20	Keyestudio 4WD Smart Car V2.0 Bottom PCB	1	

21	Fixed Parts	4	
22	Wheel	4	
23	M3*10MM Dual-pass Copper Bush	10	
24	M3*40MM Dual-pass Copper Bush	4	
25	M3*30MM Round Head Screws	8	
26	M3*6MM Round Head Screws	40	
27	M3 Nickel Plated Nuts	16	
28	M2X8MM Round Head Screws	6	

29	M3*8MM Round Head Screws	4	
30	M2 Nickel Plated Nuts	6	
31	M3*10MM Flat Screws	3	
32	Motor (with welding wire)	4	
33	3*40MM Screwdriver	1	
34	Black Nylon Ties 3*100MM	6	
35	Winding Pipe	1	
36	3Pin F-F Dupont Wire (20CM)	3	
37	Decorative Board		

5. Assembly Guide

Note: Peel the plastic film off the board first when installing the smart car. To be honest, we never intend to send wood to you.

Step 1:Mount the Bottom PCB

- Prepare the parts as follows:

Gear Motor *4

Fixed Part *4

M3 Nickel Plated Nut *10

M3*6mm Round Head Screw *14

4WD Bottom PCB *1

Tracking Sensor *1

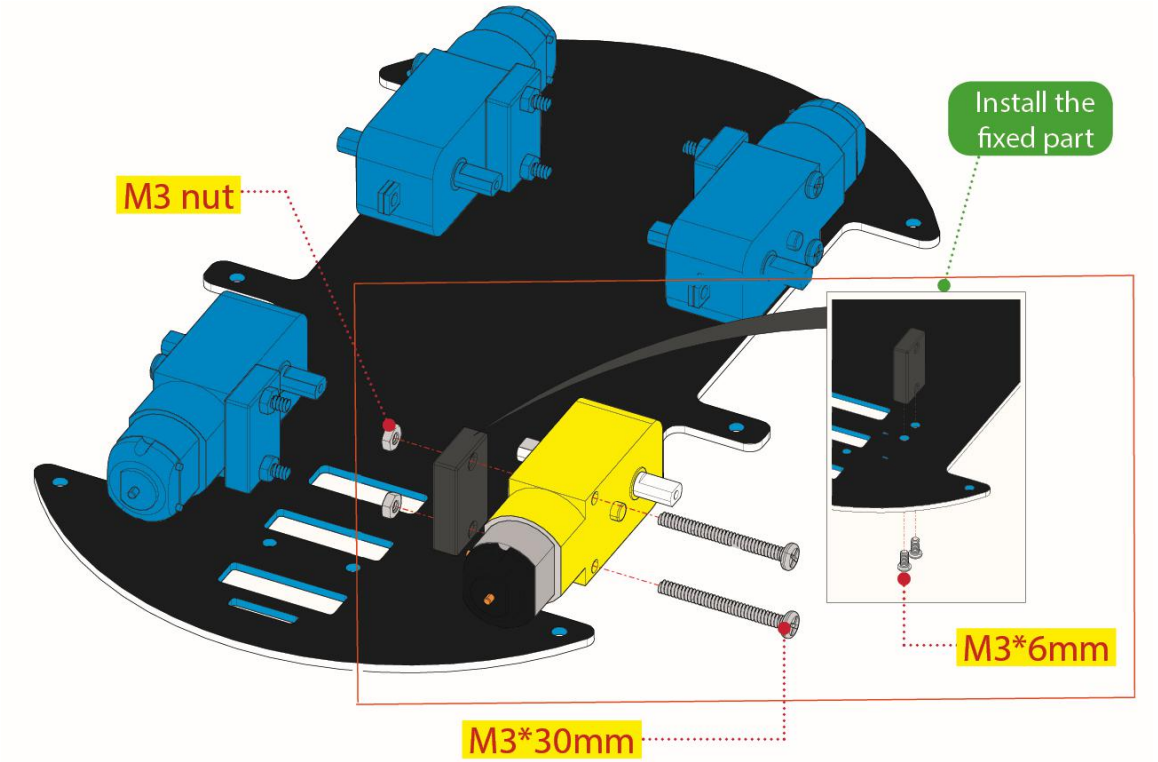
Wheel *4

5P Dupont Wire *1

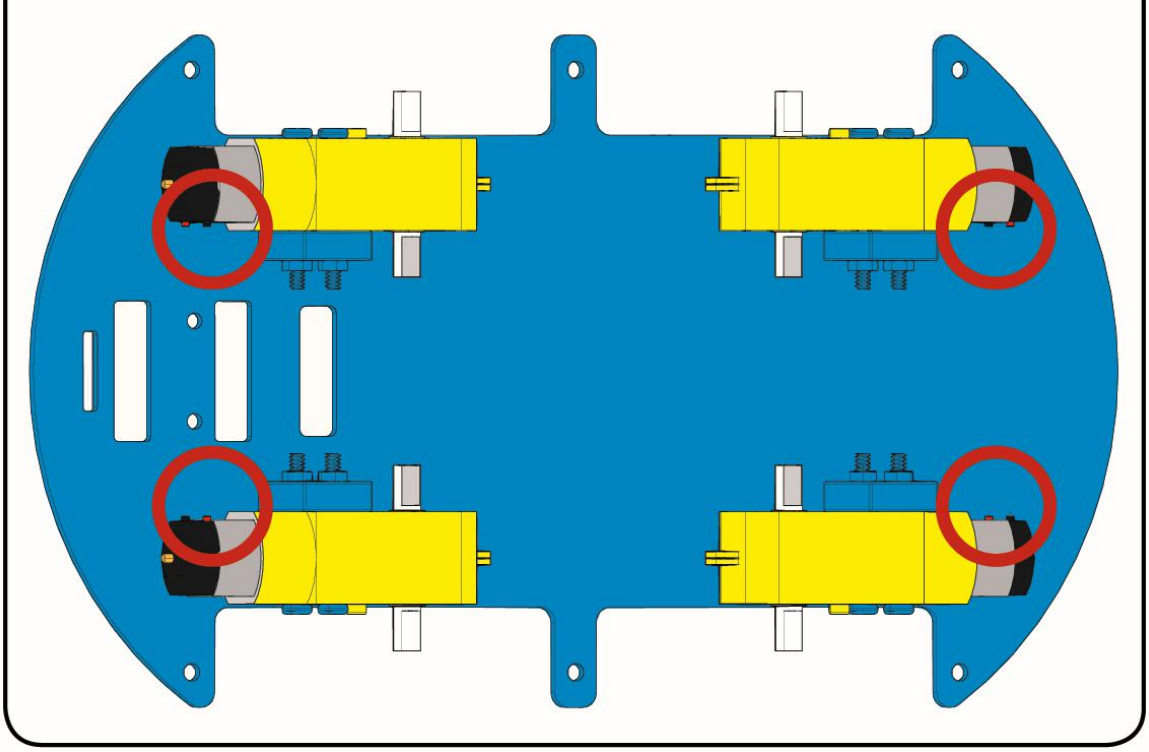
M3*40mm Copper Pillar*6

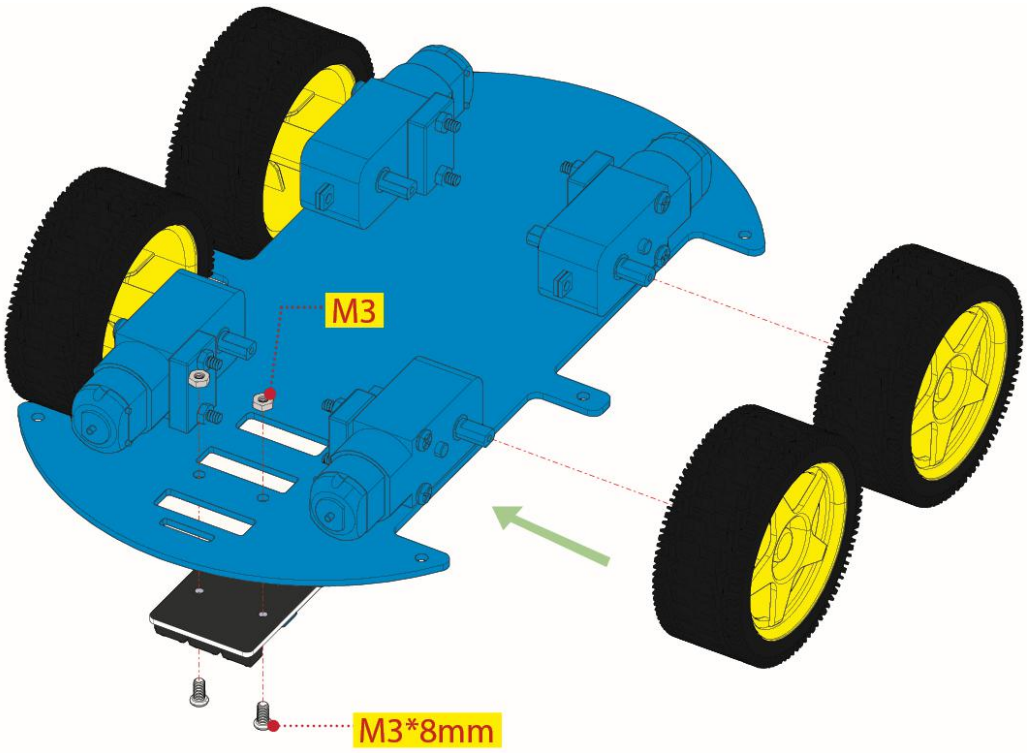
M3*30m Round Head Screw *8

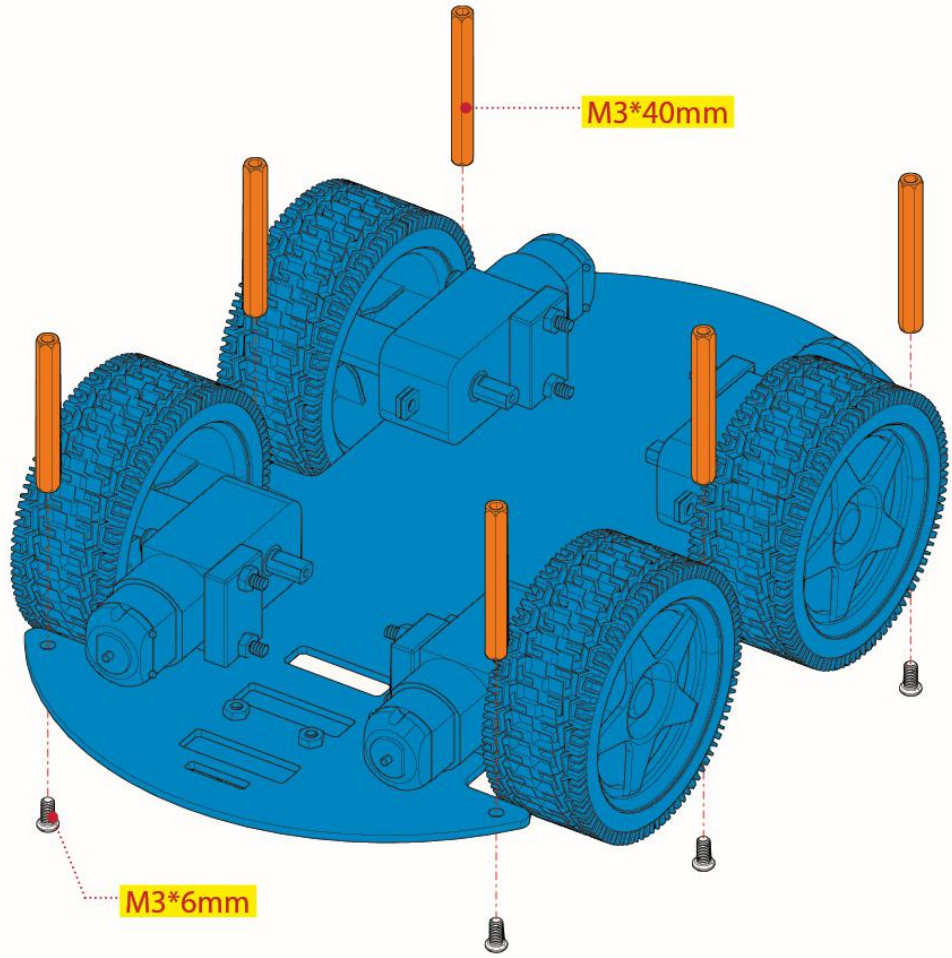
M3*8mm Round Head Screw *2

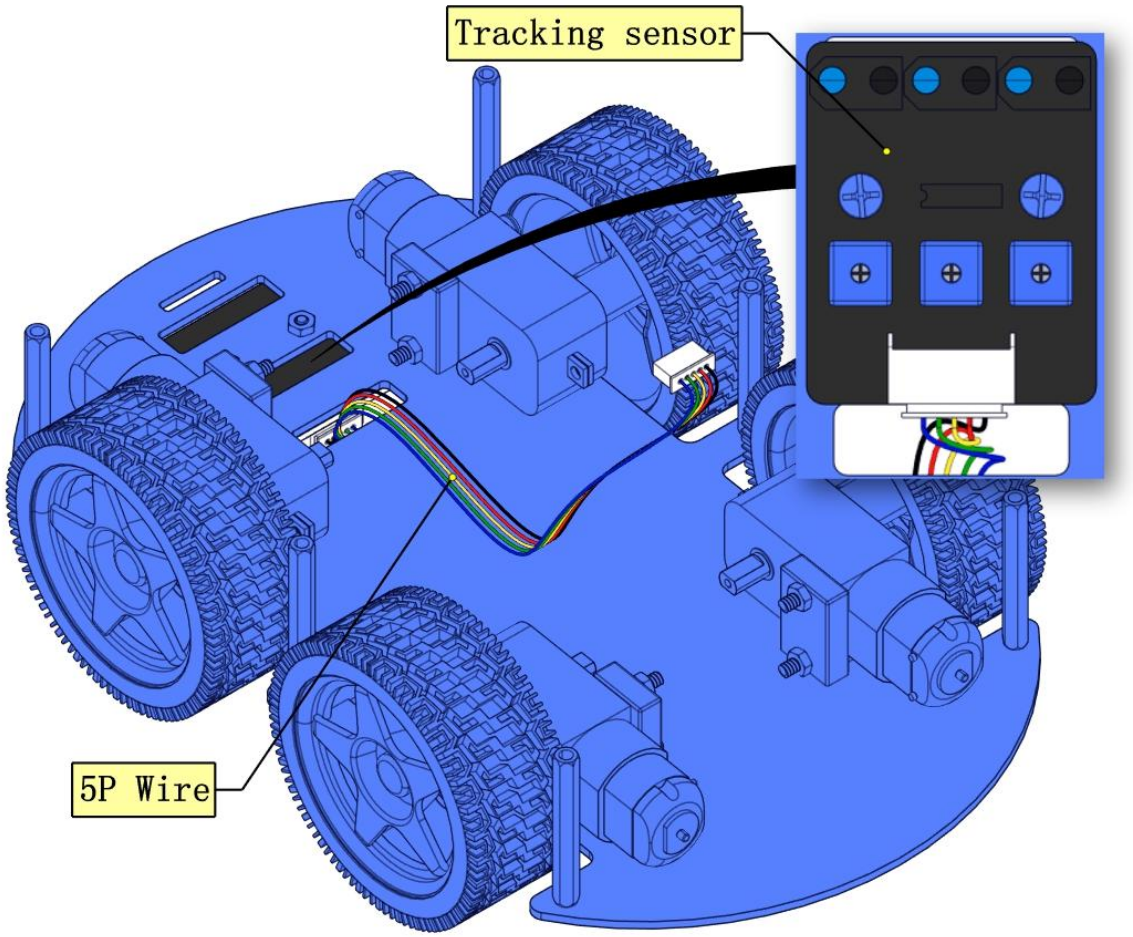


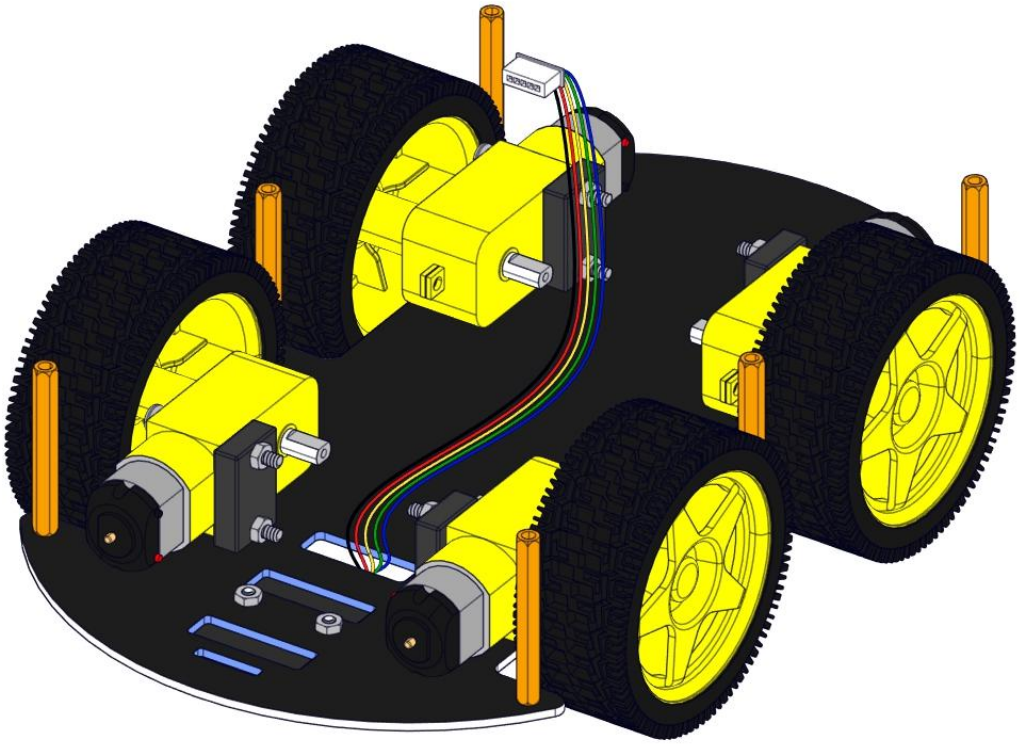
Note: The ways to install motors are same, pay attention to the direction please. Their position holes are downward and the red and black lines are inward




M3
M3*8mm


M3*40mm
M3*6mm


Tracking sensor
5P Wire



Step 2: Install Dot Matrix

- Prepare the parts as follows:

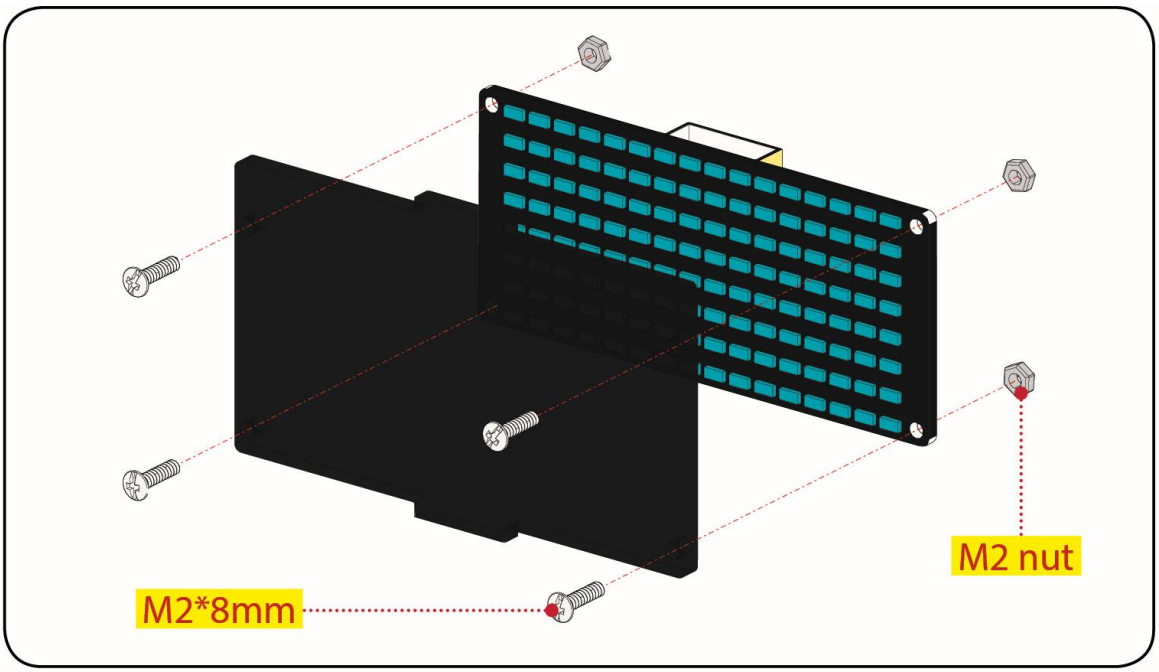
8X16 LED Panel *1

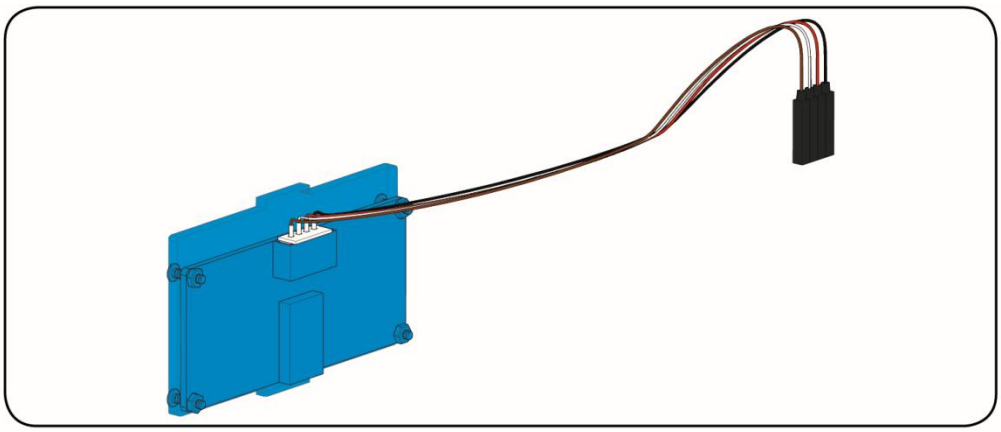
4WD Baffle

4P Wire *1

M2*8mm Round Head Screw *4

M2 Nut *4





Step 3: Install the Plastic Platform of Servo

- Prepare the parts as follows:

Servo *1

M2*4 Screw *1

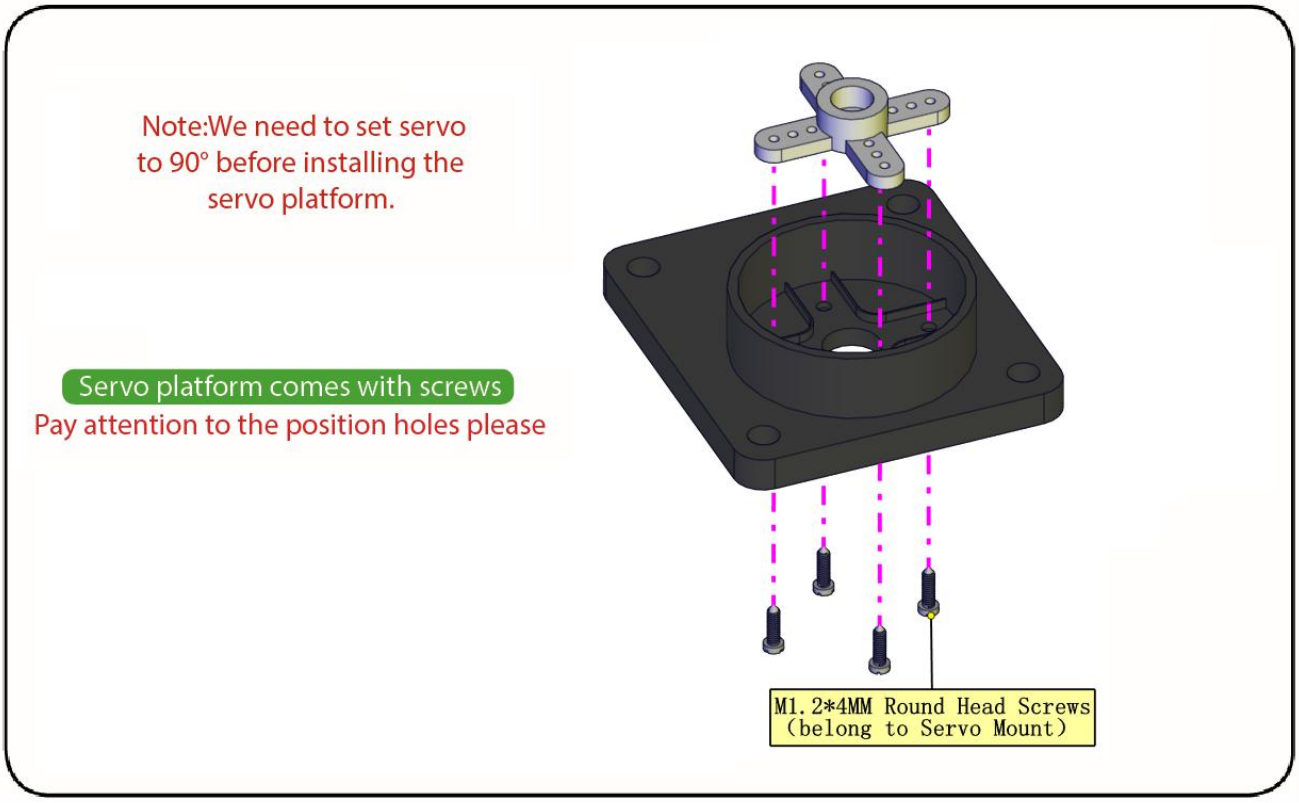
Black Cable Tie*2

Ultrasonic Sensor*1

Black Plastic Platform *1

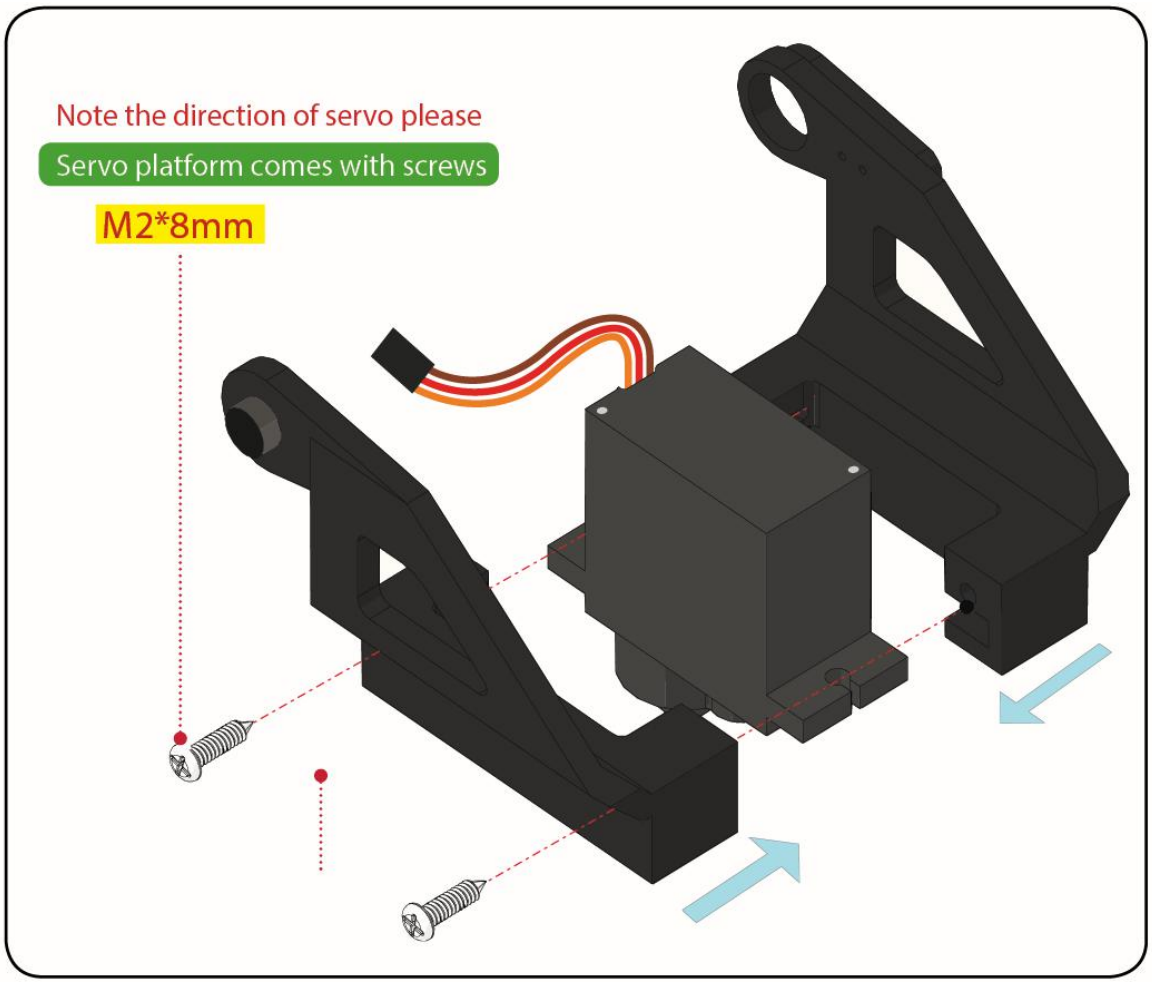
M1.2*4 Tapping Screw *4

M2*8 Tapping Screw *2

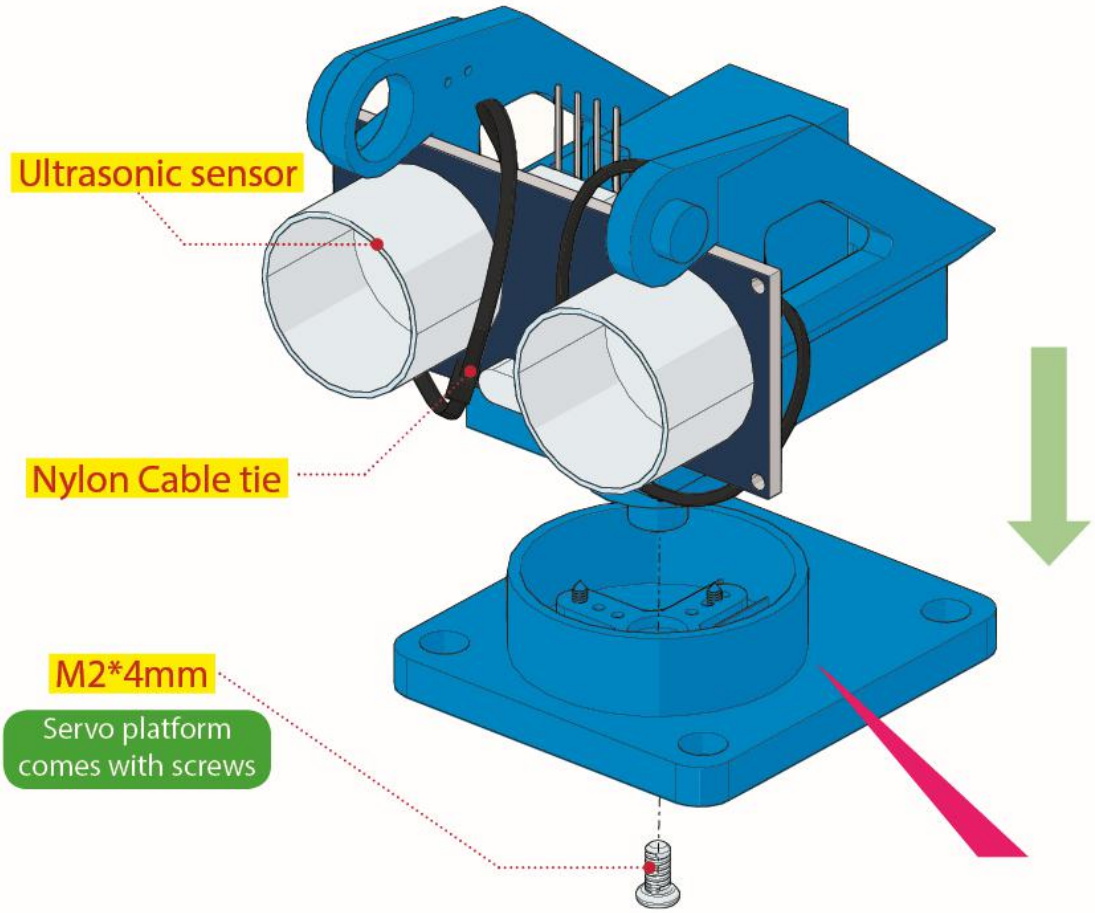


You can find M1.2*4 screws inside the bag of the servo platform

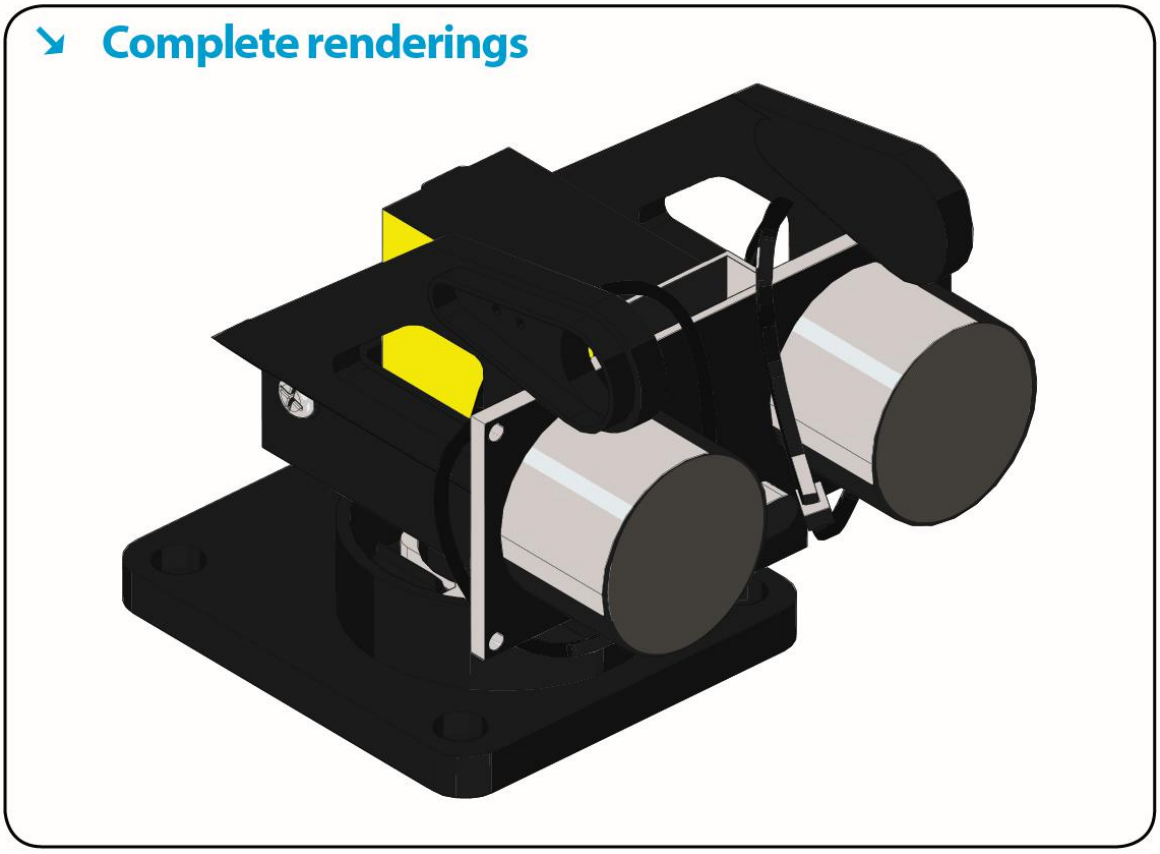
14	Servo Platform	1	


Note the direction of servo please
Servo platform comes with screws
M2*8mm

↘ Complete renderings



Note: The direction of base should be complied with the diagram,
For convenient debugging, confirm that the ultrasonic module is in front of tank robot



Step 4: Assemble Battery Holder

- Prepare the parts as follows:

Top PCB *1

M3 Nut *3

Motor Driver Board *1

Control Board *1

IR Receiver Module *1

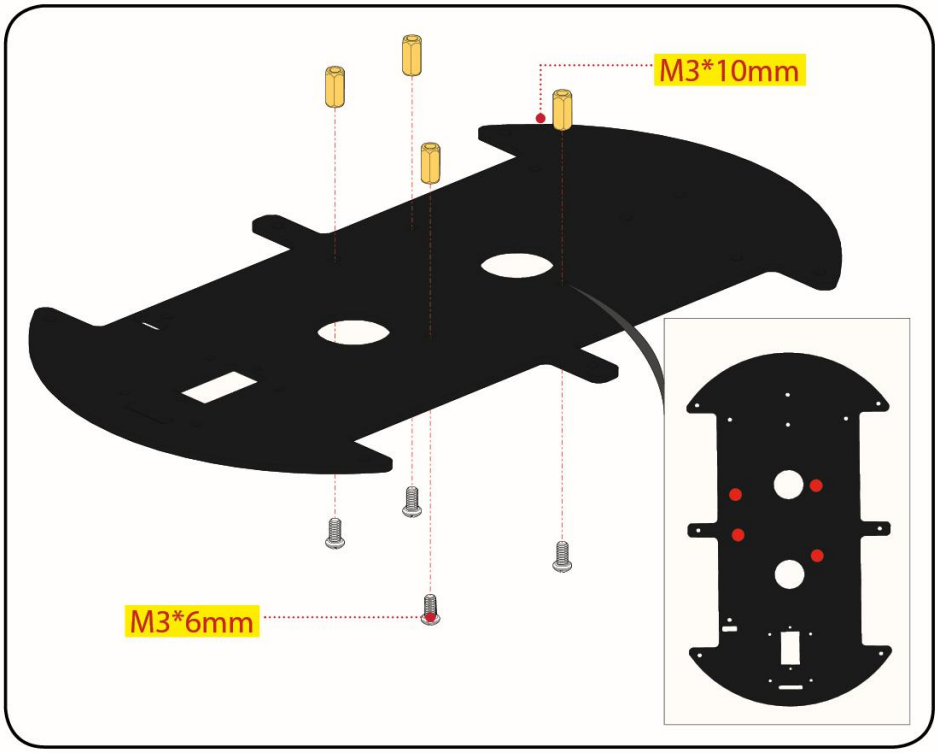
M3*10mm Copper Pillar *8

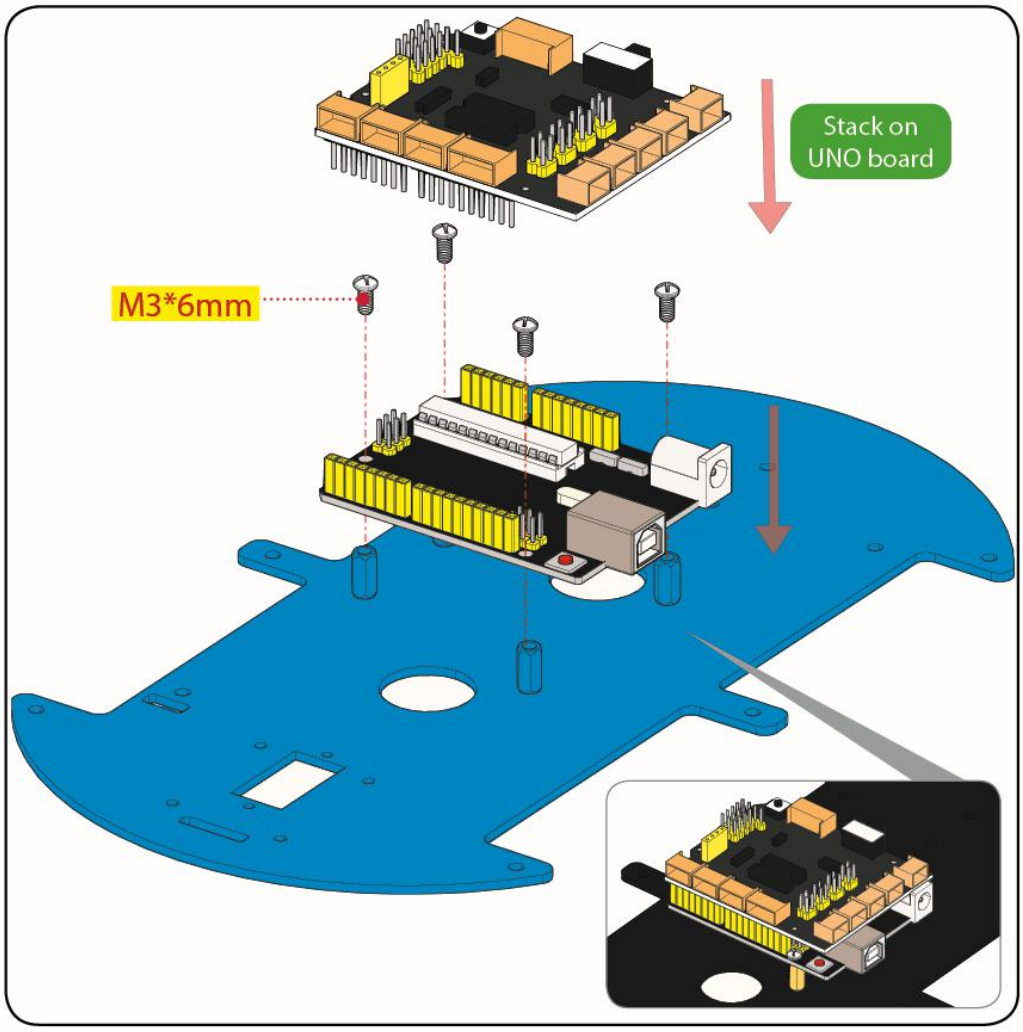
M3*8mm Round Head Screw *1

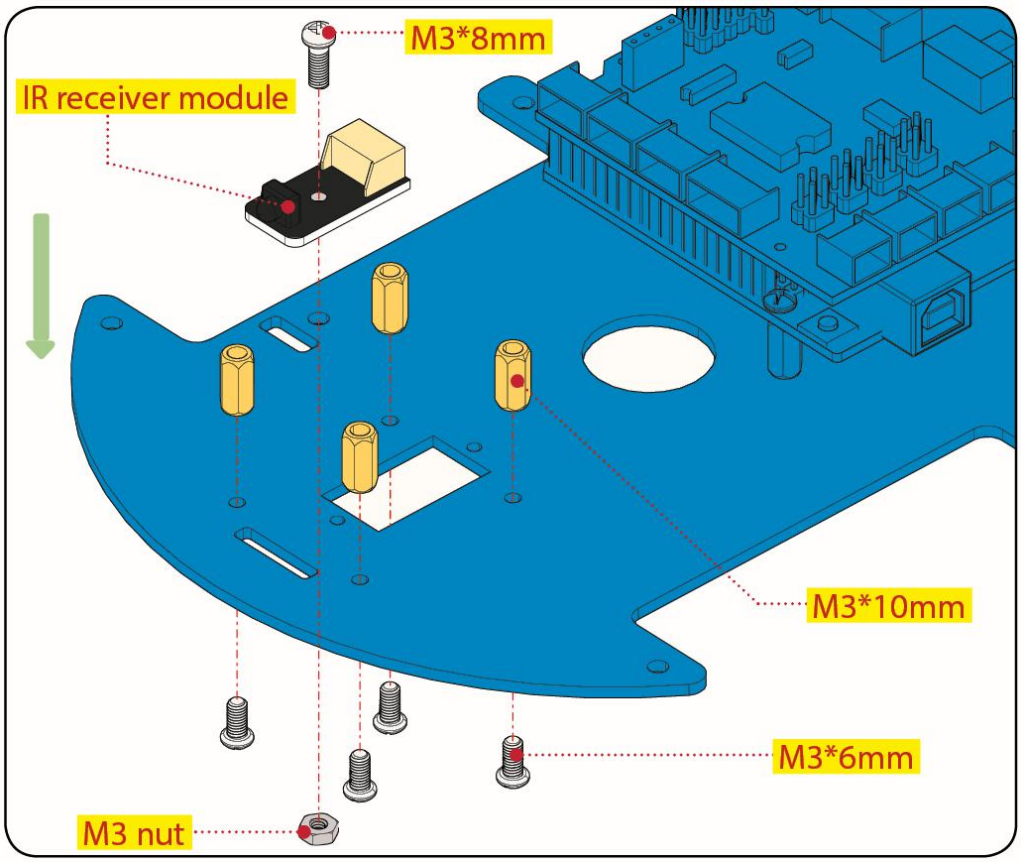
M3*6mm Round Head Screw *16

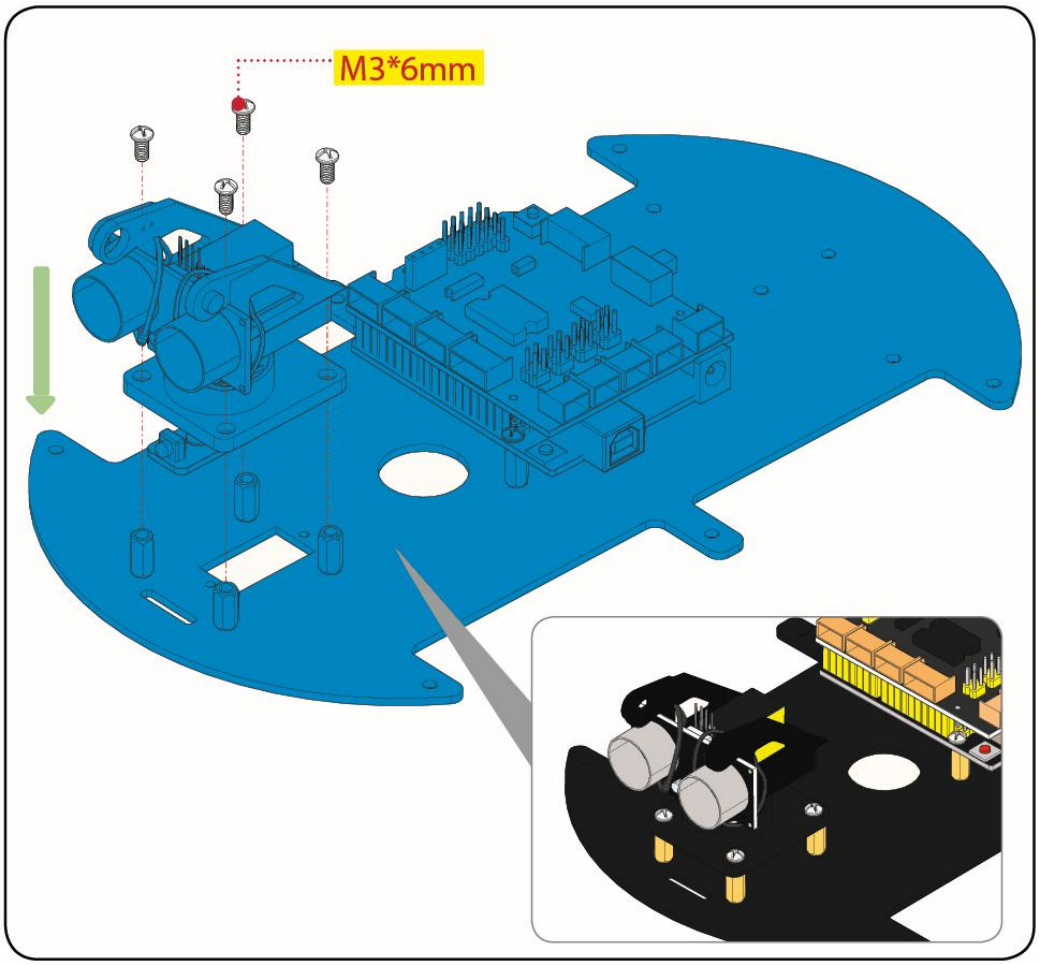
M3*10mm Flat Screw *2

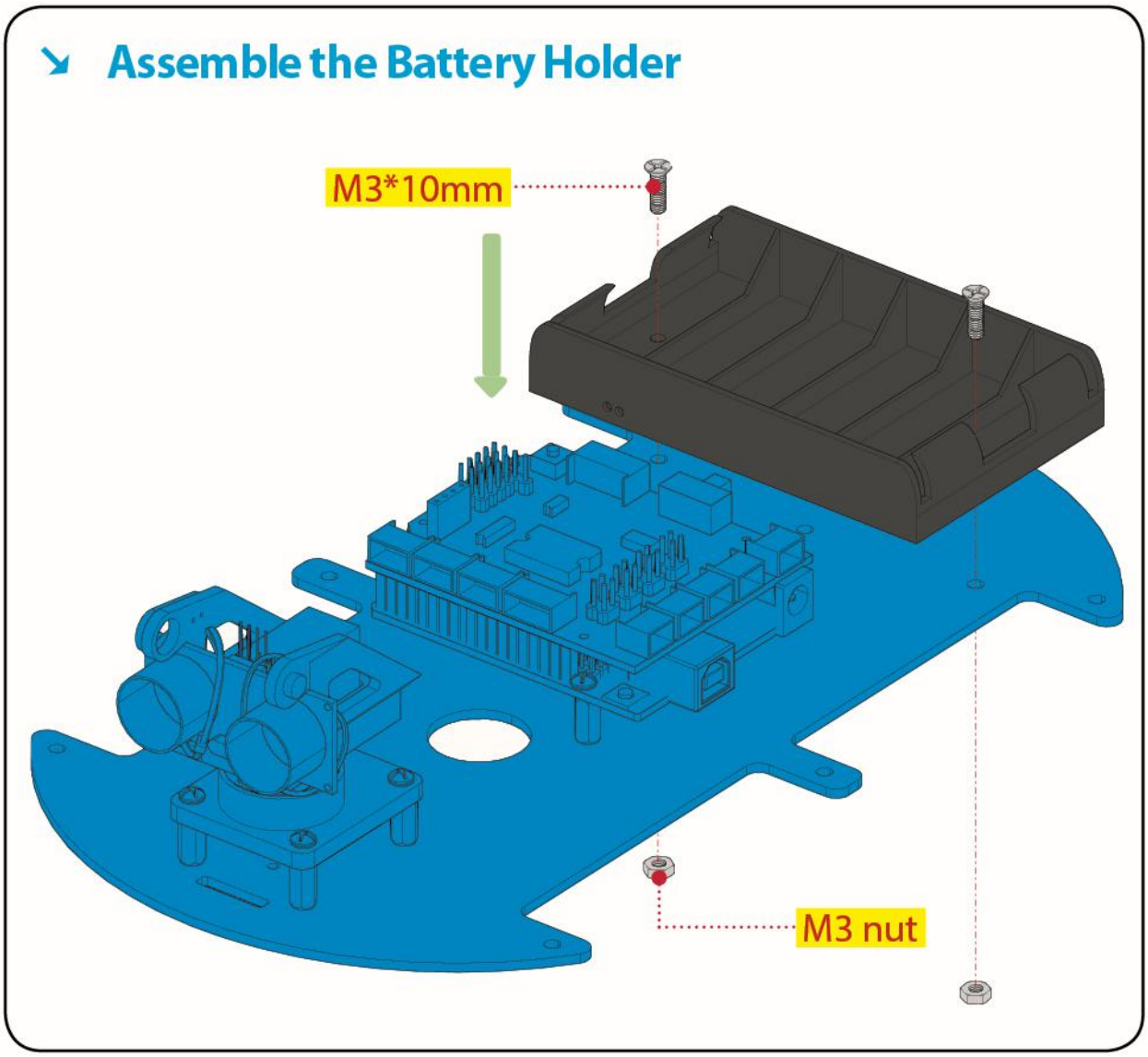
6 AA Battery Holder *1




Stack on
UNO board
M3*6mm


M3*8mm
IR receiver module
M3*10mm
M3*6mm
M3 nut


M3*6mm



Step 5: Mount the Top PCB

- Prepare the parts as follows:

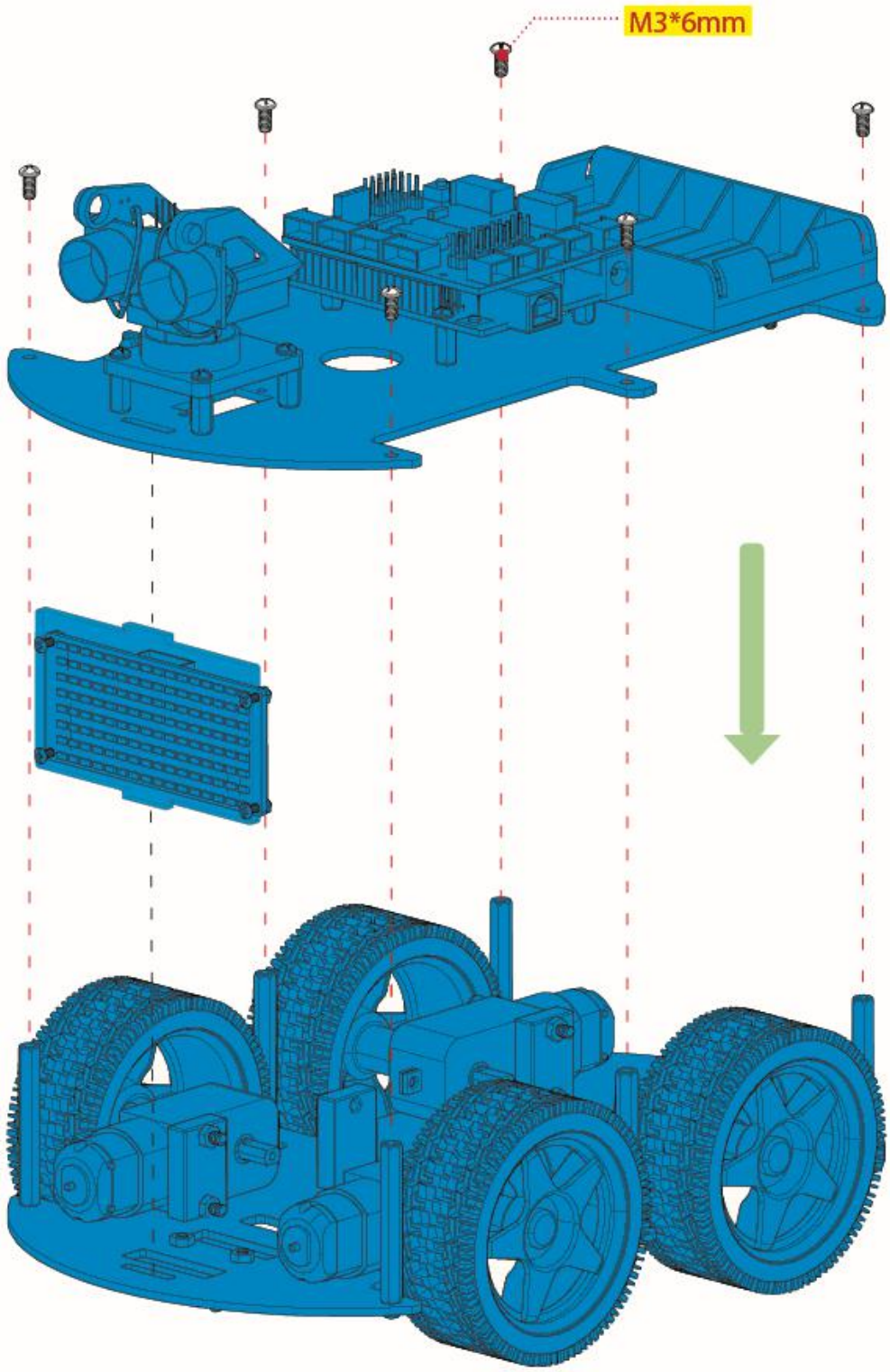
  Bluetooth Module *1

  M3*6MM Round Head Screw *6

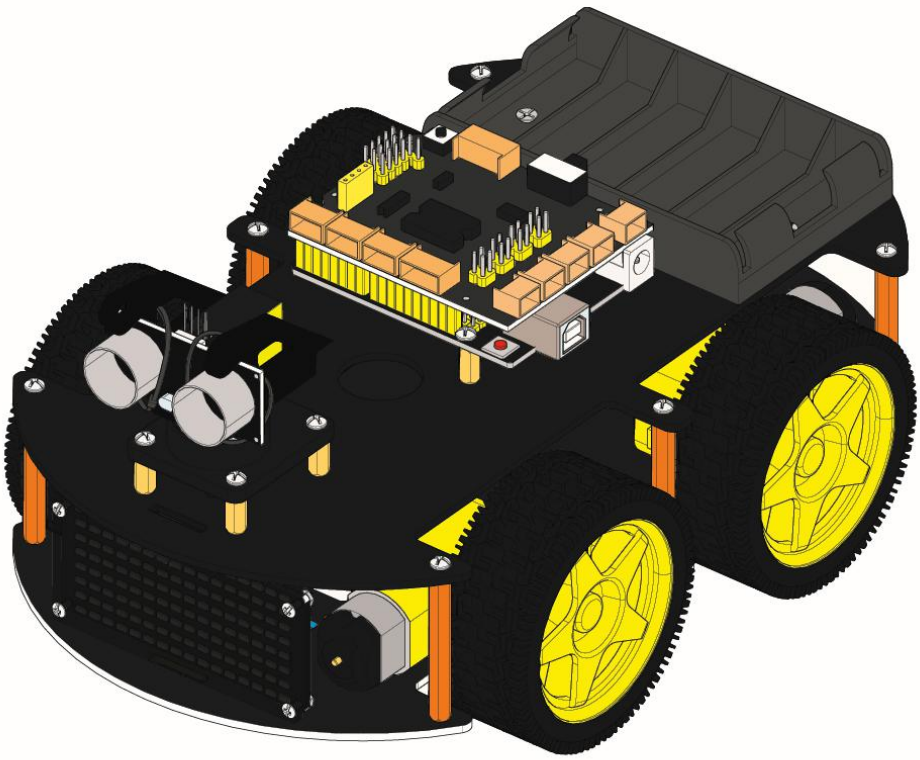
  Jumper Cap*8

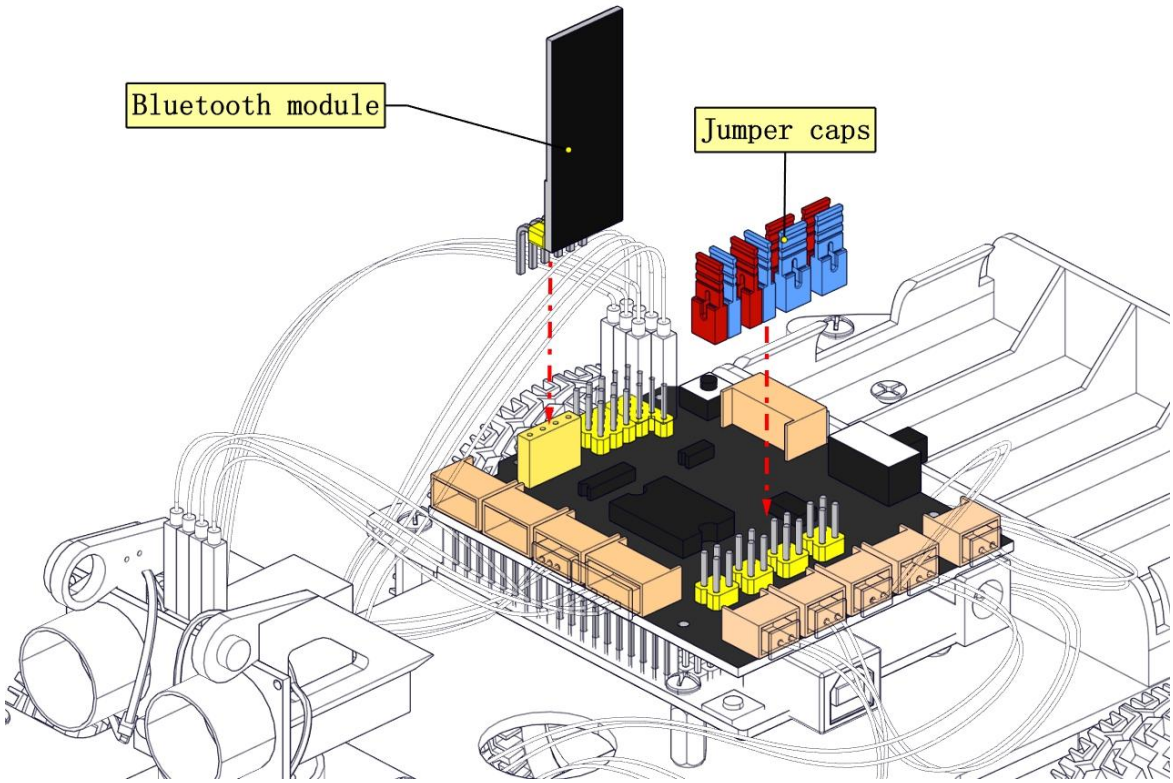
Note: you need to operate the following steps first before stacking

1. Insert 4P wire of dot matrix and lines (M2, M3) of motor into the front hole
2. Insert 5P wire of line tracking sensor and the lines (M1, M4)of motor into the back hole

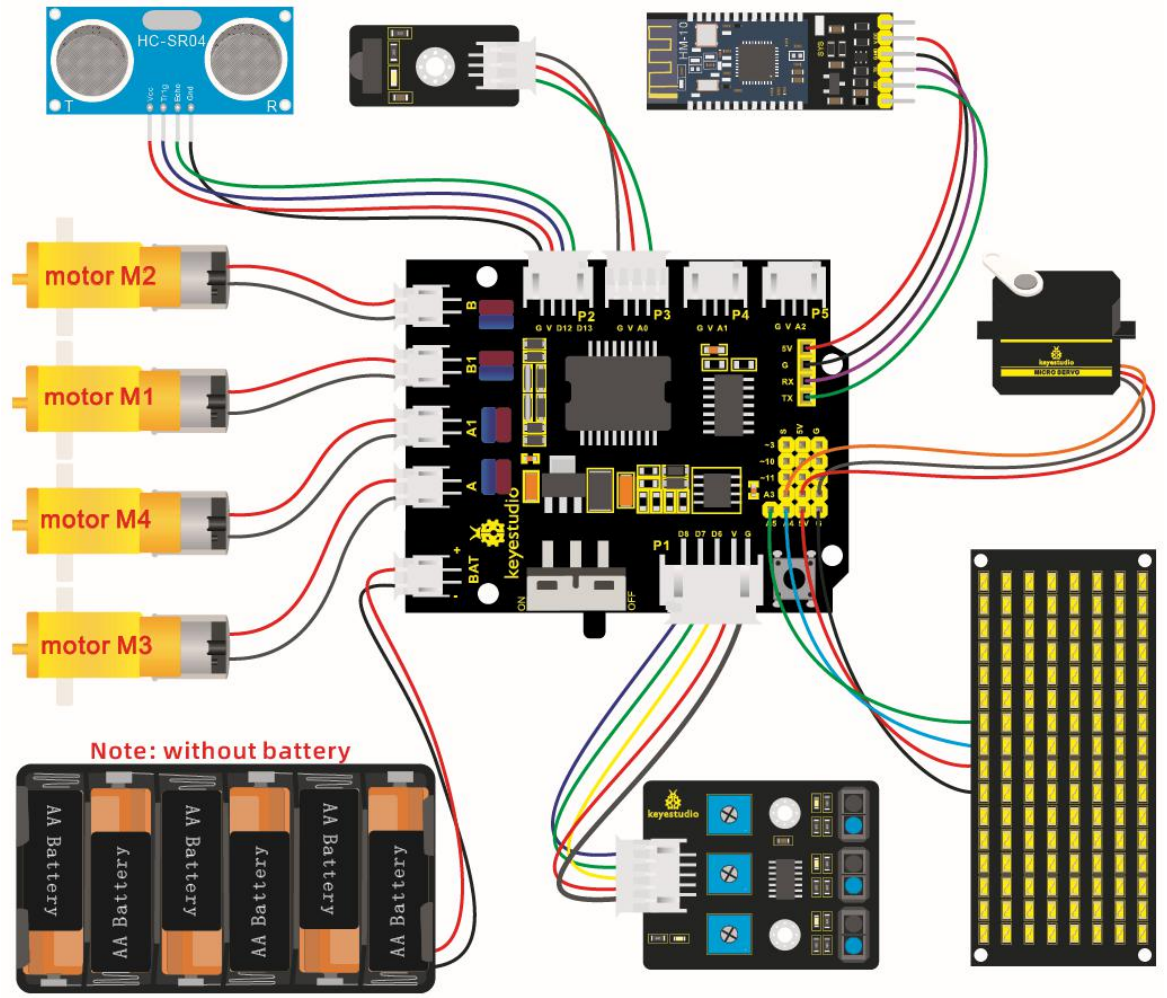


Complete renderings

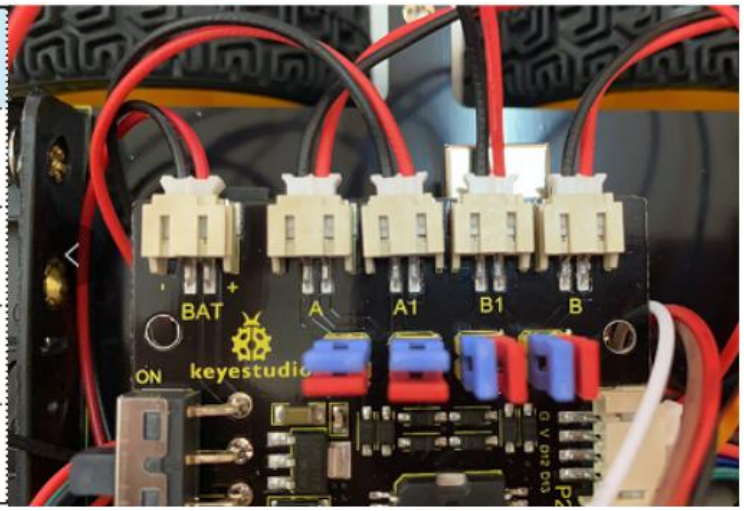




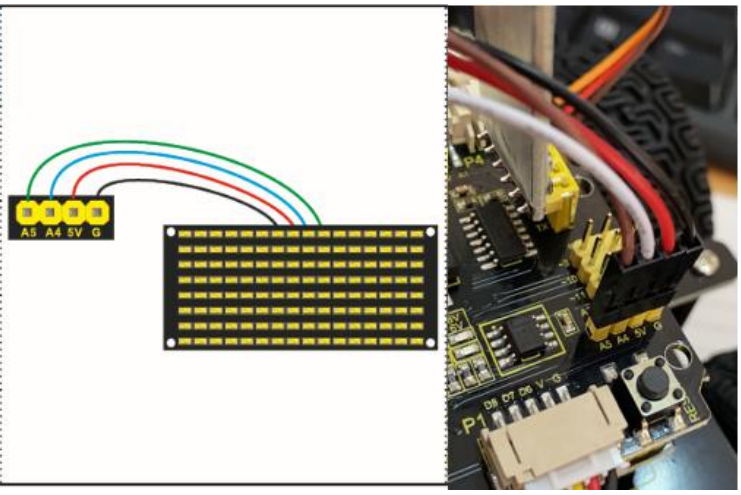
Step 6: Hook-up Guide



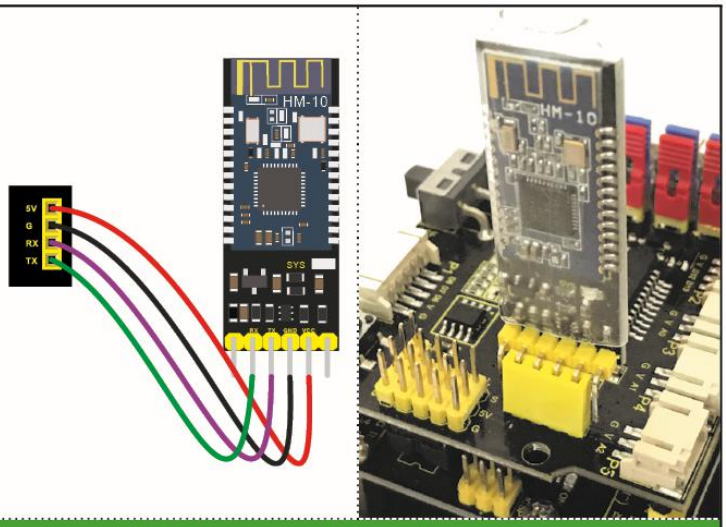
Motor	L298P Shield
M1	B1
M2	B
M3	A
M4	A1



LED Panel	L298P Shield
GND	G
VCC	5V
SDA	A4
SCL	A5

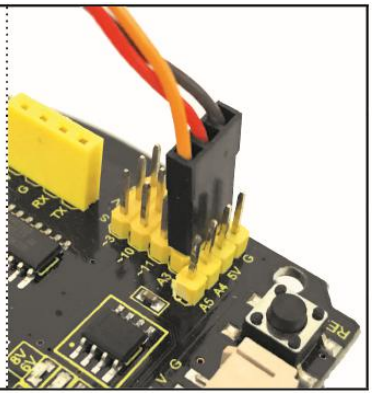


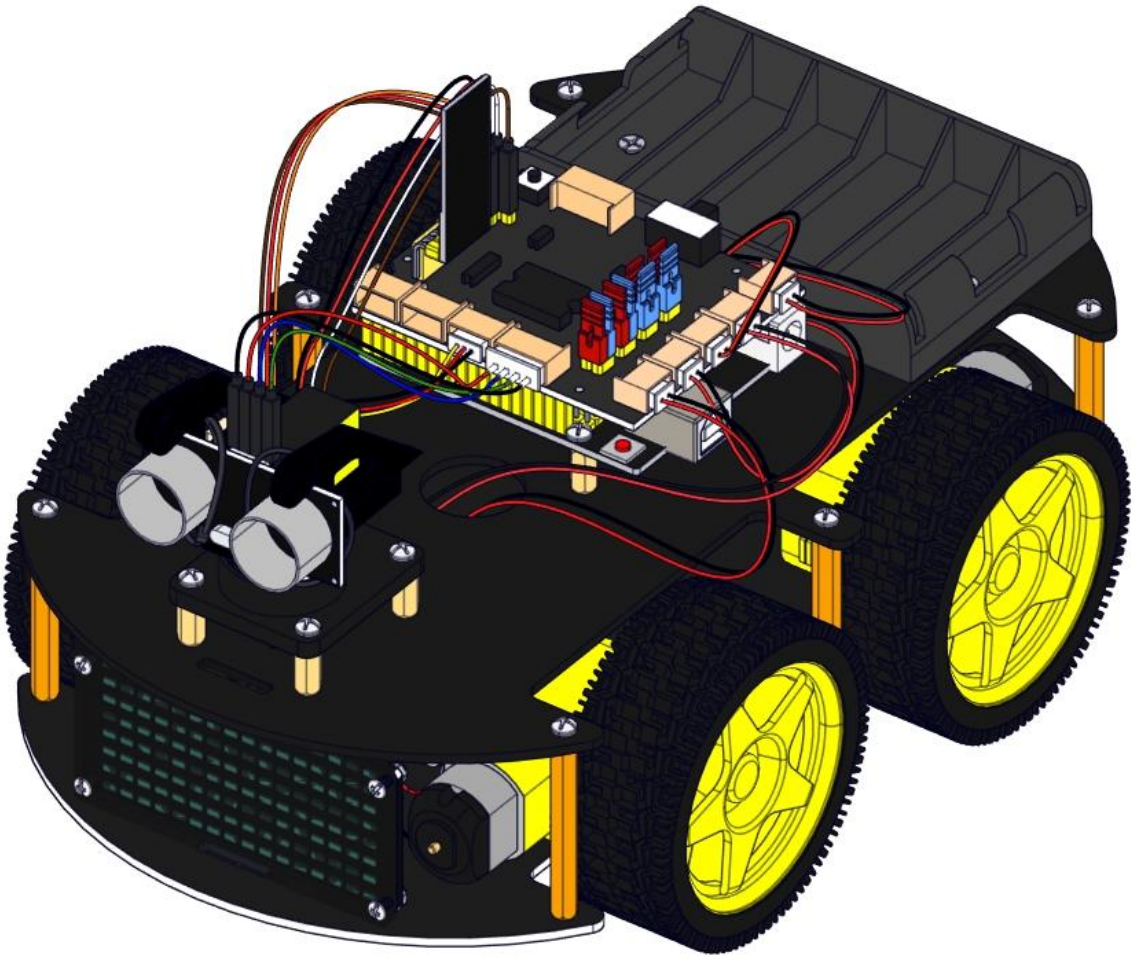
Bluetooth	L298P Shield
RXD	TX
TXD	RX
GND	G
VCC	5V



Note: Remove the Bluetooth module before uploading the program

Servo	L298P Shield
Brown wire	G
Red wire	5V
Orange wire	A3





6.Install Mixly Software and Driver

(1) Download and Install Mixly

①Description

Mixly is a free open-source graphical Arduino programming software, based on Google ' s Blockly graphical programming framework, and developed by Mixly Team@ BNU.

It is a complete support ecosystem for creative e-education, a stage for maker educators to realize their dreams.

② Download Mixly1.0

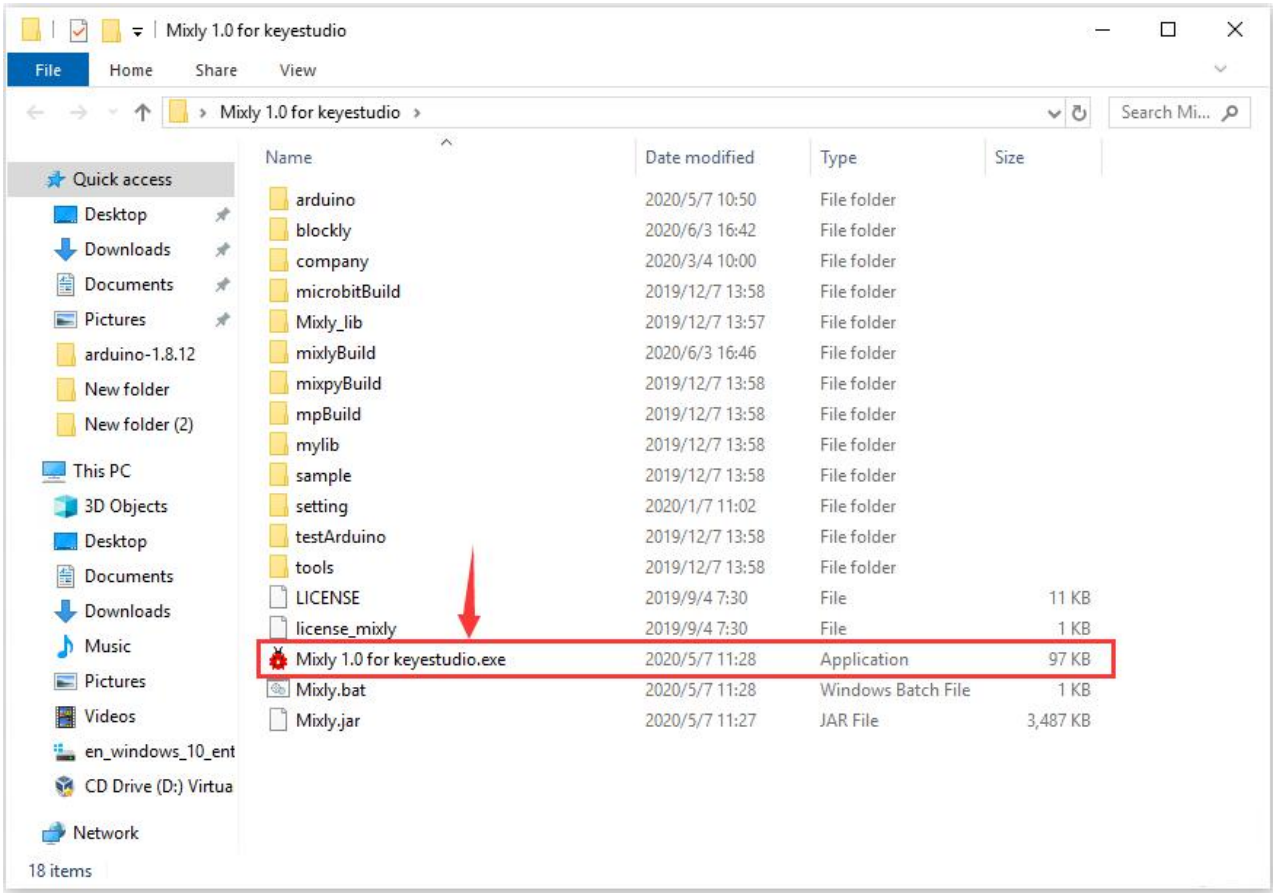
Windows System : https://fs.keyestudio.com/Mixly1-Windows

MACOS System : https://fs.keyestudio.com/Mixly1-MACOS

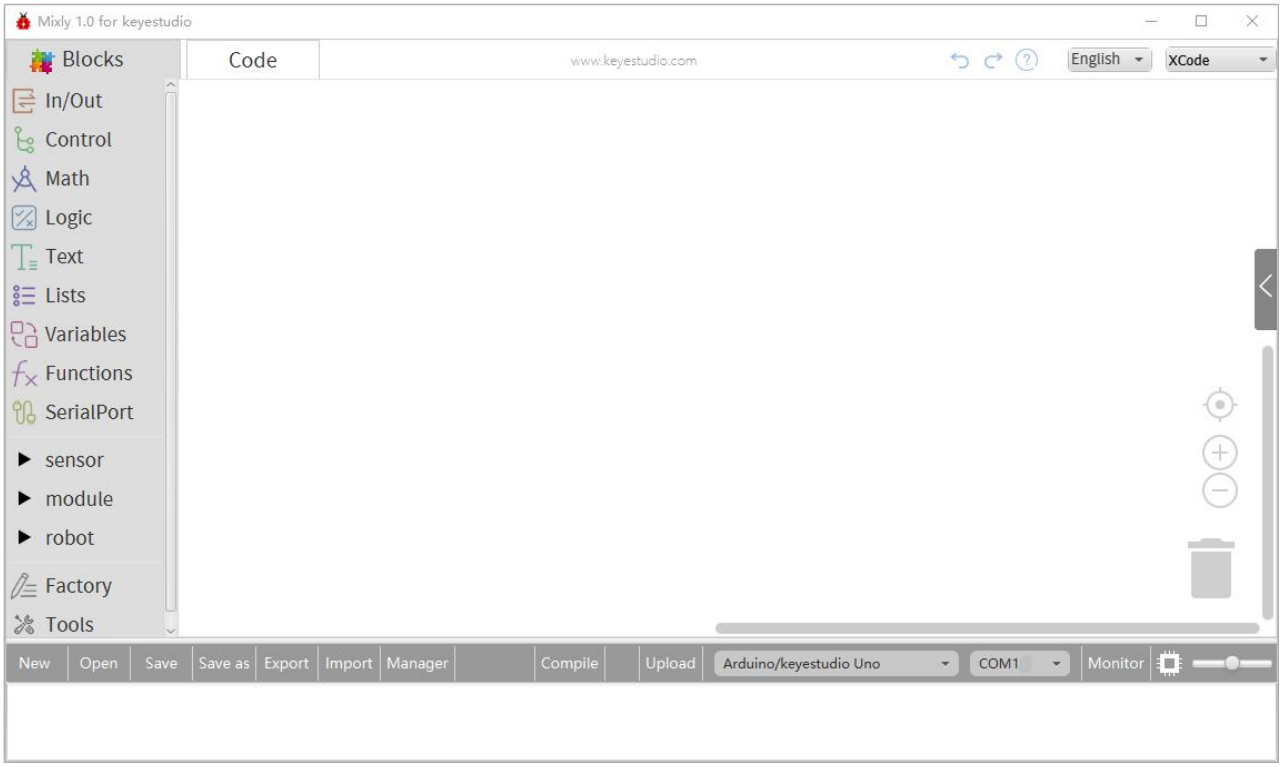
We will take Mixly1.0 (Windows version) as example, and the installation method of MAC version is similar with it.

You will get installation package after downloading. As shown below:

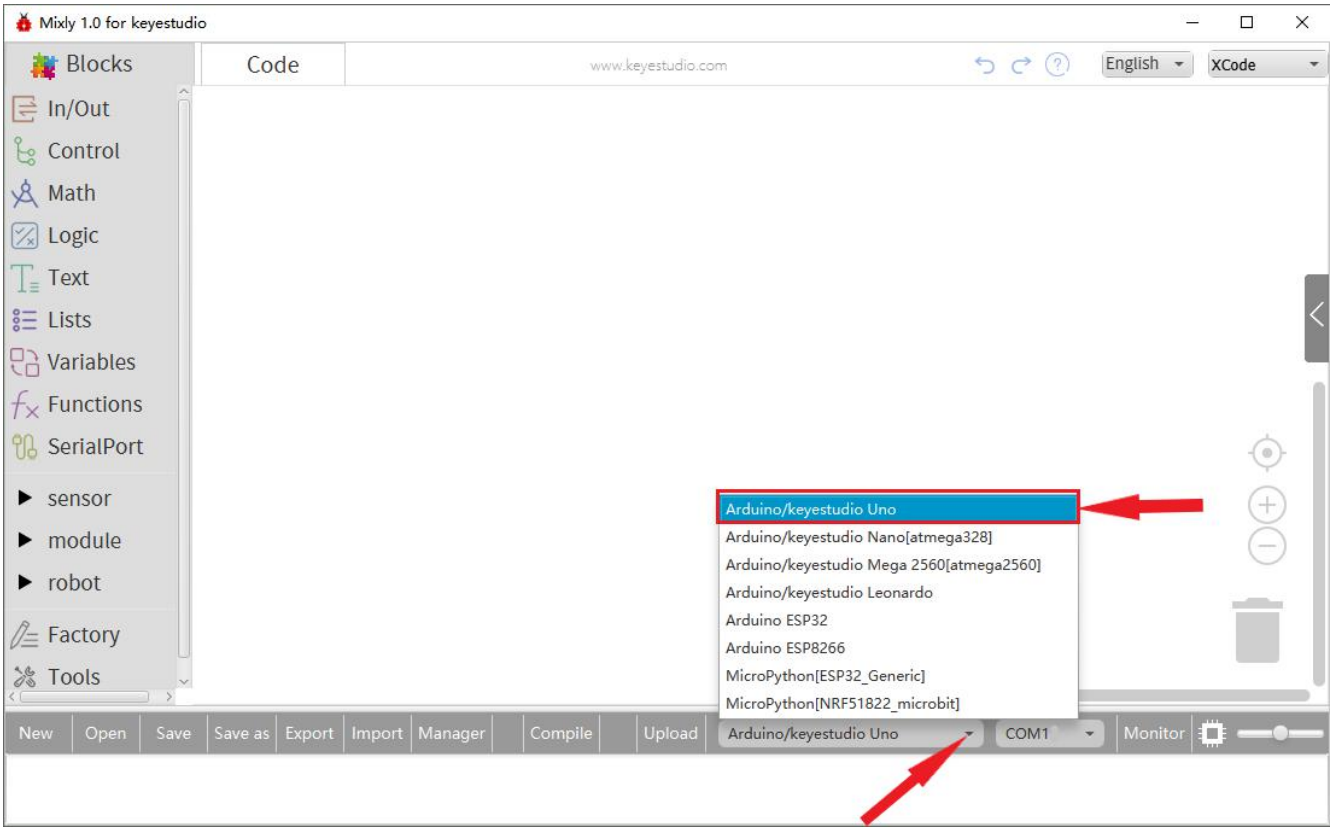
Unzip the package, you will see “Mixly 1.0 for keyestudio.exe”



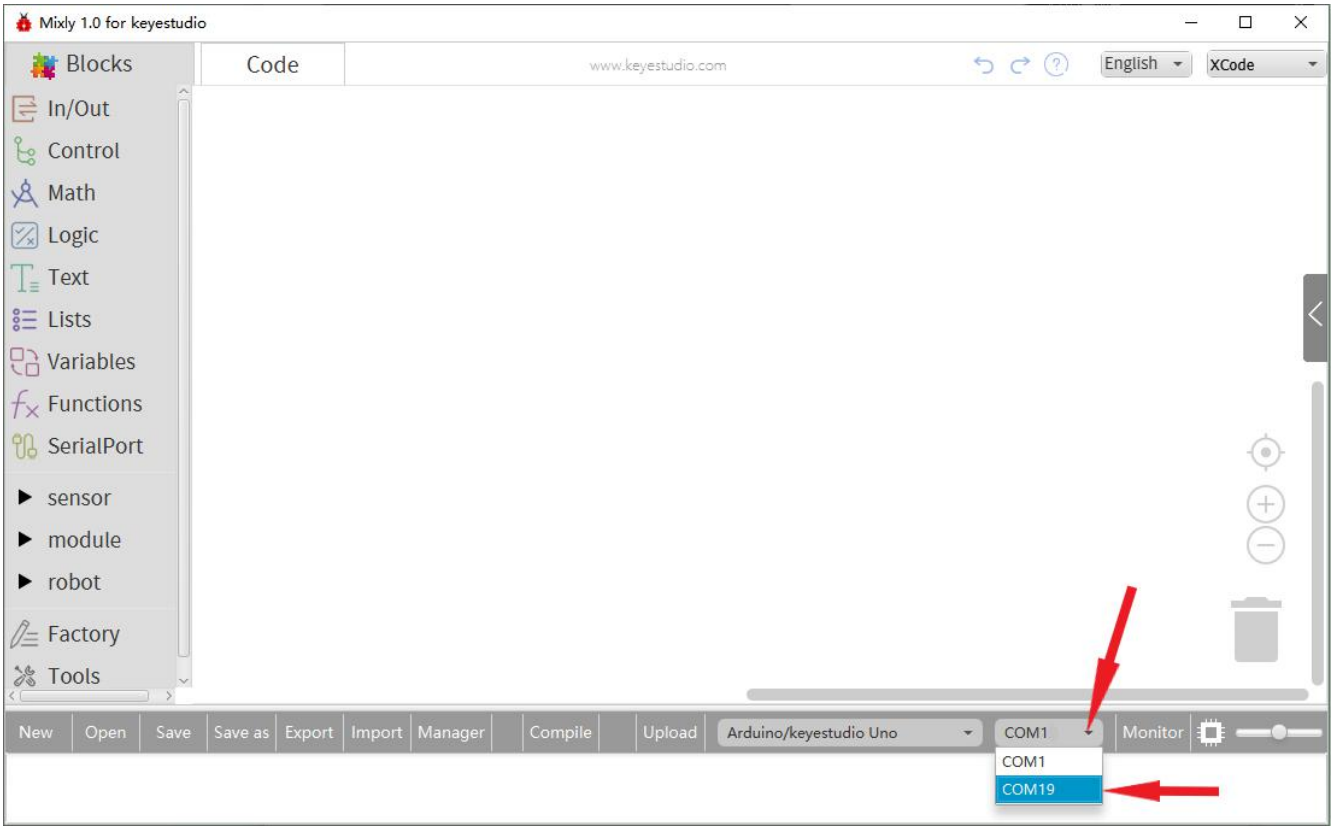
Double-click "Mixly 1.0 for keyestudio.exe", the following interface pops up.



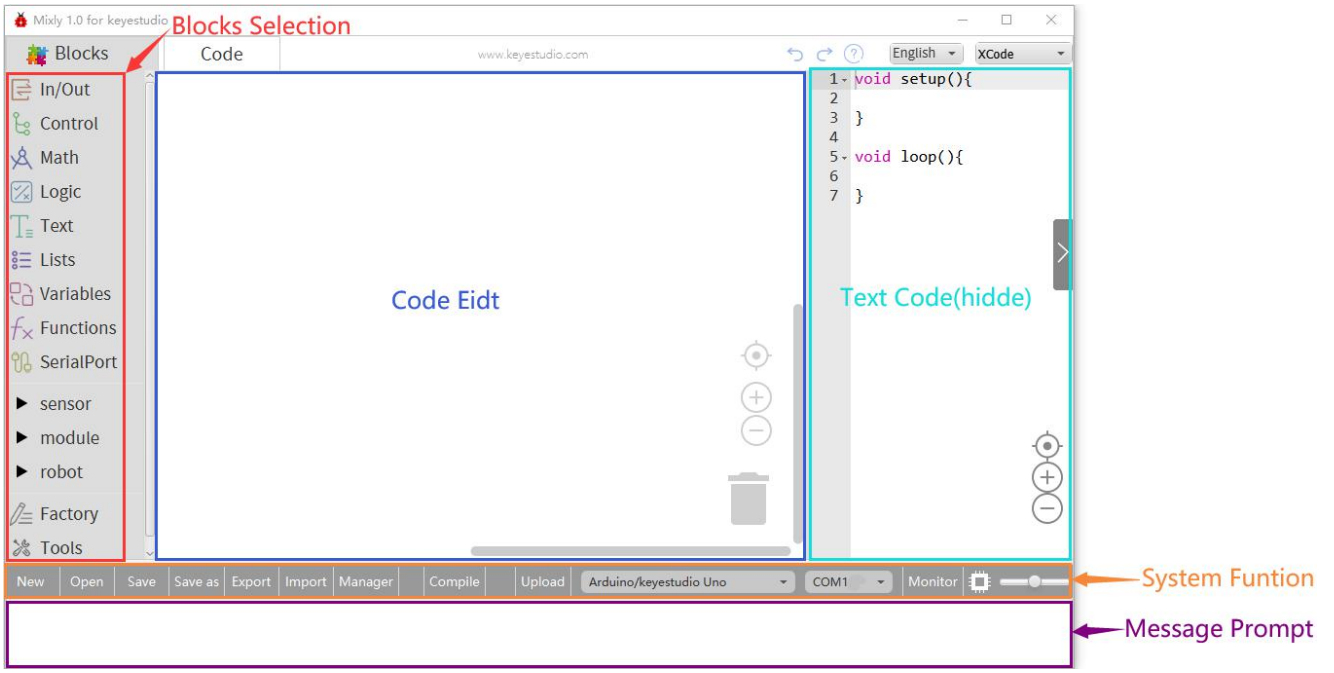
We have to choose correct Arduino development board and name, as shown below:

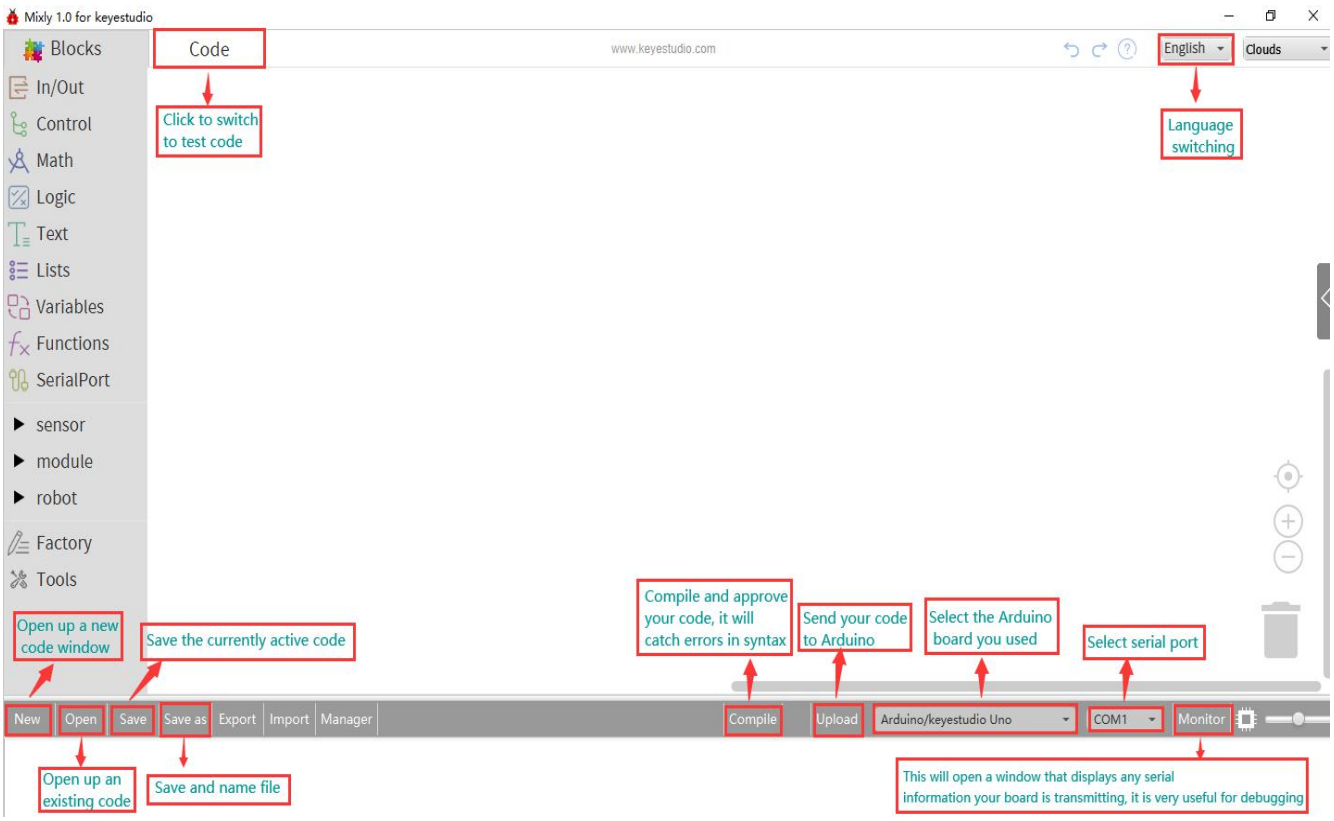


Select correct COM port(the corresponding port will be shown after installing driver successfully)



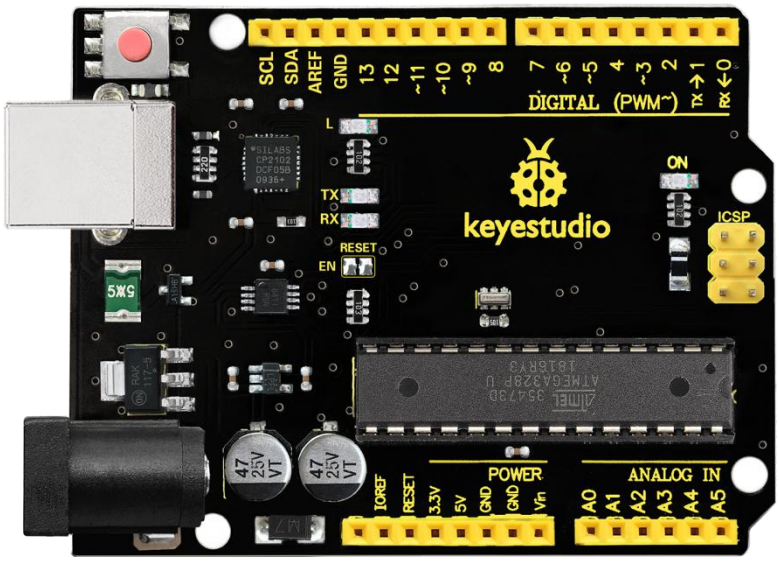
You have to know the function of every area and interface on Mixly software before uploading program on Arduino development board



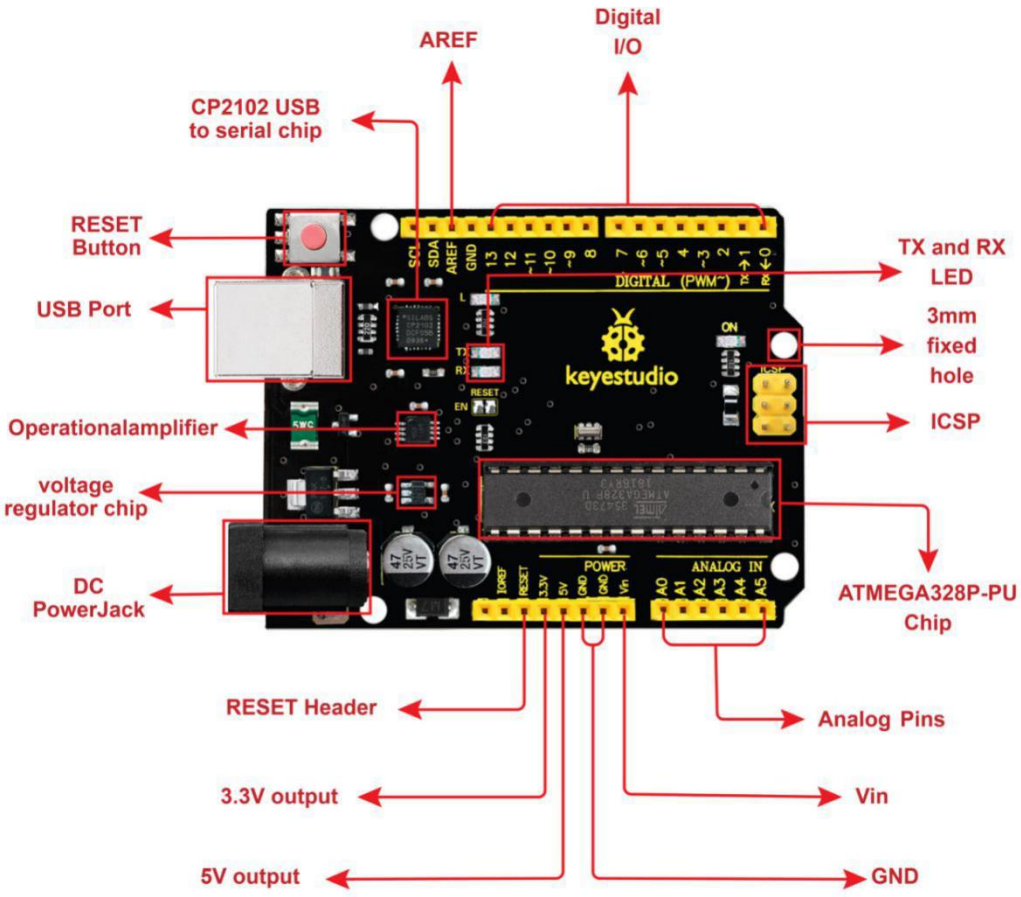


(2) Keyestudio V4.0 Development Board

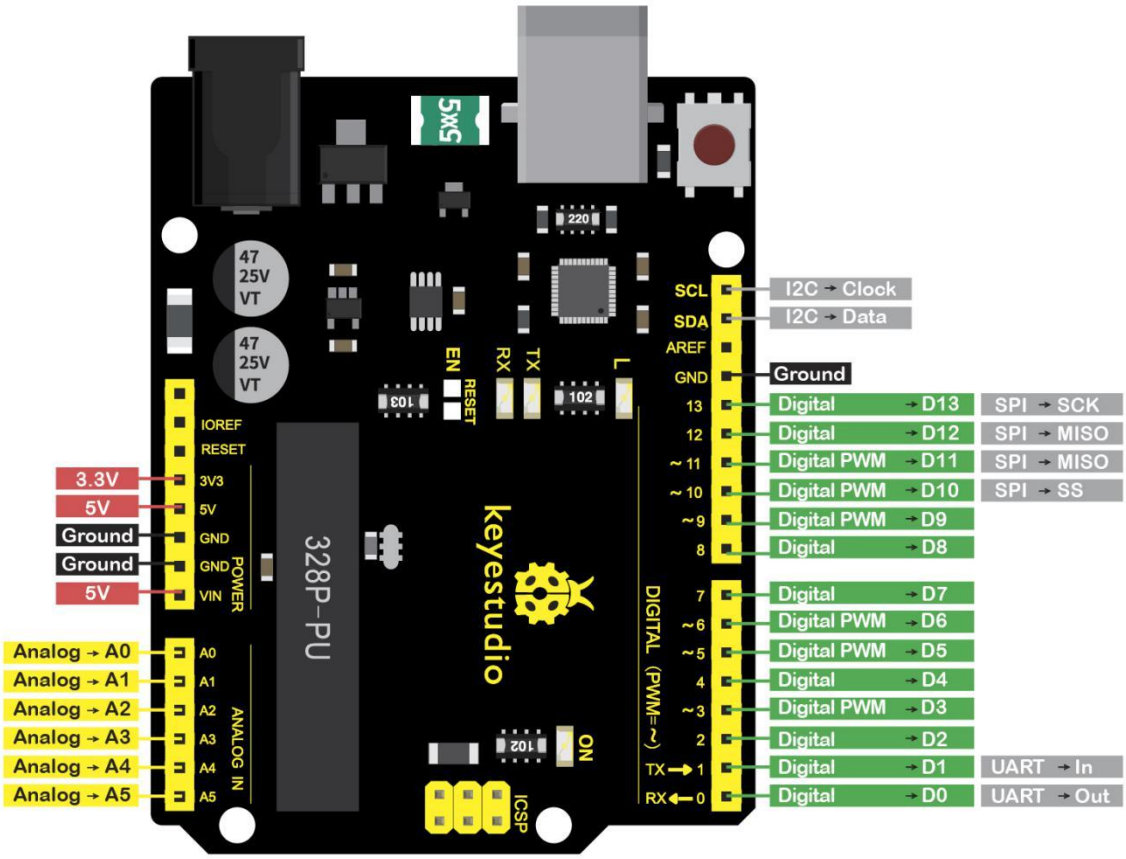
We need to know keyestudio V4.0 development board, as a core of this smart car.



Keyestudio V4.0 development board is an Arduino uno -compatible boar d, which is based on ATmega328P MCU, and with a cp2102 Chip as a UA RT-to-USB converter.



It has 14 digital input/output pins (of which 6 can be used as PWM outp uts), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a pow er jack, 2 ICSP headers and a reset button.



It contains everything needed to support the microcontroller; simply con nect it to a computer with a USB cable or power it via an external DC po wer jack (DC 7-12V) or via female headers Vin/ GND(DC 7-12V) to get started.

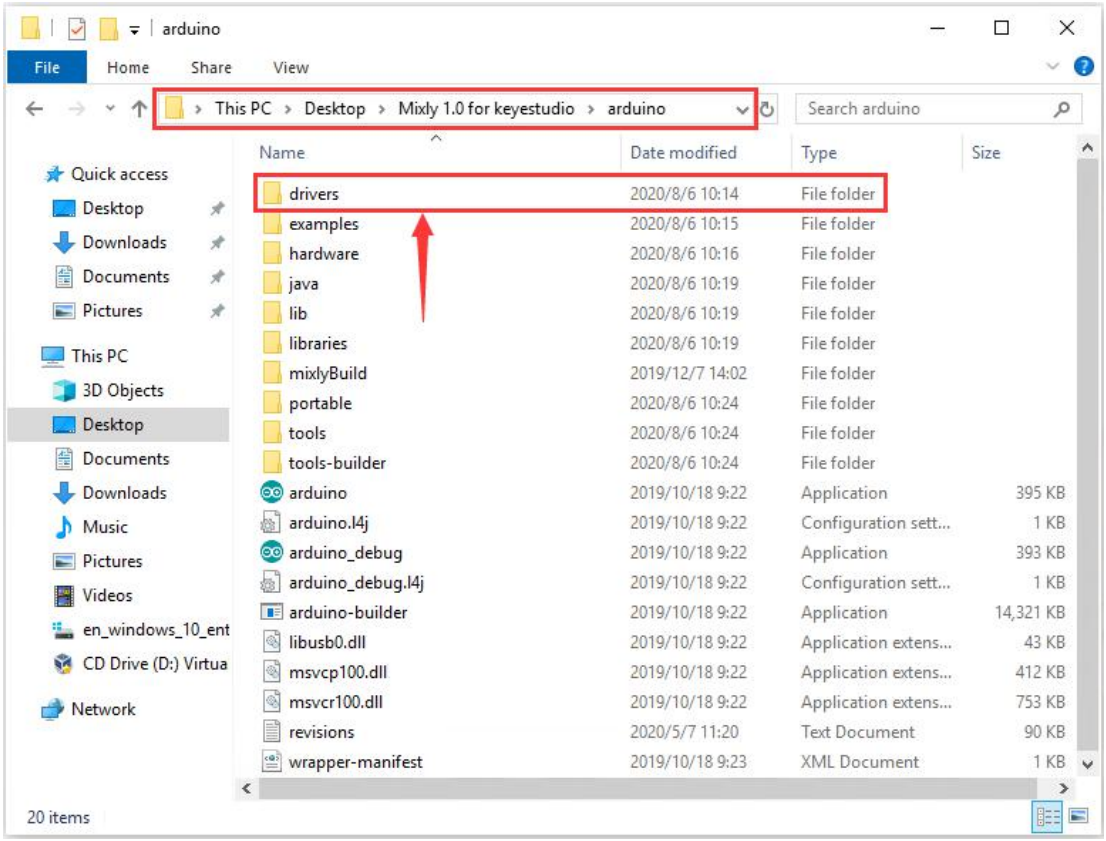
Microcontroller	ATmega328P-PU
Operating Voltage	5V
Input Voltage	DC7-12V

(recommended)	
Digital I/O Pins	14 (D0-D13) (of which 6 provide PWM output)
PWM Digital I/O Pins	6 (D3, D5, D6, D9, D10, D11)
Analog Input Pins	6 (A0-A5)
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P-PU) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P-PU)
EEPROM	1 KB (ATmega328P-PU)
Clock Speed	16 MHz
LED_BUILTIN	D13

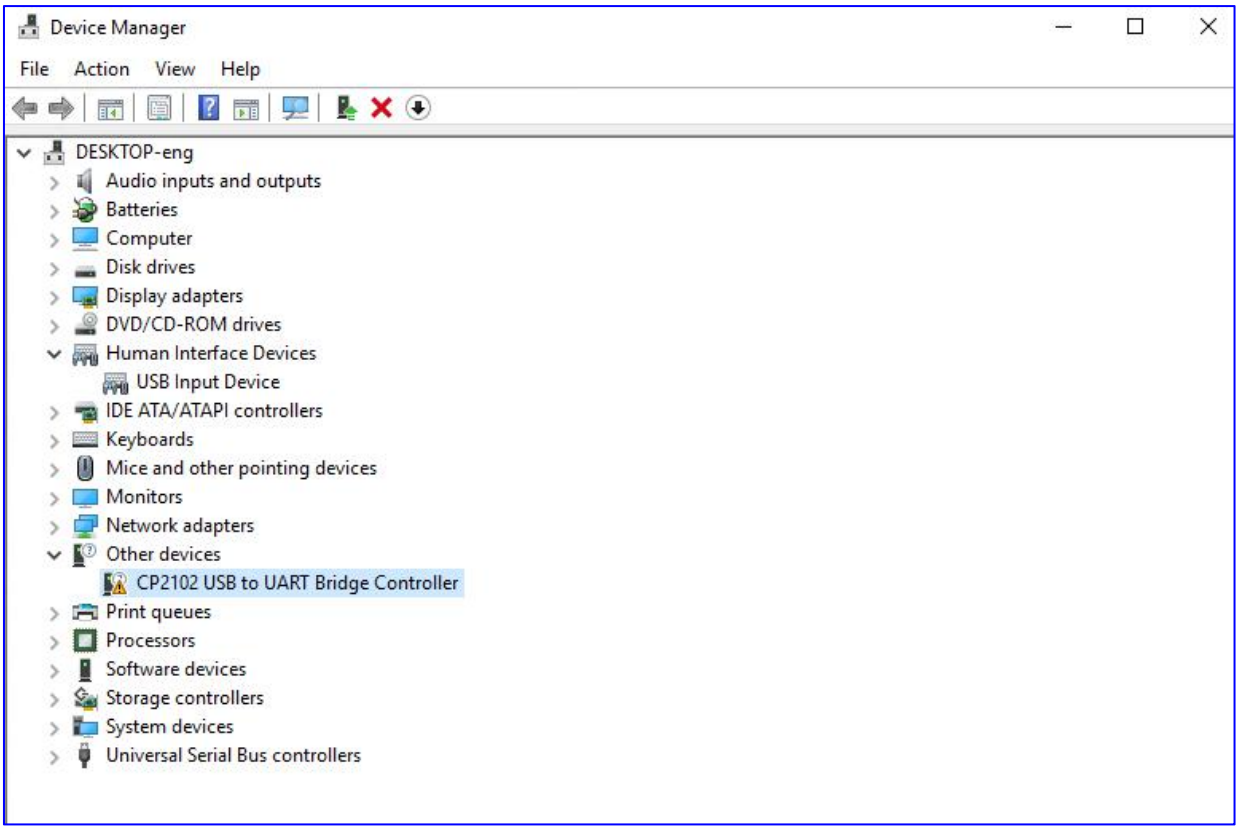
(3) Installing V4.0 board Driver

Let’s install the driver of keyestudio V4.0 board. The USB-TTL chip on V4.0 board adopts CP2102 serial chip. The driver program of this chip is

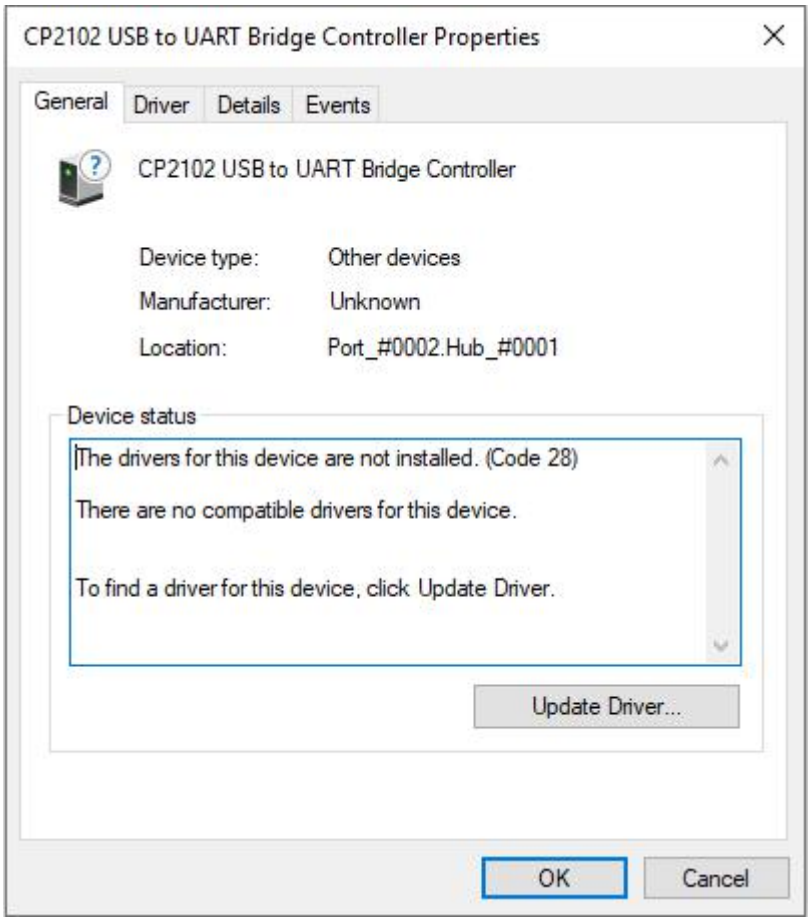
included in Arduino 1.8 version and above, which is convenient. Plug on USB port of board, the computer can recognize the hardware and automatically install the driver of CP2102.



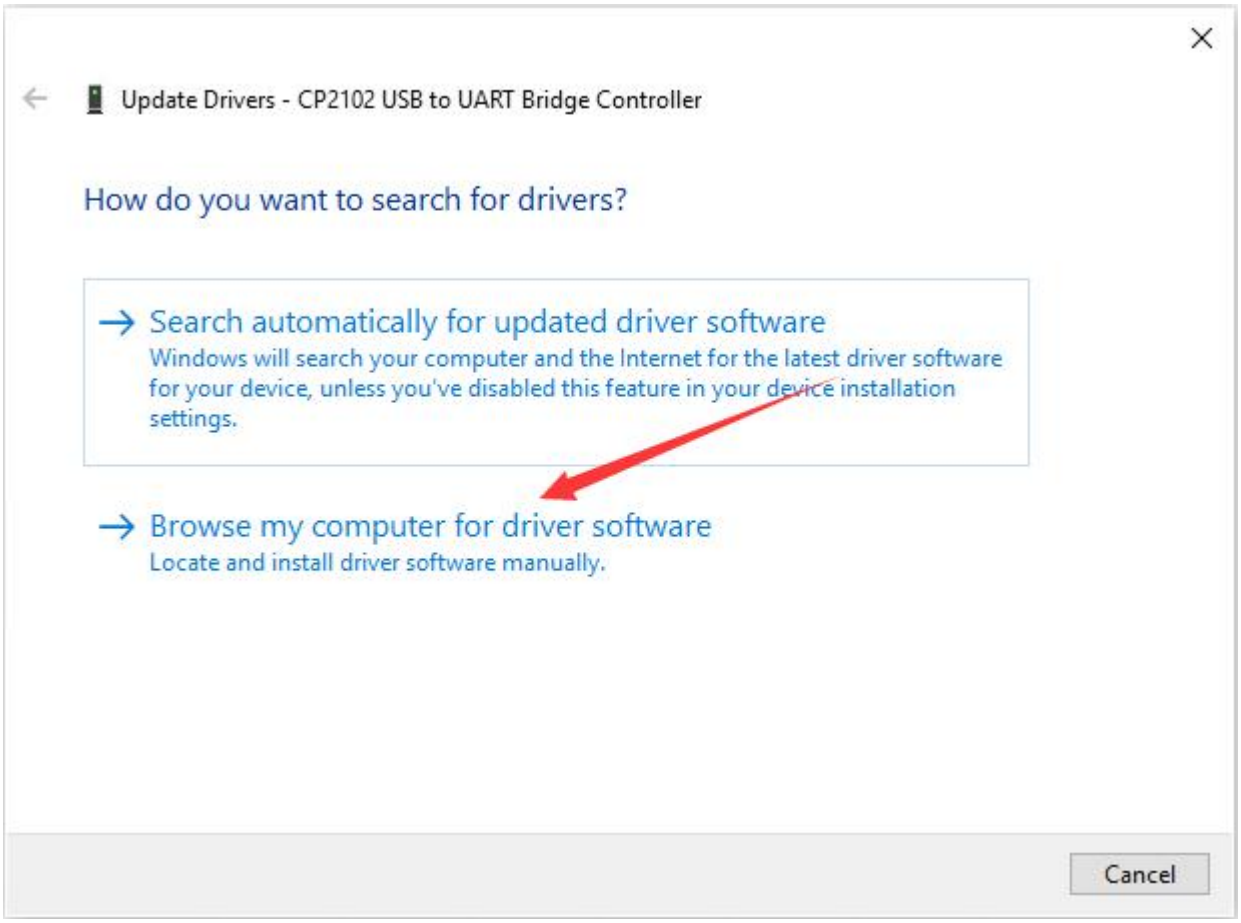
If install unsuccessfully, or you intend to install manually, open the device manager of computer. Right click Computer----- Properties----- Device Manager



There is a yellow exclamation mark on the page, which implies installing unsuccessfully. Then we double click the hardware and update the driver.

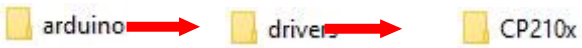


Click “OK” to enter the following page, click “browse my computer for updated driver software”, find out the installed or downloaded ARDUINO software. As shown below:

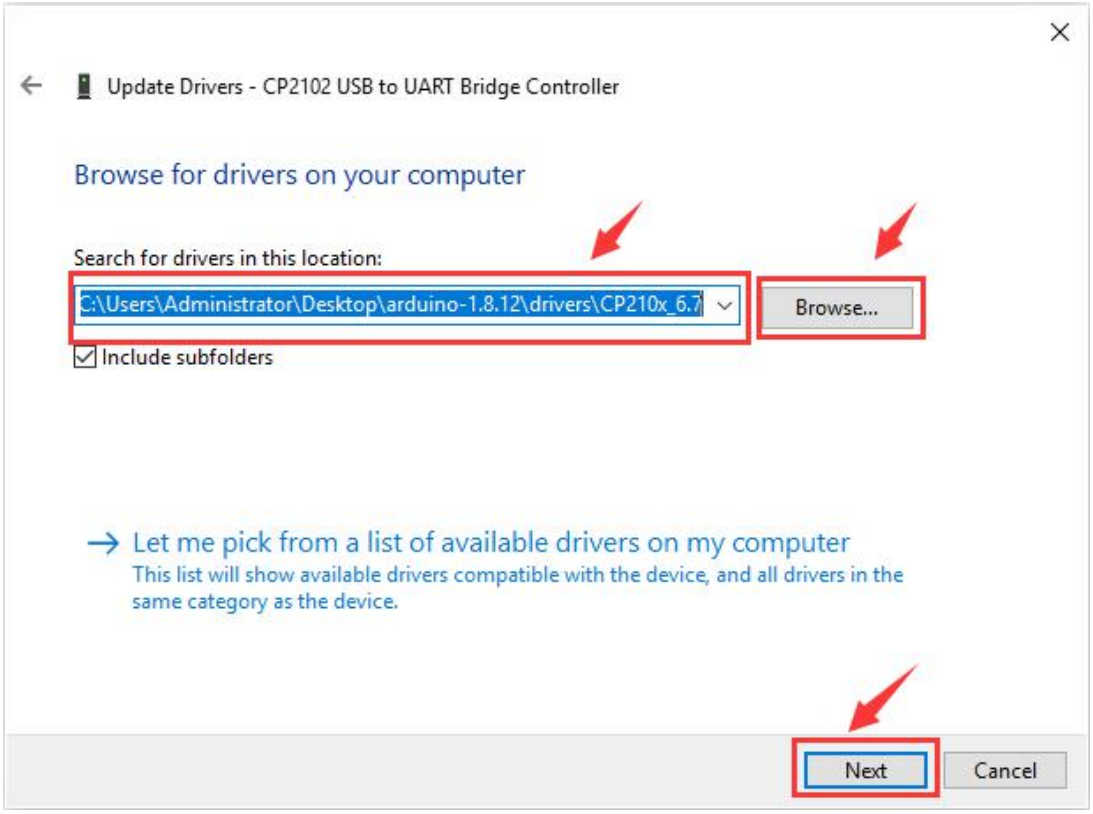


There is a DRIVERS folder in Arduino software installed package

(arduino-1.8.12) , open driver folder and you can see the driver of CP210X series chips.

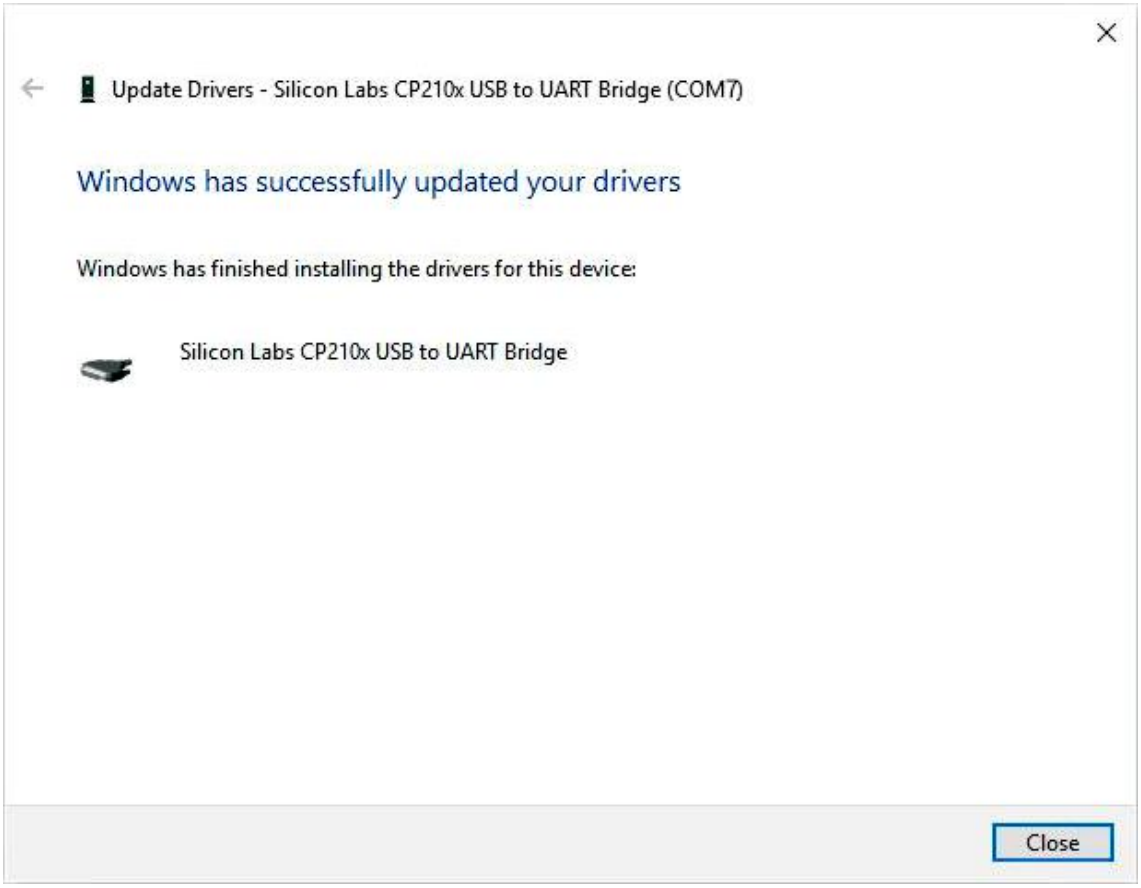


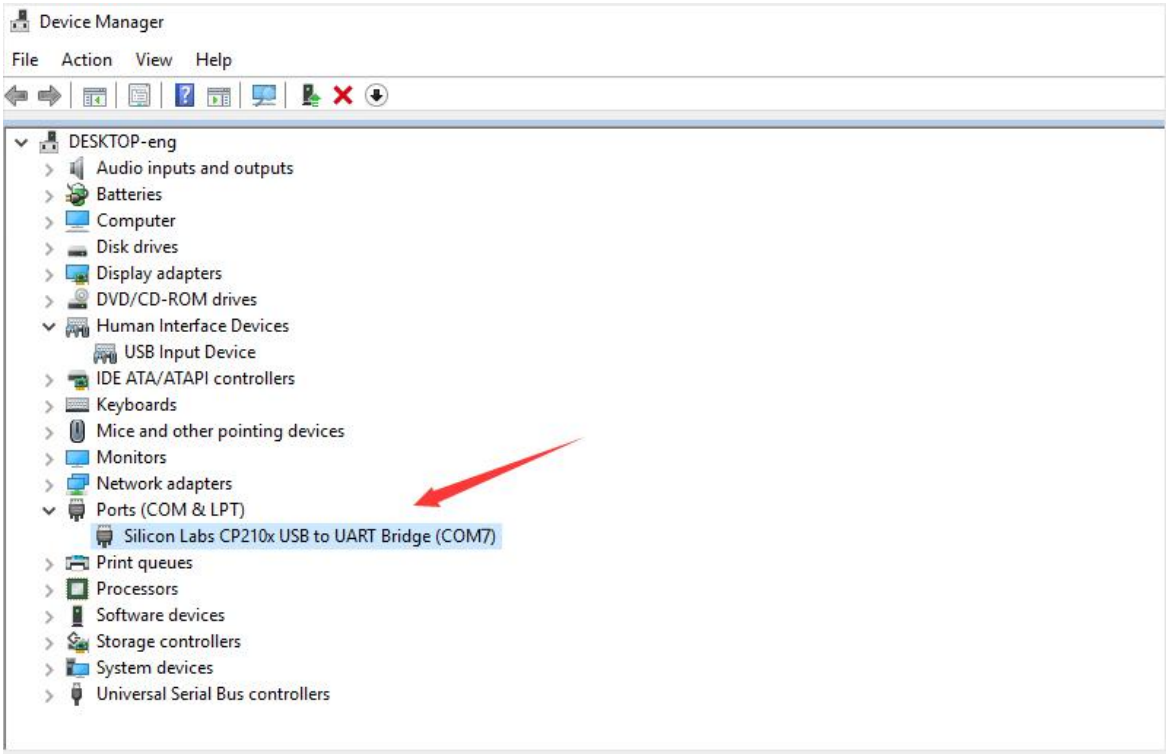
We click “Browse”, then find out the driver folder, or you could enter “driver” to search in rectangular box, then click “next”, the driver will be installed successfully. (I place Arduino software folder on the desktop, you could follow my way)



Open device manager, we will find the yellow exclamation mark disappear.

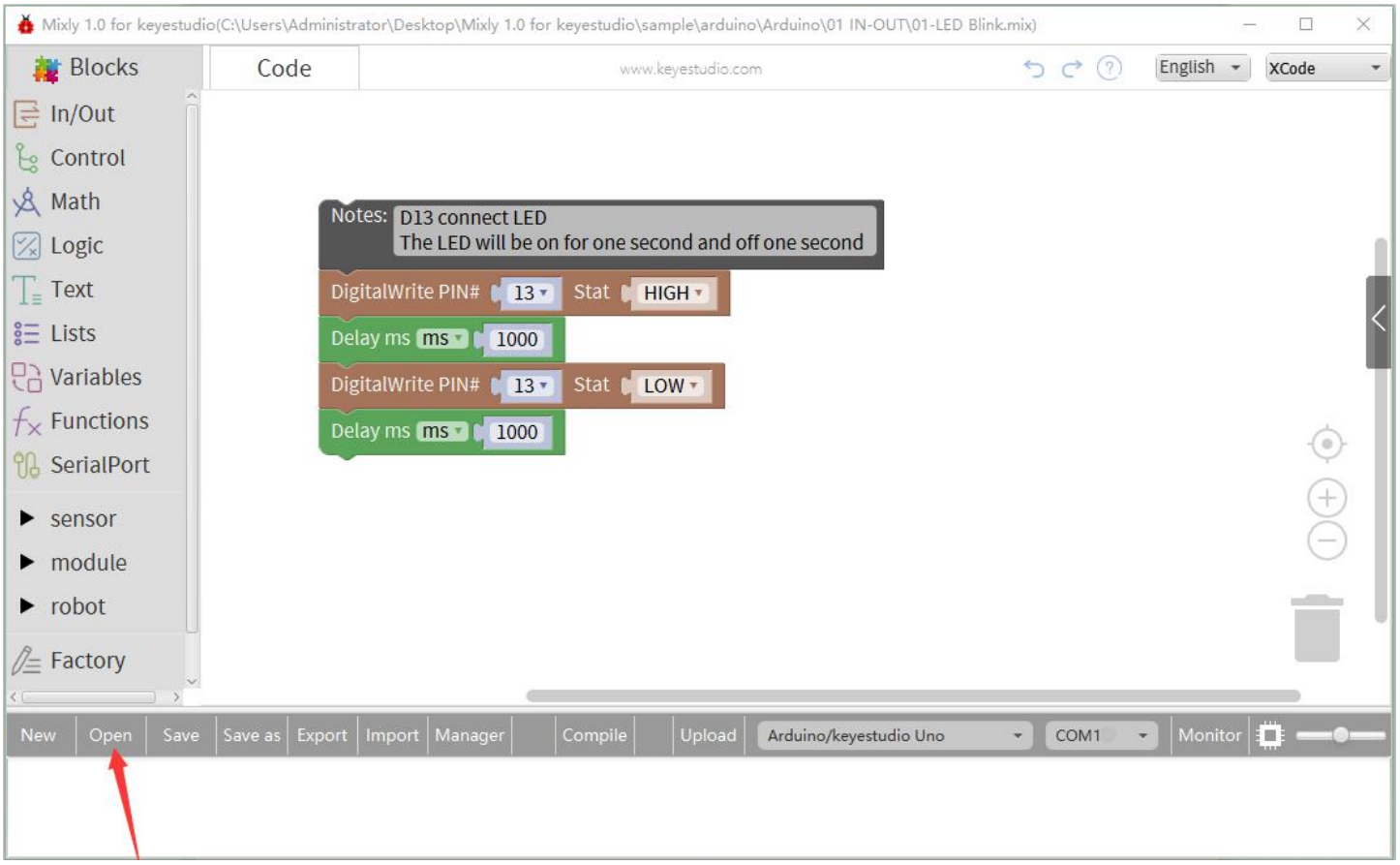
The driver of CP2102 is installed successfully.


Update Drivers - Silicon Labs CP210x USB to UART Bridge (COM7)
Windows has successfully updated your drivers
Windows has finished installing the drivers for this device:
Silicon Labs CP210x USB to UART Bridge
Close

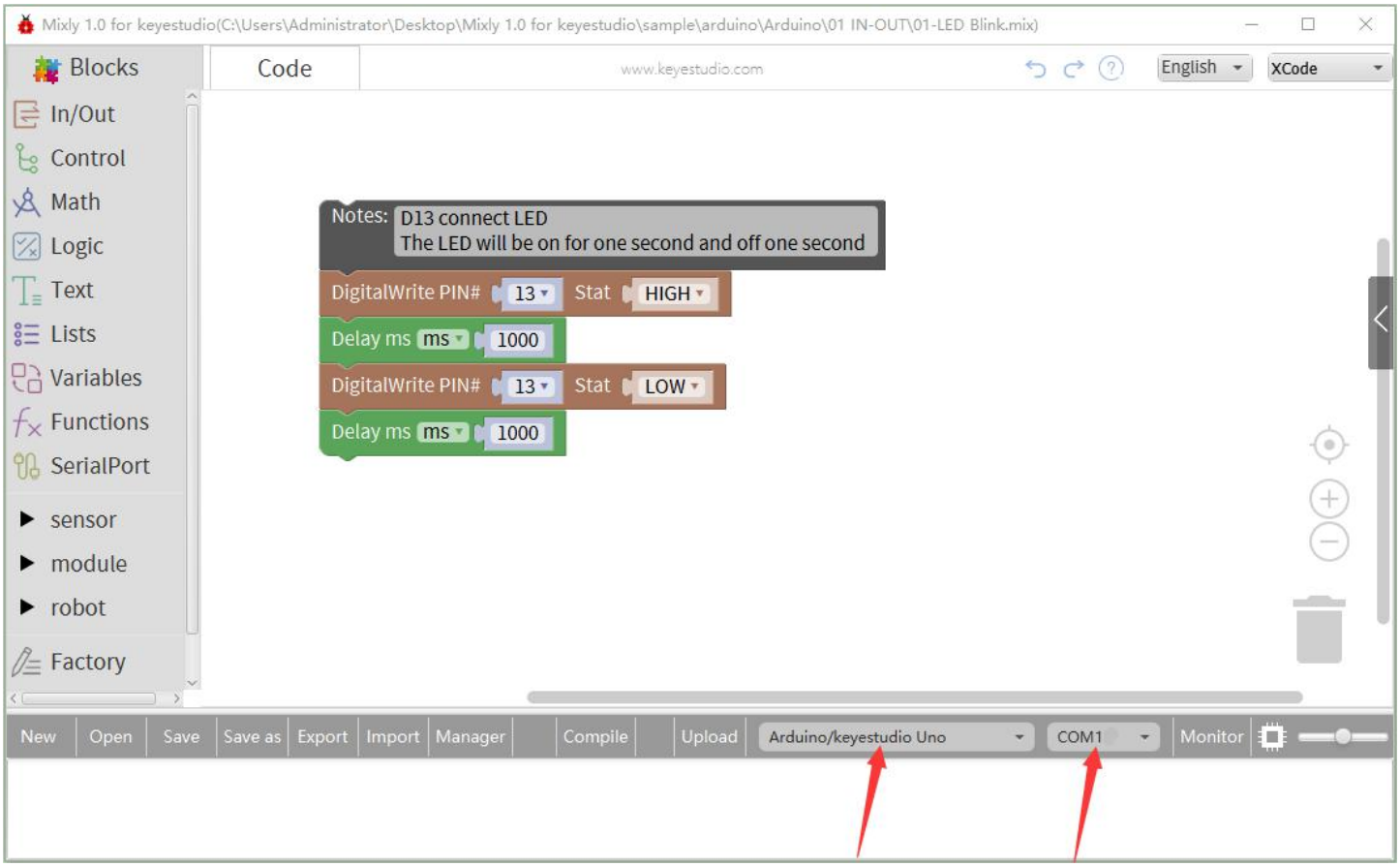

Device Manager
File Action View Help
DESKTOP-eng
Audio inputs and outputs
Batteries
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Human Interface Devices
USB Input Device
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Silicon Labs CP210x USB to UART Bridge (COM7)
Print queues
Processors
Software devices
Storage controllers
System devices
Universal Serial Bus controllers

(4) Start the first program

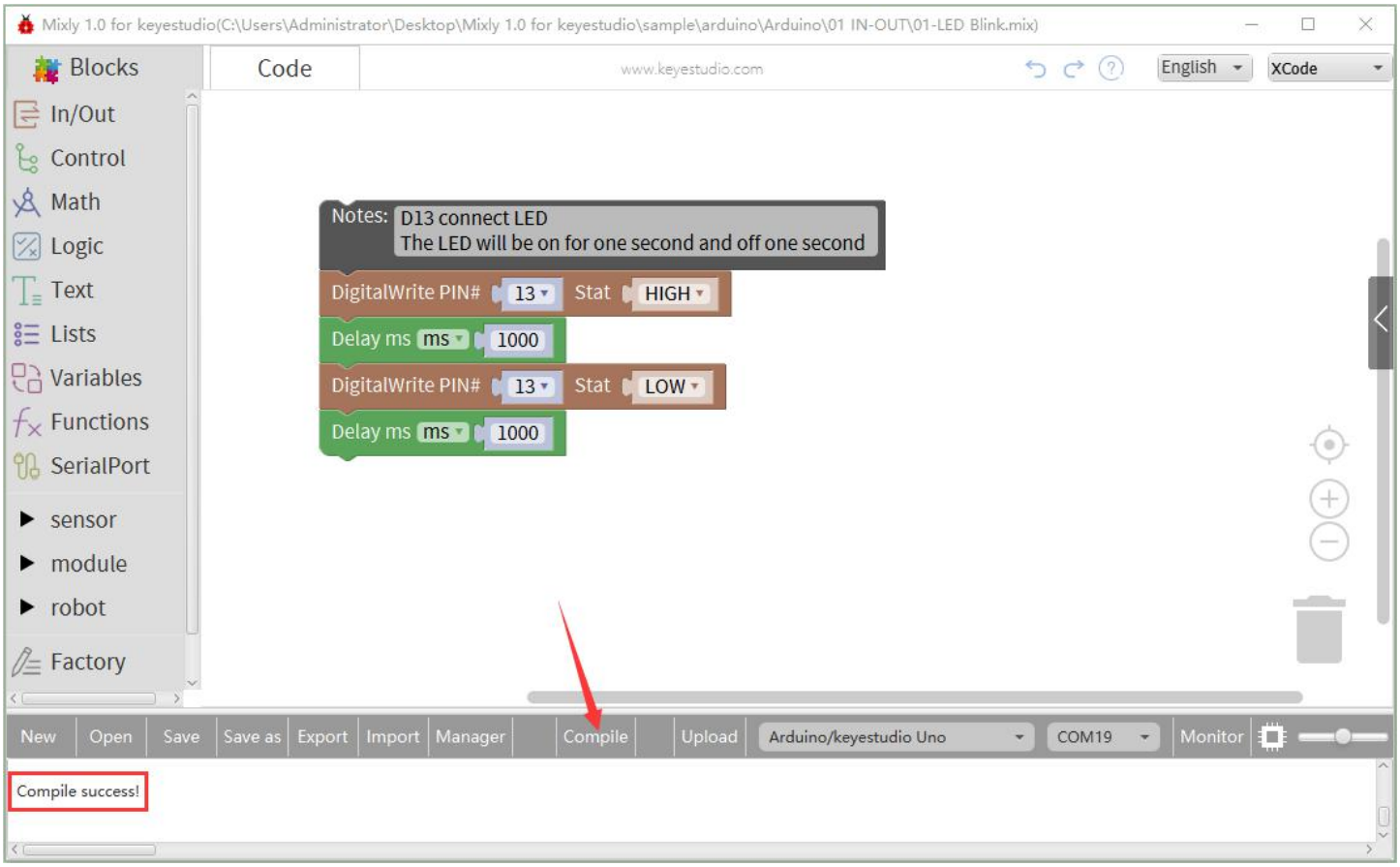
Click “Open”→ sample → arduino→ Arduino → 01 IN-OUT→01-LED Blink.mix



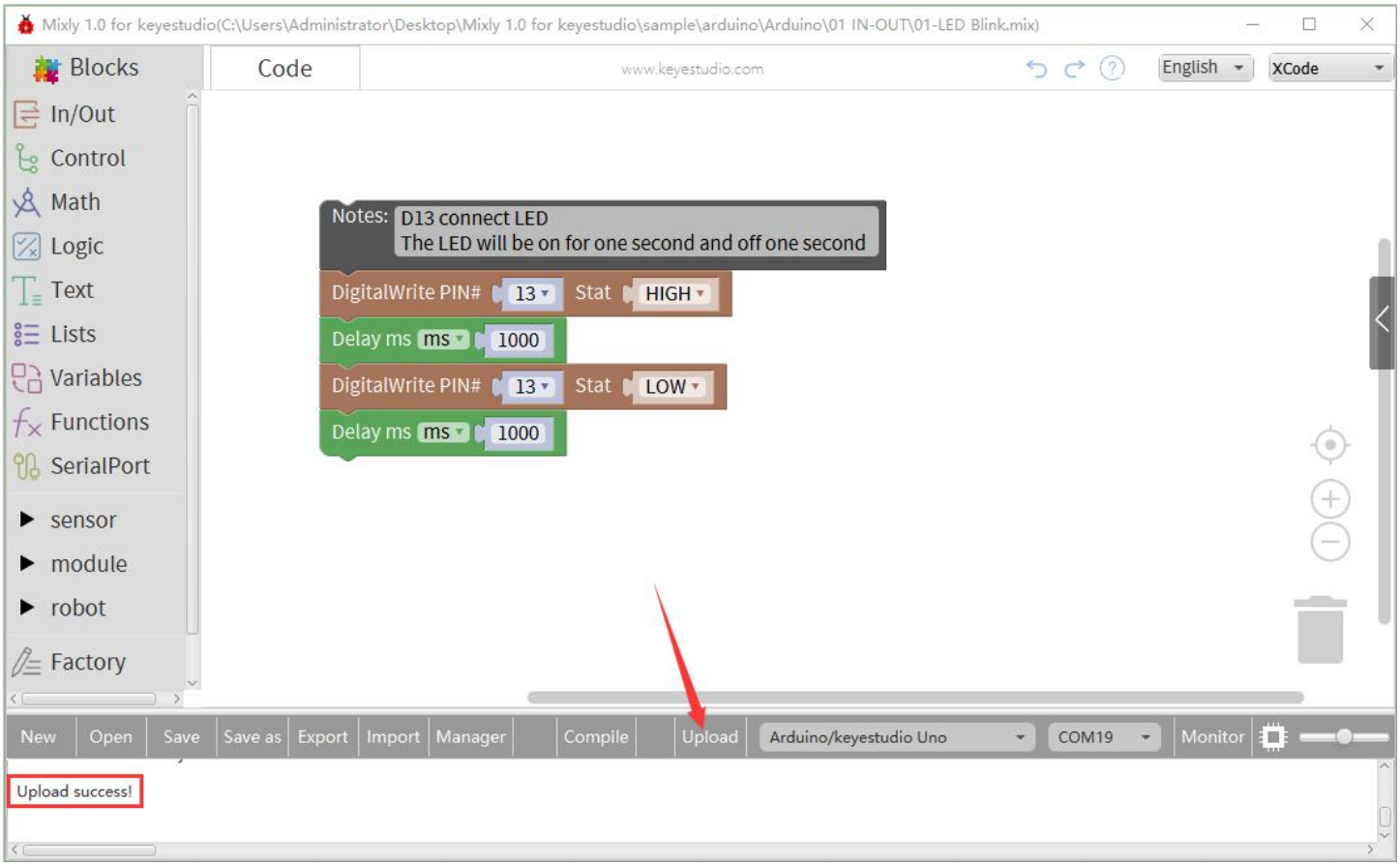
The corresponding board and COM port will be shown after setting board and COM port.



Click Compile to start compiling the program, check errors.



Click Upload to upload the program, upload successfully.



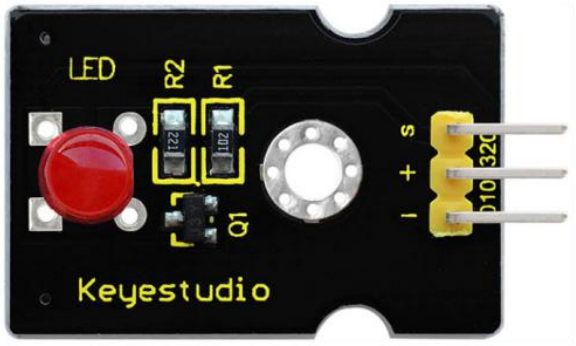
After uploading the program successfully, the onboard LED lights up for 1s, lights off for 1s. Congratulation, you finish the first program.

7. Projects :

The whole project begins with basic program. Starting from simple to complex, the lessons will guide you to assemble robot car and absorb the knowledge of electronic and machinery step by step. I reckon that you could hardly sit still and itch to have a go, let's get started.

Note: (G), marked on each sensor and module, implies negative pole, which is connected to "G", "-"or "GND"on the sensor shield and control board ; (V) represents positive pole, which is linked with V , VCC, + or 5V on the sensor shield and control board.

Project 1: LED Blink



(1) Description

For the starter and enthusiast, this is a fundamental program---LED Blink. LED, the abbreviation of light emitting diodes, consist of Ga, As, P, N chemical compound and so on. The LED can flash diverse color by altering

the delay time in the test code. When in control, power on GND and VCC, the LED will be on if S end is high level; nevertheless, it will go off.

(2) Specification

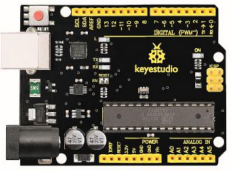
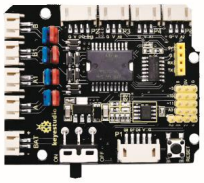
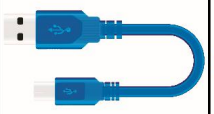
Control interface: digital port

Working voltage: DC 3.3-5V

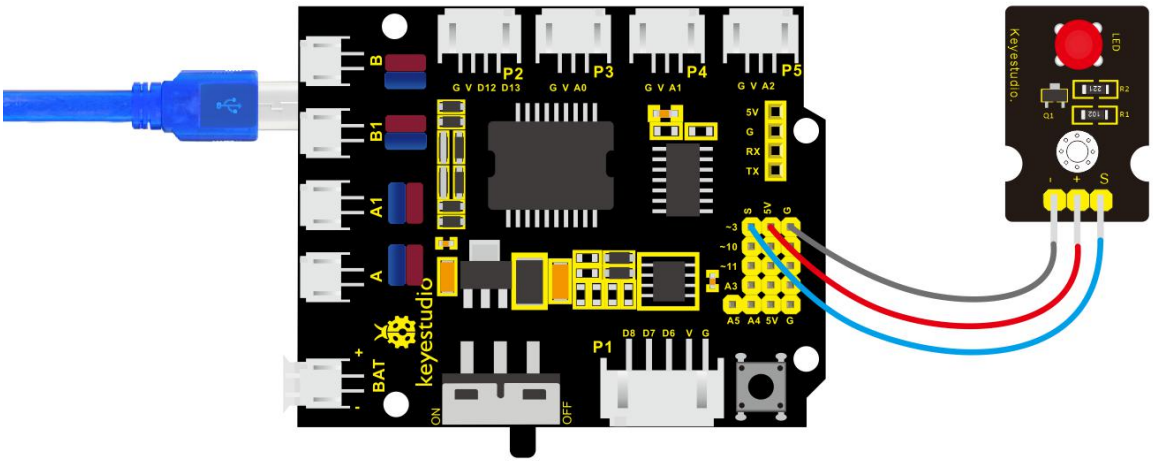
Pin spacing: 2.54mm

LED display color: red

(3) **Components**

Control Board *1	L298P Motor Shield*1	LED module *1	USB Cable *1	3pin Dupont line *1
				

(4) Wiring Diagram



The expansion board is stacked on development board, - of LED module is

connected to G of shield, “+”is linked with 5V, S end is attached to D3.

(5) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_1_LED Blink	lesson_1.1_LED Blink

You could edit code step by step as follows:

(1) Enter “Control” to get block 

(2) Click “Sensor” → “ControlOutput”→ 

,

and combine it with block 

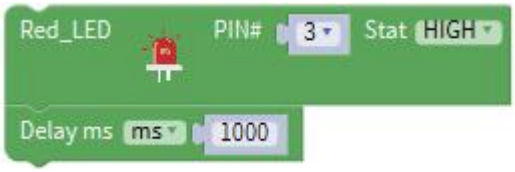
,

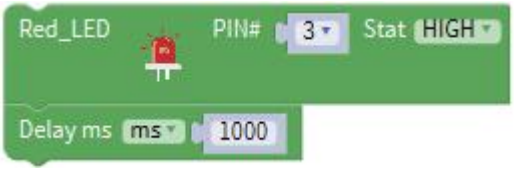
(3) S end of red LED is connected D3 of sensor shield, click the drop-down triangle button to set PIN 3 and LOW.

(4) Copy block 

once, and set to HIGH

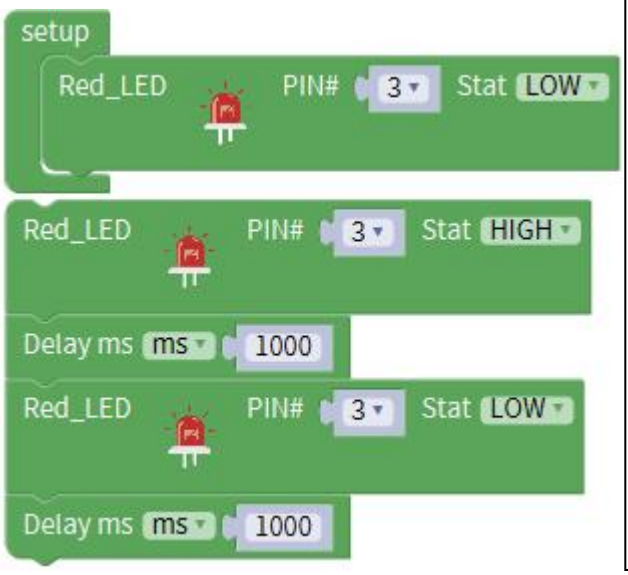
(5) Click "Control" to drag out block 



(6) Duplicate code string 

once, and set to LOW.

Complete Program :



Initialization

Turn off red LED connected to D3

Light up red LED connected to D3

Delay in 1000ms

Turn off red LED connected to D3

Delay in 1000ms

(6) Test Result :

Upload the program, LED flickers with the interval of 1s.

(7) Extension Practice :

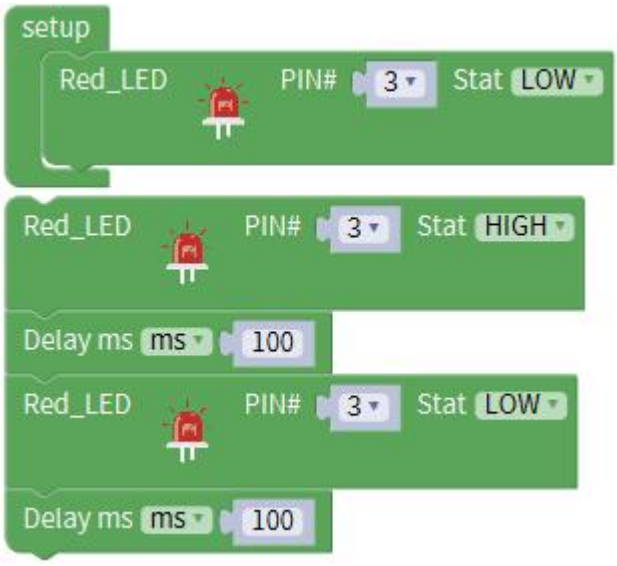
We succeed to blink LED. Next, let's observe what LED will change if we modify pins and delay time.

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_2_LED blink	lesson_1.2_Extension Practice

You could edit code step by step as follows:

Complete Program :



Initialization

Turn off red LED connected to D3

Light up red LED connected to D3

Delay in 100ms

Turn off red LED connected to D3

Delay in 100ms

The LED flickers faster through the test result, therefore, pins and delay time affect flash frequency.

Project 2: Adjust LED Brightness

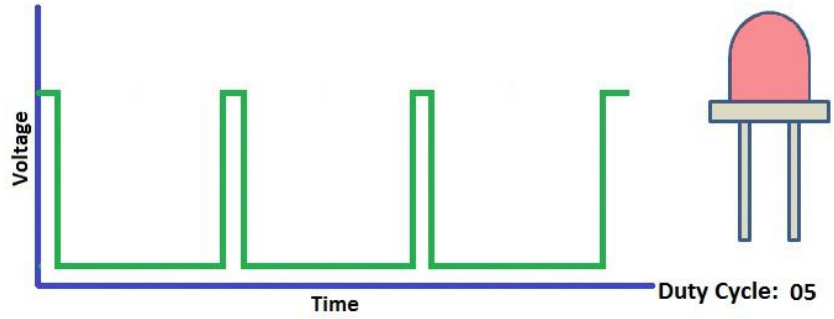
(1) Description

In previous lesson, we control LED on and off and make it blink.

In this project, we will control LED brightness through PWM to simulate breathing effect. Similarly, you can change the step length and delay time

in the code so as to demonstrate different breathing effect.

PWM is a means of controlling the analog output via digital means. Digital control is used to generate square waves with different duty cycles (a signal that constantly switches between high and low levels) to control the analog output.In general, the input voltage of port are 0V and 5V. What if the 3V is required? Or what if switch among 1V, 3V and 3.5V? We can't change resistor constantly. For this situation, we need to control by PWM.

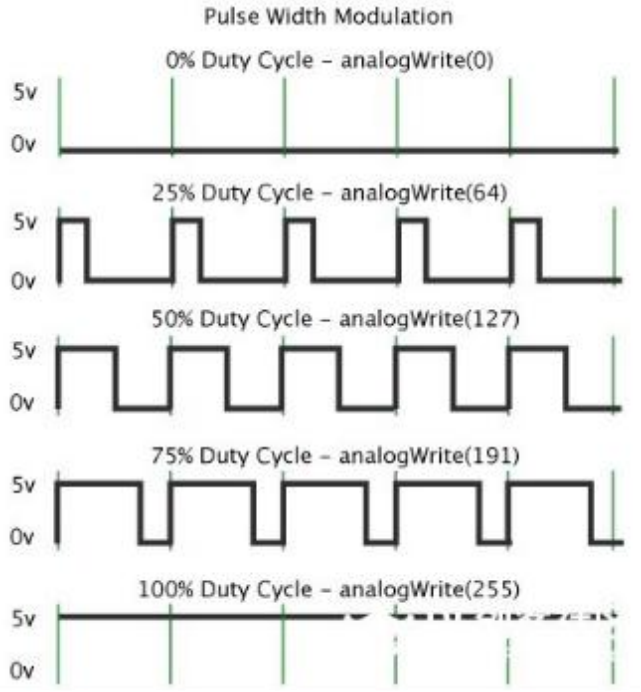


For the Arduino digital port voltage output, there are only LOW and HIGH, which correspond to the voltage output of 0V and 5V. You can define LOW as 0 and HIGH as 1, and let the Arduino output five hundred 0 or 1 signals within 1 second.

If output five hundred 1, that is 5V; if all of which is 1, that is 0V. If output 010101010101 in this way then the output port is 2.5V, which is like showing movie. The movie we watch are not completely continuous. It

actually outputs 25 pictures per second. In this case, the human can't tell it, neither does PWM. If want different voltage, need to control the ratio of 0 and 1. The more 0,1 signals output per unit time, the more accurately control.

PWM is a technology to obtain analog quantity through digital method. Digital control forms a square wave, and the square wave signal only has two states of turning on and off (that is, high or low levels). By controlling the ratio of the duration of turning on and off, a voltage varying from 0 to 5V can be simulated. The time turning on(academically referred to as high level) is called pulse width, so PWM is also called pulse width modulation. Through the following five square waves, let' s acknowledge more about PWM.

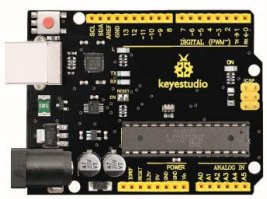
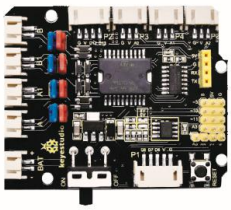
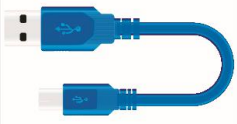


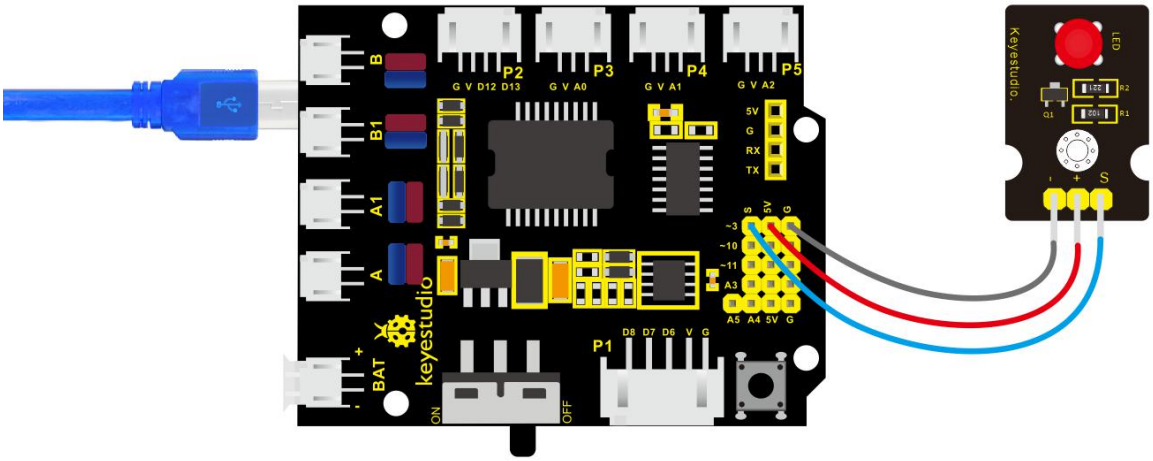
In the above figure, the green line represents a period, and value of analogWrite() corresponds to a percentage which is called Duty Cycle as well. Duty cycle implies that high-level duration is divided by low-level duration in a cycle. From top to bottom, the duty cycle of first square wave is 0% and its corresponding value is 0. The LED brightness is lowest, that is, turn off. The more time high level lasts, the brighter the LED. Therefore, the last duty cycle is 100%, which correspond to 255, LED is brightest. 25% means darker.

PWM mostly is used for adjusting the LED brightness or rotation speed of motor.

It plays vital role in controlling smart robot car. I believe that you can’ t wait to enter next project.

(2) Components

Control Board *1	L298P Motor Shield*1	LED module *1	USB Cable *1	3pin Dupont line *1
				



(3) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_2_Adjust LED Brightness	lesson_2.1_Adjust LED Brightness

You could edit code step by step as follows:

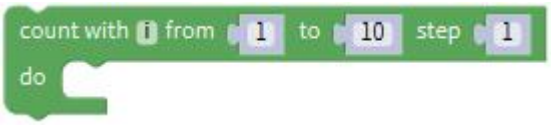
(1) Click "Control" to get block 

(2) Go to " sensor " → " ControlOutput " → 

→

,

(3) Red LED is connected to D3, so set to PIN 3 and LOW

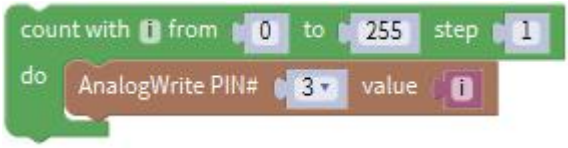
(4) Enter“Control”to get block 

, set block:

(5) Enter “In/Out” to get block 

,

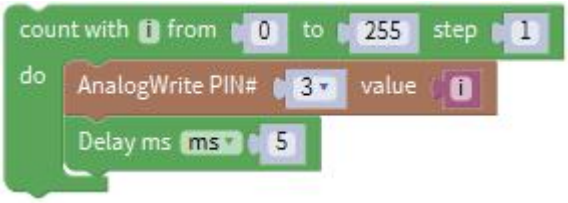
(6) Click “Variables” to move block 
into 0 box behind value.

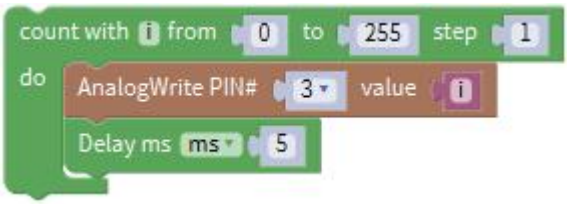


(7) Click ”Control ” to move block 

into block 

, delay in 5ms.

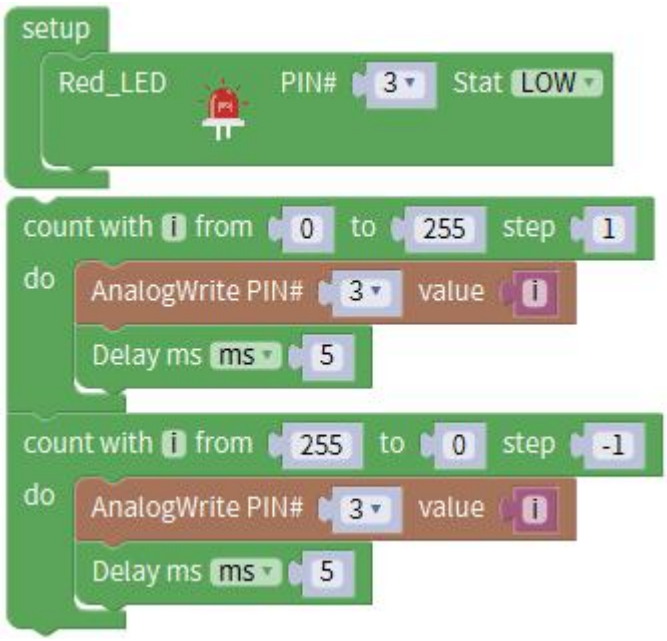


(8) Copy code string 

once , set code

string as follows:

Complete Program :



Initialization

Turn off red LED connected to D3

If i varies from 0 to 255 and increase 1 every time, the next code will be executed.

Analog output pin 3 assigns value to variable i

Delay in 5ms

If i varies from 255 to 0 and increase -1 every time, the next code will be executed.

Analog output pin 3 assigns value to variable i

Delay in 5ms

(4) Test Result :

Upload test code successfully, LED gradually becomes brighter then darker,

like human breath, rather than light on and off immediately.

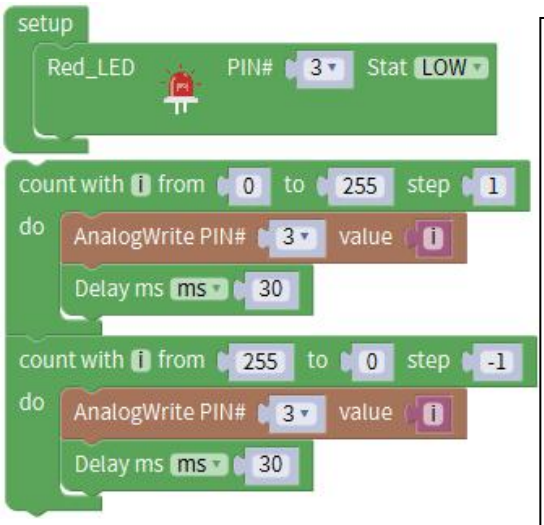
(5) Extension Practice :

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_2_Adjust LED Brightness	lesson_2.2_Extension Practice

You could edit code step by step as follows:

Complete Program :



Initialization
Turn off red LED
If i varies from 0 to 255 and increase 1 every time, the next code will be executed.
Analog output pin 3 assigns value to variable i
Delay in 30ms
If i varies from 255 to 0 and increase -1 every time, the next code will be executed.
Analog output pin 3 assigns value to variable i
Delay in 30ms

Upload code on the development board and the time interval of LED getting dark is longer.

Project 3 : The working Principle of Line Tracking Sensor

(1) Description :

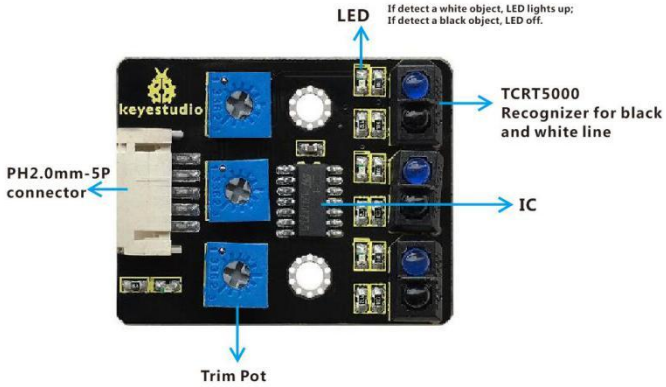
The tracking sensor is actually an infrared sensor. The component used here is the TCRT5000 infrared tube. Its working principle is to use the different reflectivity of infrared light to the color, then convert the strength of the reflected signal into a current signal.

During the process of detection, black is active at HIGH level, but white is active at LOW level. The detection height is 0-3 cm.

Keyestudio 3-channel line tracking module has integrated 3 sets of TCRT5000 infrared tube on a single board, which is more convenient for wiring and control.

By rotating the adjustable potentiometer on the sensor, it can adjust the detection sensitivity of the sensor.

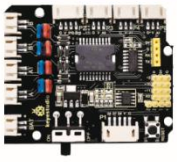
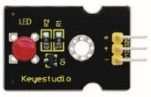
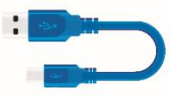
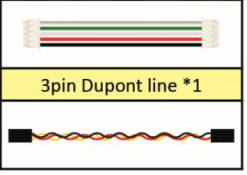
(2) Specification :



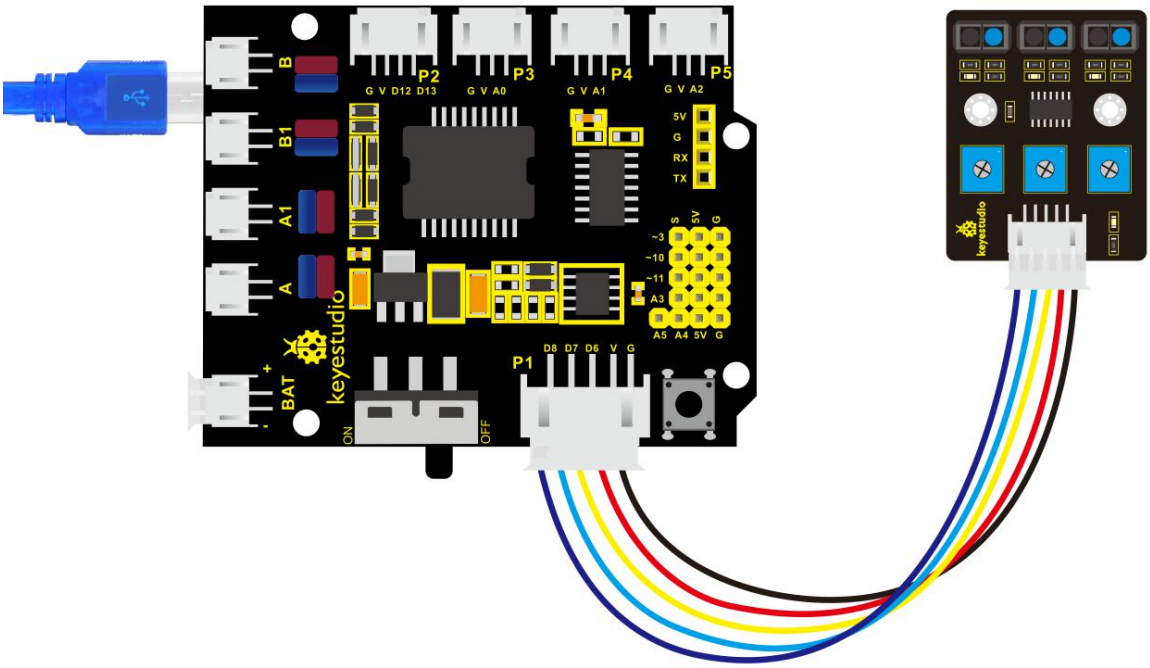
- Operating Voltage: 3.3-5V (DC)
- Interface: 5PIN
- Output Signal: Digital signal
- Detection Height: 0-3 cm

Special note: before testing, turn the potentiometer on the sensor to adjust the detection sensitivity. When adjust the LED at the threshold between ON and OFF, the sensitivity is the best.

(3) Equipment :

Control Board *1	L298P Motor Shield *1	Channel Line Tracking *1	LED module *1	USB Cable *1	5P Double-ended line *1
					 3pin Dupont line *1

(4) Connection Diagram :



(5) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_3_Line Tracking Sensor	lesson_3.1_Line Tracking Sensor

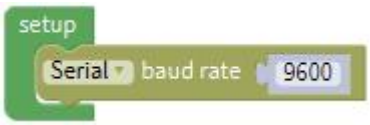
You could edit code step by step as follows:

(1) Click “Control” to get block 

(2) Enter “Serial port” to move block 

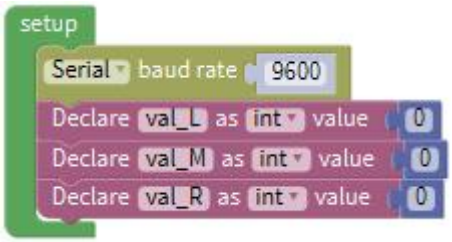
into 

block.



(3) Go to "Variables" to move  into block  for three times ; then enter "Math" to drag block  and copy it for 3 times.

(4) Combine  with  block, and separately set to val_L, val_M and val_R.



(5) Enter "Variables" to move out block  ,

(6) Go to "Sensor" → "DigitalRead" → 

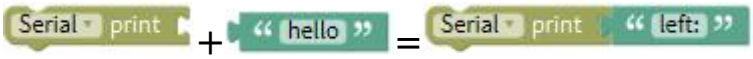
(7) Integrate block  with  block.

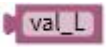
(8) The line tracking sensor is linked with D6, so set to PIN 6.

(9) Replicate block  twice , and separately change val_L into val_M and val_R;

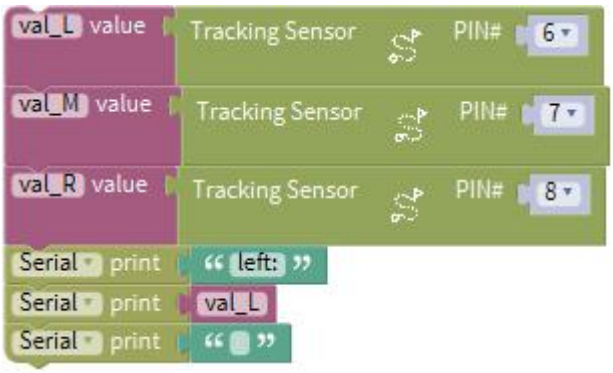
(10) The tracking sensor is also connected to D7 and D8, therefore, set to PIN 7 and PIN 8

(11) Click “Serial Port”to drag out block  , and go to “Text” to move block  into block  ,

(12) Change hello into left : 

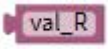
(13) Copy block  again , and enter “ Variables ” to move block  into  .

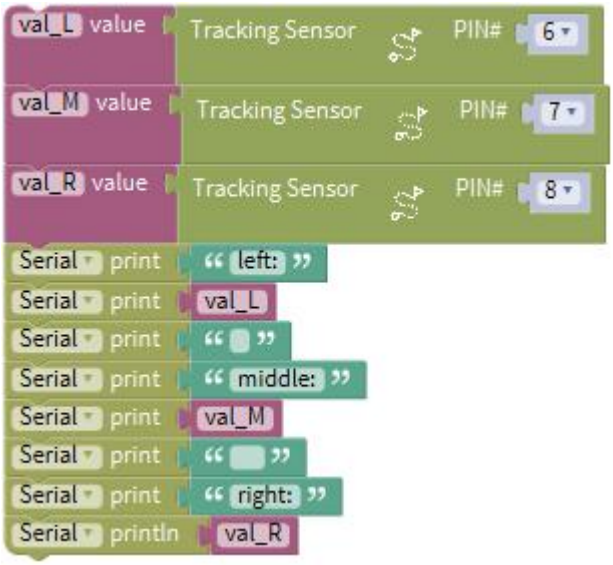
(14) Replicate code  once and delete left : , as shown below:



(15) Duplicate code string  once, and change left : into middle : , val_L into val_M.

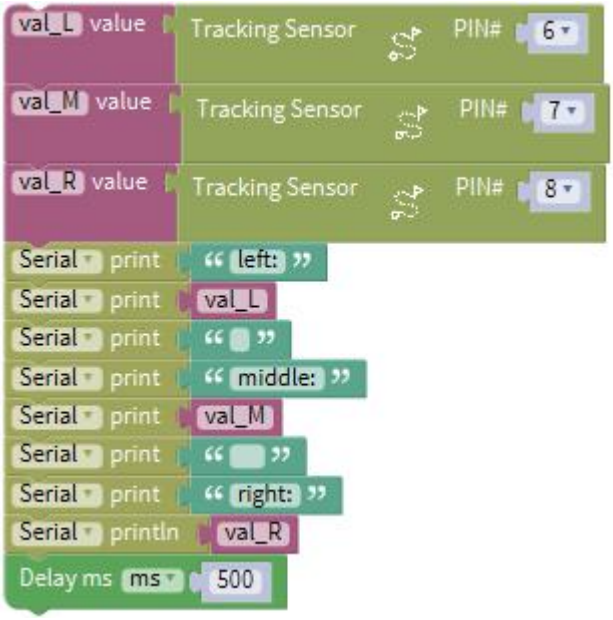
(16) Copy block  , and alter left : into right :

(17) Go to “ Serial Port ” to drag out  block , and enter “Variables” to drag  into block  .

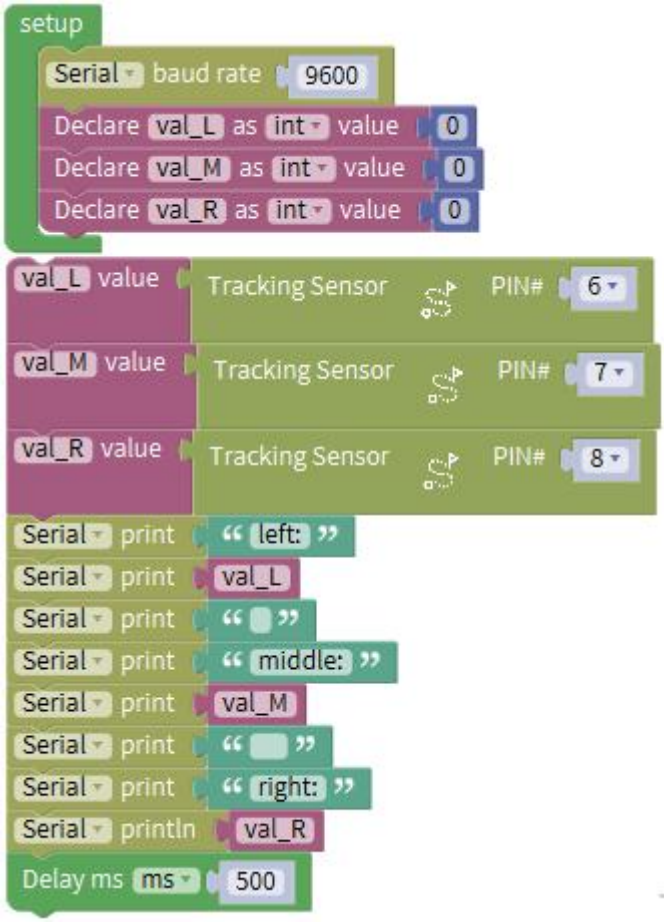


(18) Enter "Control" to get block 

, and delay in 500ms.



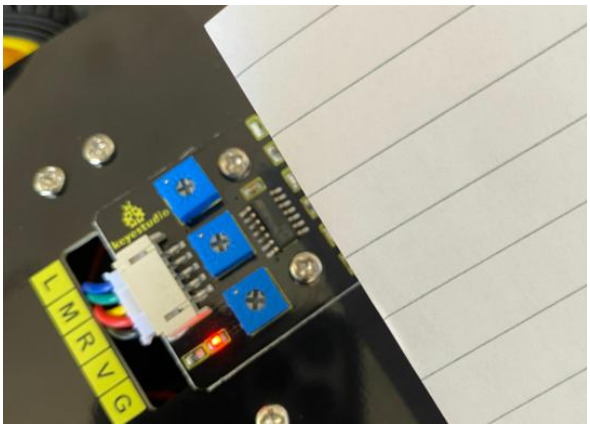
Complete Program :

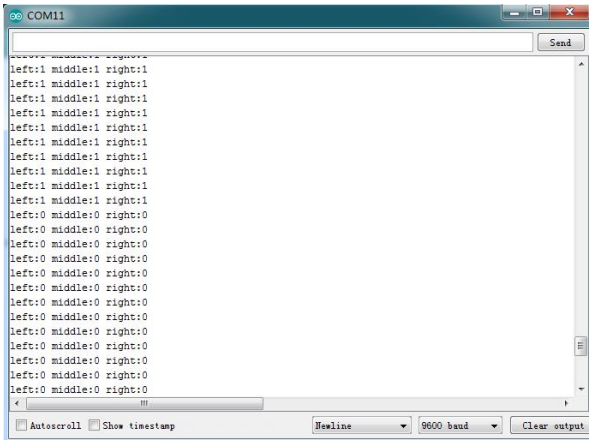


Initialization
Set baud rate to 9600
Set val_L to integer 0
Set val_M to integer 0
Set val_R to integer 0
Set the digital signals read by left TCRT5000 IR tube to val_L
Set the digital signals read by middle TCRT5000 IR tube to val_M
Set the digital signals read by right TCRT5000 IR tube to val_R
Serial port prints left:
Serial port displays the digital signals read by left TCRT5000 IR tube
Serial port prints blank space
Serial port shows middle
Serial port prints the digital signals read by middle TCRT5000 IR tube
Serial port shows blank space
Serial port displays right:
Serial port shows the digital signals read by right TCRT5000 IR tube
Delay in 500ms

(6) Test Result :

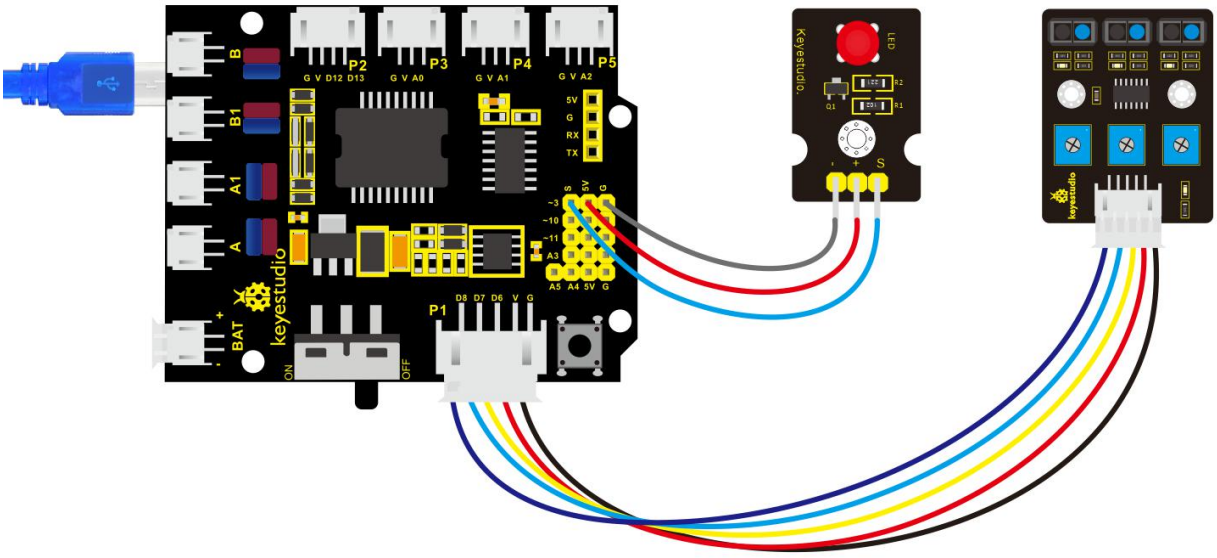
Upload the code on development board, open serial monitor to check line tracking sensors. And the displayed value is 1(high level) when no signals are received. The value becomes int o 0 when covering sensor with paper.





(7) Extension Practice :

After knowing its working principle, connect an LED to D3. We could control LED by line tracking sensor.



The program will be generated if you find the following file and drag it into Mixly software.

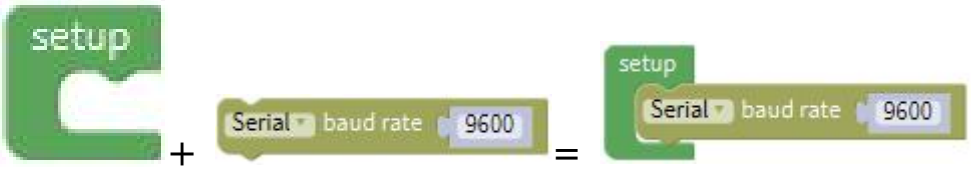
File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_3_Line Tracking Sensor	lesson_3.1_Extension Practice

You could edit code step by step as follows:

(1) Click “Control” to get block 

(2) Enter “Serial Port” to move block 

into 



(3) Go to “Variables” to drag out block 

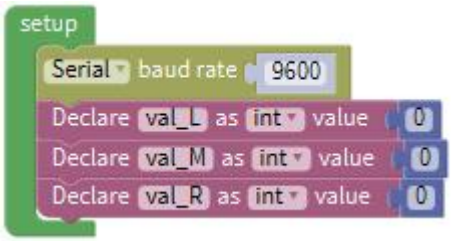
and copy it twice

(4) Separately change item into val_L, val_M and val_R.

(5) Go to “Math” to move out block 

and replicate it twice.

Edit code string as follows:



(6) Go to “Variables” to drag out block 

,

(7) Then click “Sensor” → “DigitalRead” → 

(8) Move block 

into 

block,

(9) The line tracking sensor is linked with D6, so set to PIN 6.

(10) Replicate block 

twice , and separately change val_L into val_M and val_R;

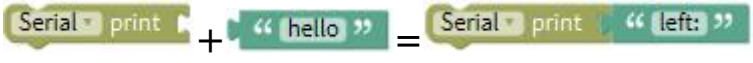
(11) The tracking sensor is also connected to D7 and D8, therefore, set to PIN 7 and PIN 8

(12) Click “Serial Port” to drag out block 

, and go to “Text” to move block 

into block 

,

(13) Change hello into left : 

(14) Copy block 

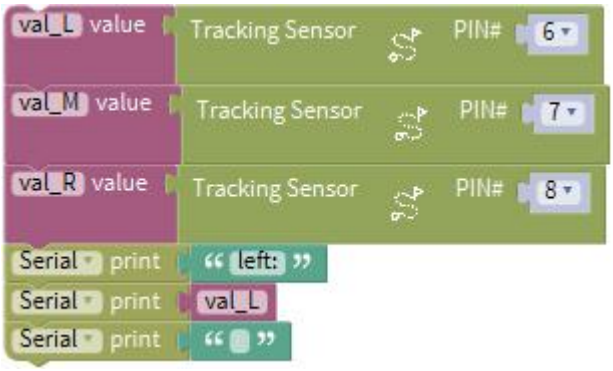
again , and enter “ Variables ” to move block 

into 

.

(15) Replicate code 

once and delete left : , as shown below:



(16) Duplicate code string 

once, and change left : into middle : , val_L into val_M.

(17) Copy block 

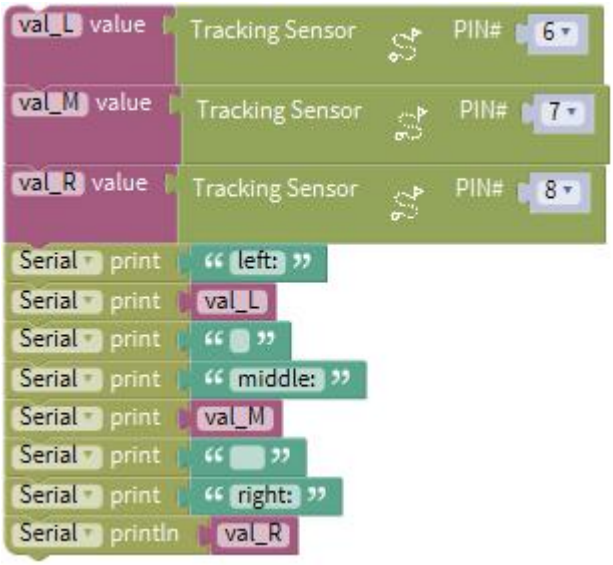
, and alter left : into right :

(18) Go to " Serial Port " to drag out 

block , and enter "Variables" to drag 

into block 

.



(19) Enter " Control " to move out block 

, click  to move

block 

into 

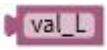
block, then 

block turns into 

.

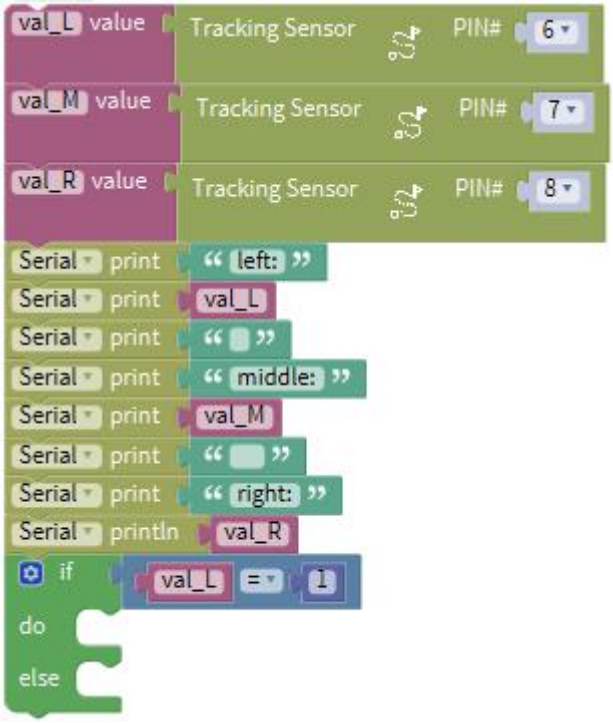
(20) Go to "Logic" to move block 

into if block.

(21) Enter "Variables" to drag block 

into the left box of block"=", click "Math" to move block 

into right box of"=", and change 0 into 1.



(22) Go to "sensor" → "ControlOutput" → 

(23) Place it into do block, and set to LOW

(24) LED is linked with D3, therefore, set to PIN 3.

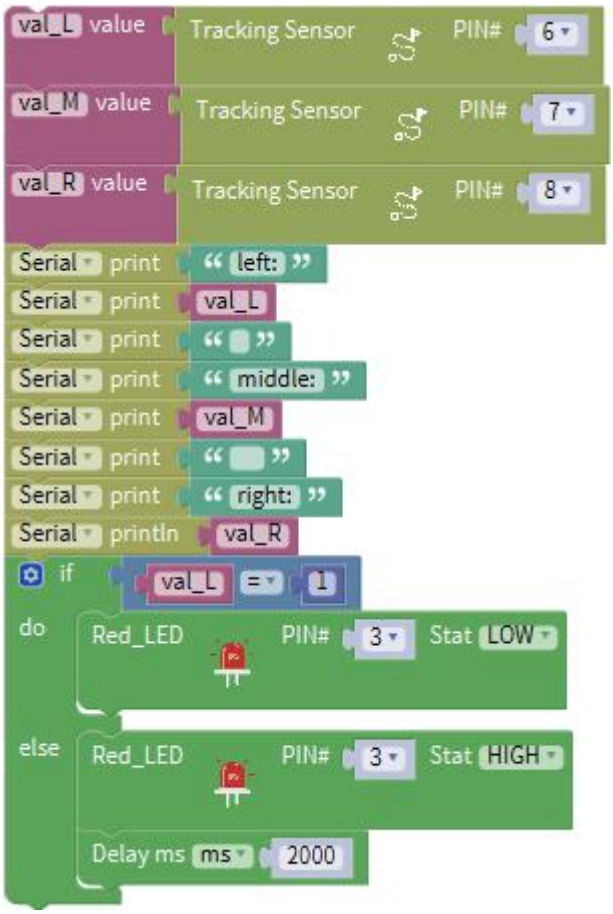
(25) Copy block 

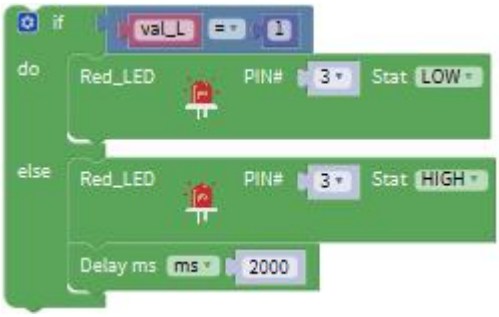
again and keep it into else

(26) Then set to HIGH, click "Control" to move out

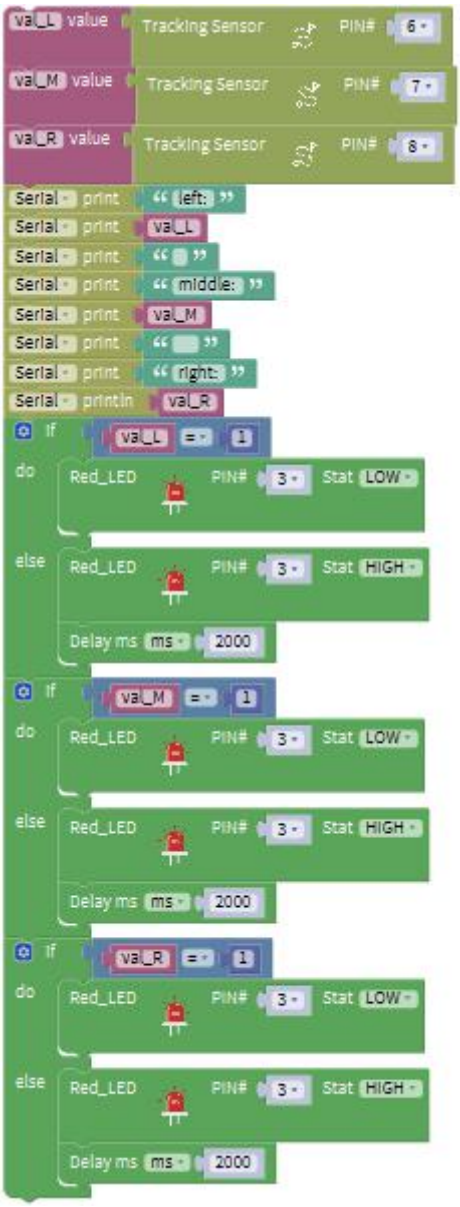
block 

, and delay in 2000ms.



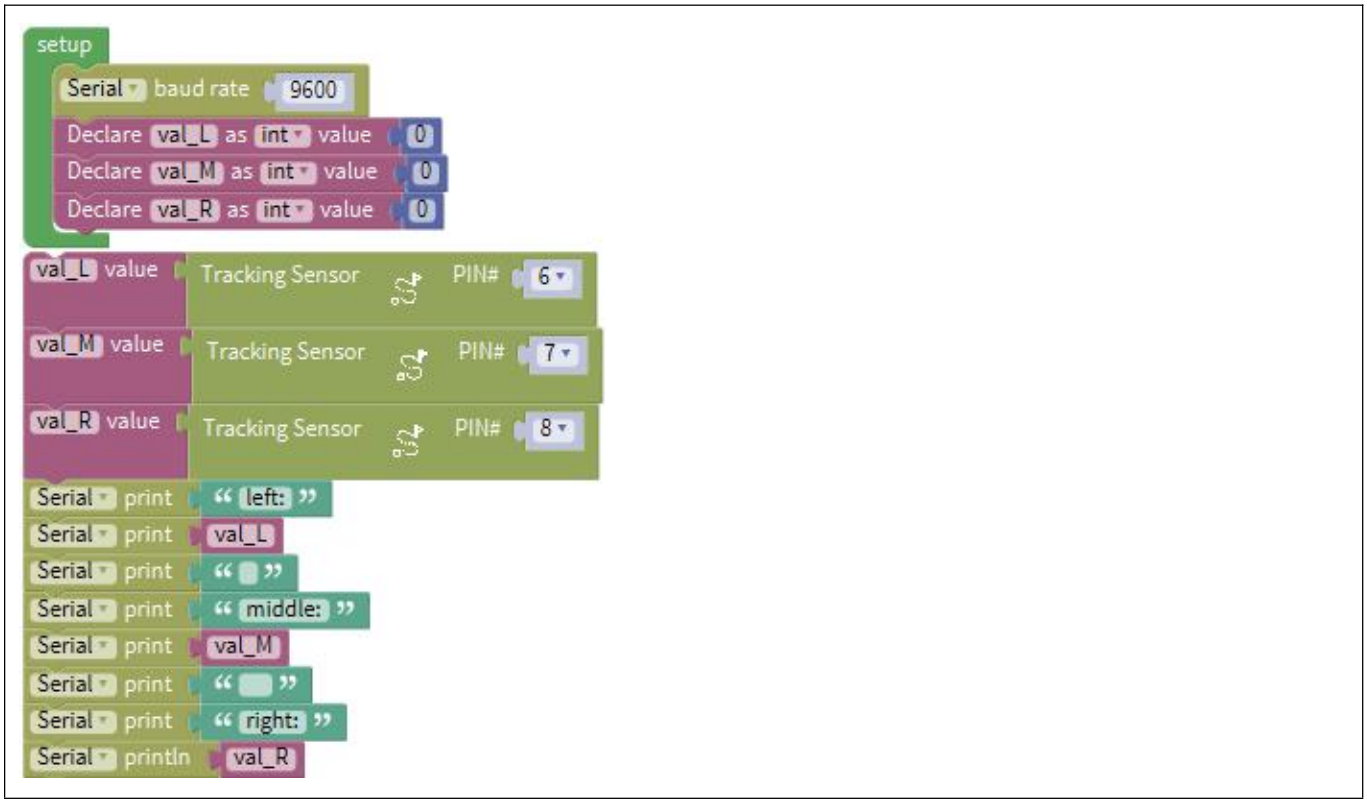
(27) Replicate code string 

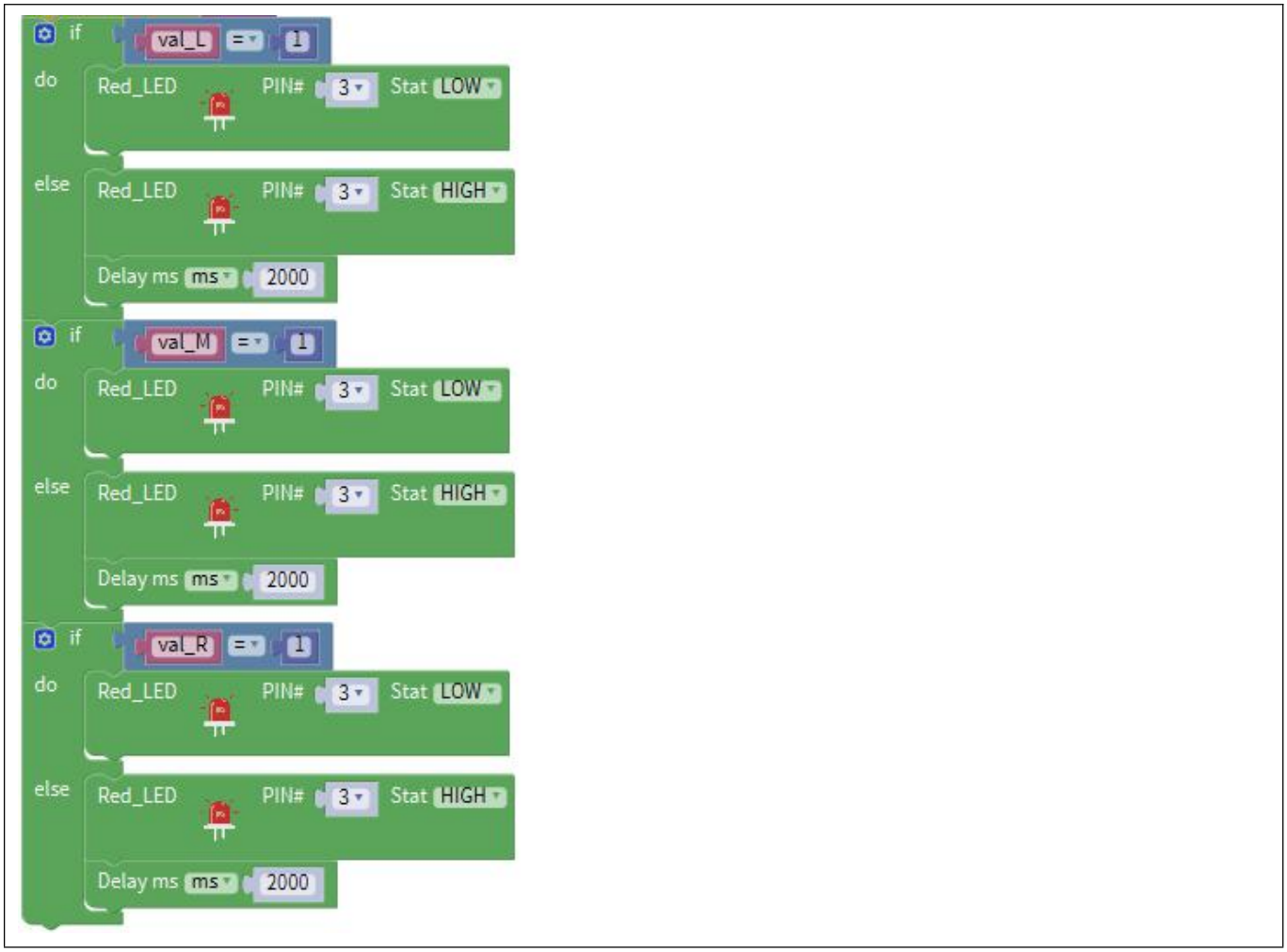
twice , and



respectively set to val_M and val_R.

Complete Program :




if val_L = 1
do Red_LED PIN# 3 Stat LOW
else Red_LED PIN# 3 Stat HIGH
Delay ms ms 2000
if val_M = 1
do Red_LED PIN# 3 Stat LOW
else Red_LED PIN# 3 Stat HIGH
Delay ms ms 2000
if val_R = 1
do Red_LED PIN# 3 Stat LOW
else Red_LED PIN# 3 Stat HIGH
Delay ms ms 2000

Initialization
Set baud rate to 9600
Set val_L to integer 0
Set val_M to integer 0
Set val_R to integer 0
Set the digital signals read by left TCRT5000 IR tube to val_L
Set the digital signals read by middle TCRT5000 IR tube to val_M
Set the digital signals read by right TCRT5000 IR tube to val_R
Serial port prints left:
Serial port displays the digital signals read by left TCRT5000 IR tube
Serial port prints blank space
Serial port prints middle:
Serial Port shows the digital signals read by middle TCRT5000 IR tube
Serial port prints blank space
Serial port prints right:
Serial port displays the digital signals read by right TCRT5000 IR tube
If the digital signal read by right TCRT5000 IR tube is 1, the program under do block will be executed.
Turn off red LED
If the digital signal read by left TCRT5000 IR tube is not 1, the program under else block will be executed.
Light up red LED
Delay in 2000ms
If the digital signal read by middle TCRT5000 IR tube is 1, the program under do block will be executed.
LED Turn off red LED
If the condition of digital signal read by left TCRT5000 IR tube=1 is not met, the program under else block will be executed.
Delay in 2000ms
If the condition of digital signal read by right TCRT5000 IR tube=1 is met, the program under do block will be executed.
LED Turn off red LED
If the condition of digital signal read by middle TCRT5000 IR tube=1 is not met, the program under else block will be executed.
Light up red LED
Delay in 2000ms

Upload the code to development board, we could see LED light up when covering the line tracking sensor by hand.

Project 4: Servo Control

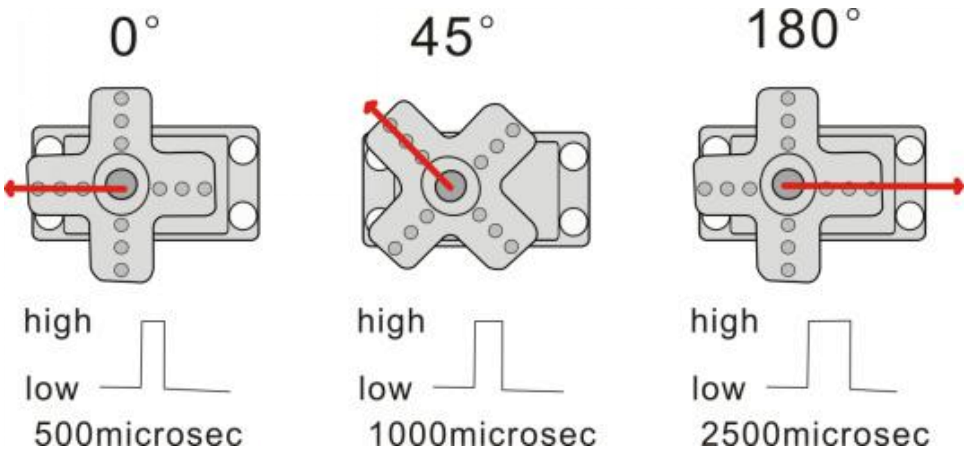
(1) Description

Servo motor is a position control rotary actuator. It mainly consists of housing, circuit board, core-less motor, gear and position sensor. Its working principle is that the servo receives the signal sent by MCU or receiver and produces a reference signal with a period of 20ms and width of 1.5ms, then compares the acquired DC bias voltage to the voltage of the potentiometer and obtain the voltage difference output.

When the motor speed is constant, the potentiometer is driven to rotate through the cascade reduction gear, which leads that the voltage difference is 0, and the motor stops rotating. Generally, the angle range of servo rotation is 0° --180 °

The rotation angle of servo motor is controlled by regulating the duty cycle of PWM (Pulse-Width Modulation) signal. The standard cycle of PWM signal is 20ms (50Hz). Theoretically, the width is distributed between 1ms-2ms, but in fact, it's between 0.5ms-2.5ms. The width

corresponds the rotation angle from 0° to 180°. But note that for different brand motor, the same signal may have different rotation angle.



The corresponding servo angles are shown below:

High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

(2) Specification

Working voltage: DC 4.8V ~ 6V

Operating angle range: about 180 ° (at 500 → 2500 μsec)

Pulse width range: 500 → 2500 μsec

No-load speed: 0.12 ± 0.01 sec / 60 (DC 4.8V) 0.1 ± 0.01 sec / 60 (DC

6V)

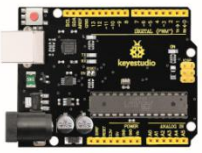
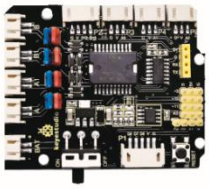
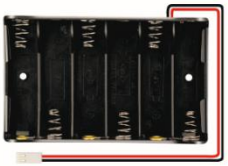
No-load current: 200 ± 20mA (DC 4.8V) 220 ± 20mA (DC 6V)

Stopping torque: 1.3 ± 0.01kg · cm (DC 4.8V) 1.5 ± 0.1kg · cm (DC 6V)

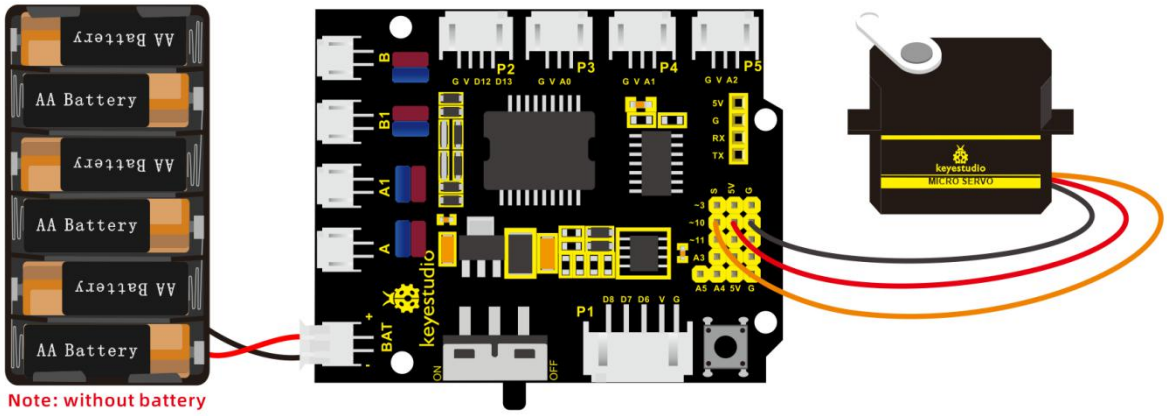
Stop current: ≦ 850mA (DC 4.8V) ≦ 1000mA (DC 6V)

Standby current: 3 ± 1mA (DC 4.8V) 4 ± 1mA (DC 6V)

(3) Equipment :

Control Board *1	L298P Motor Shield *1	Servo motor *1	6 AA battery Holder *1	USB Cable *1
				

(4) Wiring Diagram :



Wiring note: the brown line of servo is linked with Gnd(G), the red line is connected to 5v(V) and orange line is attached to digital 10.

The servo has to be connected to external power due to its high demand for driving servo current. Generally, the current of development board is

not enough. If without connected power, the development board could be burnt.

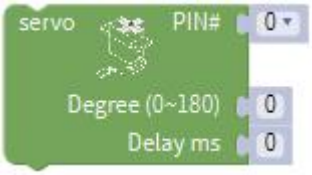
(5) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_4_Servo Control	lesson_4_Servo Control

You could edit code step by step as follows:

(1) Go to“Control”to get block 

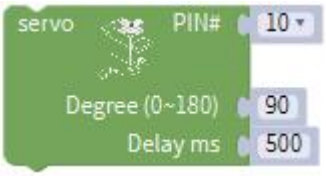
(2) Enter “Module”→“Drive_Module”to get block 

and place it into block 

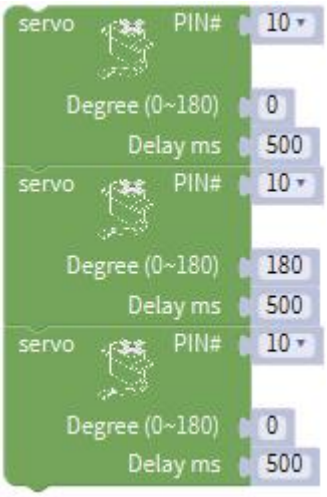
,

(3) The servo is linked with D10, so set to PIN 10.

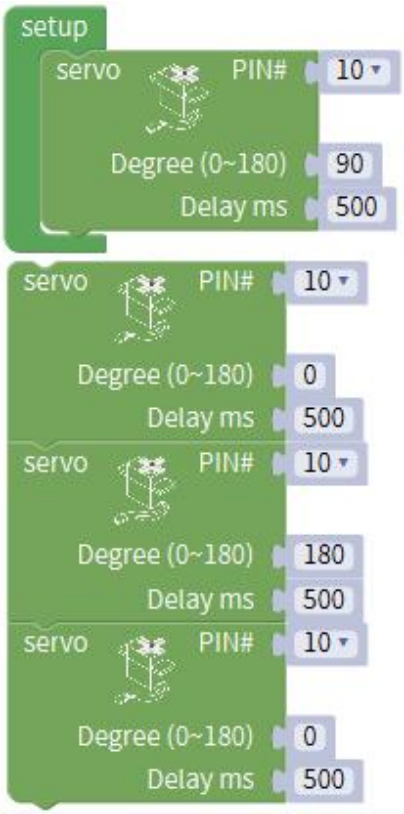
(4) Set servo to 90°, and delay in 500ms.

(5) Replicate code string 

for three times , respectively change 90 into 0, 180 and 0



Complete Program :



Initialization

Set initial angle of servo to 90°

Set angle of servo to 0°

Set angle of servo to 180°

Set angle of servo to 0°

(6) Test Result :

Upload code successfully and power on, servo swings in the range of 0° to 180°.

Project 5: Ultrasonic Sensor

(1) Description

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.

The HC-SR04 or the ultrasonic sensor is being used in a wide range of electronics projects for creating obstacle detection and distance measuring application as well as various other applications. Here we have brought the simple method to measure the distance with arduino and ultrasonic sensor and how to use ultrasonic sensor wit h arduino.

(2) Specification

Power Supply :+5V DC

Quiescent Current : <2mA

Working Current: 15mA

Effectual Angle: <15°

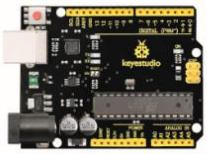
Ranging Distance : 2cm – 400 cm

Resolution : 0.3 cm

Measuring Angle: 30 degree

Trigger Input Pulse width: 10uS

(3) Equipment :

Control Board *1	L298P Motor Shield *1	Ultrasonic module *1	LED module *1	USB Cable *1	4pin Dupont line *1
					 3pin Dupont line *1 

(4) Ultrasonic Sensor

As the above picture shown, it is like two eyes. One is transmitting end, the other is receiving end.

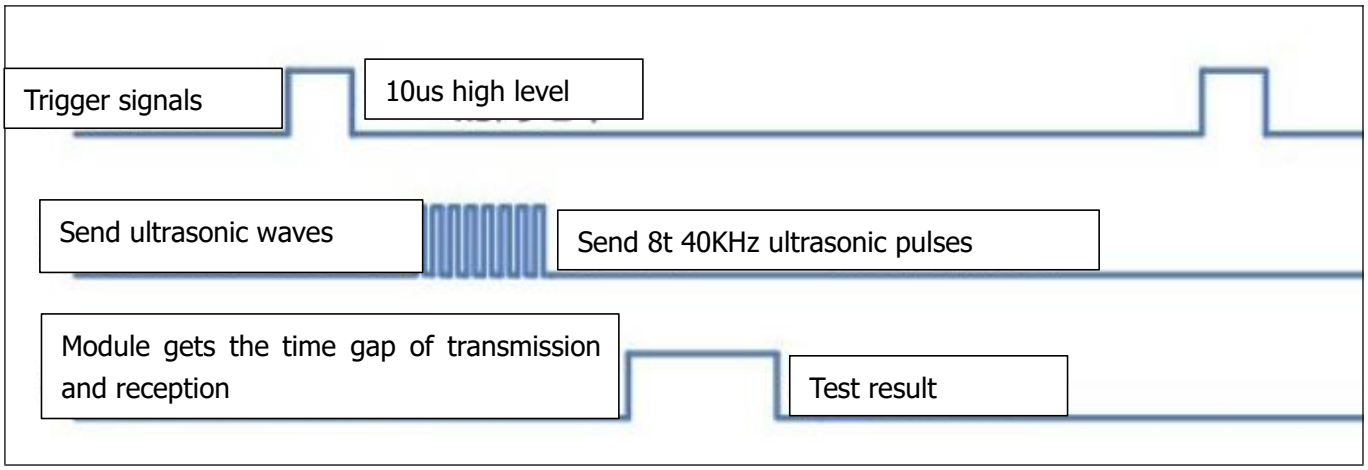
The ultrasonic module will emit the ultrasonic waves after trigger signal. When the ultrasonic waves encounter the object and are reflected back, the module outputs an echo signal, so it can determine the distance of object from the time difference between trigger signal and echo signal. The t is the time that emitting signal meets obstacle and returns.
and the propagation speed of sound in the air is about 343m/s, therefore, distance = speed * time, because the ultrasonic wave emits and comes back, which is 2 times of distance, so it needs to be divided by 2, the distance measured by ultrasonic wave = (speed * time)/2

1. Use method and timing chart of ultrasonic module:

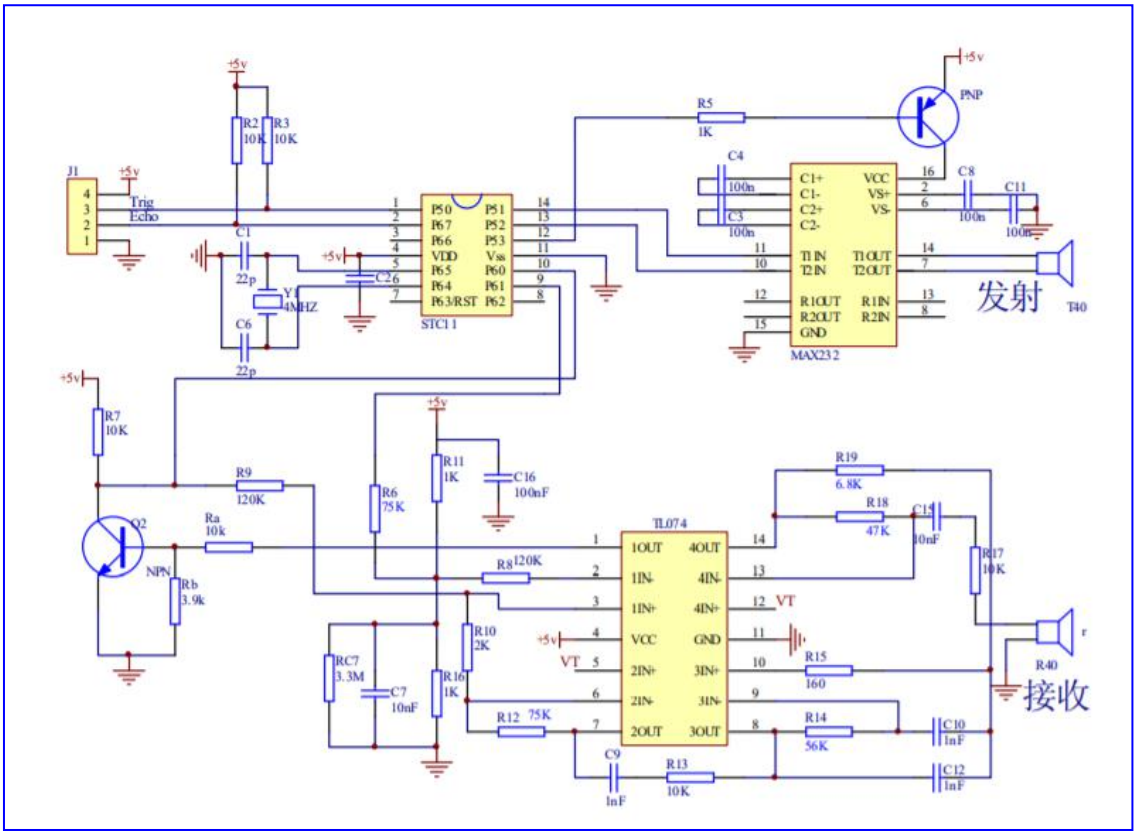
Setting the delay time of Trig pin of SR04 to 10μs at least, which can trigger it to detect distance.

2. After triggering, the module will automatically send eight 40KHz ultrasonic pulses and detect whether there is a signal return. This step will be completed automatically by the module.

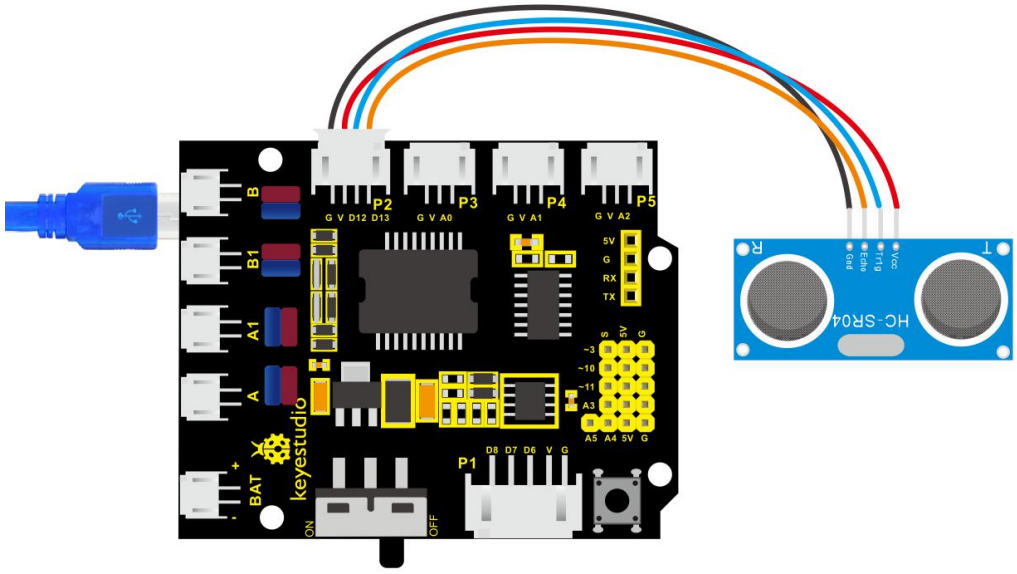
3. If the signal returns, the Echo pin will output a high level, and the duration of the high level is the time from the transmission of the ultrasonic wave to the return.



Circuit diagram of ultrasonic sensor:



(5) Hook-up Diagram :



Wiring guide:

Ultrasonic sensor		keyestudio V5 sensor shield
VCC	→	5v(V)
Trig	→	12(S)
Echo	→	13(S)
Gnd	→	Gnd(G)

(6) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_5_Ultrasonic Sensor	lesson_5.1_Ultrasonic Sensor

You could edit code step by step as follows:

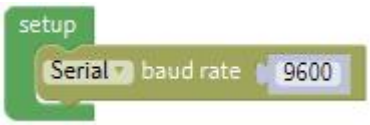
(1) Click “Control” Module to get block 

.

(2) Enter “Serial Port” to move block 

into 

block.

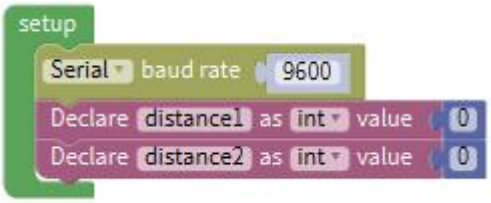


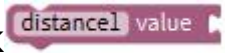
(3) Go to "Variables" to get block 

and copy it twice. Respectively change item into distance 1 and distance 2.

(4) Click "Math" to move out block 

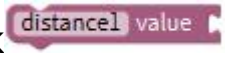
and replicate it twice.



(5) Click "Variables" to drag out block 

, and enter "sensor" → "OtherSensor" → 

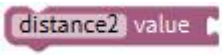
.

(6) Combine block 

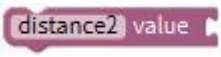
with block 

,

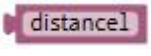
Trig pin of ultrasonic sensor is connected to D12(S) of shield, Echo is attached to D13(S), so set Trig 12 and Echo 13.

(7) Click "Variables" to get block 

, and drag out 

from "Math" and keep it behind block 

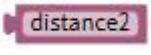
.

(8) Go to "Variables" to move out block 

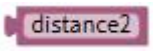
left 1 box".

(9) Then change another 1 into 0.3937, and set to "×".

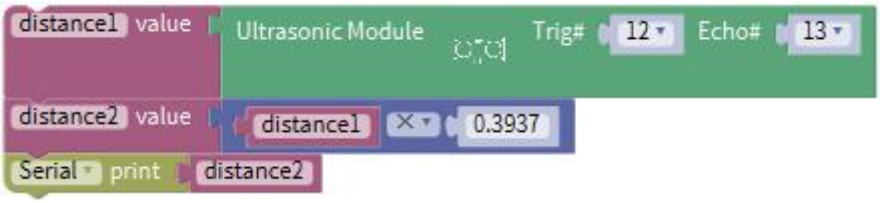
(10) Go to "Serial Port" to drag out block 

, and click "Variables" to get block 

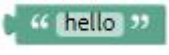
.

(11) Combine 

with 

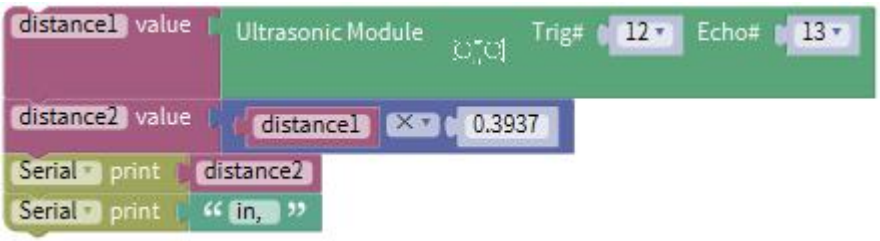


(12) Go to "Serial Port" to get block 

, then click "Text" module to drag block 

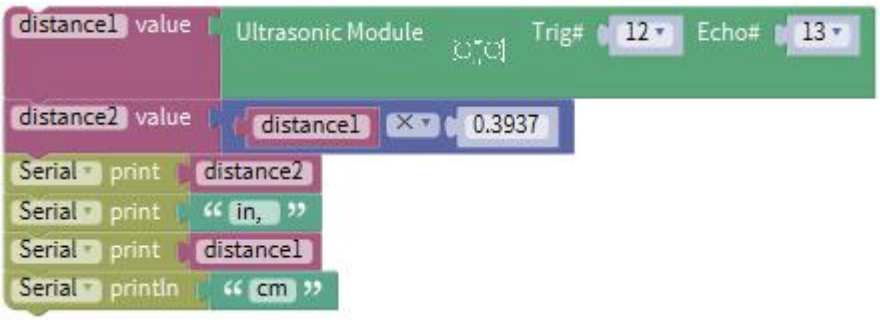
into 

中, and change hello into in.



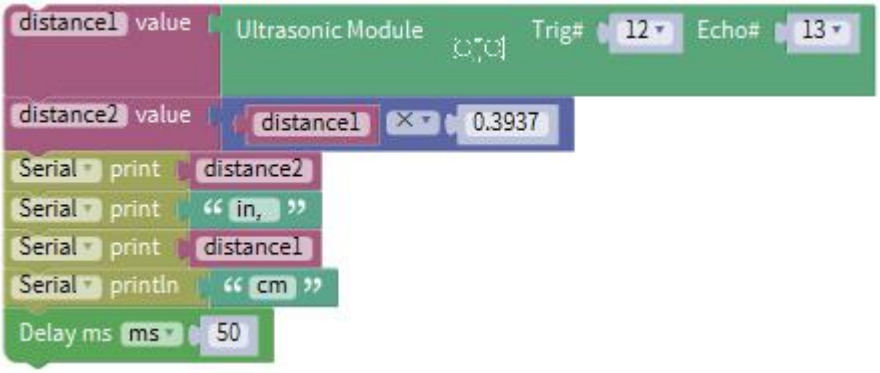
(13) Copy code string 

once, change distance2 into distance1, in into cm

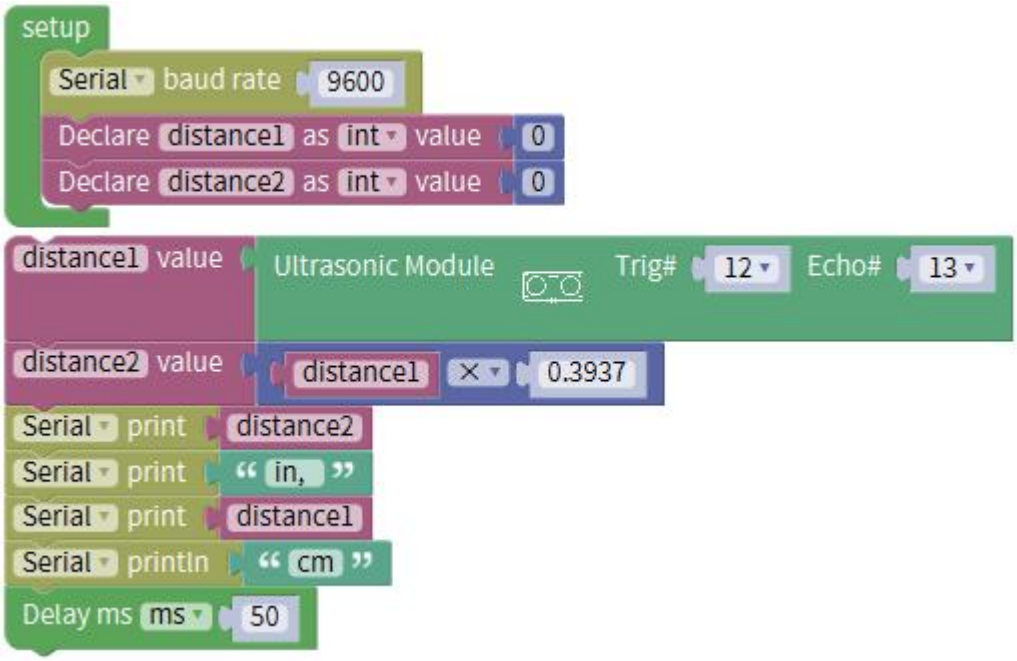


(14) Click "Control" to get block 

, delay in 50ms.



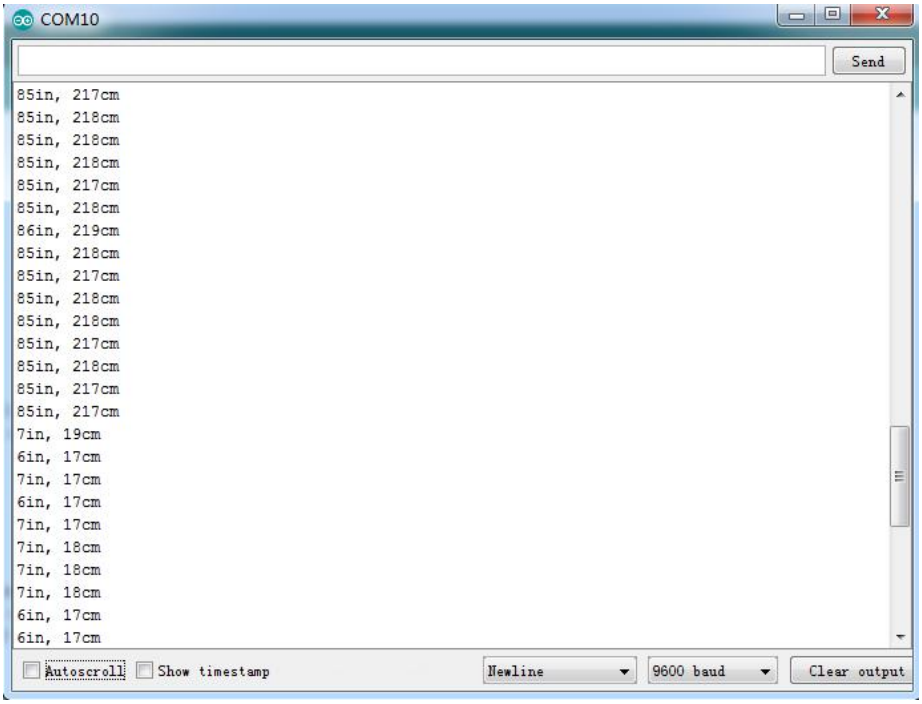
Complete Program :



Initialization
Set baud rate to 9600
Set distance1 to integer and 0
Set distance2 to integer and 0
Set the value of distance1 to the distance detected by ultrasonic sensor.
Set the distance2 to distance1×0.3937
Serial Port prints the value of distance2
Serial port prints in,
Serial port prints the value of distance1
Serial port prints cm
Delay in 50ms

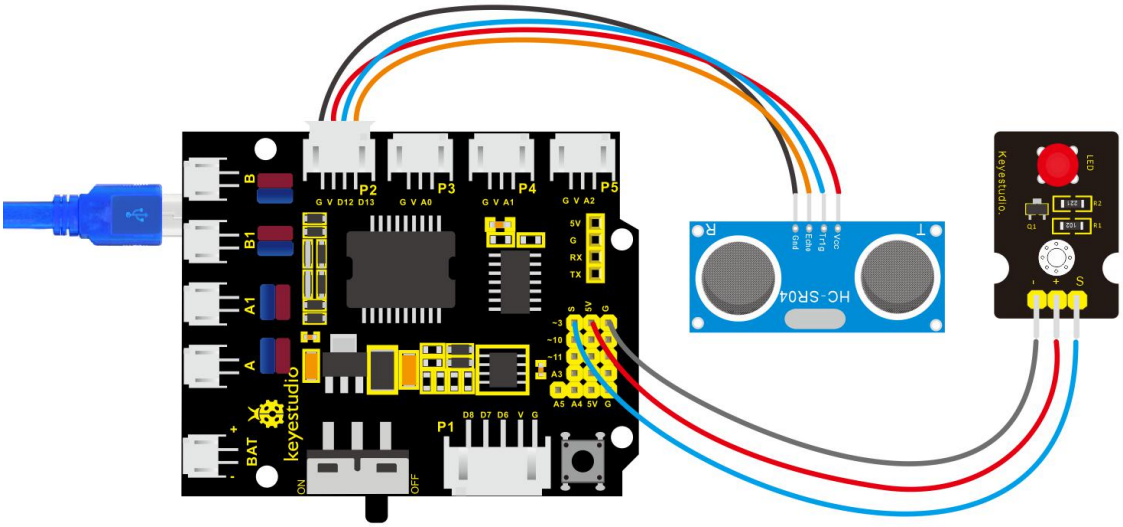
(7) Test Result :

Upload test code on the development board, open serial monitor and set baud rate to 9600. The detected distance will be displayed, unit is cm and inch. Hinder the ultrasonic sensor by hand, the displayed distance value gets smaller.



(8) Extension Practice :

We have measured the distance displayed by ultrasonic sensor. How about controlling the LED with the measured distance? Let's try it, connect an LED light module to the D3 pin.



The program will be generated if you find the following file and drag it into Mixly software.

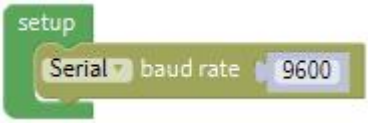
File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_5_Ultrasonic Sensor	lesson_5.2_Extension Practice

You could edit code step by step as follows:

(1) Enter “Control” to get block 

(2) Click “Serial Port” to drag out block 

into 



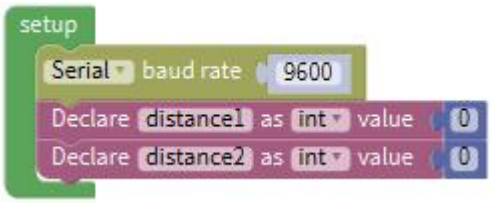
(3) Go to "Variables" to move out block 

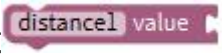
and copy it twice.

(4) Enter "Math" to drag out block 

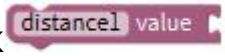
and replicate twice, and change item into distance 1 and distance 2.

(5) Edit the code string as follows:

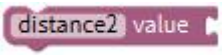


(6) Go to "Variables" to move out 

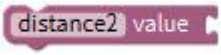
block, then click"sensor" → "OtherSensor" → 

(7) Combine it with block 

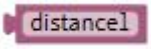
, the pin Trig of ultrasonic sensor is linked with D12(S) of expansion board, pin Echo is attached to D13(S); therefore, click the triangle button to select 12 and 13.

(8) Click "Variables" to get block 

, and drag out 

from "Math" and keep it behind block 

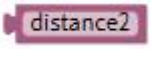
.

(9) Go to "Variables" to move out block 

left 1 box".

(10) Then change another 1 into 0.3937, and set to "×".

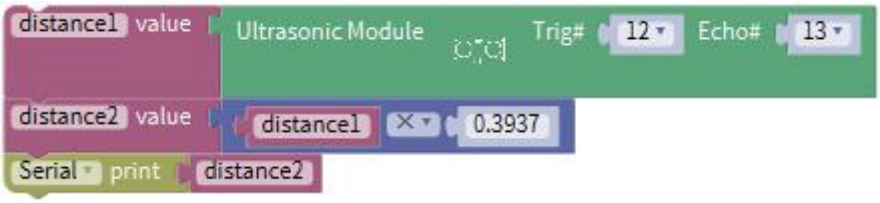
(11) Go to "Serial Port" to drag out block 

, and click "Variables" to get block 

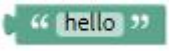
.

(12) Combine 

with 

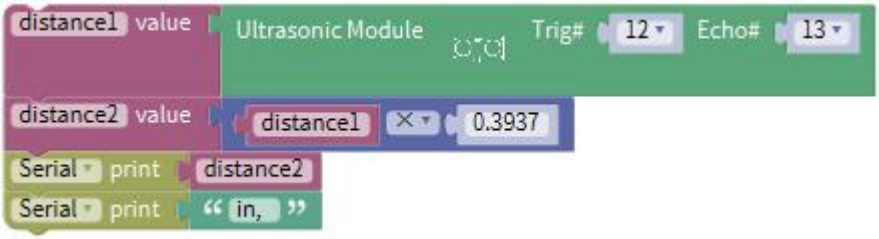


(13) Go to "Serial Port" to get block 

, then click "Text" module to drag block 

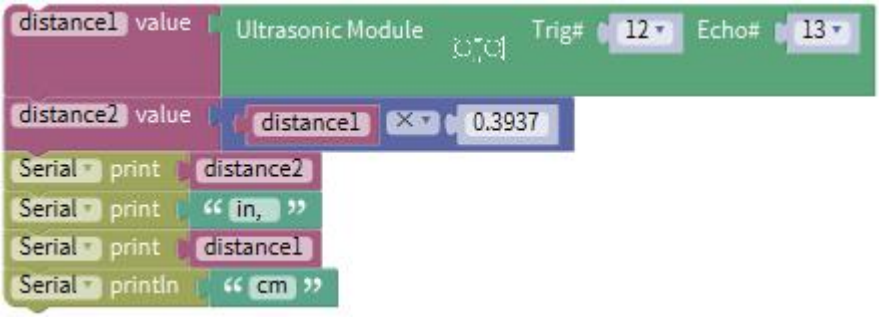
into 

中, and change hello into in.



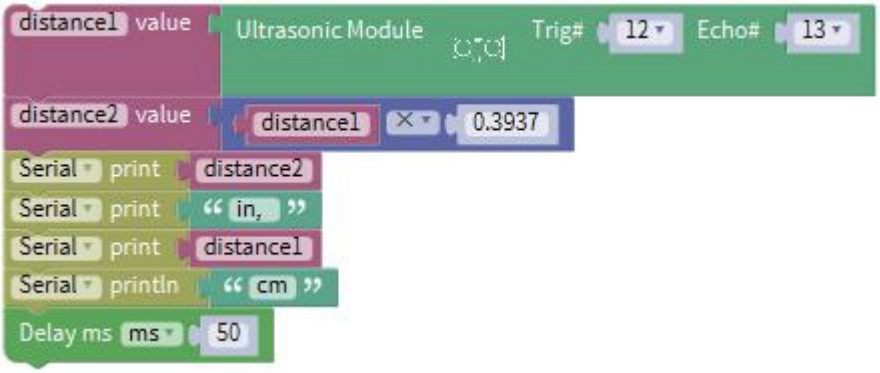
(14) Replicate code string 

once, and change distance2 into distance1, and in into cm.



(15) Go to"Control"to move out block 

and delay in 50ms



(16) Click " Control " and drag it into block 

, click  and move 

into 

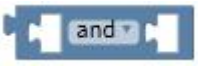
then we get block 

.

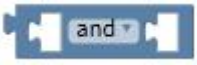
(17) Go to "Logic" to get 

and 

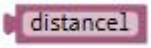
.

(18) Place 

into if block, and leave 

in the left box of block 

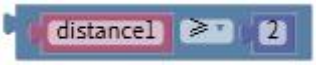
.

(19) Enter "Variables" to get block 

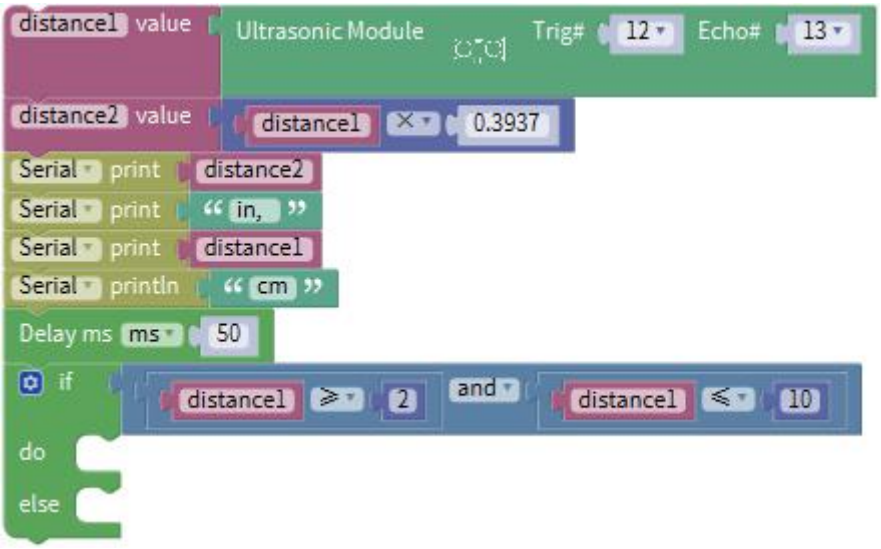
and keep it at left box of block "=".

(20) Go to "Math" to move out 

into right box of block "=", and change 0 into 2, "=" into "≥".

(21) Replicate block 

once and set 



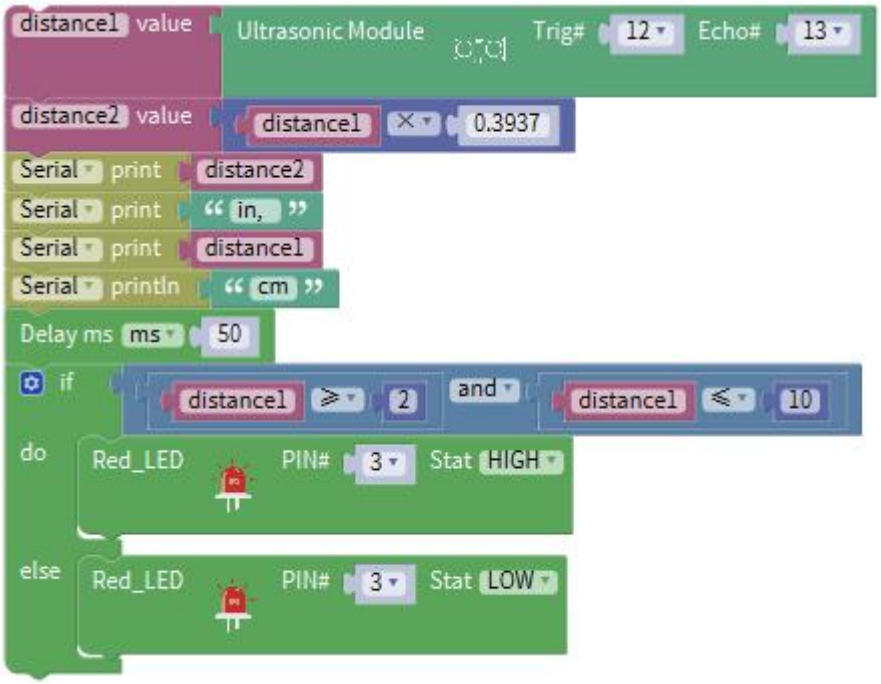
(22) Click "sensor" →"ControlOutput"→ 

and place it into do block,

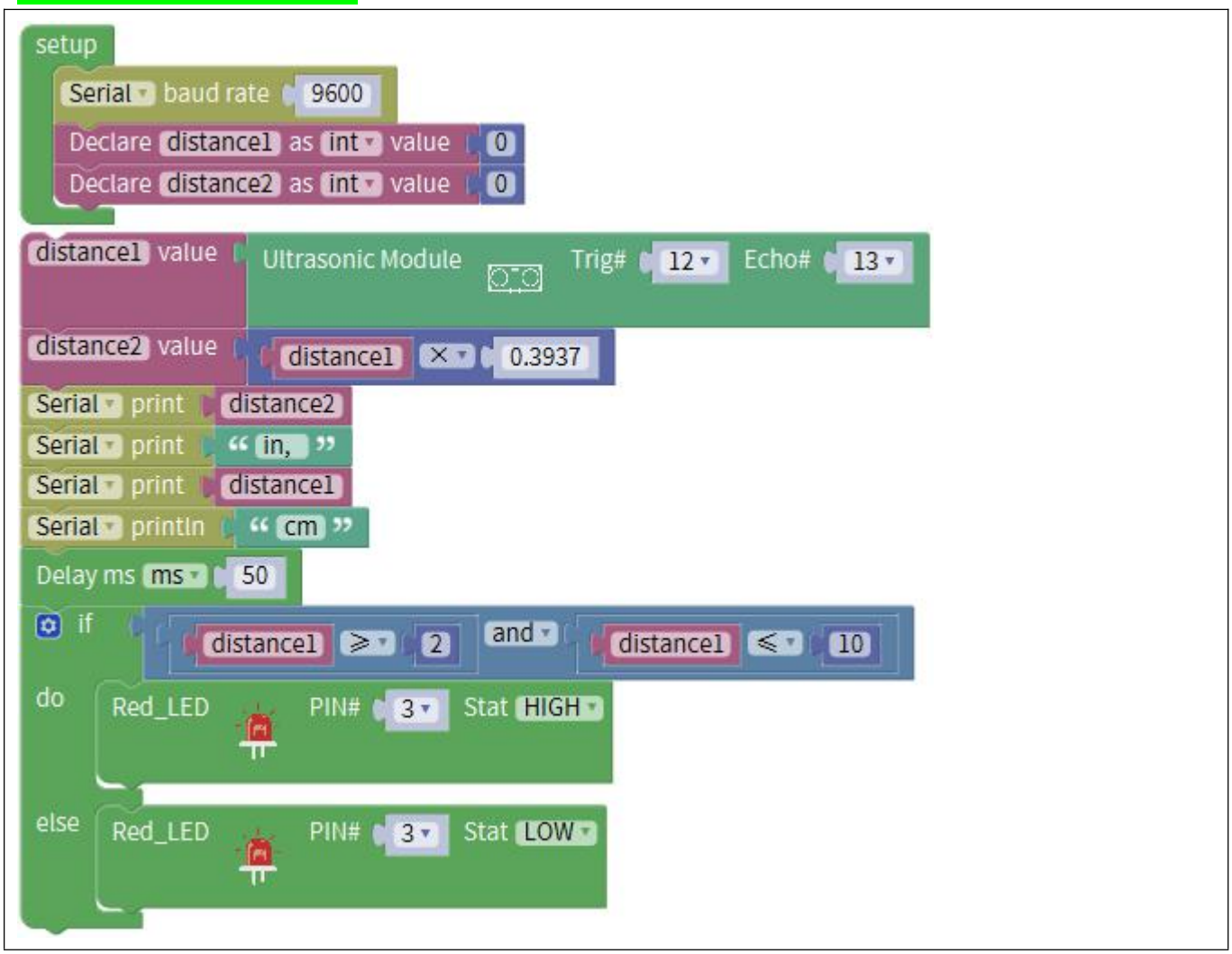
(23) The S end of LED Module is connected to D3 of expansion board, therefore, click the triangle button to select 3.

(24) Replicate block 

once and leave it into "else" and set to LOW.



Complete Program :



Initialization
Set baud rate to 9600
Set distance1 to integer 0
Set distance2 to integer 0
Set the value of distance1 to the distance(cm) detected by ultrasonic sensor
Set the value of distance2 to distance1×0.3937
Serial port shows the value of distance2
Serial port prints in
Serial port displays the value of distance1
Print cm on serial port
Delay in 50ms
If distance1≥2 and distance1≤10，the code under do block will be executed.
Light up red LED connected to D3
If the condition of distance1≥2 and distance1≤10 is not met，the code under the else block will be executed.
Turn off red LED connected to D3

Upload test code to development board and block ultrasonic sensor by hand, then check if LED is on

Project 6: IR Reception

(1) Description

There is no doubt that infrared remote control is ubiquitous in daily life. It is used to control various household appliances, such as TVs, stereos, video recorders and satellite signal receivers. Infrared remote control is composed of infrared transmitting and infrared receiving systems, that is, an infrared remote control and infrared receiving module and a single-chip

microcomputer capable of decoding.

The 38K infrared carrier signal emitted by remote controller is encoded by the encoding chip in the remote controller. It is composed of a section of pilot code, user code, user inverse code, data code, and data inverse code. The time interval of the pulse is used to distinguish whether it is a 0 or 1 signal and the encoding is made up of these 0, 1 signals.

The user code of the same remote control is unchanged. The data code can distinguish the key.

When the remote control button is pressed, the remote control sends out an infrared carrier signal. When the IR receiver receives the signal, the program will decode the carrier signal and determines which key is pressed. The MCU decodes the received 01 signal, thereby judging what key is pressed by the remote control.

Infrared receiver we use is an infrared receiver module. Mainly composed of an infrared receiver head, it is a device that integrates reception, amplification, and demodulation. Its internal IC has completed demodulation, and can achieve from infrared reception to output and be compatible with TTL signals. Additionally, it is suitable for infrared remote control and infrared data transmission. The infrared receiving module

made by the receiver has only three pins, signal line, VCC and GND. It is very convenient to communicate with arduino and other microcontrollers.

(2) Specification

Operating Voltage: 3.3-5V（DC)

Interface: 3PIN

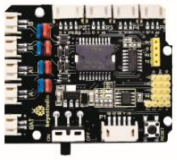
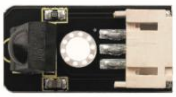
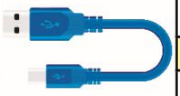
Output Signal: Digital signal

Receiving Angle: 90 degrees

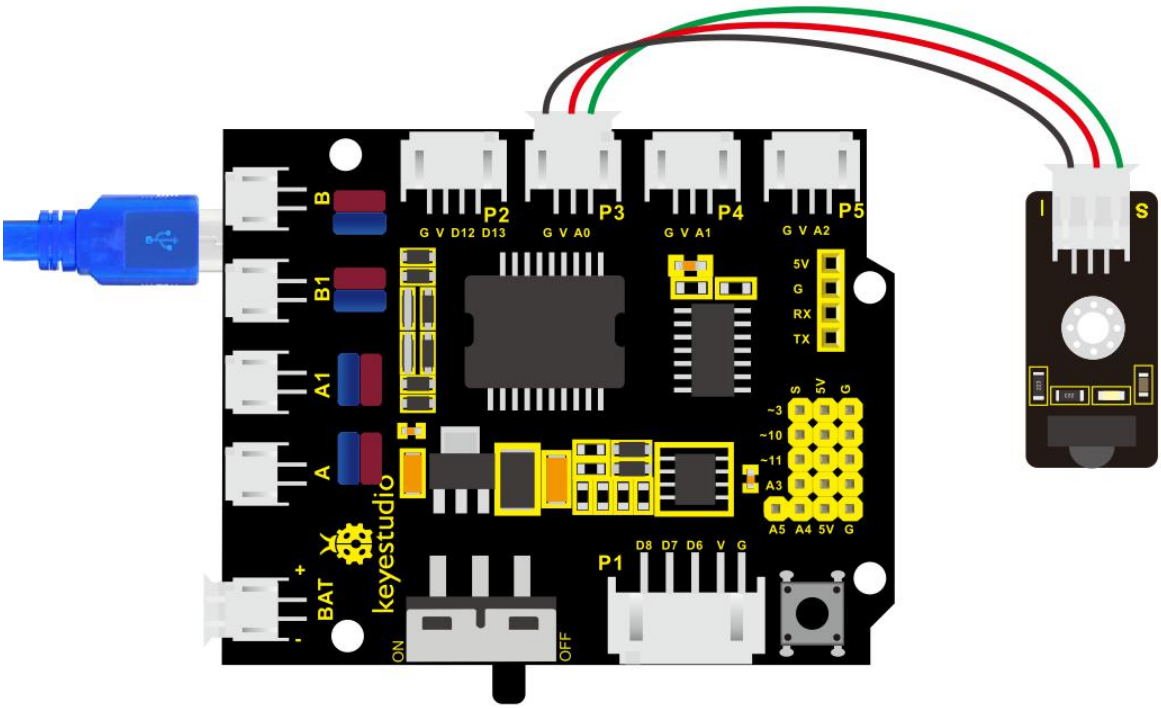
Frequency: 38khz

Receiving Distance: 10m

(3) Equipment :

Control Board *1	L298P Motor Shield *1	IR receiver module *1	LED module *1	USB Cable *1	3pin Dupont line *1
					 3pin Dupont line *1 

(4) Connection Diagram



Respectively link “-”, “+” and S of IR receiver module with G(GND）, V（VCC）and A0 of keyestudio development board.

Attention: On the condition that digital ports are not available, analog ports can be regarded as digital ports. A0 equals to D14, A1 is equivalent to digital 15.

(5) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_6_IR Reception	lesson_6.1_IR Reception

You could edit code step by step as follows:

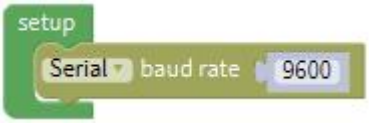
(1) Enter “Control” Module to get block 

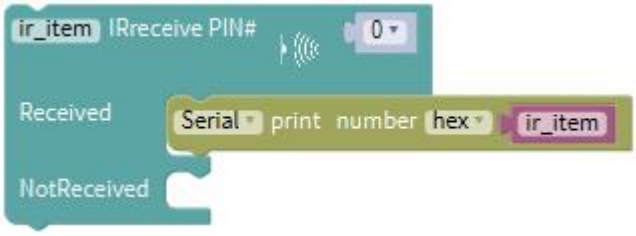
.

(2) Click “ Serial Port ” Module to drag block 

into 

block



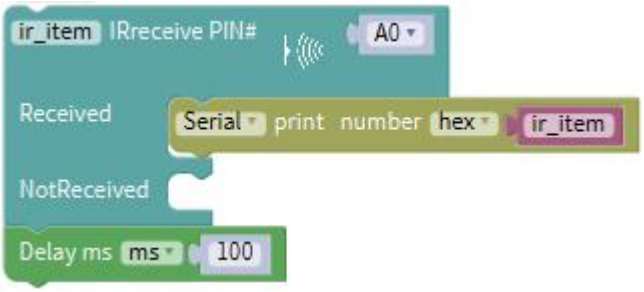
(3) Go to “ Module ” → ” Communicate_Module ” → 

,

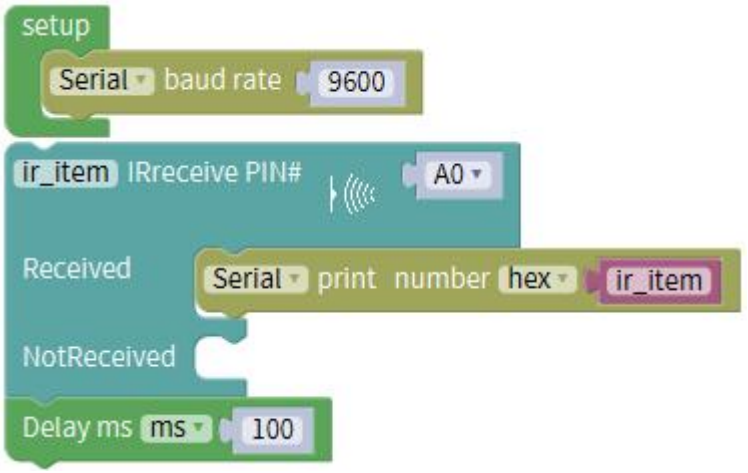
(4) Signal end of IR receiver module is connected to A0 of shield, therefore, click triangle button to select A0.

(5) Click “Control” Module, to move 

out, delay in 100ms



Complete Program :



Initialization

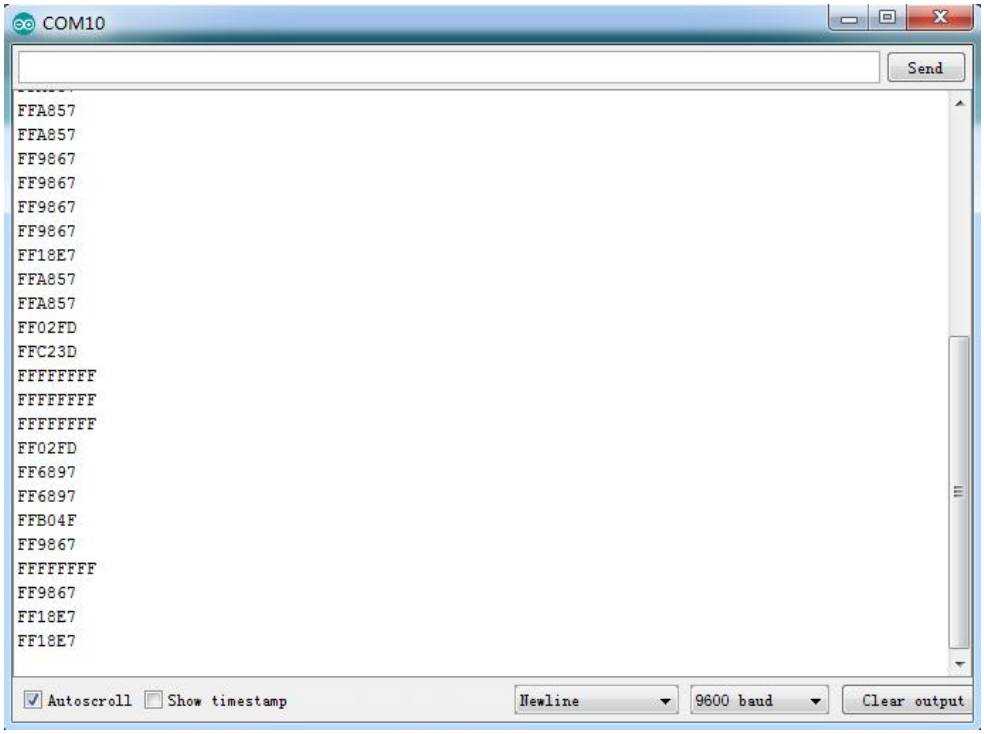
Set baud rate to 9600

Serial port prints the numbers in Hex when IR receiver module receives the signal of A0

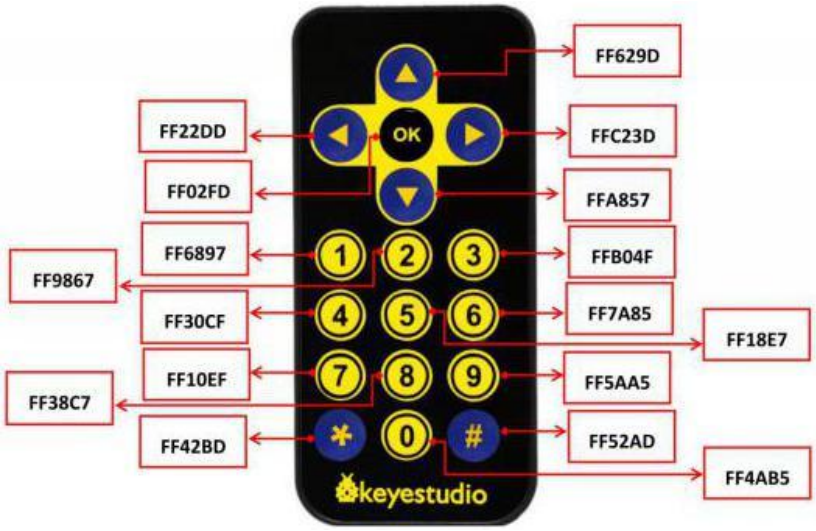
(6) Test Result :

Upload test code, open serial monitor and set baud rate to 9600, point

remote control to IR receiver and the corresponding value will be shown, if pressing so long, the error codes will appear

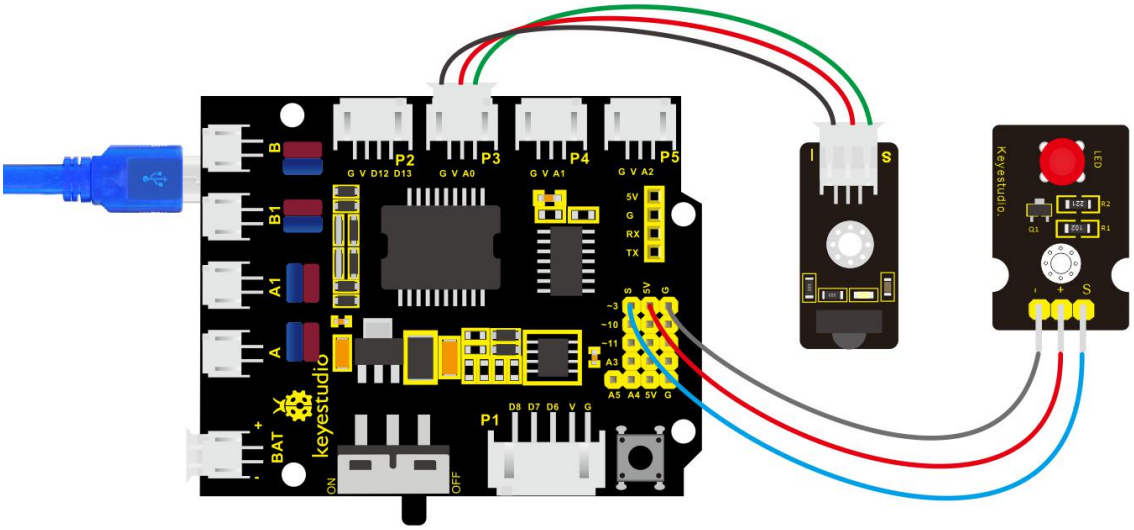


Below we have listed out each button value of keyestudio remote control. So you can keep it for reference.



(7) Extension Practice:

We decoded the key value of IR remote control. How about controlling LED by the measured value? We could operate an experiment to affirm. Attach an LED to D3, then press the keys of remote control to make LED light up and off.



The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_6_IR Reception	lesson_6.2_Extension Practice

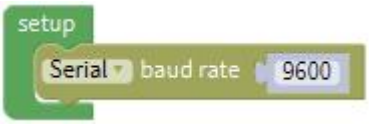
You could edit code step by step as follows:

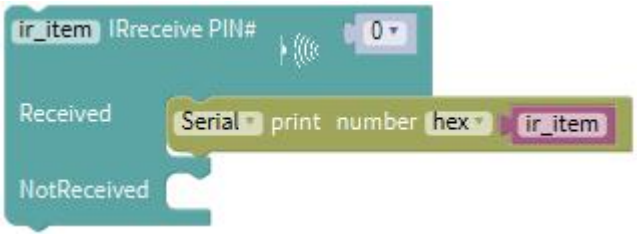
(1) Enter “Control” Module to get block 

(2) Go to “Serial Port” to move out block 

into block 

.



(3) Go to “ Module ” → “ Communicate_Module ” → 

,

(4) S end of IR receiver module is linked with A0. therefore, click triangle button to set A0.

(5) Go to “Variables” to move block 

into block 

,

(6) Go to “ Math ” to drag out block 

and integrate with 

, then change item into a.

(7) Click “ Control ” module to get block 

, click 
,

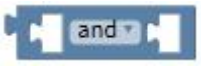
move 

into 

block, then 

turns into 

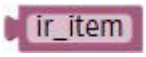
block.

(8) Go to "Logic", and find out block 

. Leave it behind if block

(9) Go to "Logic" to place 

in the left box.

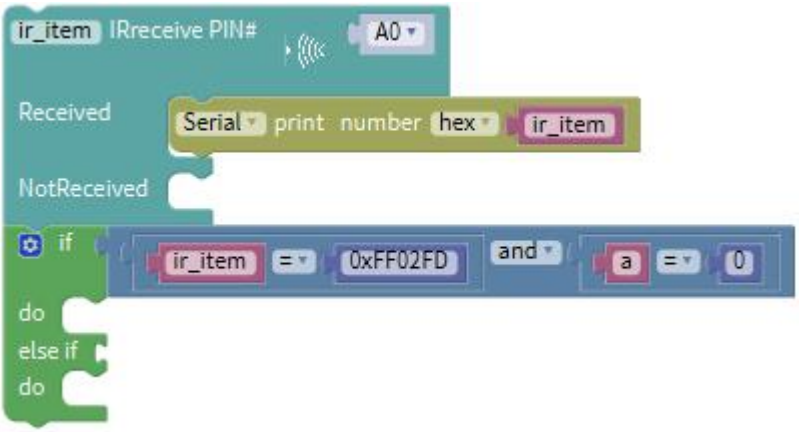
(10) Click "Variables" to move 

in the left box of "=" block, go to "Math" to drag out 

into right box of "=", then change 0 into 0xFF02FD.

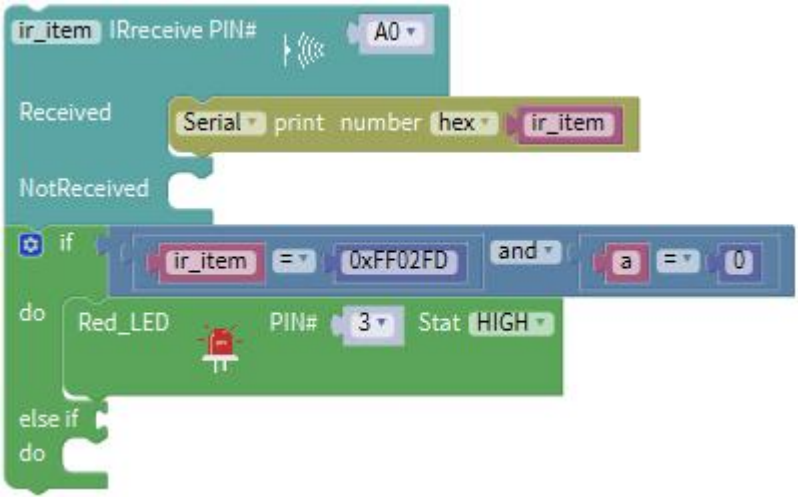
Copy 

once and keep it into right box of "and" block, change ir_item into a, 0xFF02FD into 0.



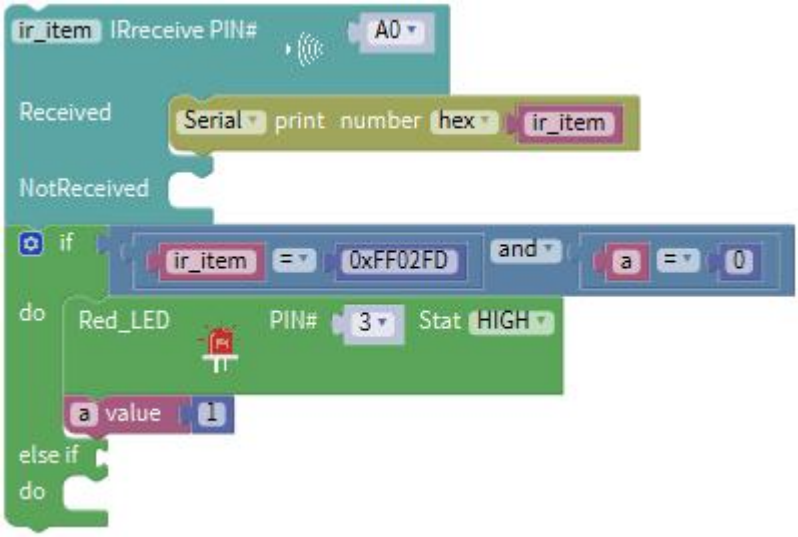
(11) Click "sensor"→ "ControlOutput"→ 

(12) Keep it into the first do block, S end of red LED module is linked with D3 of expansion board, and set to Pin 3.

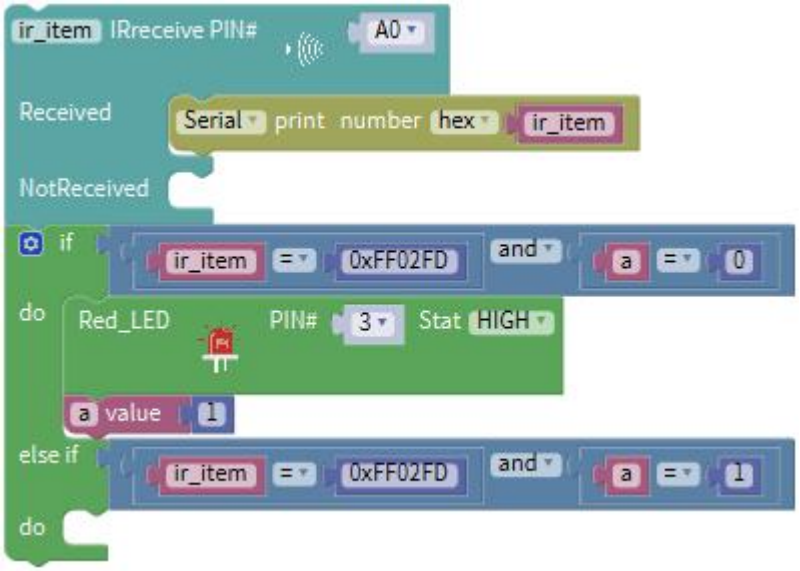


(13) Go to "Variables" to move  below "Red_LED...HIGH" block.

(14) Click "Math" to get block  into  , and change 0 into 1.



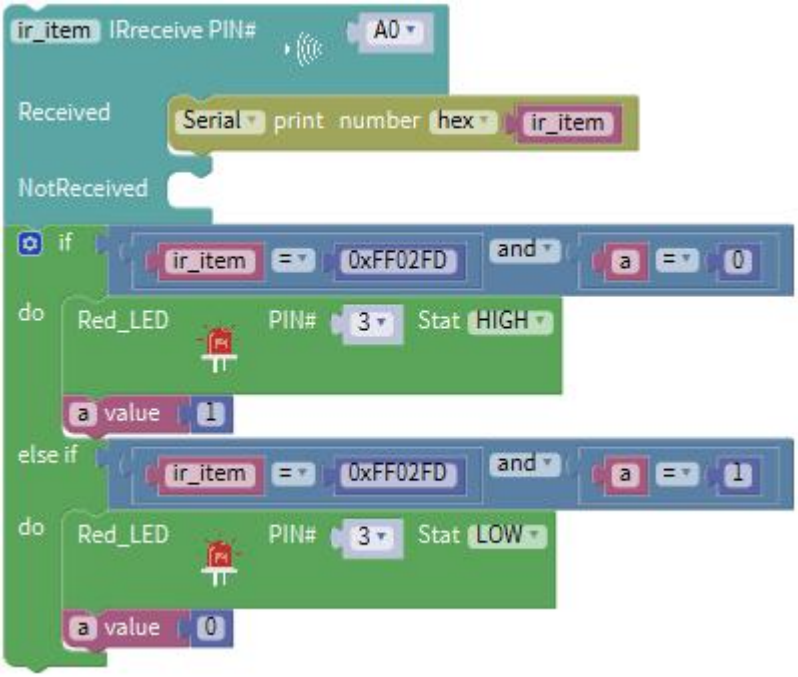
(15) Replicate  once and move into "else if", change 0 into 1.



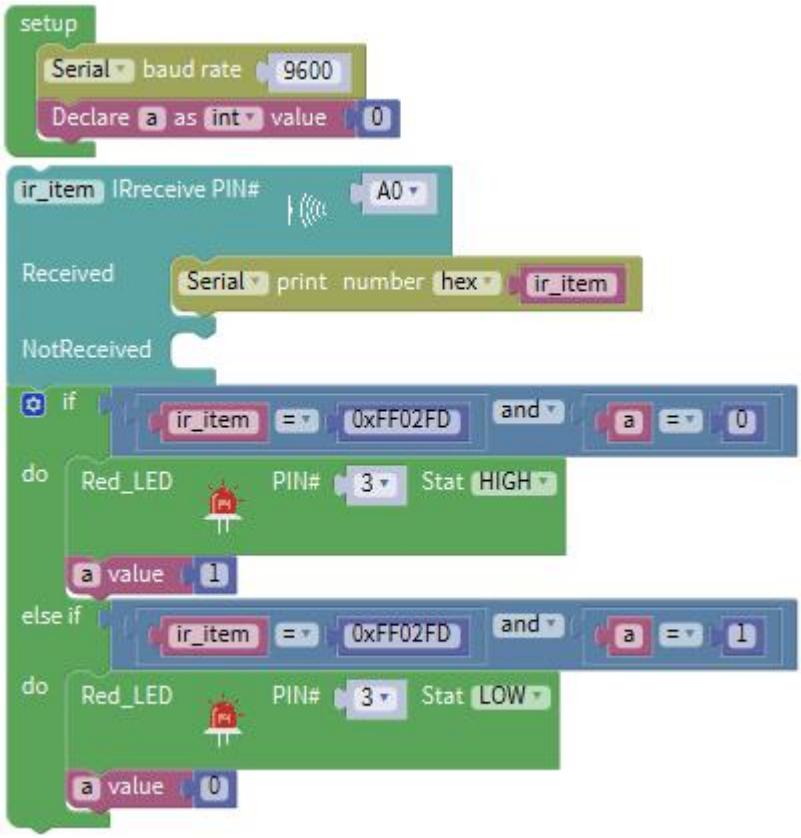
(16) Replicate code string 

once and place it into second do block.

(17) Click “HIGH” into “LOW”, 1 into 0.



Complete Program :



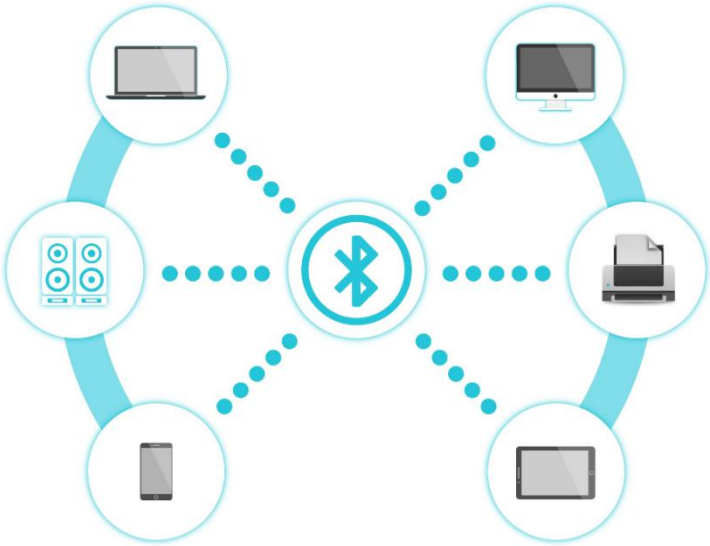
Initialization
Set baud rate to 9600
Set a to integer and assign 0 to it
Serial port prints the numbers in Hex when IR receiver receives the signal of A0
If ir_item=0xFF02F and a=0, the problem under do block will be executed
Light up red LED connected to D3
Set the value of a to 1
If ir_item=0xFF02F and a=1, the problem under do block will be executed
Turn off red LED connected to D3
Set the value of a to 0

Upload code to development board, press “OK” key on remote control to make LED on and off.

Project 7: Bluetooth Remote Control

(1) Description :

Bluetooth, a simple wireless communication module most popular since the last few decades and easy to use are being used in most of the battery-powered devices.



Over the years, there have been many upgrades of Bluetooth standard to keep fulfil the demand of customers and technology according to the need of time and situation.

Over the few years, there are many things changed including data transmission rate, power consumption with wearable and IoT Devices and Security System.

Here we are going to learn about HM-10 BLE 4.0 with Arduino Board. The HM-10 is a readily available Bluetooth 4.0 module. This module is used for establishing wireless data communication. The module is designed by using the Texas Instruments CC2540 or CC2541 Bluetooth low energy (BLE) System on Chip (SoC).

(2) Parameters:

Bluetooth protocol: Bluetooth Specification V4.0 BLE

No byte limit in serial port Transceiving

In open environment, realize 100m ultra-distance communication with iphone4s

Working frequency: 2.4GHz ISM band

Modulation method: GFSK(Gaussian Frequency Shift Keying)

Transmission power: -23dbm, -6dbm, 0dbm, 6dbm, can be modified by AT command.

Sensitivity: ≤-84dBm at 0.1% BER

Transmission rate: Asynchronous: 6K bytes ; Synchronous: 6k Bytes

Security feature: Authentication and encryption

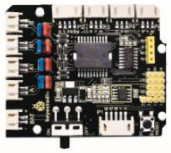
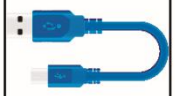
Supporting service: Central & Peripheral UUID FFE0, FFE1

Power consumption: Auto sleep mode, stand by current 400uA~800uA, 8.5mA during transmission.

Power supply: 5V DC

Working temperature: –5 to +65 Centigrade

(3) Equipment :

Control Board *1	L298P Motor Shield *1	4.0 Bluetooth module *1	LED module *1	USB Cable *1	3pin Dupont line *1
					

(4) Hook-up diagram :

1. STATE: state test pins, connected to internal LED, generally keep it unconnected.

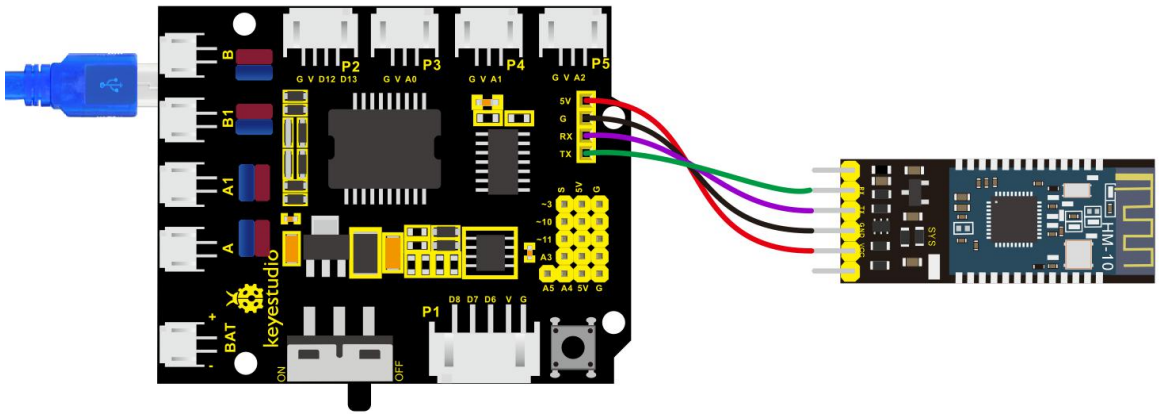
2. RXD: serial interface, receiving terminal.

3. TXD: serial interface, transmitting terminal.

4. GND: Ground.

5. VCC: positive pole of the power source.

6. EN/BRK: break connect, it means breaking the Bluetooth connection, generally, keep it unconnected.



Pay attention to the pin direction when inserting Bluetooth module, and

don't insert it before uploading test code

(5) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

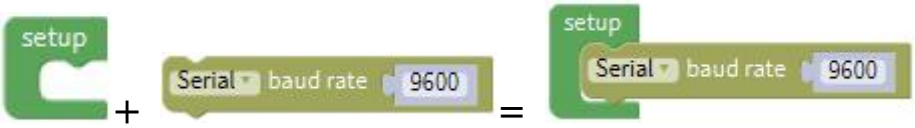
File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_7_Bluetooth Remote Control	lesson_7.1_Bluetooth Remote Control

You could edit code step by step as follows:

(1) Click "Control"to get block 

(2) Enter "Serial Port"to move block 

into 



(3) Click "Variables" to drag out 

into 

, enter "Math" block to get block 

.

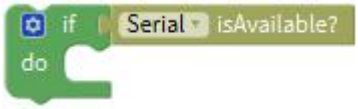
(4) Place it behind "value", and change item into ble_val, click the

drop-down triangle button to select char.

(5) Go to "Control" to get block 

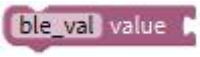
, and click "Serial Port" to move out block 

into if block.

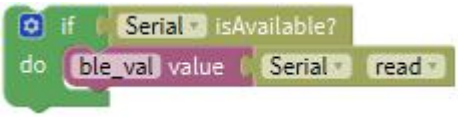


(6) Enter "Variables" to drag out block 

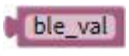
into do block, click "Serial Port" to move out block 

into 

block.



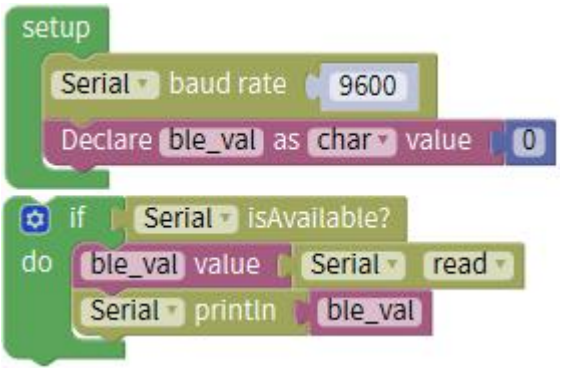
(7) Enter "Serial Port" to move block 

into do block, then click "Variables" to drag out 

into block 

.

Complete Program :



Initialization

Set serial baud rate to 9600

Set to val and assign 0 to it

Set to ble_val and assign value of 0

Serial port waits data from Bluetooth module, if there is data, serial will prints data, otherwise, only 0 is displayed.

Set ble_val

serial port prints variable ble_val

(There will be contradiction between serial communication of code and communication of Bluetooth when uploading code, therefore, don't link with Bluetooth module before uploading code.)

After uploading code on development board, then insert Bluetooth module, wait for the command from cellphone.

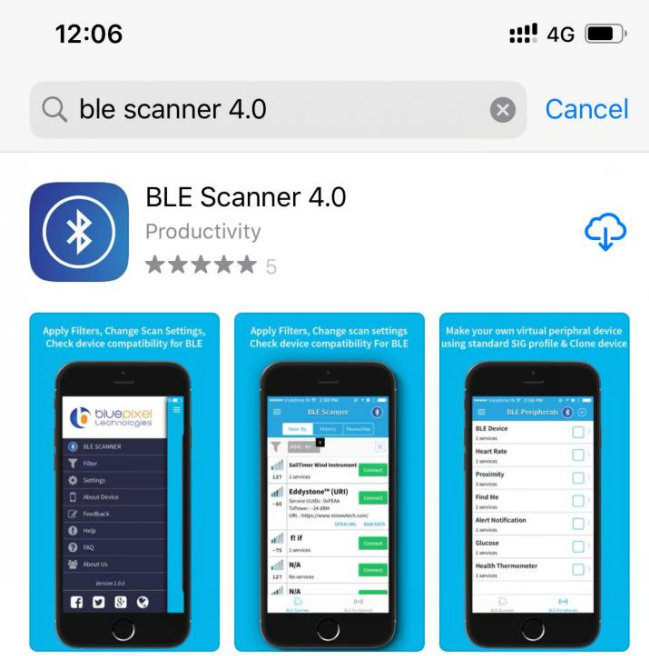
(6) Download APP

The code is the received signal by serial monitor, in this project, we send signal to control robot car via cellphone.

Then we need to download the APP.

1. iOS system

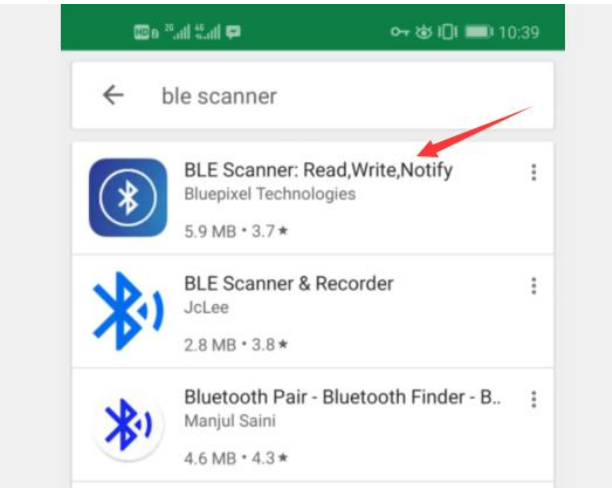
Enter APP STORE to search BLE Scanner 4.0, then download it.



2. Android system

Enter Google Play to find out BLE Scanner, then download.

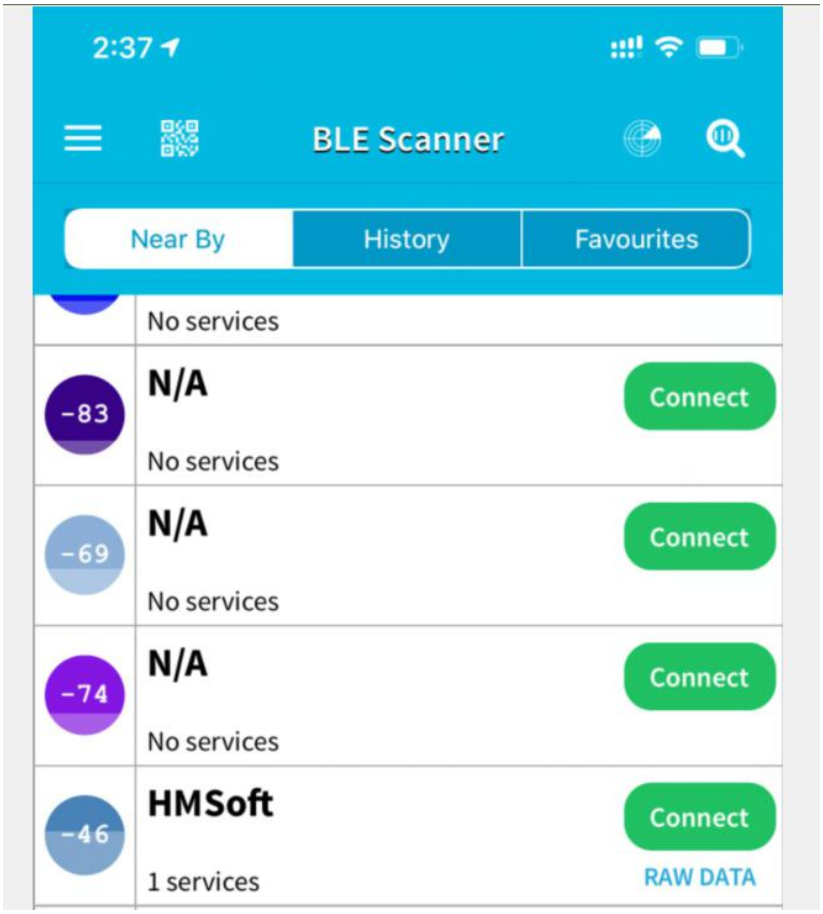
(Enable "location" in settings of your cellphone; otherwise, app may not be searched.)



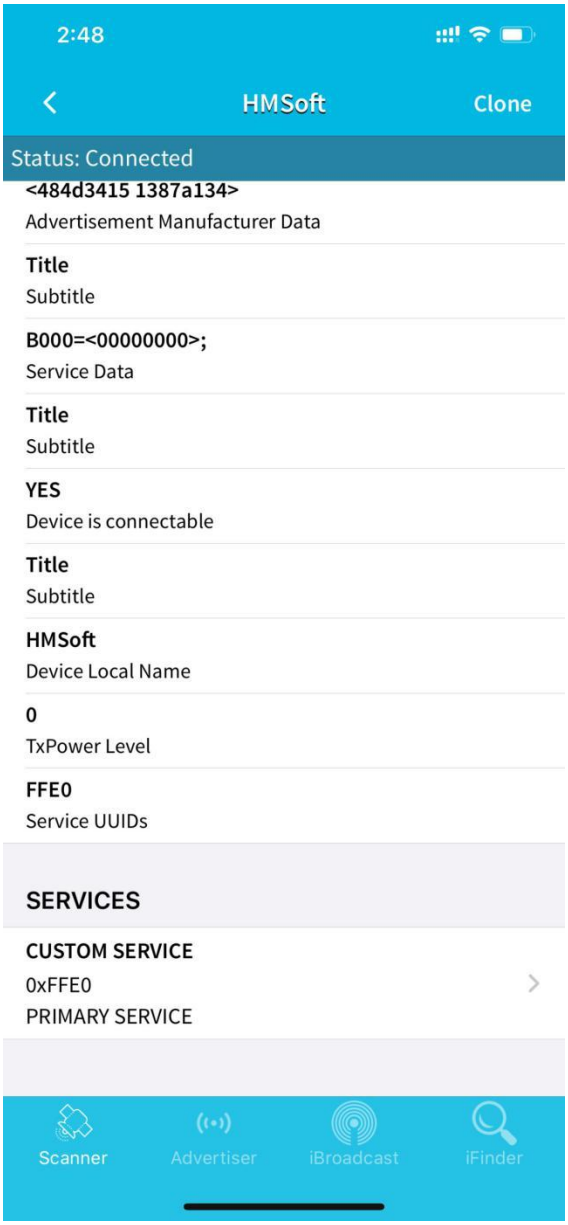
3. After installation, open App and enable "Location and Bluetooth" permission.

4. Open App, the name of Bluetooth module is HMSoft.

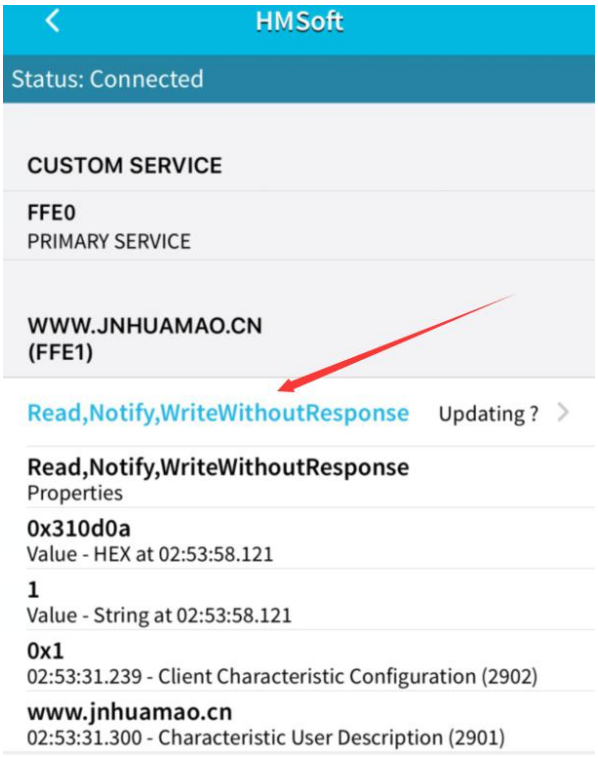
Then click "connect" to link with Bluetooth



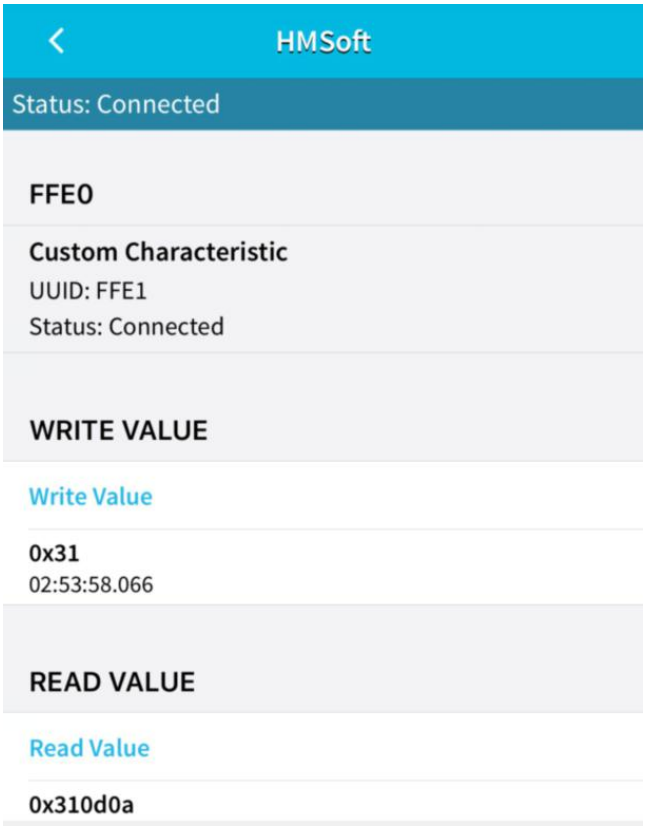
5. After connecting to HMSoft, click it to get multiple options, such as device information, access permission, general and custom service. Choose “CUSTOM SERVICE”



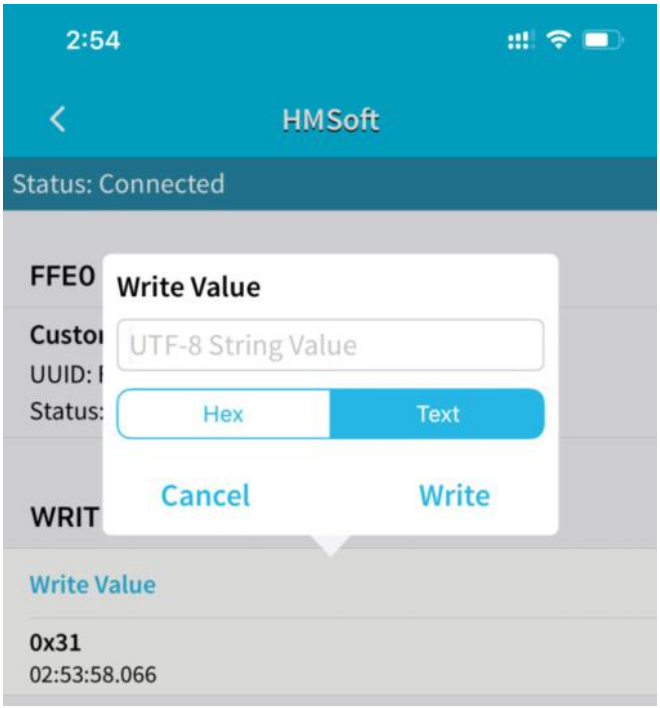
6. Then pop up the following page.



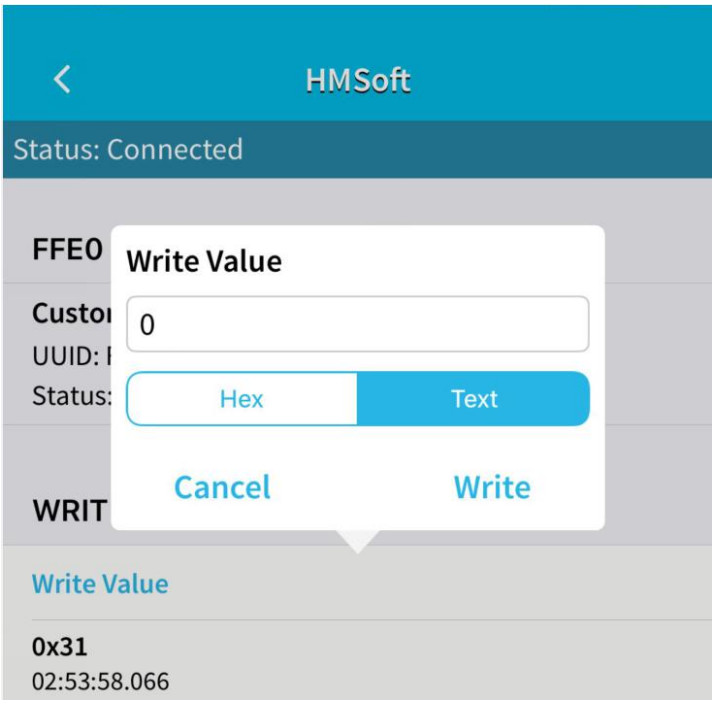
7. Click (Read,Notify,WriteWithoutResponse)to enter the following page



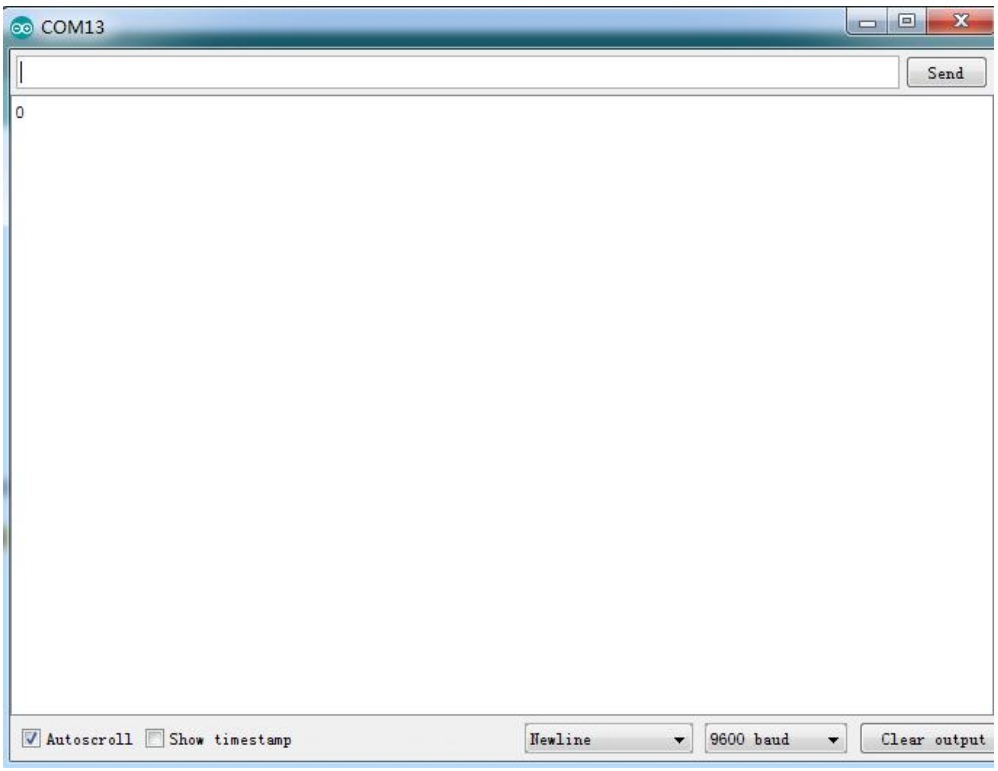
8. Click Write Value to enter HEX or Text.



9. Open the serial monitor on Arduino, enter a 0 or other character on Text interface.



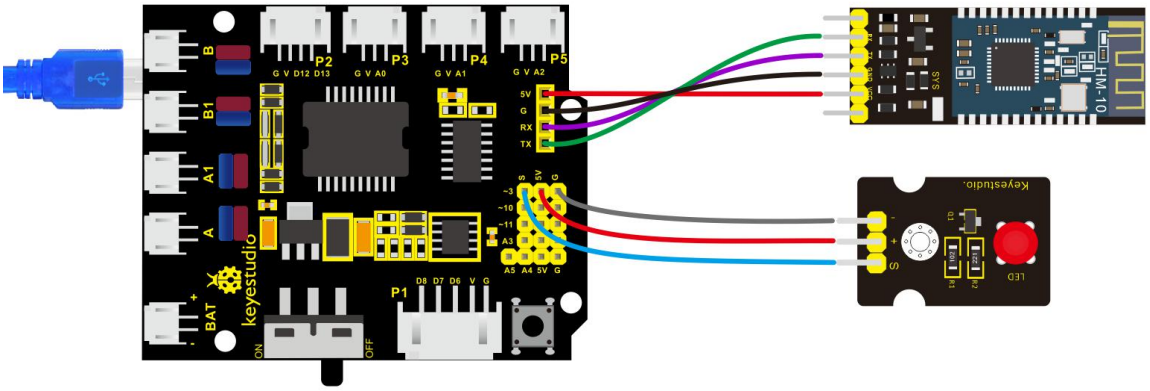
10. Then click "Write", open serial monitor to view if there is a "0" signal



(7) Extension Practice :

We could send a command via Bluetooth to turn on and off a LED.

D3 is connected to a LED, as shown below:



The program will be generated if you find the following file and drag it into

Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_7_ Bluetooth Remote Control	lesson_7.2_ Extension Practice

You could edit code step by step as follows:

(1) Click “Control” to get block 

(2) Enter “Serial Port”to move block 

into 

(3) Click “Variables” to move block 

into 

block,

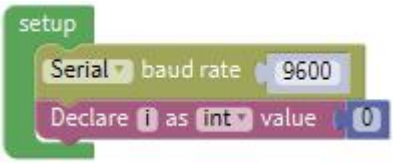
then go to “Math” to find out 

,

(4) Combine 

with 

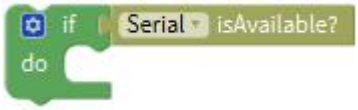
and change item into i.



(5) Click “Control” to get block 

, and enter “Serial port” to move 

into if block.

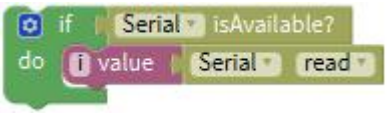


(6) Go to “Variables” to drag block 

into do block,

(7) enter “Serial port” to move out 

block and edit code string as follows:



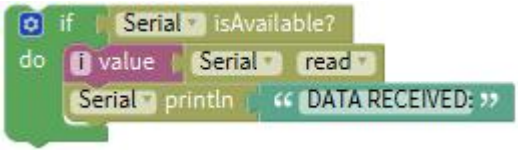
(8) Enter “Serial port” to drag out 

,

(9) Click “Text” to move out 

into 

block, and change “hello” into DATA RECEIVED : .



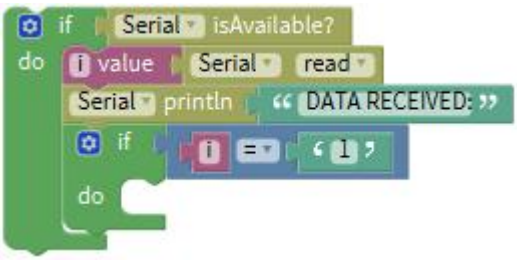
(10) Click “Control” to move out block 

into do, click “Logic” to move 

block into if,

(11) Enter “Variables” to move 
into left box of “=”, then enter “Text” to get block 

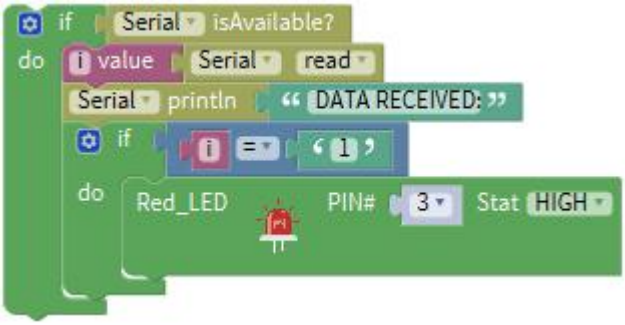
and leave it into right box“=”, change a into 1.



(12) Enter “Sensor”→ “ControlOutput”→ 

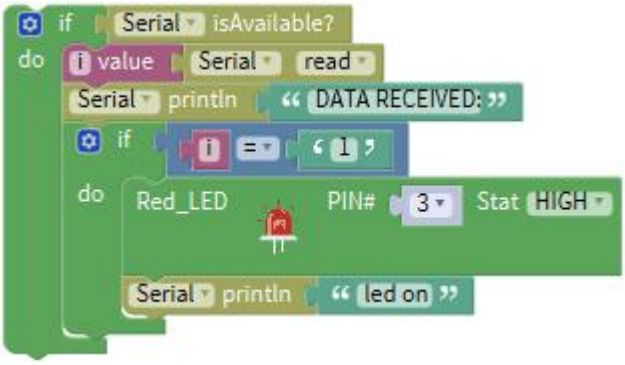
(13) Drag it into the second do block, S end of red LED is connected to D3,

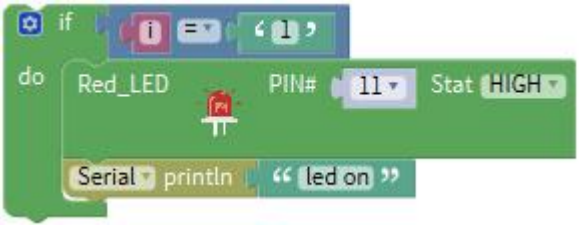
therefore, set to Pin 3.



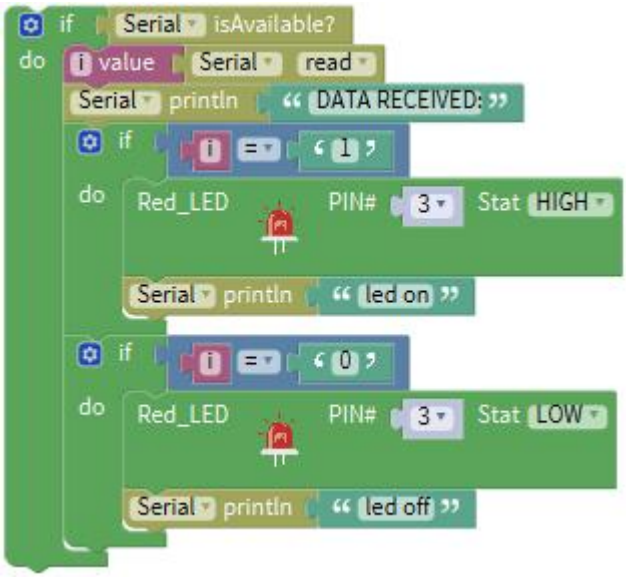
(14) Replicate 

once and place it into the second do block. Change DATA RECEIVED: into led on.



(15) Copy code string 

once and change 1 into 0, click "HIGH" to select "LOW", and change led on into led off.

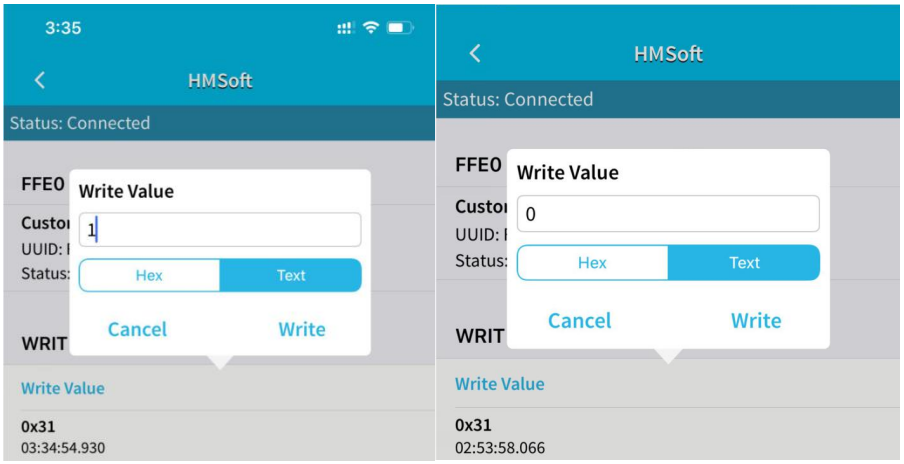


Complete Program :

Initialization
Set baud rate to 9600

Set i to integer and assign 0 to it.

Set the value of i to be read
Serial port prints DATA RECEIVED:
If i= '1', the program under do block will be executed
Light up a red LED connected to D3
Print led on serial port
When i= '0", the program under do block will be executed
Light off a red LED connected to D3
Serial port prints " led off"



Click "Write" on APP, when you enter 1, LED will be on, when you input 0, LED will be off. (Remember to remove the Bluetooth module after finishing experiment; otherwise, burning code will be affected)

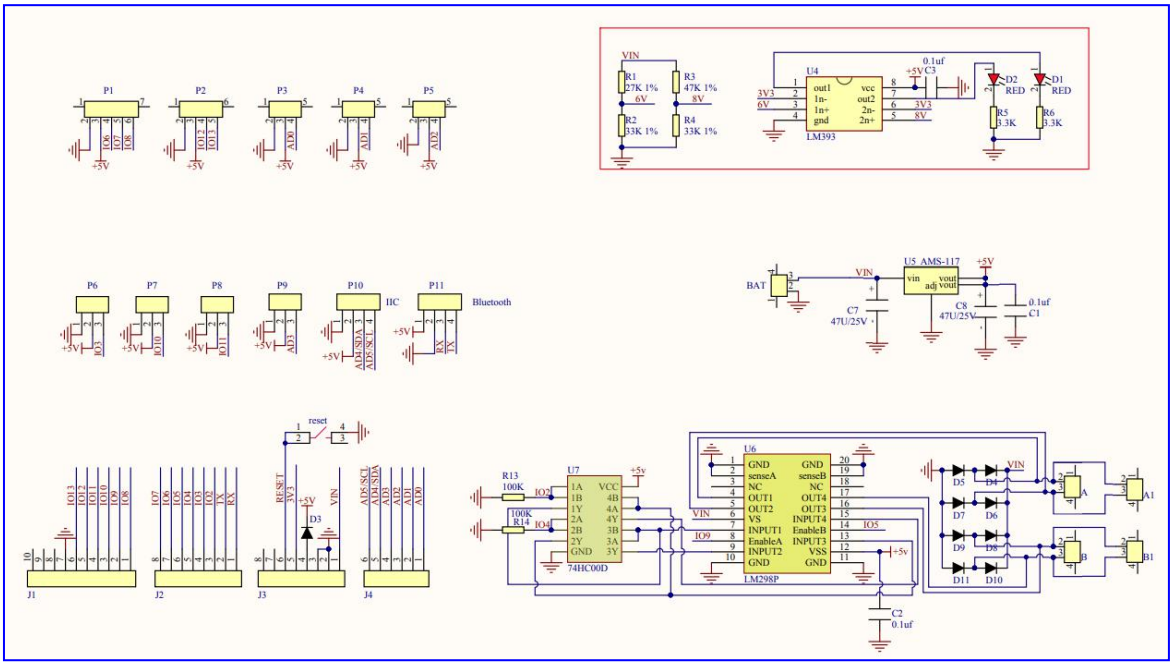
Project 8: Motor Driving and Speed Control

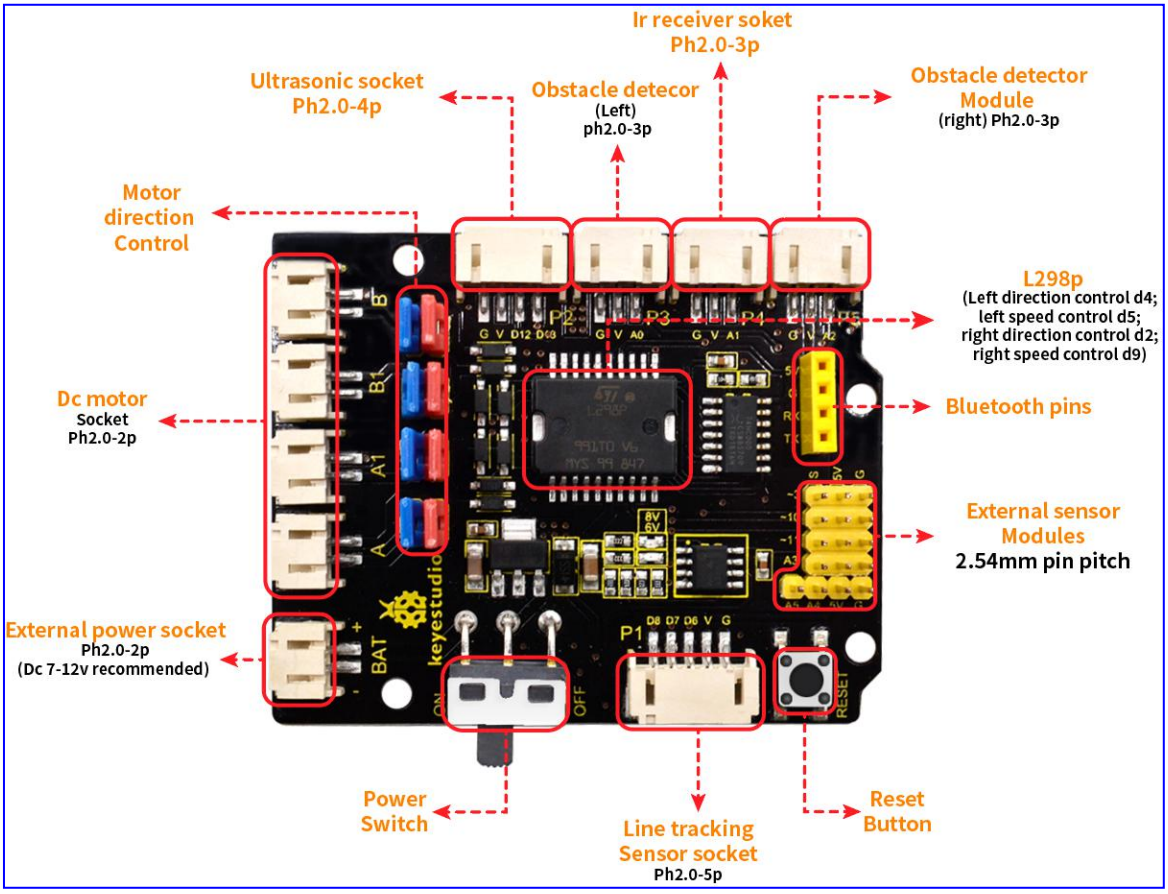
(1) Description

There are many ways to drive the motor. Our robot car uses the most commonly used L298P solution. L298P is an excellent high-power motor driver IC produced by STMicroelectronics. It can directly drive DC motors, two-phase and four-phase stepping motors. The driving current up to 2A, and output terminal of motor adopts eight high-speed Schottky diodes as protection.

We designed a shield based on the circuit of L298p.

The stacked design reduces the technical difficulty of using and driving the motor.





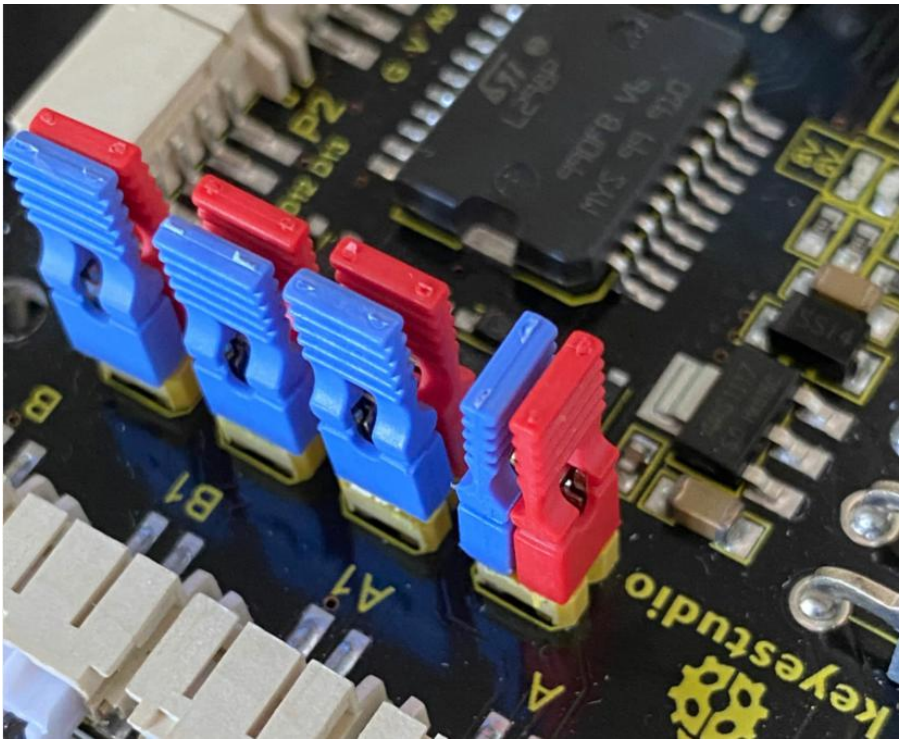
(2) Specification

Circuit Diagram for L298P Board

1) Logic part input voltage: DC5V
2) Driving part input voltage: DC 7-12V
3) Logic part working current: <36mA
4) Driving part working current: <2A
5) Maximum power dissipation: 25W (T=75℃)
6) Working temperature: -25℃～+130℃
7) Control signal input level: high level 2.3V<Vin<5V, low level -0.3V<Vin<1.5V

(3) Drive Robot to Move

The driver of motor driver shield is in parallel connection. You could control the direction of motors by altering the orientation of jumper caps(seen in the picture).

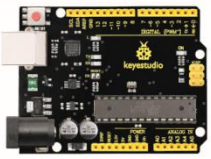
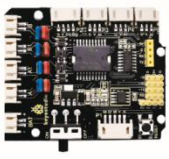
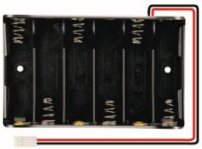


Through the above diagram, the direction pin of B motor is D4, and speed pin is D5; D2 is the direction pin of A motor, D9 is speed pin.

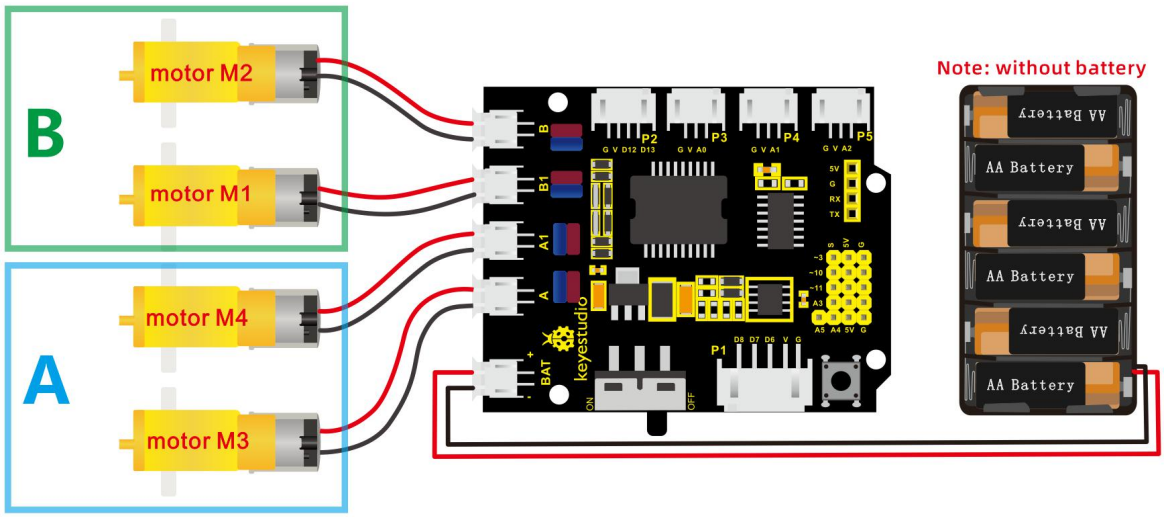
PWM decides 2 motors to rotate so as to drive robot car. The PWM value is in the range of 0-255, the larger the number, the faster the motor rotates

4WD Robot	Motor (A)	Motor (B)
Forward	Rotate clockwise	
Backward	Rotate anticlockwise	
Rotate to left	Rotate anticlockwise	Rotate clockwise
Rotate to right	Rotate clockwise	Rotate anticlockwise
Stop	Stop	Stop

(4) Equipment :

Control Board *1	L298P Motor Shield *1	Motor *4	USB Cable *1	6 AA battery Holder *1
				

(5) Hook-up Diagram :



Attention: connect motors in compliance with the above connection diagram

(6) Test Code :

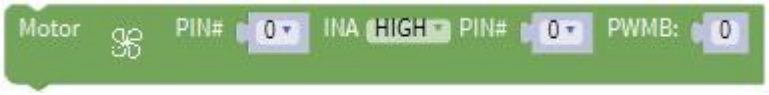
The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_8_Motor Driving and Speed Control	lesson_8.1_Motor Driving and Speed Control

You could edit code step by step as follows:

(1) Click "Control" to get block 

.

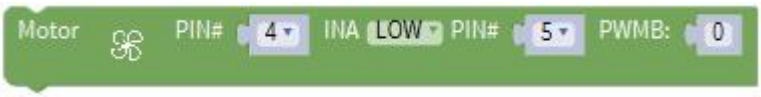
(2) Go to "Module" → "Drive_Module" → 

,

(3) Place it into 

block,

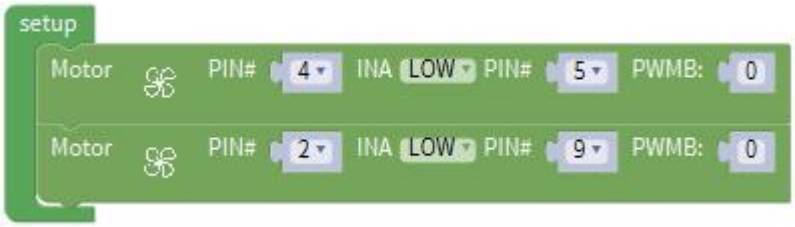
(4) The direction pin and speed control pin of B motor(left) are connected to D4 and D5. So set PIN 4 and PIN 5 as follows:

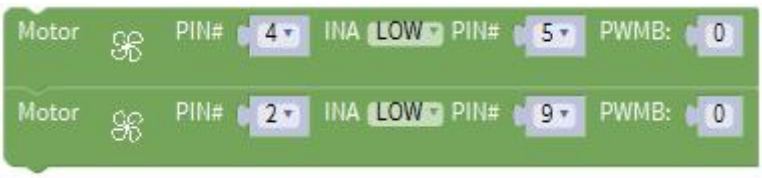
(5) Copy 

once and place it into block 

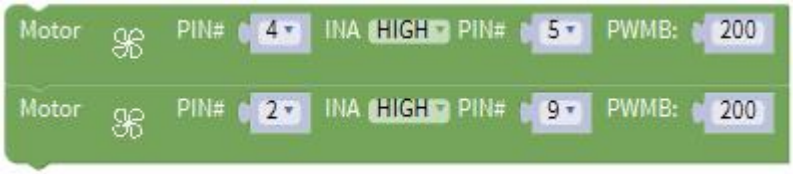
,

(6) The direction pin and speed control pin of A motor (right) are connected to D2 and D9. So set PIN 2 and PIN 9 as follows:



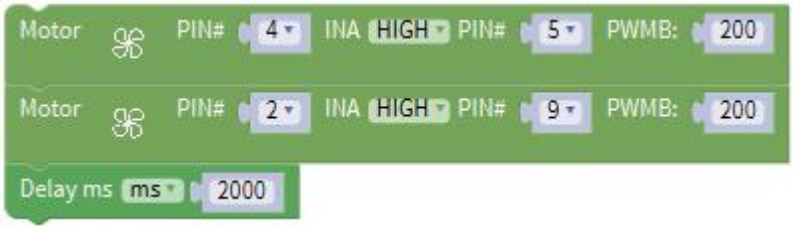
(2) Duplicate code string 

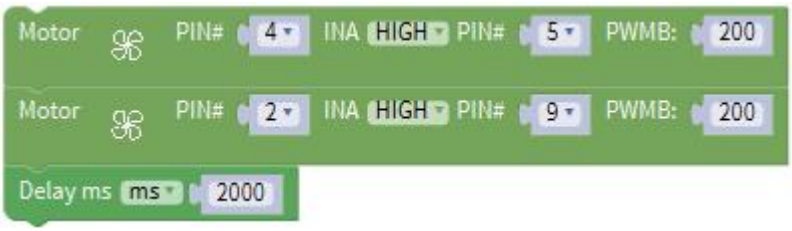
once, set INA to HIGH and PWB to 200 :



(3) Click "Control" to move out block 

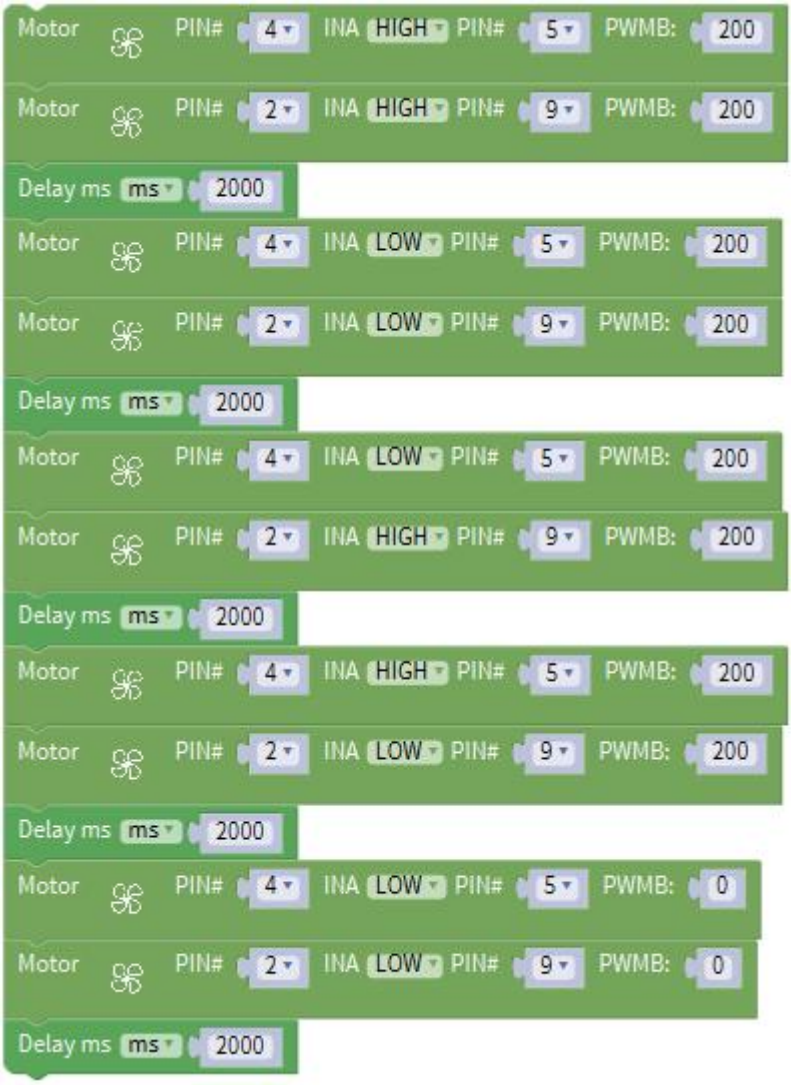
, delay in 2000ms.



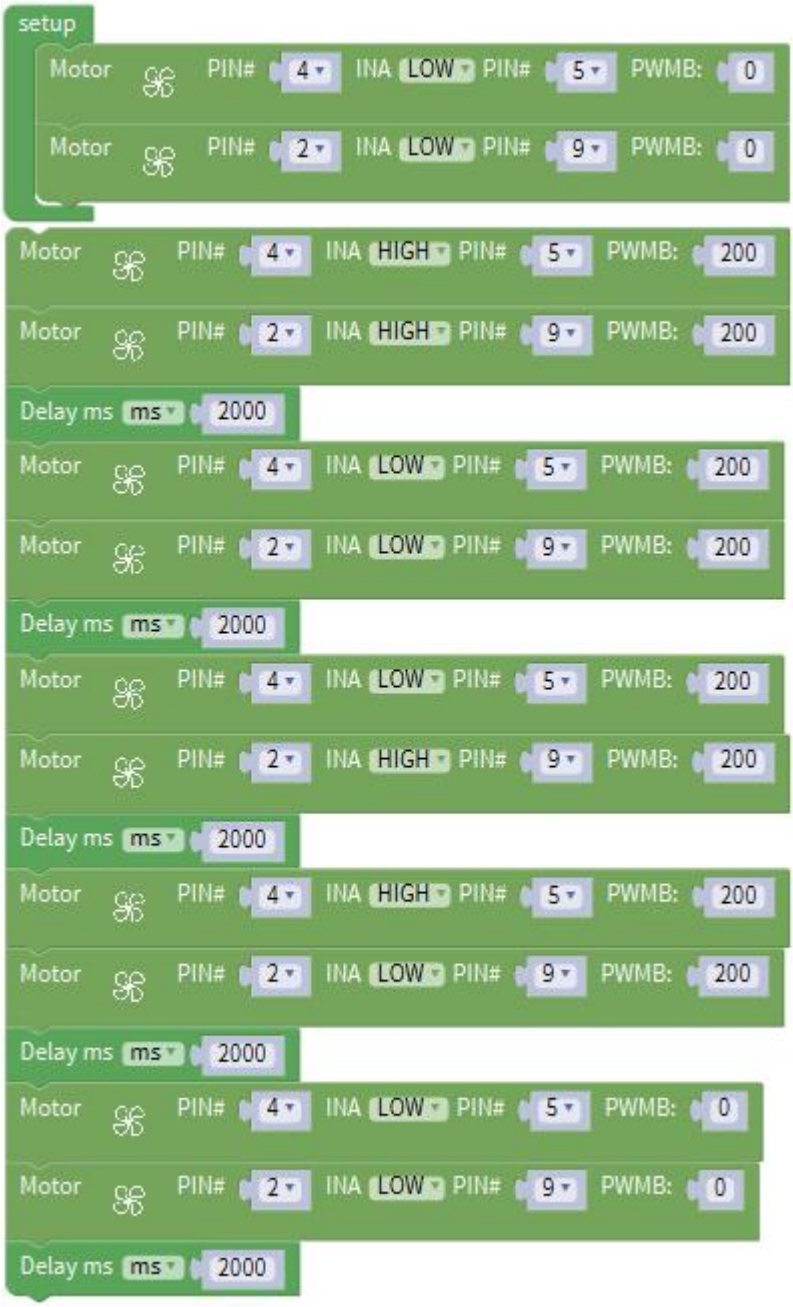
(4) Replicate 

for four times,

then set the code string as follows:



Complete Program :


setup
Motor PIN# 4 INA LOW PIN# 5 PWMB: 0
Motor PIN# 2 INA LOW PIN# 9 PWMB: 0
Motor PIN# 4 INA HIGH PIN# 5 PWMB: 200
Motor PIN# 2 INA HIGH PIN# 9 PWMB: 200
Delay ms ms 2000
Motor PIN# 4 INA LOW PIN# 5 PWMB: 200
Motor PIN# 2 INA LOW PIN# 9 PWMB: 200
Delay ms ms 2000
Motor PIN# 4 INA LOW PIN# 5 PWMB: 200
Motor PIN# 2 INA HIGH PIN# 9 PWMB: 200
Delay ms ms 2000
Motor PIN# 4 INA HIGH PIN# 5 PWMB: 200
Motor PIN# 2 INA LOW PIN# 9 PWMB: 200
Delay ms ms 2000
Motor PIN# 4 INA LOW PIN# 5 PWMB: 0
Motor PIN# 2 INA LOW PIN# 9 PWMB: 0
Delay ms ms 2000

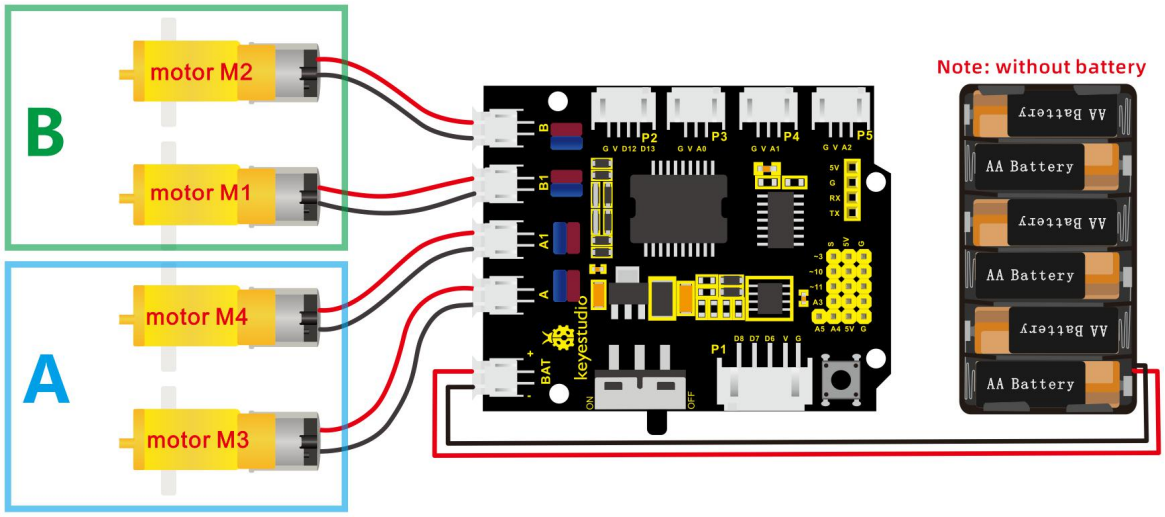
```
Initialization
Set the direction control pin4 of B motor to LOW and PWM control pin5 to 0
Set the direction control pin2 of A motor to LOW and PWM control pin9 to 0

Set the direction control pin4 of B motor to HIGH and PWM control pin5 to 200
Set the direction control pin2 of A motor to HIGH and PWM control pin9 to 200
Delay in 2000ms
Set the direction control pin4 of B motor to LOW and PWM control pin 5to 200
Set the direction control pin2 of A motor to LOW and PWM control pin 9 to 200
Delay in 2000ms

Set the direction control pin4 of B motor to LOW and PWM control pin5 to 200
Set the control pin2 of A motor to HIGH and PWM control pin 9 to 200

Delay in 2000ms
Set the direction control pin4 of B motor to HIGH and PWM control pin 5 to 200
Set the direction control pin2 of A motor to LOW and PWM control pin9 to 200


Delay in 2000ms
Set the direction control pin4 of B motor to LOW and PWM control pin 5 to 0
Set the direction control pin2 of A motor to LOW and PWM control pin 9 to 0
Delay in 2000ms
```

(7) Test Result:

Hook up by connection diagram, upload code and power on, smart car goes forward and back for 2s, turns left and right for 2s, stops for 2s and alternately.

(8) Extension Practice :

Adjust the speed that PWM controls the motor, hook up in same way.

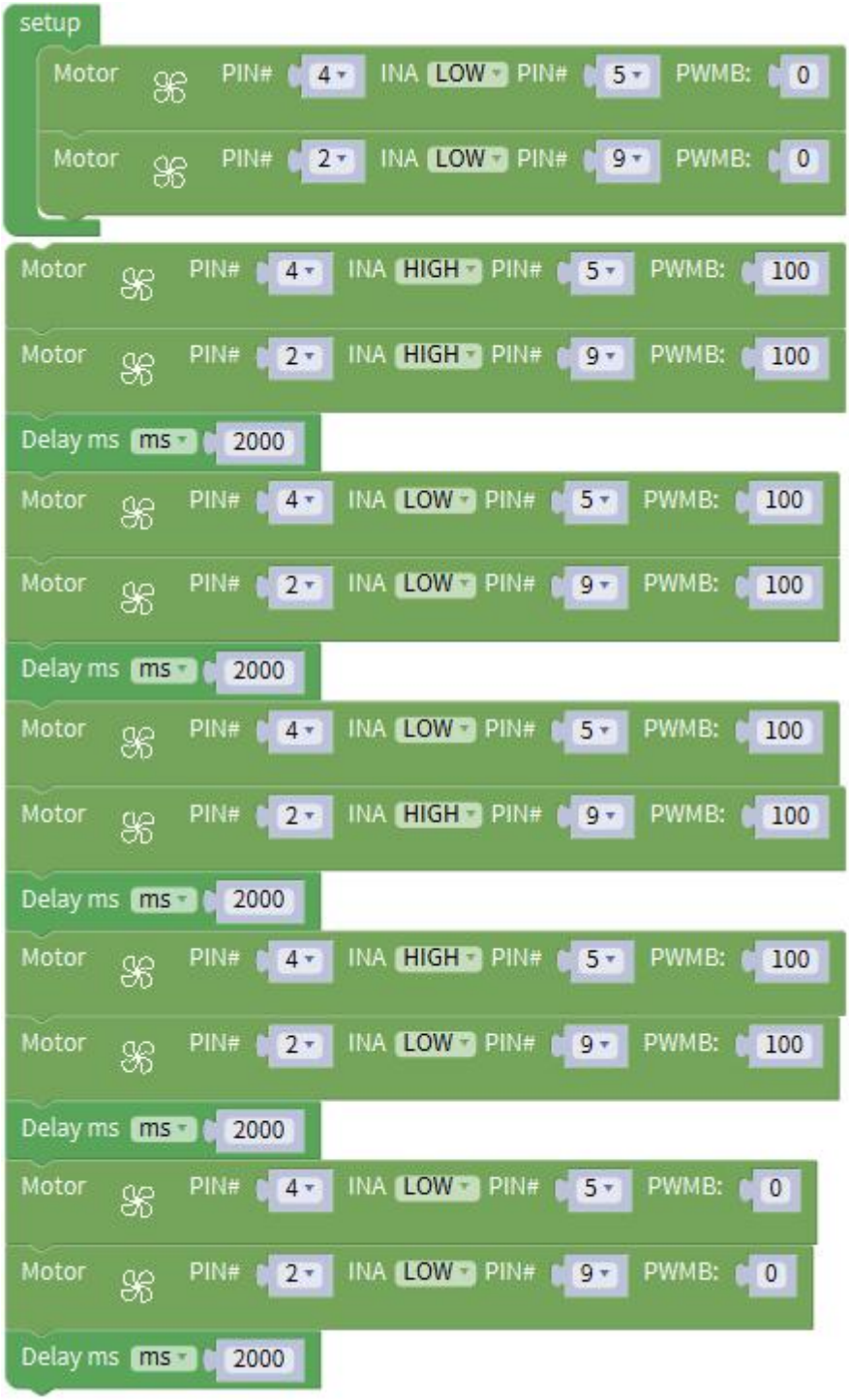


The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_8_Motor Driving and Speed Control	lesson_8.2_Extension Practice

You could edit code step by step.

Complete Program :


setup
Motor PIN# 4 INA LOW PIN# 5 PWMB: 0
Motor PIN# 2 INA LOW PIN# 9 PWMB: 0
Motor PIN# 4 INA HIGH PIN# 5 PWMB: 100
Motor PIN# 2 INA HIGH PIN# 9 PWMB: 100
Delay ms ms 2000
Motor PIN# 4 INA LOW PIN# 5 PWMB: 100
Motor PIN# 2 INA LOW PIN# 9 PWMB: 100
Delay ms ms 2000
Motor PIN# 4 INA LOW PIN# 5 PWMB: 100
Motor PIN# 2 INA HIGH PIN# 9 PWMB: 100
Delay ms ms 2000
Motor PIN# 4 INA HIGH PIN# 5 PWMB: 100
Motor PIN# 2 INA LOW PIN# 9 PWMB: 100
Delay ms ms 2000
Motor PIN# 4 INA LOW PIN# 5 PWMB: 0
Motor PIN# 2 INA LOW PIN# 9 PWMB: 0
Delay ms ms 2000

Initialization
Set the direction control pin4 of B motor to LOW and PWM control pin5 to 0
Set the control pin2 of A motor to LOW and PWM control pin 9 to 0

Set the direction control pin4 of B motor to HIGH and PWM control pin5 to 100
Set the control pin2 of A motor to HIGH and PWM control pin 9 to 100
Delay in 2000ms

Set the direction control pin4 of B motor to LOW and PWM control pin5 to 100
Set the control pin2 of A motor to LOW and PWM control pin 9 to 100
Delay in 2000ms

Set the direction control pin4 of B motor to LOW and PWM control pin5 to 100
Set the control pin2 of A motor to HIGH and PWM control pin 9 to 100
Delay in 2000ms

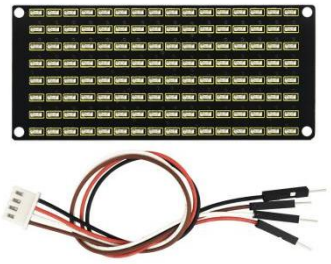
Set the direction control pin4 of B motor to HIGH and PWM control pin5 to 100
Set the control pin2 of A motor to LOW and PWM control pin 9 to 100
Delay in 2000ms

Set the direction control pin4 of B motor to LOW and PWM control pin5 to 0
Set the control pin2 of A motor to LOW and PWM control pin 9 to 0
Delay in 2000ms

Upload code successfully, the motors rotate slower.

Project 9: 8*16 LED Board

(1) Description



If we add a 8*16 LED board to the robot, it will be amazing. Keyestudio's 8*16 dot matrix can meet your requirements. You can create facial emoticons, patterns or other interesting displays yourself. 8*16 LED light board comes with 128 LEDs. The data of the microprocessor (arduino) communicates with the AiP1640 through the two-wire bus interface, so as to control the 128 LEDs on the module, which produce the patterns you need on dot matrix. To facilitate wiring, we also provide a HX-2.54 4Pin wiring.

(2) Specification

Working voltage: DC 3.3-5V

Power loss: 400mW

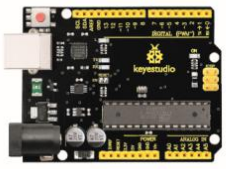
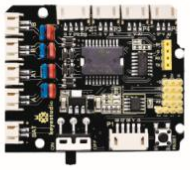
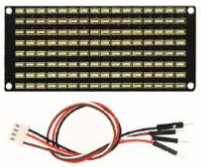
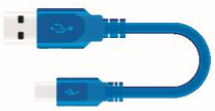
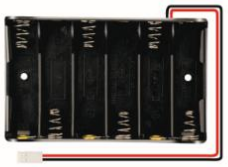
Oscillation frequency: 450KHz

Drive current: 200mA

Working temperature: -40~80℃

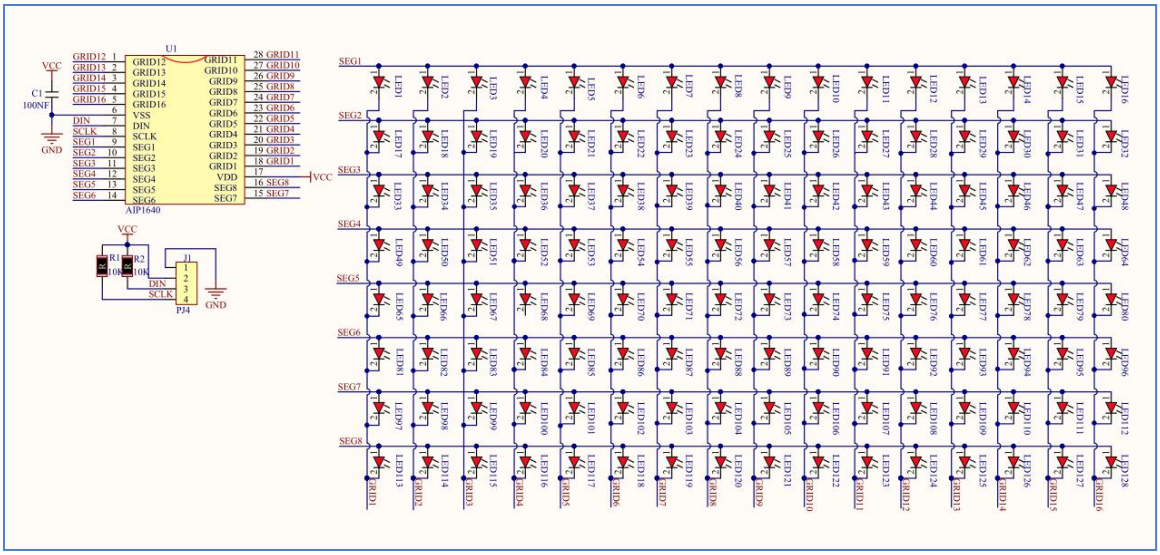
Communication method: two-wire bus

(3) Equipment :

Control Board *1	L298P Motor Shield *1	8x16 LED panel board *1	USB Cable *1	6 AA battery Holder *1
				

(4) 8*16 Dot Matrix Display

Circuit Graph :



The principle of 8*16 dot matrix:

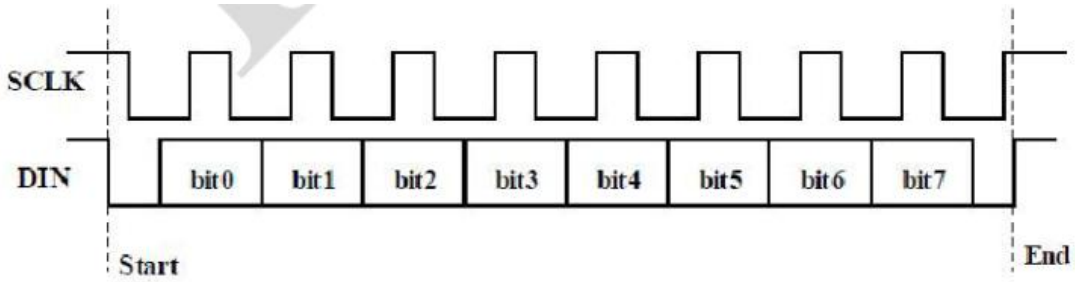
How to control each led light of 8*16 dot matrix? We know that a byte has 8 bits, each bit is 0 or 1. When a bit is 0, turn off LED and when a bit is 0, turn on LED. Thereby, one byte can control the LED in a row of dot matrix, so 16 bytes can control 16 columns of led lights, that is, 8*16 dot matrix.

Interface Description and Communication Protocol:

The data of the microprocessor (arduino) communicates with the AiP1640 through the two-wire bus interface.

The communication protocol diagram is shown below:

(SCLK) is SCL, (DIN) is SDA:



①The starting condition for data input: SCL is high level and SDA changes from high to low.

②For data command setting, there are methods as shown in the figure below

In our sample program, select the way to **add 1 to the address automatically**, the binary value is 0100 0000 and the corresponding hexadecimal value is 0x40

B7	B6	B5	B4	B3	B2	B1	B0	Description
0	1	Irrelevant choice, fill in 0			0	Irrelevant choice, fill in 0		add 1 to the address automatically
0	1				1			Fixed address Universal mode
0	1			0				
0	1			1				Test mode

③For address command setting, the address can be selected as shown below.

The first 00H is selected in our sample program, and the binary number 1100 0000 corresponds to the hexadecimal 0xc0

B7	B6	B5	B4	B3	B2	B1	B0	Display address
1	1	Irrelevant choice, fill in 0		0	0	0	0	00H
1	1			0	0	0	1	01H
1	1			0	0	1	0	02H
1	1			0	0	1	1	03H
1	1			0	1	0	0	04H
1	1			0	1	0	1	05H
1	1			0	1	1	0	06H
1	1			0	1	1	1	07H
1	1			1	0	0	0	08H
1	1			1	0	0	1	09H
1	1			1	0	1	0	0AH
1	1			1	0	1	1	0BH
1	1			1	1	0	0	0CH
1	1			1	1	0	1	0DH
1	1			1	1	1	0	0EH
1	1			1	1	1	1	0FH

④The requirement for data input is that SCL is high level when inputting data, the signal on SDA must remain unchanged. Only when the clock

signal on SCL is low level, the signal on SDA can be altered. The data input is low-order first, high-order is behind

⑤ The condition to end data transmission is that when SCL is low, SDA is low, and when SCL is high, the SDA level also becomes high.

⑥ Display control, set different pulse width, the pulse width can be selected as shown below

In the example, we choose pulse width 4/16, and the hexadecimal corresponds to 1000 1010 is 0x8A

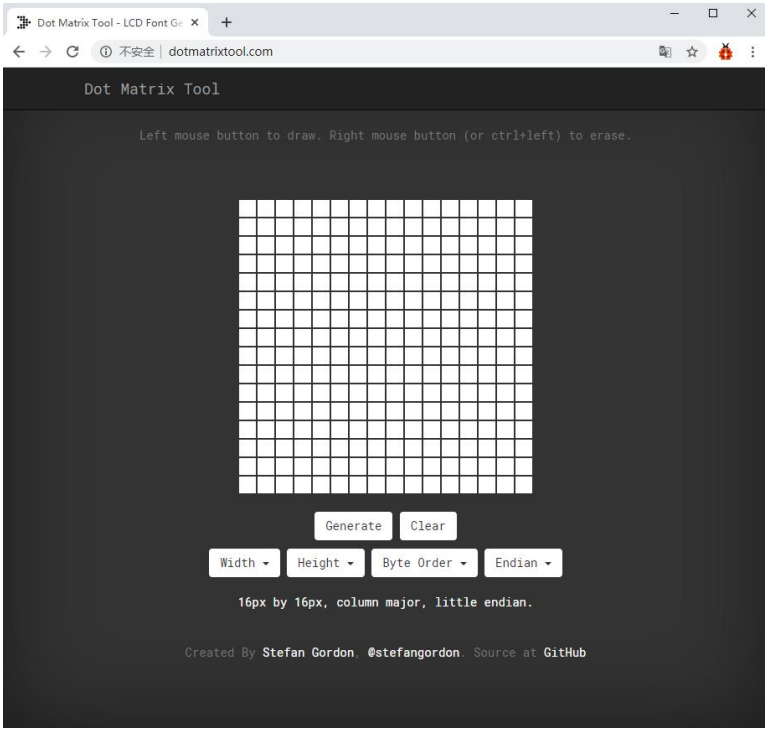
B7	B6	B5	B4	B3	B2	B1	B0	Function	Description
1	0	Irrelevant choice, fill in 0		1	0	0	0	Clear quantity setting (Brightness setting)	Set pulse width to 1/16
1	0			1	0	0	1		Set pulse width to 2/16
1	0			1	0	1	0		Set pulse width to 4/16
1	0			1	0	1	1		Set pulse width to 10/16
1	0			1	1	0	0		Set pulse width to 11/16
1	0			1	1	0	1		Set pulse width to 12/16
1	0			1	1	1	0		Set pulse width to 13/16
1	0			1	1	1	1		Set pulse width to 14/16
1	0			0	X	X	X	Display switch setting	On
1	0			1	X	X	X		off

4. Introduction for Modulus Tool

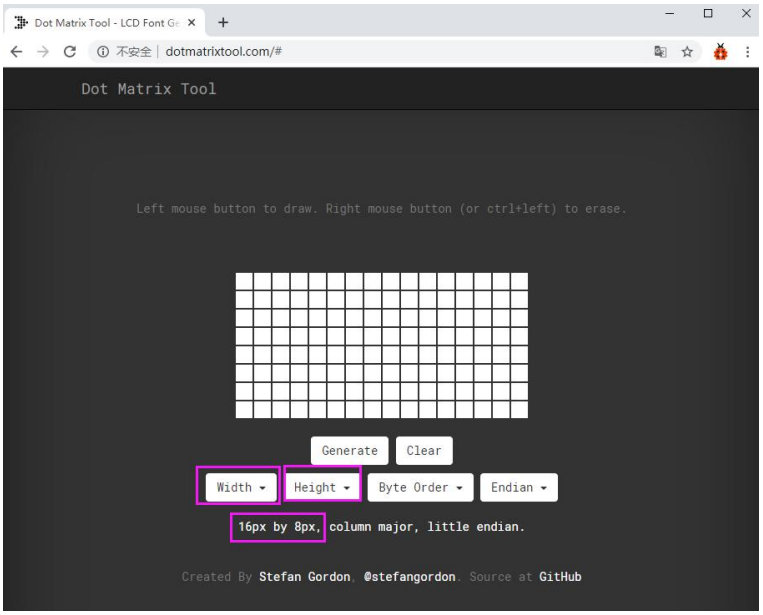
The online version of dot matrix modulus tool:

http://dotmatrixtool.com/#

①Open links to enter the following page.

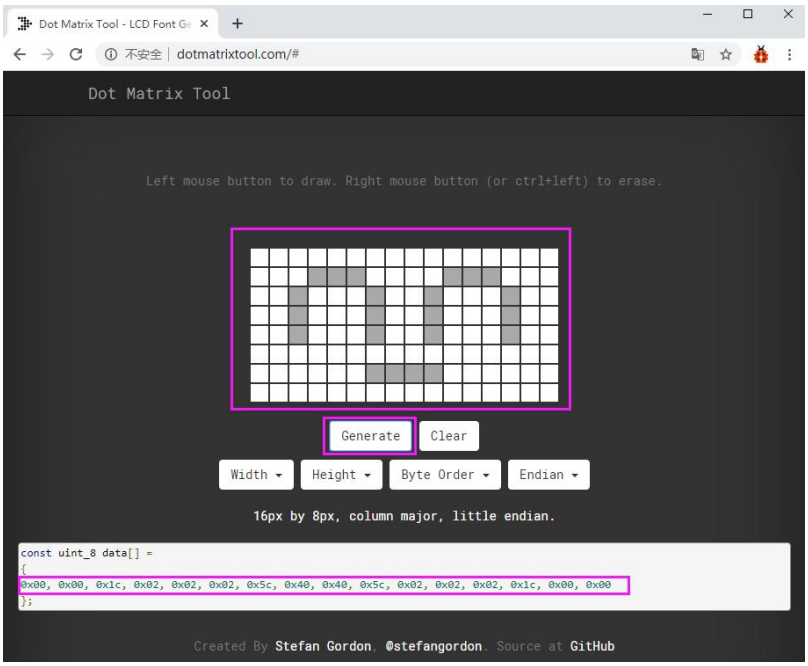


②The dot matrix is 8*16 in this project, so set the height to 8, width to 16, as shown below.

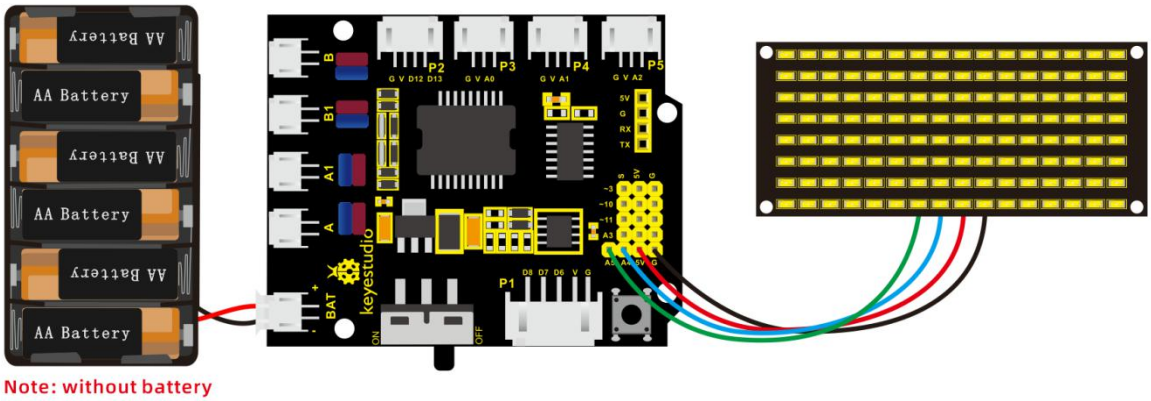


③ Generate hexadecimal data from the pattern

As shown below, press the left mouse button to select, the right button to cancel, draw the pattern you want, click **Generate**, and the hexadecimal data we need will be produced.



(5) Connection Diagram



Wiring note: The GND, VCC, SDA, and SCL of the 8*16 LED panel are respectively connected to -(GND), + (VCC), A4 and A5 of the keyestudio

sensor expansion board for two-wire serial communication. (Note: This pin is connected to arduino IIC, but this module is not IIC communication, it can be linked with any two pins.)

(6) Test Code

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_8*16 LED Board	lesson_9.1_8*16 LED Board

You could edit code step by step as follows:

(1) Click “Control” to get block 

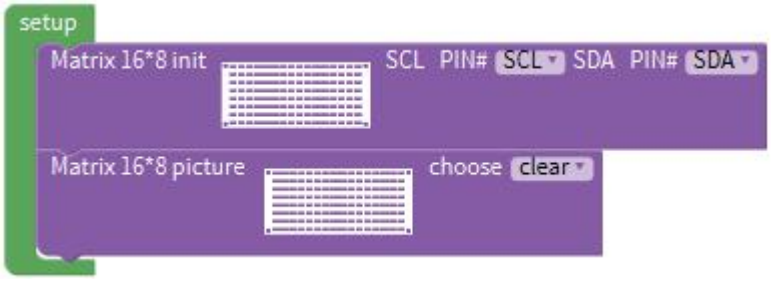
(2)Enter “ Module ” → “ Display_Module ” →

(3) Leave it into block 

(4) Go to “ Display_Module ” to get block 

and leave it into block 

, and click drop-down triangle button to set “clear”.



(5) Move out block 

, and set “∩_∩”.

Complete Program :

Initialization

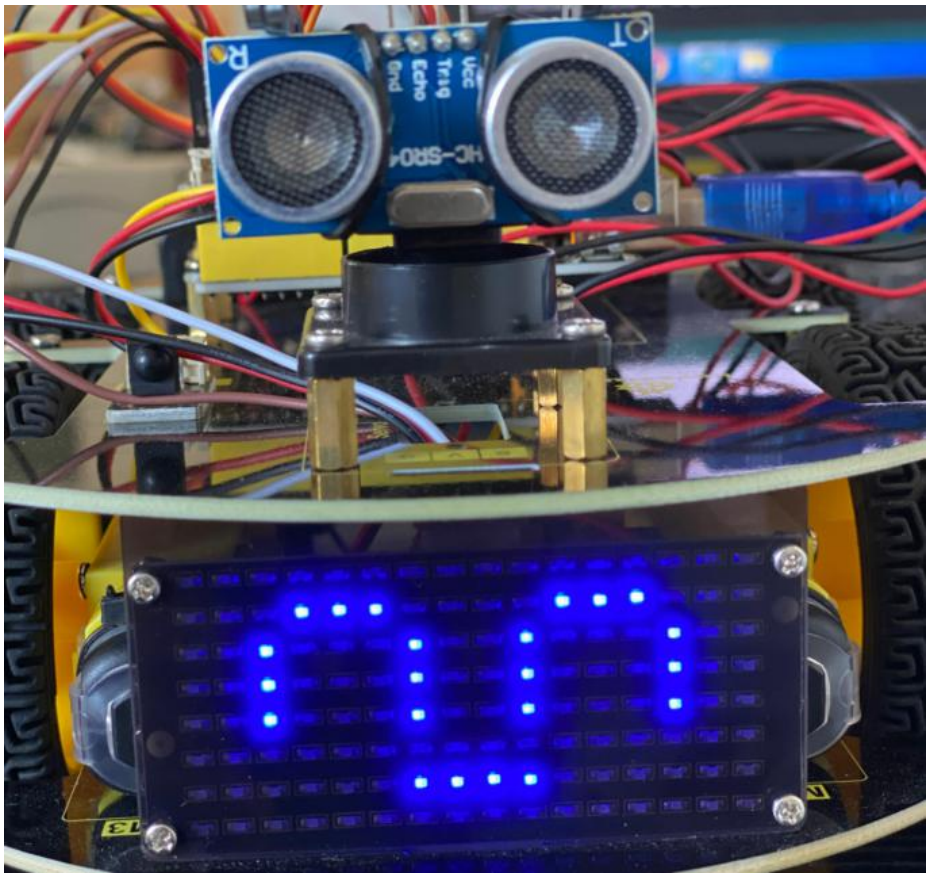
Set to SCL and SDA, SCL corresponds to A5, SDA to A4.

Clear screen

8x16 dot matrix shows ∩_∩pattern

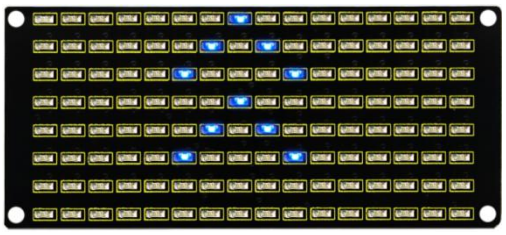
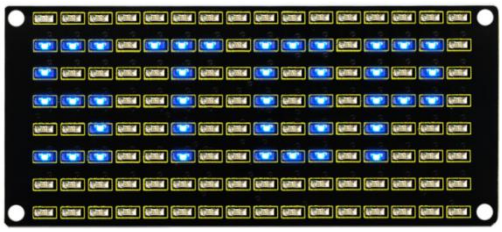
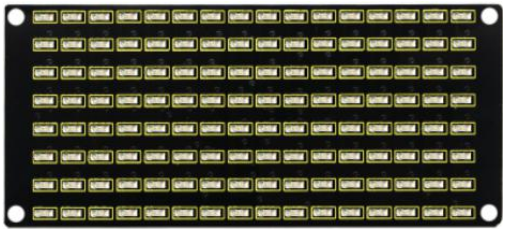
(7) Test Result :

Wire according to connection diagram. The DIP switch is dialed to right end and power on, the smile face appears on dot matrix.



(8) Extension Practice:

We use the modulo tool (http://dotmatrixtool.com/#)to make the dot matrix alternately display start, forward and stop patterns then clear the patterns. The time interval is 2000 milliseconds.

Get the graphical code to be displayed via modulus tool

Start :

Go front :

Go back :

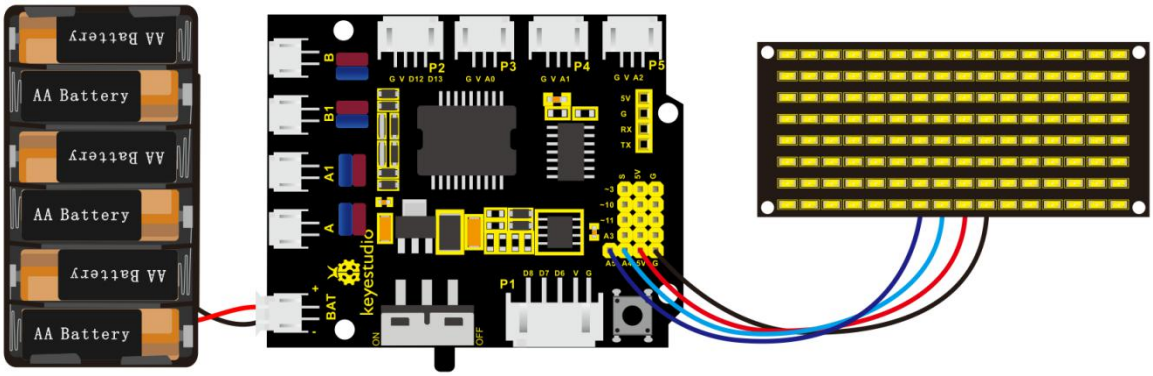
Turn left :

Turn right :

Stop :

Clear:

Hook-up Diagram :



The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_9_8×16 LED Board	lesson_9.2_Extension Practice

You could edit code step by step as follows:

You could edit code step by step as follows:

(1) Click “Control” to get block 

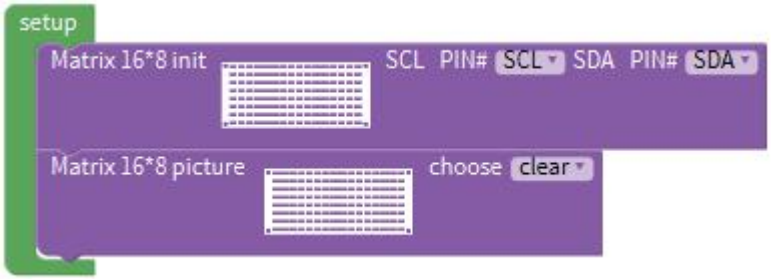
(2) Enter “ Module ” → "Display_Module" to move block 

into 

(3) Then drag block 

into 
.

(4) Click the drop-down triangle button behind “↑”to select “clear”.

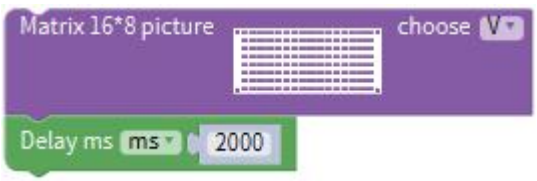


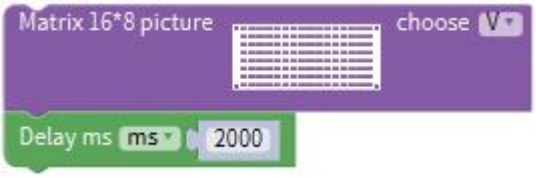
(5) Copy 

again, click the drop-down triangle button behind “↑” to set “V”.

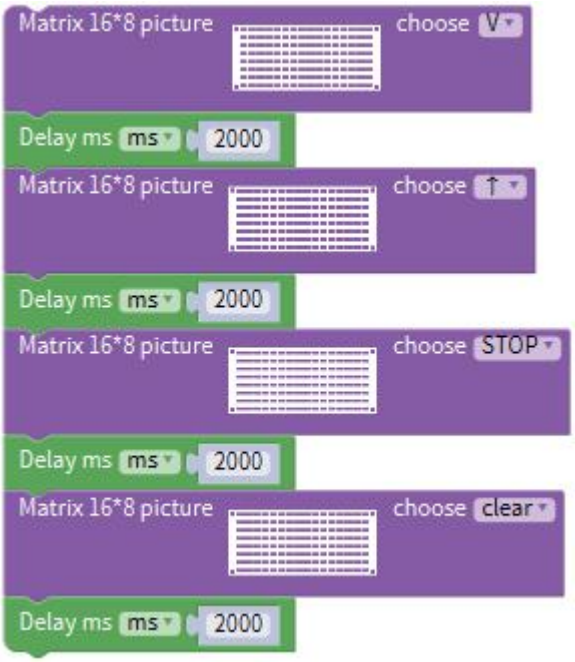
(6) Click “Control” to get block 

, delay in 2000ms.

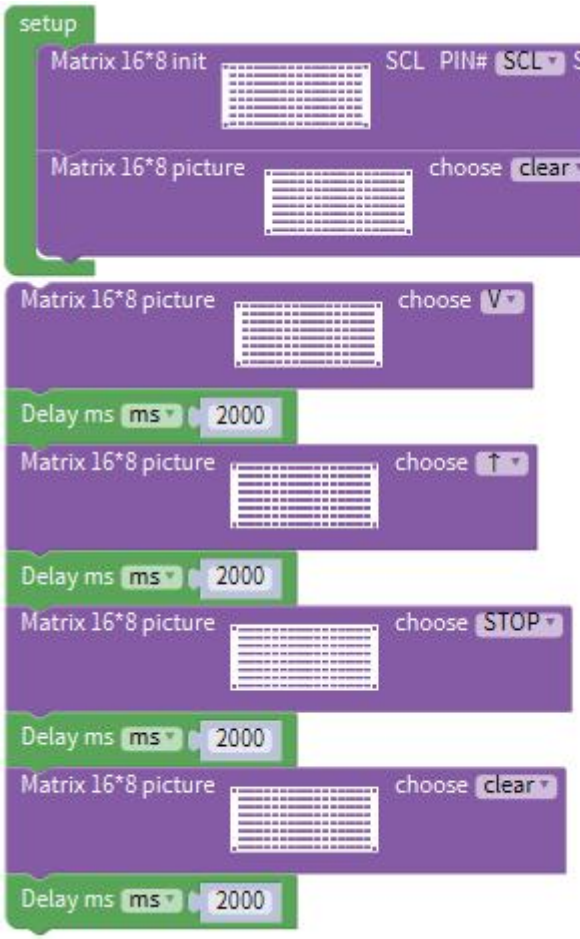


(7) Replicate code string 

for three times, separately click the triangle button to set "↑", "STOP" and "clear".



Complete Program :


setup
Matrix 16*8 init
SCL PIN# SCL SDA PIN# SDA
Matrix 16*8 picture
choose clear
Matrix 16*8 picture
choose V
Delay ms ms 2000
Matrix 16*8 picture
choose ↑
Delay ms ms 2000
Matrix 16*8 picture
choose STOP
Delay ms ms 2000
Matrix 16*8 picture
choose clear
Delay ms ms 2000

Initialization

Set to SCL and SDA, SCL corresponds to A5, SDA to A4.

Clear screen

8x16 Dot matrix shows “V”

Delay in 2000ms

8x16 Dot matrix shows “↑”

Delay in 2000ms

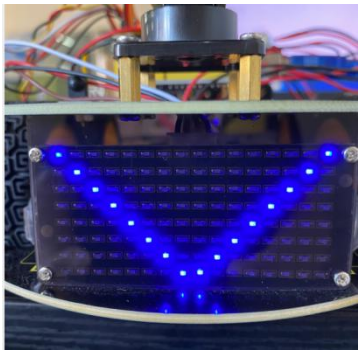
8x16 Dot matrix shows “STOP”

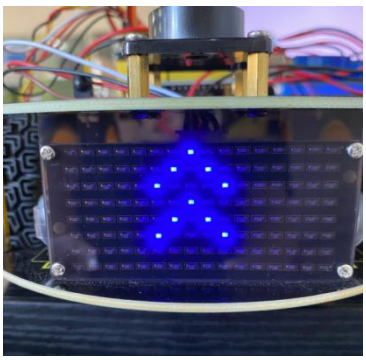
Delay in 2000ms

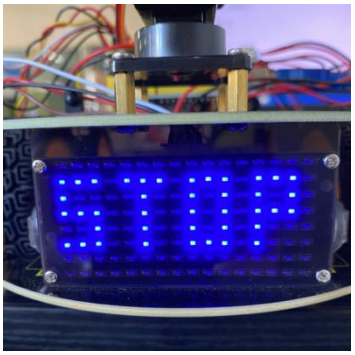
Clear screen

Delay in 2000ms

Upload code on development board, 8*16 dot matrix displays front and stop patterns, alternately.







Project 10: Line Tracking Robot

(1) Description

The previous projects are inclusive of the knowledge of multiple sensors and modules. Next, we will work on a little challenging task.

We could make a line tracking car on top of the working principle of line tracking sensor.

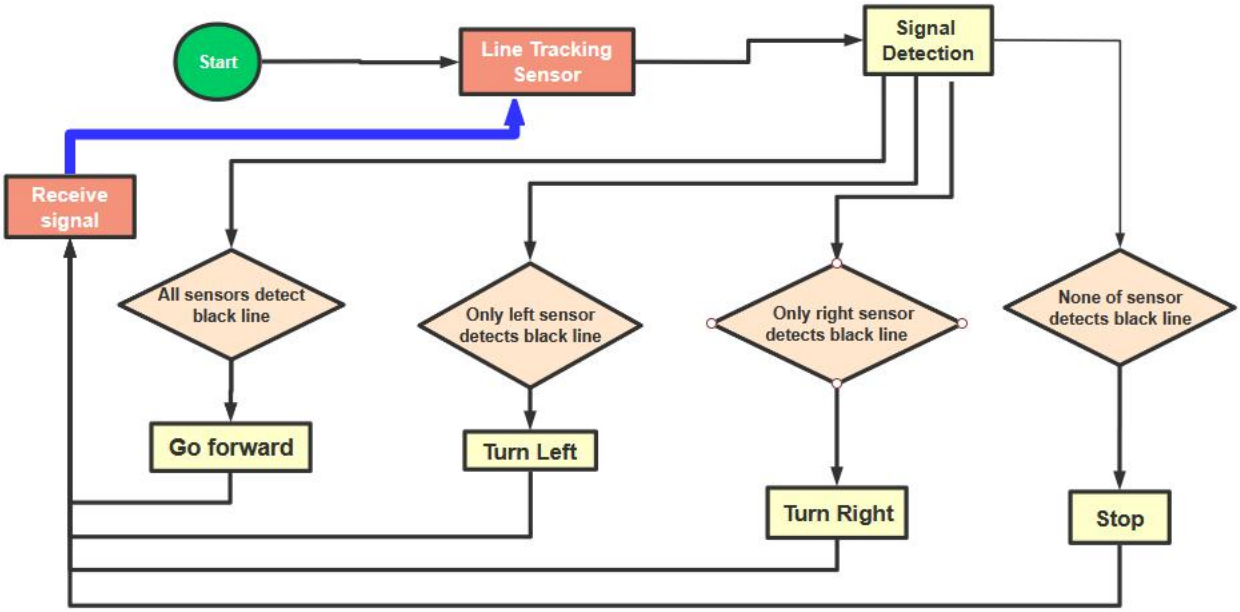
Line tracking robot car:

Detection	Left tracking sensor	detects black line :

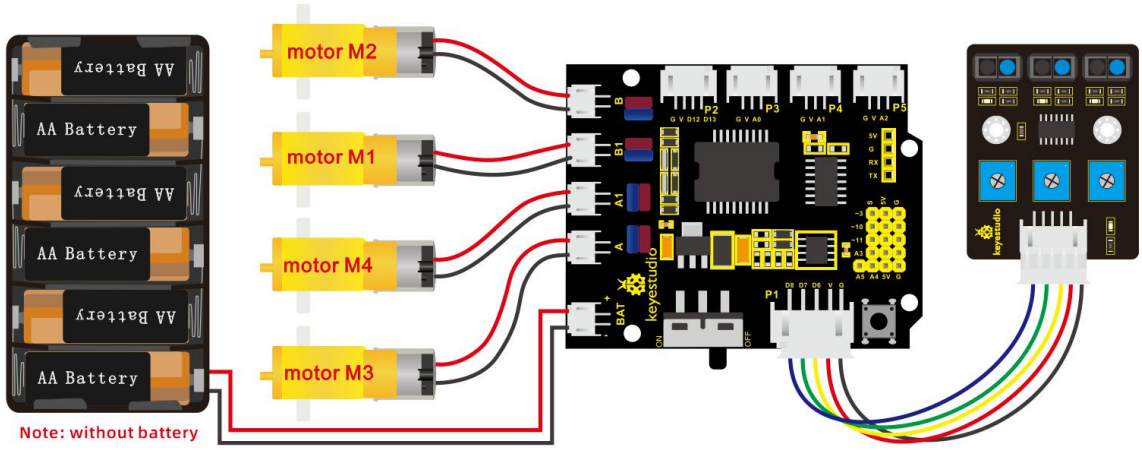
<table>
<tr><td rowspan="6"></td><td rowspan="2"></td><td>HIGH</td></tr>
<tr><td>detects white line : LOW</td></tr>
<tr><td rowspan="2">Middle tracking sensor</td><td>detects black line : HIGH</td></tr>
<tr><td>detects white line : LOW</td></tr>
<tr><td rowspan="2">Right tracking sensor</td><td>detects black line : HIGH</td></tr>
<tr><td>detects white line : LOW</td></tr>
<tr><td>Condition 1</td><td colspan="2">Status</td></tr>
<tr><td>Middle tracking sensor detects black line</td><td colspan="2">go front
(PWM set to 70)</td></tr>
<tr><td rowspan="4">Middle tracking sensor detects white line</td><td colspan="2">Status</td></tr>
<tr><td colspan="2">detecting the left and the right tracking sensor</td></tr>
<tr><td>Condition 2</td><td>Status</td></tr>
<tr><td>left tracking sensor detects black line;</td><td>Rotate to left (set PWM to 200)</td></tr>
</table>

	right sensor detects white line	
	left tracking sensor detects white line; right sensor detects black line	Rotate to right (set PWM to 200)
	left tracking sensor detects black line; right sensor detects black line	stop
	left tracking sensor detects white line; right sensor detects white line	stop

(2) Flow Chart



(3) Connection Diagram



(4) Test Code

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_10_Line Tracking Robot	lesson_10_Line Tracking Robot

You could edit code step by step as follows:

(1) Go to"Control"Module to find out block 

.

(2) Click"Variables"to move out block 

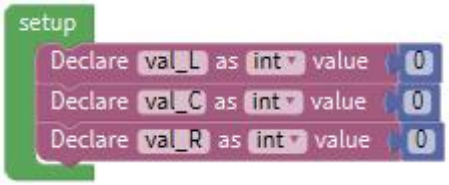
and duplicate it for two times.

(3) Separately change item into val_L, val_C and val_R.

(4) Go to"Math"to move out block 

and copy it twice.

(5) Edit the following code string:



(6) Go to "Variables" to get block 

.

(7) Enter "robot"→"4wd-SmartCar"→ 

.

(8) 

+ 

= 

(9) Replicate 

twice , then change val_L into

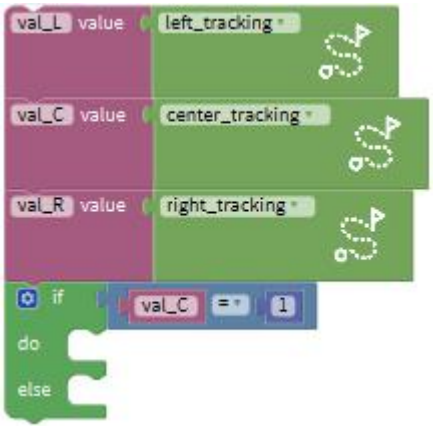
val_C and val_R ; left_tracking into center_tracking and right_tracking.

(10) Click“Control”to get block  , click  , and move into  , then  block turns into  block.

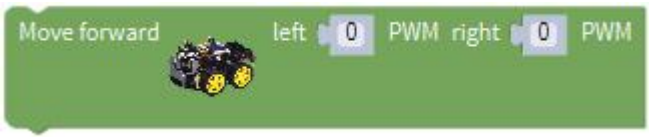
(11) Click“Logic”to move out block  and place it behind if block.

Go to“Variables”to drag out block  into left box of“=”block,

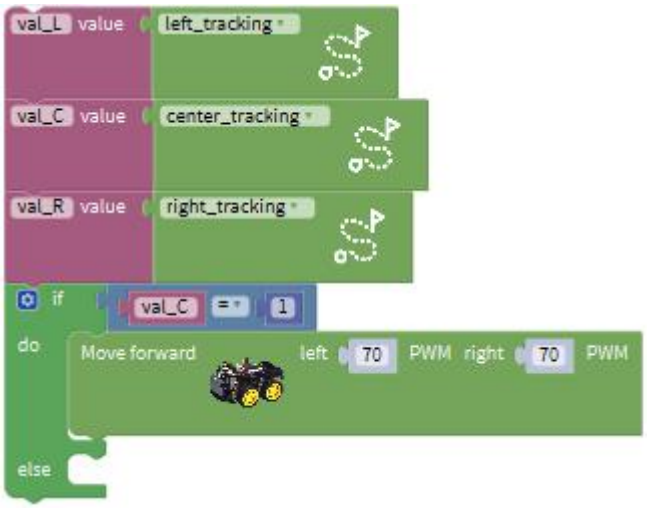
(12) Copy  again and leave it into right box of“=”b lock, and change 0 into 1.



(13) Enter “ robot ” → “ 4wd-SmartCar ” →



(14) Place it into do block and set to 70.



(15) Go to“Control”to move block  into else block, click  , drag block  into  block and move block  into  block , then  block turns into block  .

(16) Enter“Logic”Module to move block  into if block, go to “Logic”to drag block  into left box of “=”.

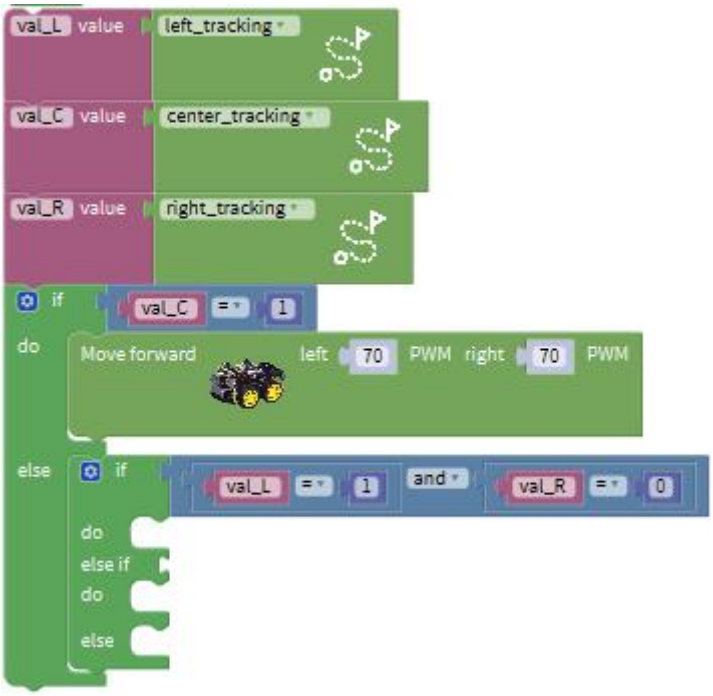
(17) Go to“Variables” to move block  into left box of “=”block,

(18) Enter“Math”to drag block  into right box of“=”block, then change 0 into 1 ;

(19) Replicate block 

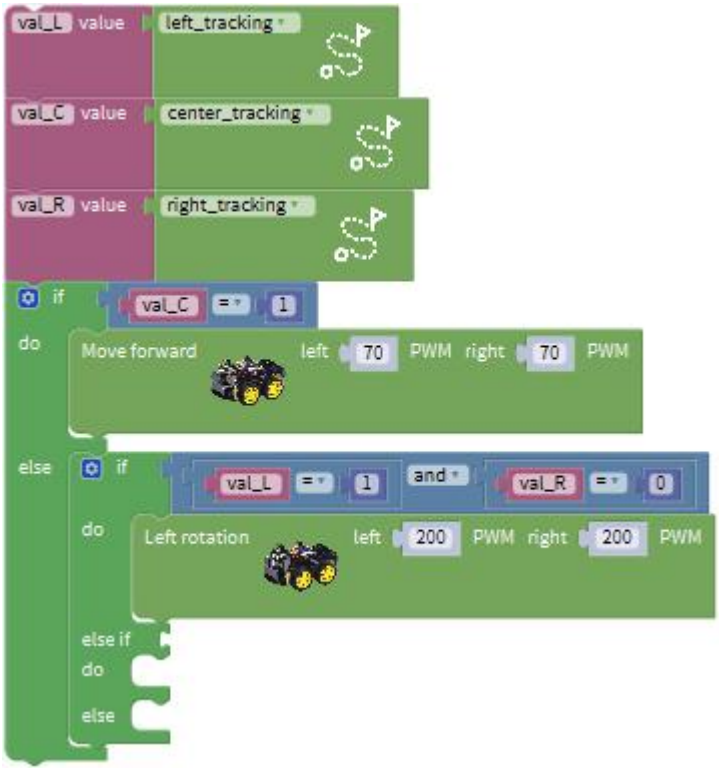
once and keep it into right box of "and"block ,

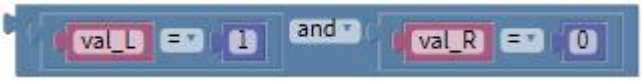
(20) Change val_L into val_R, 1 into 0.



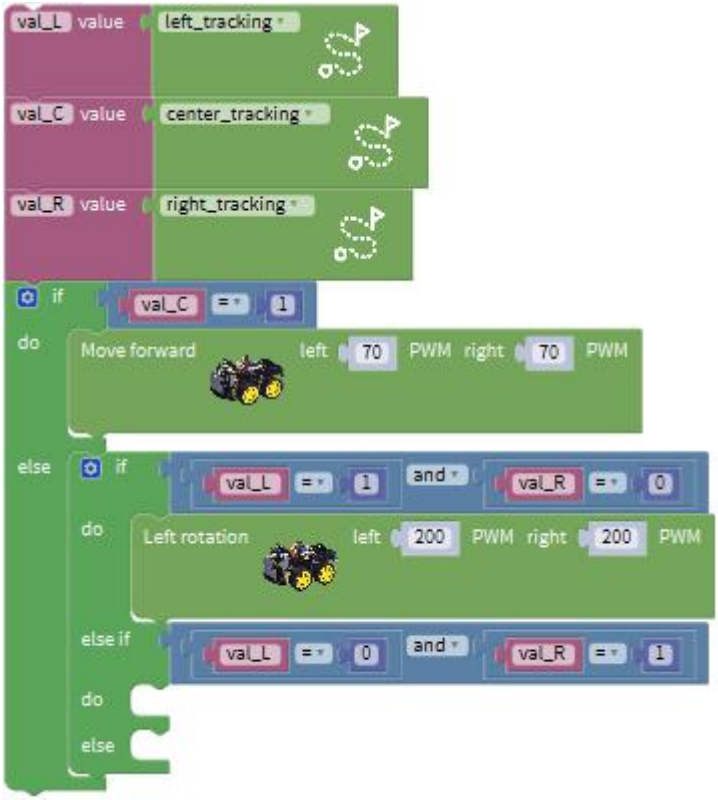
(21) Enter "robot" → "4wd-SmartCar " →

(22) Leave it into second do block, and change 0 into 200.



(23) Duplicate block 

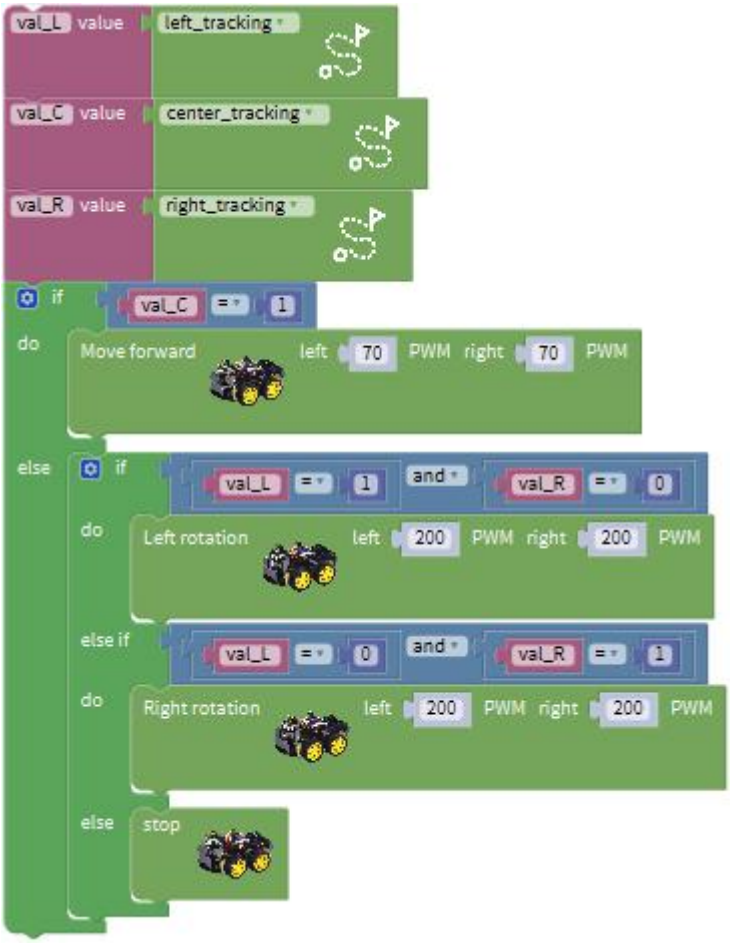
once and keep it into else if block. Refer the following code string to edit:



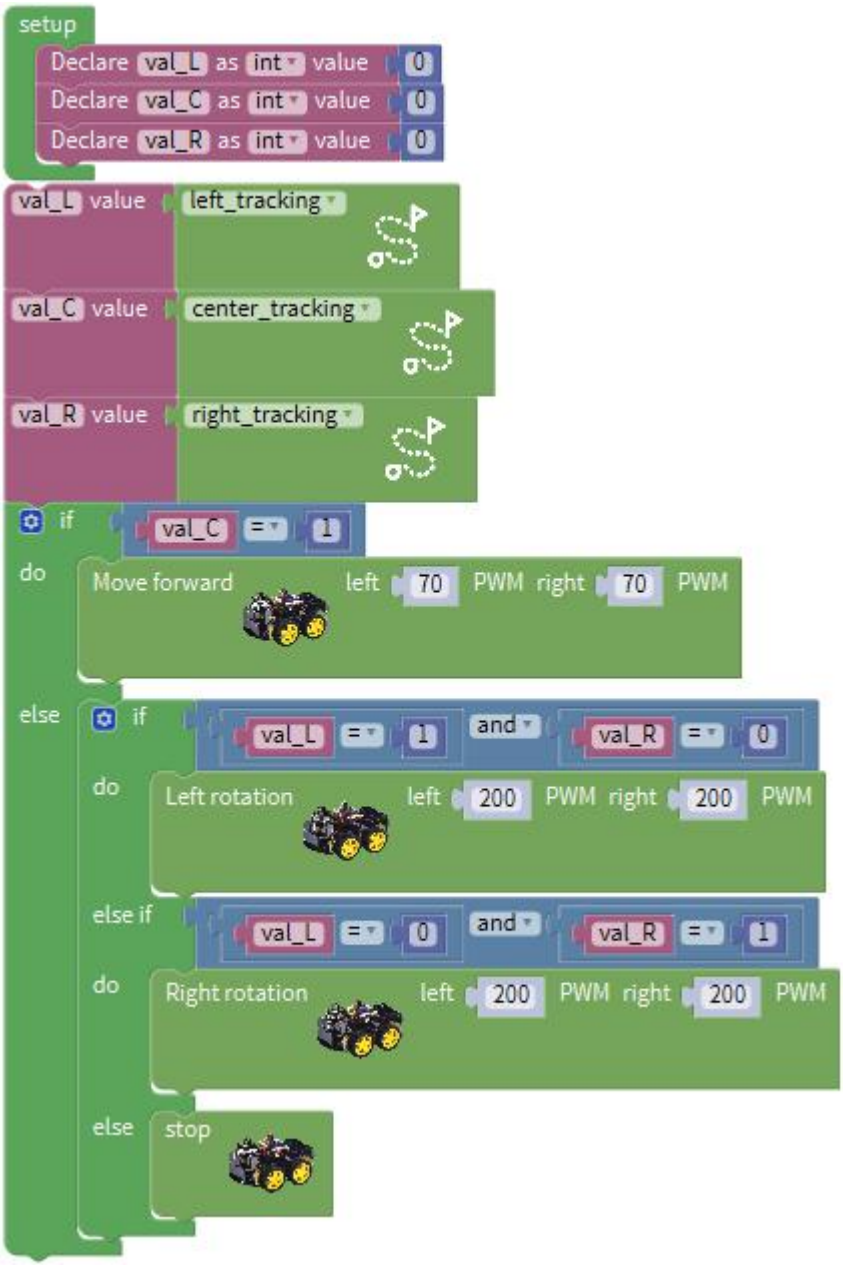
(24) Click " robot " → " 4wd-SmartCar " Module , →

(25) Place it into second do block, and change 0 into 200. Then drag

block into second else block.



Complete Program :


setup
Declare val_L as int value 0
Declare val_C as int value 0
Declare val_R as int value 0
val_L value left_tracking
val_C value center_tracking
val_R value right_tracking
if val_C = 1
do
Move forward
left 70 PWM right 70 PWM
else
if val_L = 1 and val_R = 0
do
Left rotation
left 200 PWM right 200 PWM
else if val_L = 0 and val_R = 1
do
Right rotation
left 200 PWM right 200 PWM
else
stop

Initialization
Set val_L to integer 0
Set val_C to integer 0
Set val_R to integer 0
Set the digital signals read by left TCRT5000 IR tube to val_L
Set the digital signals read by middle TCRT5000 IR tube to val_C
Set the digital signals read by right TCRT5000 IR tube to val_R

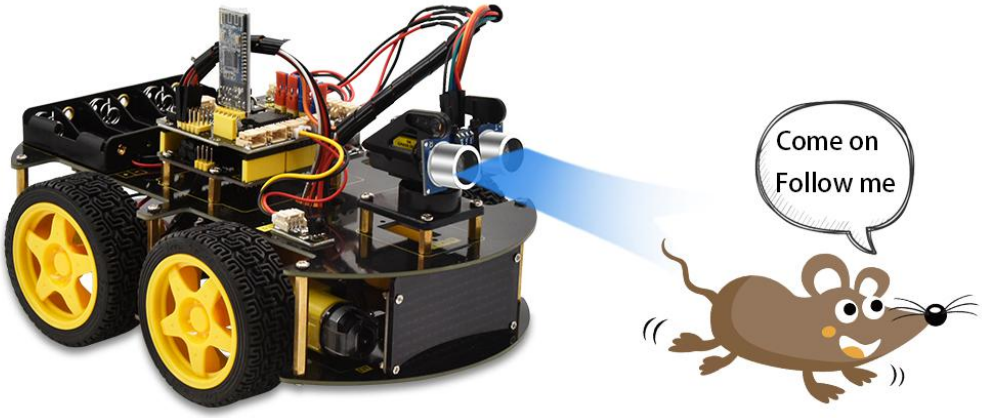
If val_C=1, the code under do block will be executed.
4wd smart car moves forward at the speed of PWM70
If val_L=1 and val_R=0, the code under do block will be executed.
4wd smart car rotates to left at the speed of PWM200
If val_L=0 and val_R=1, the code under do block will be executed.
4wd smart car rotates to right at the speed of PWM200

If the above condition is not met, the code under the else block will be executed
4wd smart car stops

(5) Test Result

Upload code on the development board, plug in power and turn on the switch of smart car. The smart turtle car will walk along the black line.

Project 11: Ultrasonic Follow Robot



(1) Description

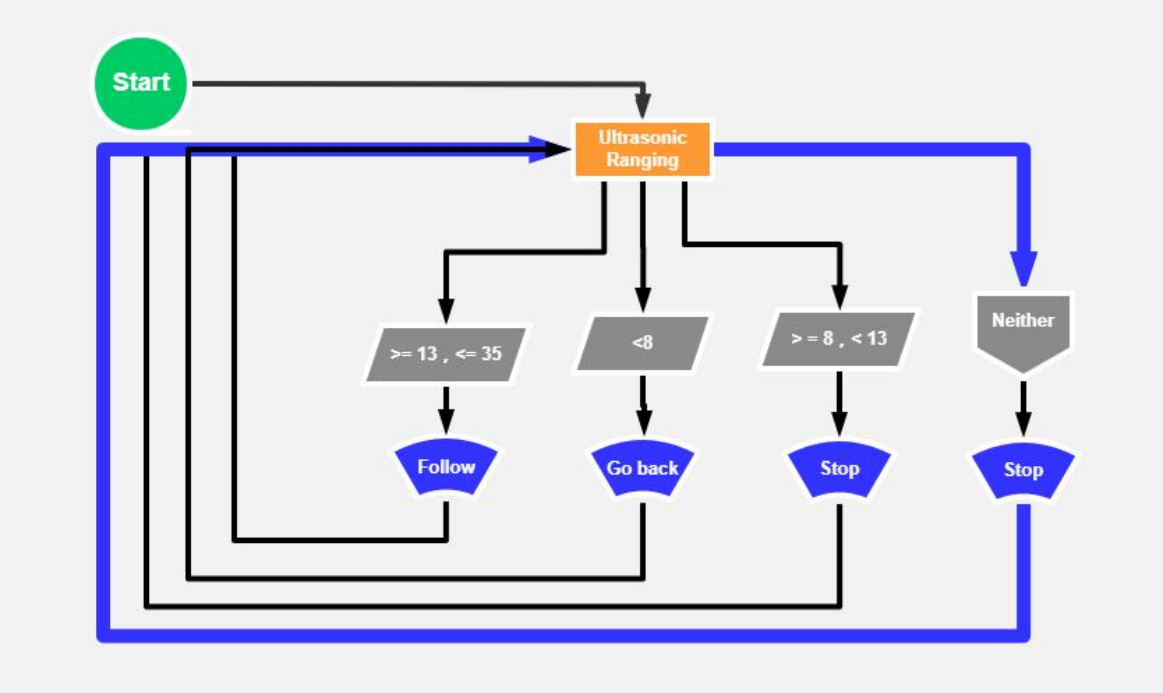
In this project, we detect the distance value of the obstacle to drive two motors so as to make robot car move and 8*8 dot matrix show smile face pattern.

The specific logic of ultrasonic follow robot car is as shown below:

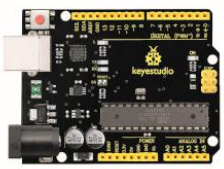
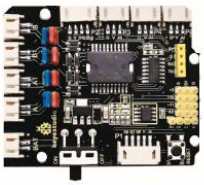
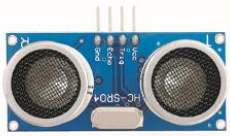
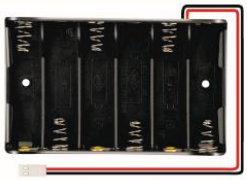
Detection	Measured distance of front obstacles	distance (unit : cm)
Condition	Distance＜8	
Status	Go back（set PWM to 100）	
Condition	distance≥8 and distance＜13	
Status	Stop	
Condition	distance≥13 and distance＜35	

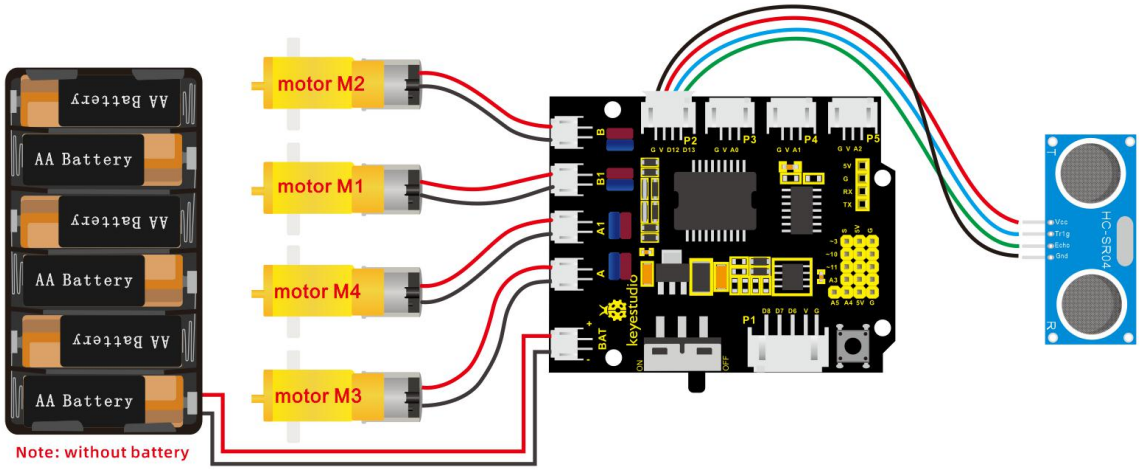
Status	Go front（set PWM to 100）
Condition	distance≥35
Status	stop

(2) Flow Chart



(3) Connection Diagram

Control Board *1	L298P Motor Shield *1	Ultrasonic module *1	Motor *4
			
USB Cable *1	**6 AA battery Holder *1**	**4pin Dupont line *1**	
			



(4) Test Code

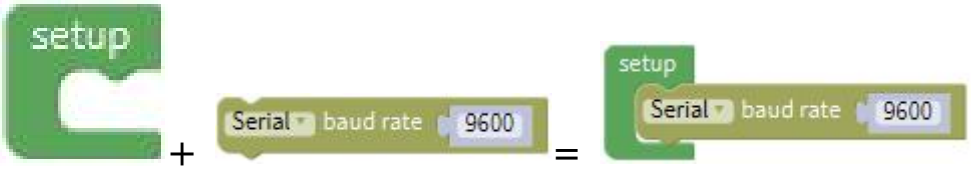
The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_11_Ultrasonic Follow Robot	lesson_11_Ultrasonic Follow Robot

You could edit code step by step as follows:

(1) Enter “Serial Port”to move block 

into 

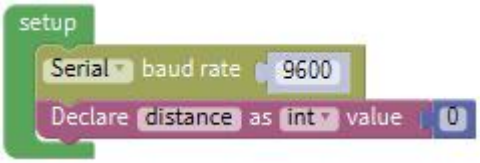


(2) Go to“Variables”to move block 

into block 

, enter“Math”to drag block 

into block 

, and change item into distance. 

(3) Click“Variables”to move out block 

,

(4) Enter “robot”→“4wd-SmartCar”→ 

(5) 

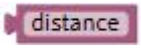
+ 

= 

.

(6) Click“Serial Port”to move out block 

,

(7) Enter“Variables”to drag out 

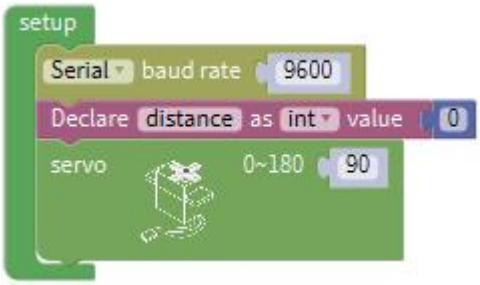
into block 

.

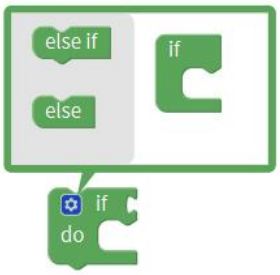
(8) Click “robot”→“4wd-SmartCar”→ 

(9) Place it into 

block , and change 0 into 90.



(10) Click " Control " to move out block 

, tap  , 

appears, drag 

into block 

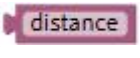
for 2 times and move 

into if block, then the block 

is produced.

(11) Enter"Logic"to move block 

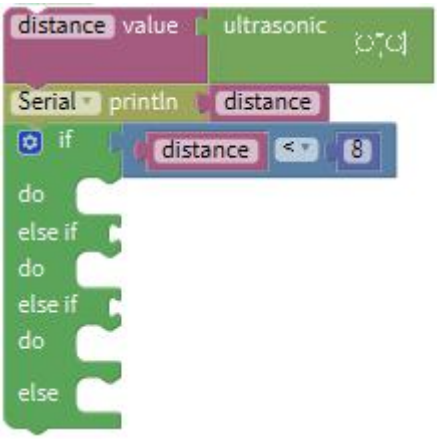
into if block

(12) Go to"Variables"to drag block 

into left box of "="

(13) Click "Math"to move out block 

into right box of "=", change 0 into 8, and "="into"<".



(14) Enter " robot " → " 4wd-SmartCar " →

(15) Leave it into the first do block and set to 100

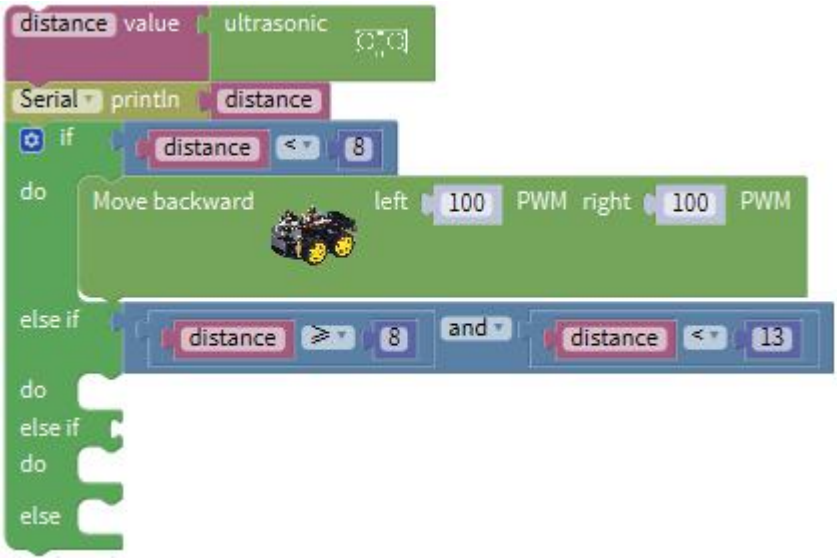
(16) Go to"Logic"to move block 

into else if block.

(17) Replicate 

block twice and place them into both side of "and" block.

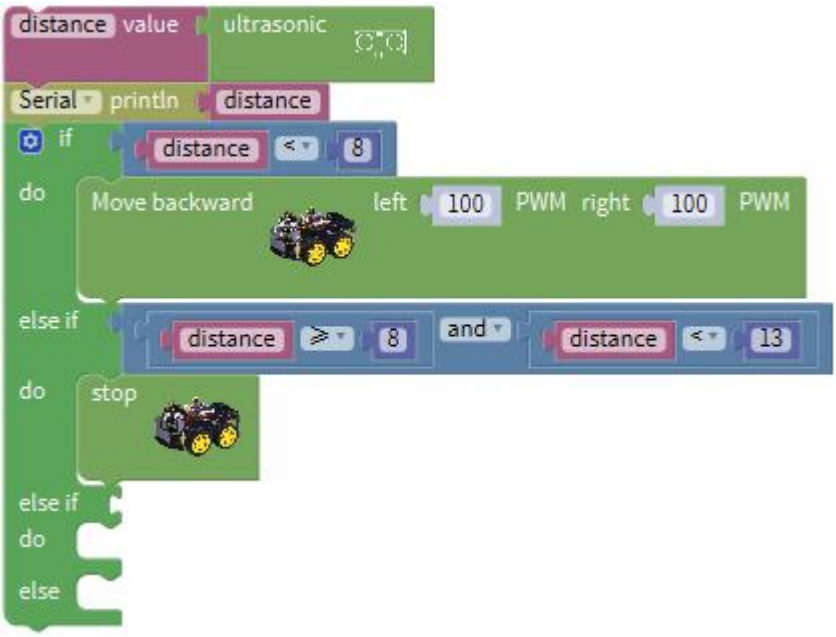
(18) Edit the code string as follows:



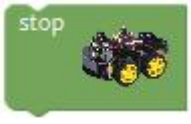
(19) Click “robot”→“4wd-SmartCar”→ 

,

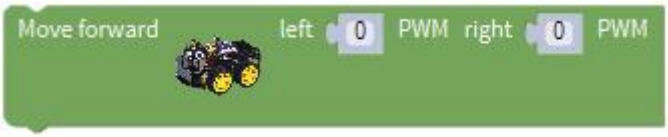
(20) Leave it into the second do block



(21) Copy block 

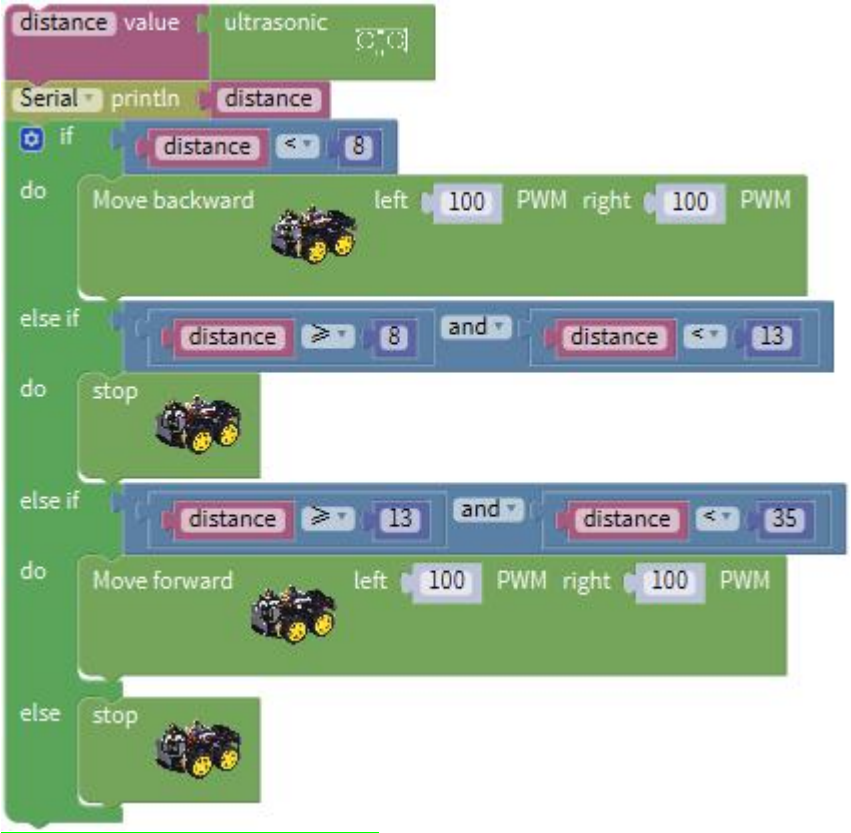
and 

once and place them behind the second“else if”block and else block.

(22) Click “ 4wd-SmartCar ” to move out block 

and leave it into the third do block and set to 100.

(23) Set code string as follows:



Complete Program :

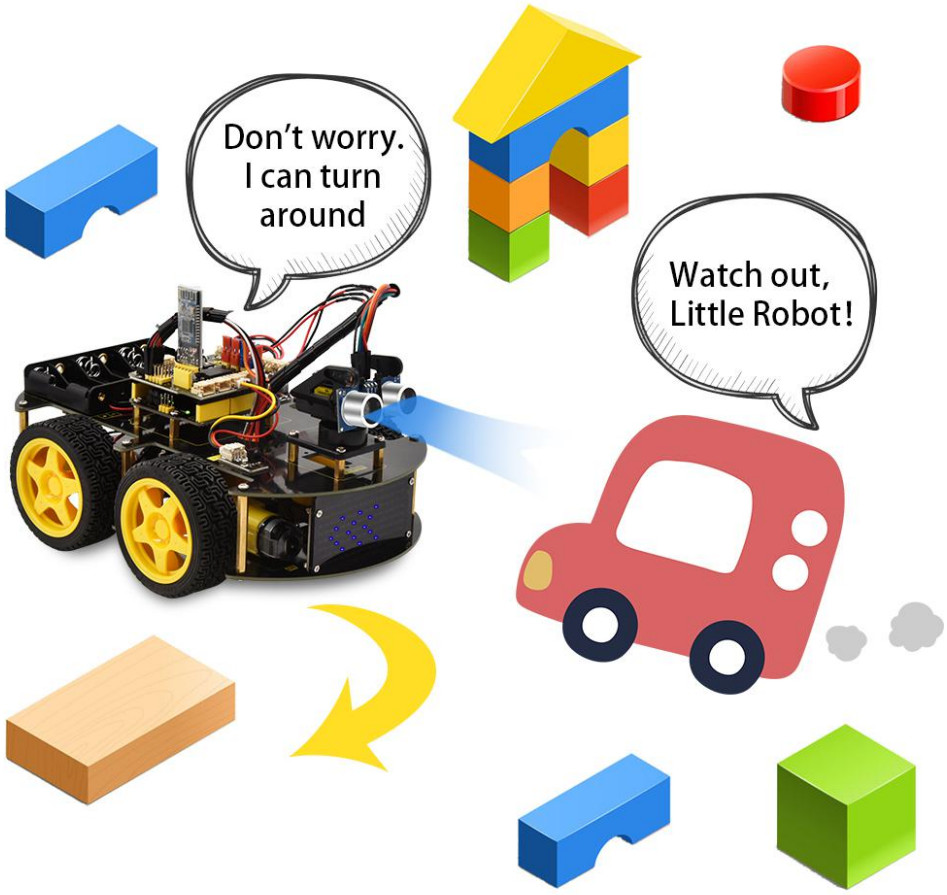

setup
Serial baud rate 9600
Declare distance as int value 0
servo 0~180 90
distance value ultrasonic
Serial println distance
if distance < 8
do
Move backward left 100 PWM right 100 PWM
else if distance ≥ 8 and distance < 13
do
stop
else if distance ≥ 13 and distance < 35
do
Move forward left 100 PWM right 100 PWM
else
stop

Initialization
Set baud rate to 9600
Set distance to integer 0
Set servo to 90°
Assign the detected distance by ultrasonic sensor to variable distance
Serial Port prints the distance detected by ultrasonic sensor.
When distance<8cm, the program under do block will be executed
4wd smart car moves backward at the speed of PWM100
When 8≤distance<13cm，the program under do block will be executed
4wd smart car stops
When 13≤distance<35cm, the code under do block will be executed
4wd smart car moves forward at the speed of PWM100
When the above condition is not met，the program under else block will be executed.
4wd smart car stops

(5) Test Result

Upload the code to the development board, plug in power and turn on the switch of smart car. Robot car will follow the obstacle to move(robot car only moves forward and backward).

Project 12: Ultrasonic Avoiding Robot



(1) Description

We combine the hardware knowledge -- LED matrix, motor drive, ultrasonic and servo, to build an ultrasonic avoiding robot!

In the circuit process, we can make use of ultrasonic sensor to detect the distance between robot and front obstacles. Control the motor rotating by measured data, thus control the robot motion and show the running state by dot matrix.

The ultrasonic avoiding capability is almost the same as the ultrasonic

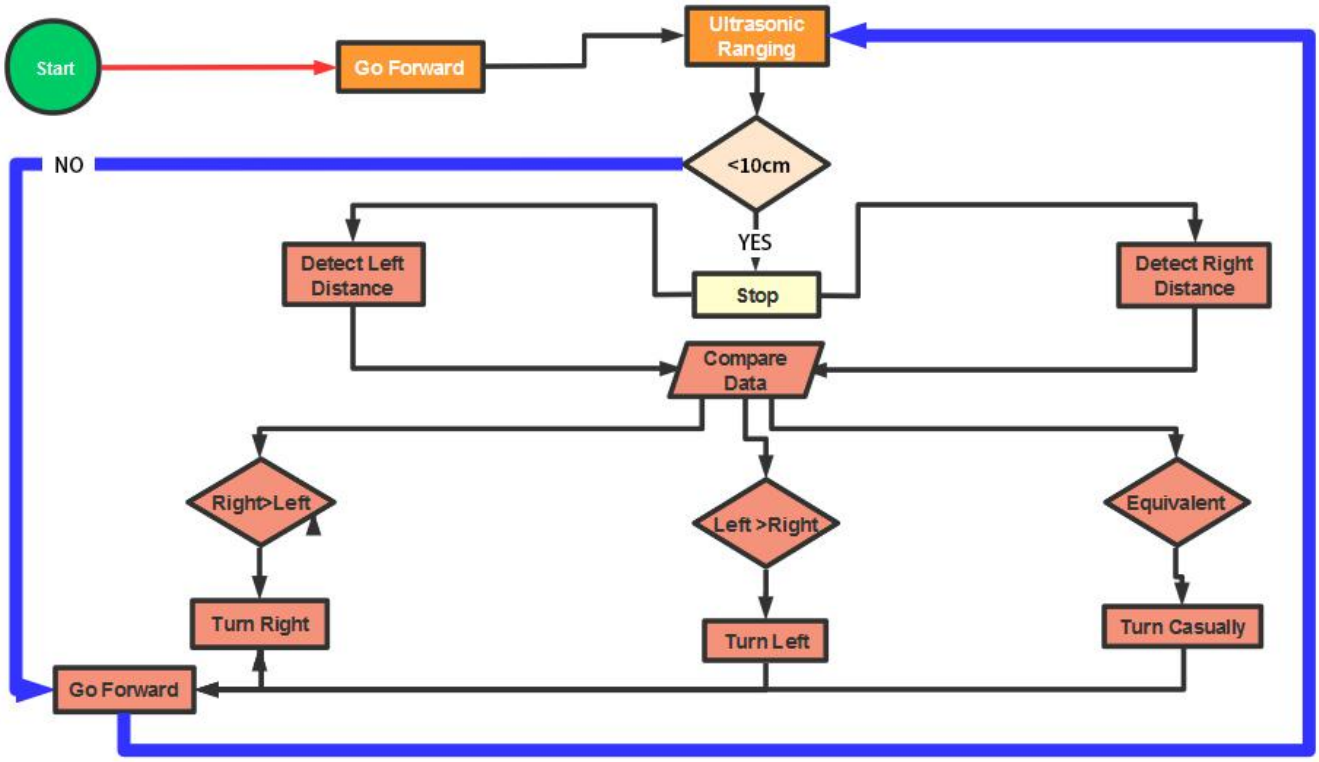
following function. We only need to change the source code.

The specific logic of ultrasonic avoiding smart car is as shown below:

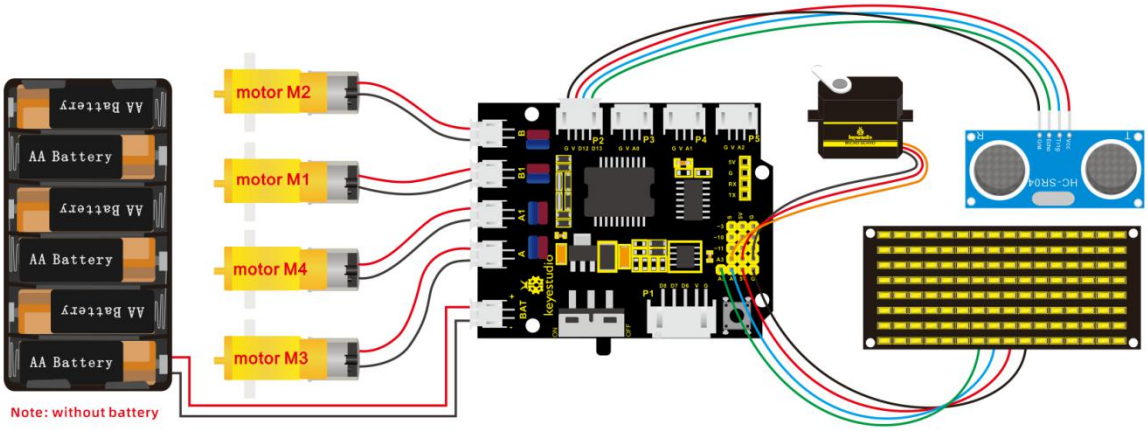
<table>
<tr><td rowspan="2">Initial Setup</td><td colspan="3">8x16 LED Matrix Clear</td></tr>
<tr><td colspan="3">Set servo to 90°</td></tr>
<tr><td rowspan="8">Loop program</td><td colspan="3">measured distance of front obstacle : distance （unit: cm)</td></tr>
<tr><td>Condition 1</td><td colspan="2">State</td></tr>
<tr><td rowspan="6">0<distance ＜10</td><td colspan="2">Smart car stops</td></tr>
<tr><td colspan="2">8x16 LED matrix shows “stop” pattern</td></tr>
<tr><td>Set the servo to 180°</td><td>measured distance of obstacle : a1 （unit : cm)</td></tr>
<tr><td>Set the servo to 0°</td><td>measured distance of obstacle : a2 （unit : cm)</td></tr>
<tr><td>Condition 2</td><td>state</td></tr>
</table>

		a1＜a2	rotate to right（PWM set to 200）
			8x16 LED matrix shows “rightward” pattern
			Set the servo to 90°
		a1≥a2	rotate to left （PWM set to 200）
			8x16 LED matrix shows “leftward” pattern
			Set servo to 90°
	distance ≥ 10	8x16 LED matrix shows “forward” pattern	
		Go front（PWM set to 150）	

(2) Flow Chart



(3) Wiring Diagram



(4) Test Code

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_12_Ultrasonic Avoiding Robot	lesson_12_Ultrasonic Avoiding Robot

You could edit code step by step as follows:

(1) Go to “Control” to find out block 

.

(2) Click “Variables” to move out block 

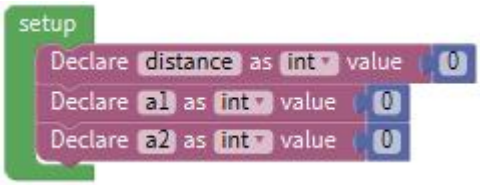
and duplicate it twice.

(3) Separately change item into distance, a1 and a2.

(4) Go to “Math” to move out block 

and copy it for three times

(5) Edit the following code string:



(6) Enter “robot” → “4wd-SmartCar” → 

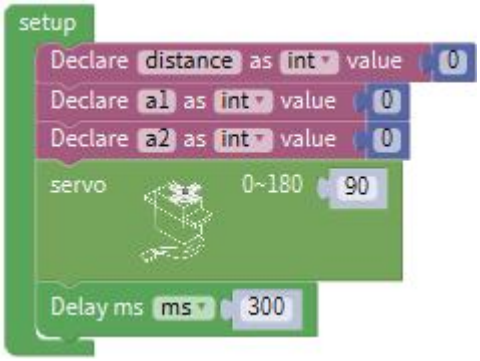
(7) Leave it into block 

, change 0 into 90.

(8) Click " Control " to move block 

into 

block and delay in 300ms.

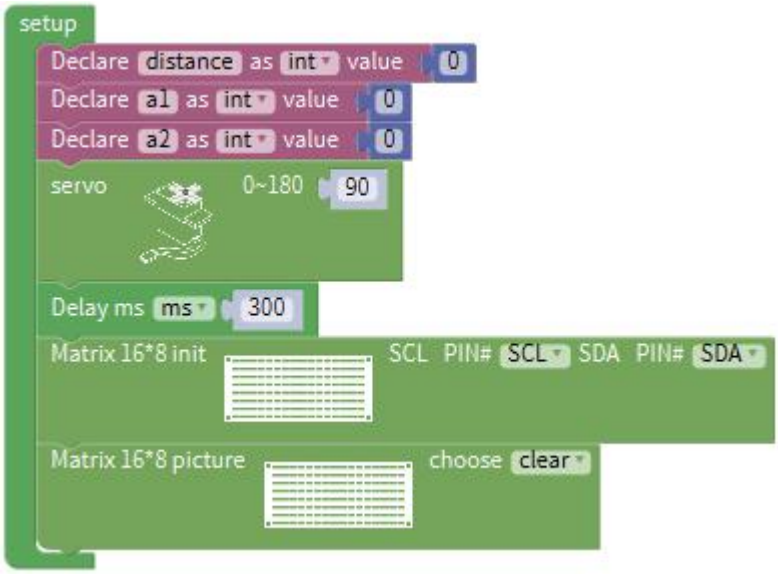


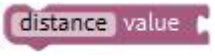
(9) Click " robot " → " 4wd-SmartCar " → 

and 

(10) Keep them into 

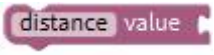
block, click"↑"to select"clear".



(11) Go to " Variables " to move out block 

, then enter

" 4wd-SmartCar " to get block 

and combine it with block 

.

(12) Click"Control"to move out block 

, tap  , and move 

block into 

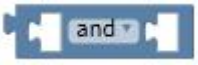
to produce bloc 

.

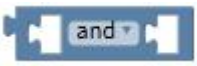
(13) Go to "Logic" to get 

and 

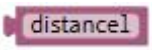
.

(14) Place 

into if block, and leave 

in the left box of block 

.

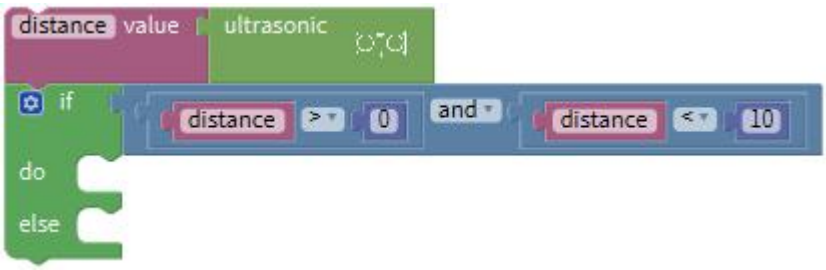
(15) Enter "Variables" to get block 

and keep it at left box of block "=".

(16) Go to "Math" to move out 

into right box of block "=", change "=" into">". Replicate block 

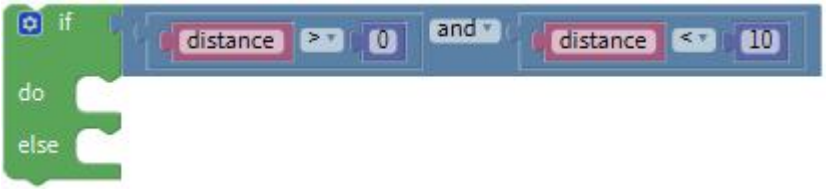
once and set code string as follows:



(17) Click “robot” → ”4wd-SmartCar” → 

and 

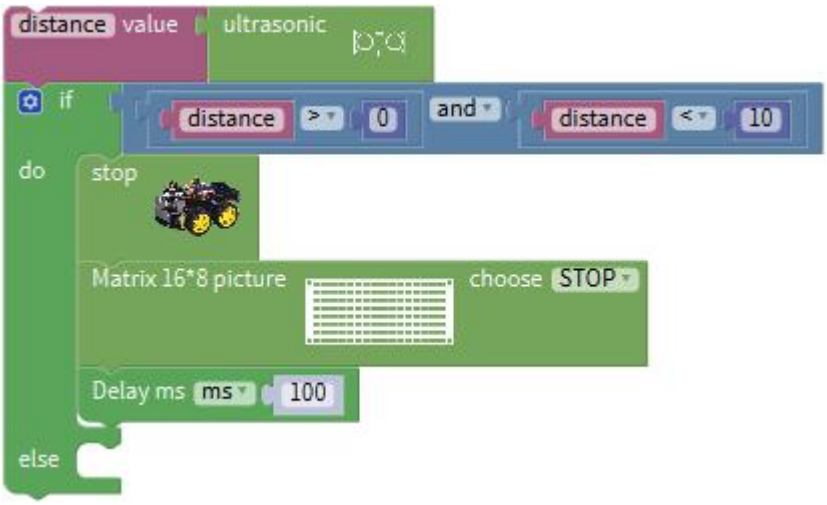
.

(18) Leave them into block 

, and click“ ↑ ”to set “STOP”, then click“Control”to move out block 

and delay in 100ms.

(19) Place it below



(20) Go to “robot” → “4wd-SmartCar” → 

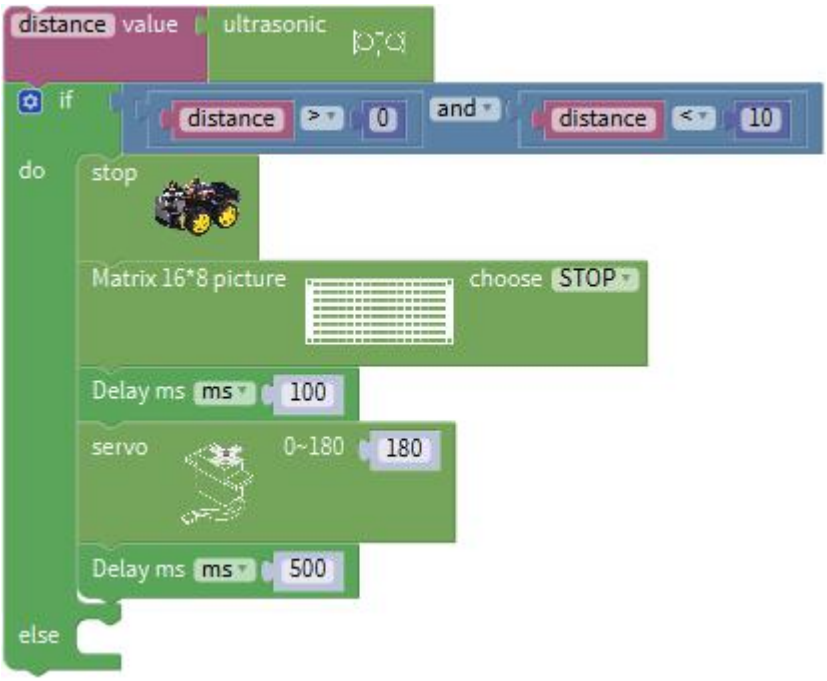
(21) Place it into “do” block, change 0 into 180,

(22) Click “ Control ” to move out block 

and delay in

500ms.

(23) Keep it into do block.

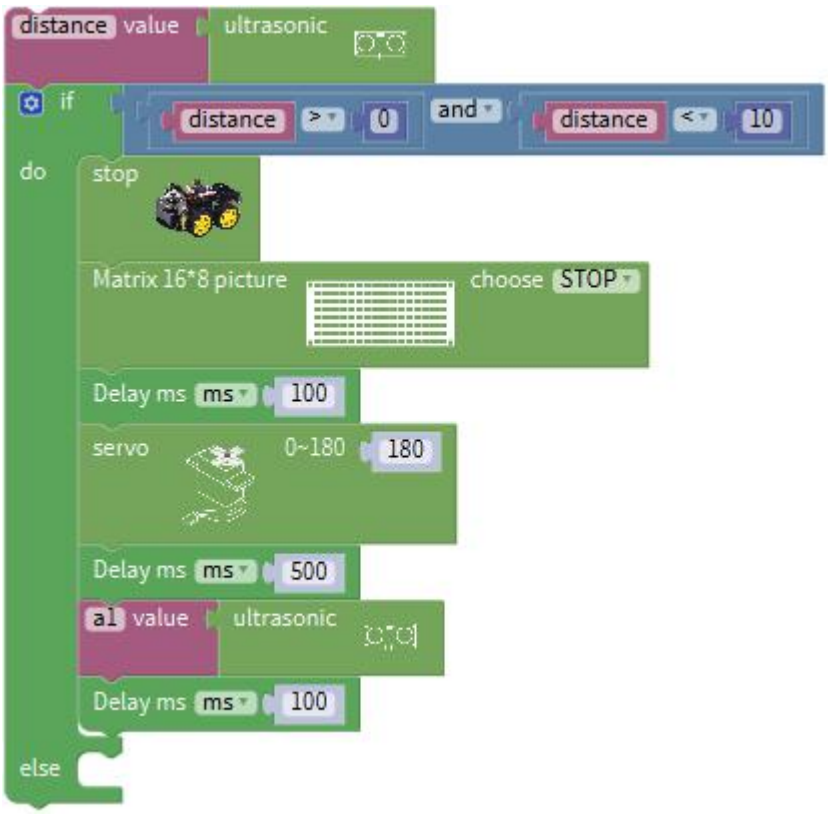


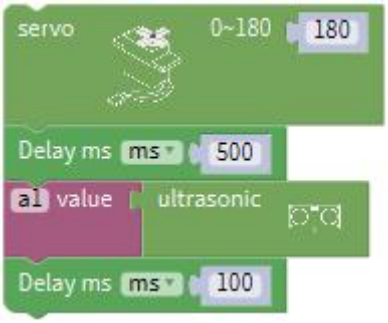
(24) Replicate code 

again and change distance into a1, then place it into do block

(25) Copy block 

and delay in 100ms.





(26) Replicate once and alter 180 into 0, a1 into a2.

(27) Finish the code as follows:

(28) Click "Control" and drag it into block 

, click ⚙ and move 

into 

then we get block 

.

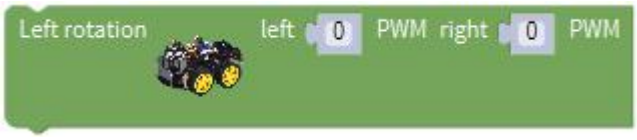
(29) Go to "Logic" to move out block 

into if block, then click "Variables" to drag block 

into left box of " = " block, and move block 

into right box of"=", and change"="into">".

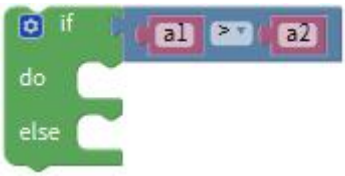
(30) Click "robot" → "4wd-SmartCar" →



,

and

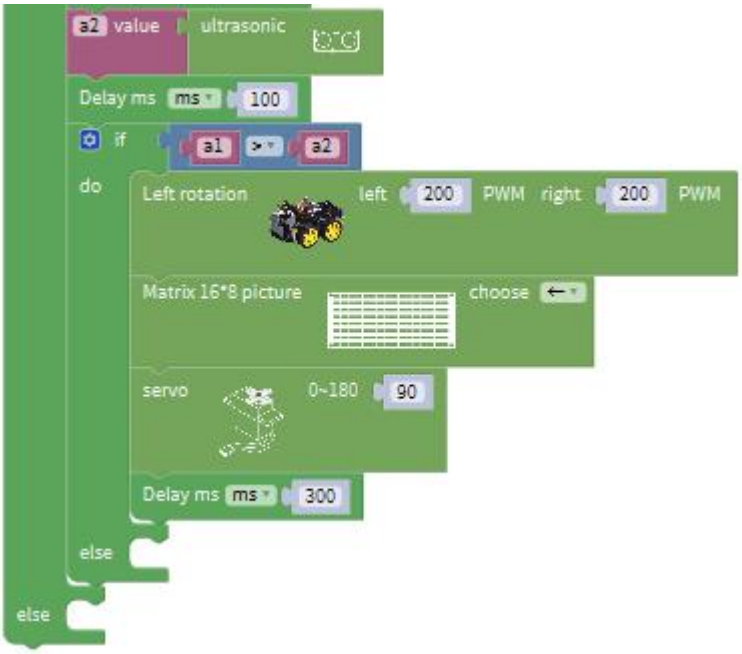
(31) Place them into



block, set to 200, click "↑" to select "←".

(32) Change 0 into 90, and drag

from "Control" into



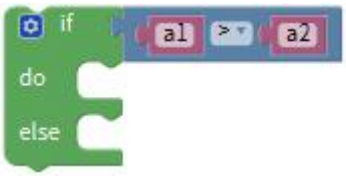
do block, delay in 300ms.

(33) Drag block

,

and 

from "4wd-SmartCar"

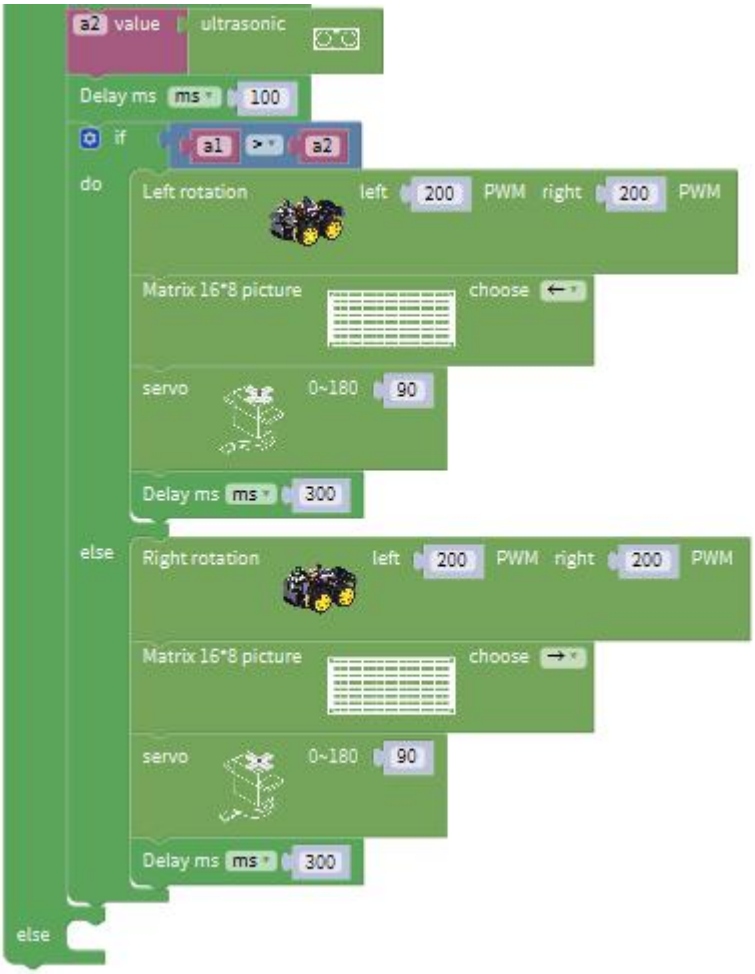
(34) Place them into block 

, set to 200, click "↑" to choose "→"

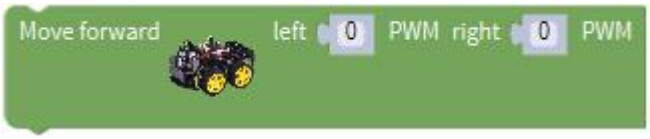
(35) And set servo to 90 and move 

from"Control" and delay in 300ms.

(36) Finish the code as follows:

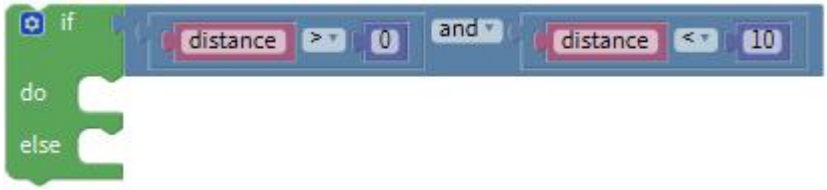


(37) Click " robot " → " 4wd-SmartCar " →

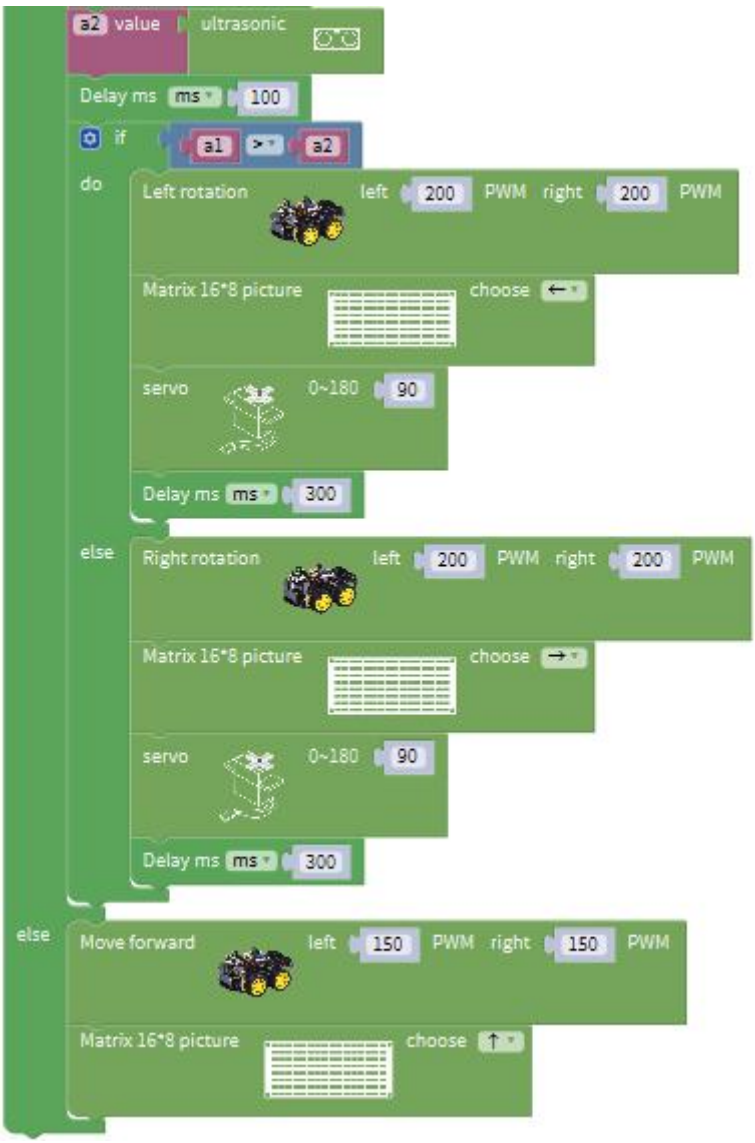


and

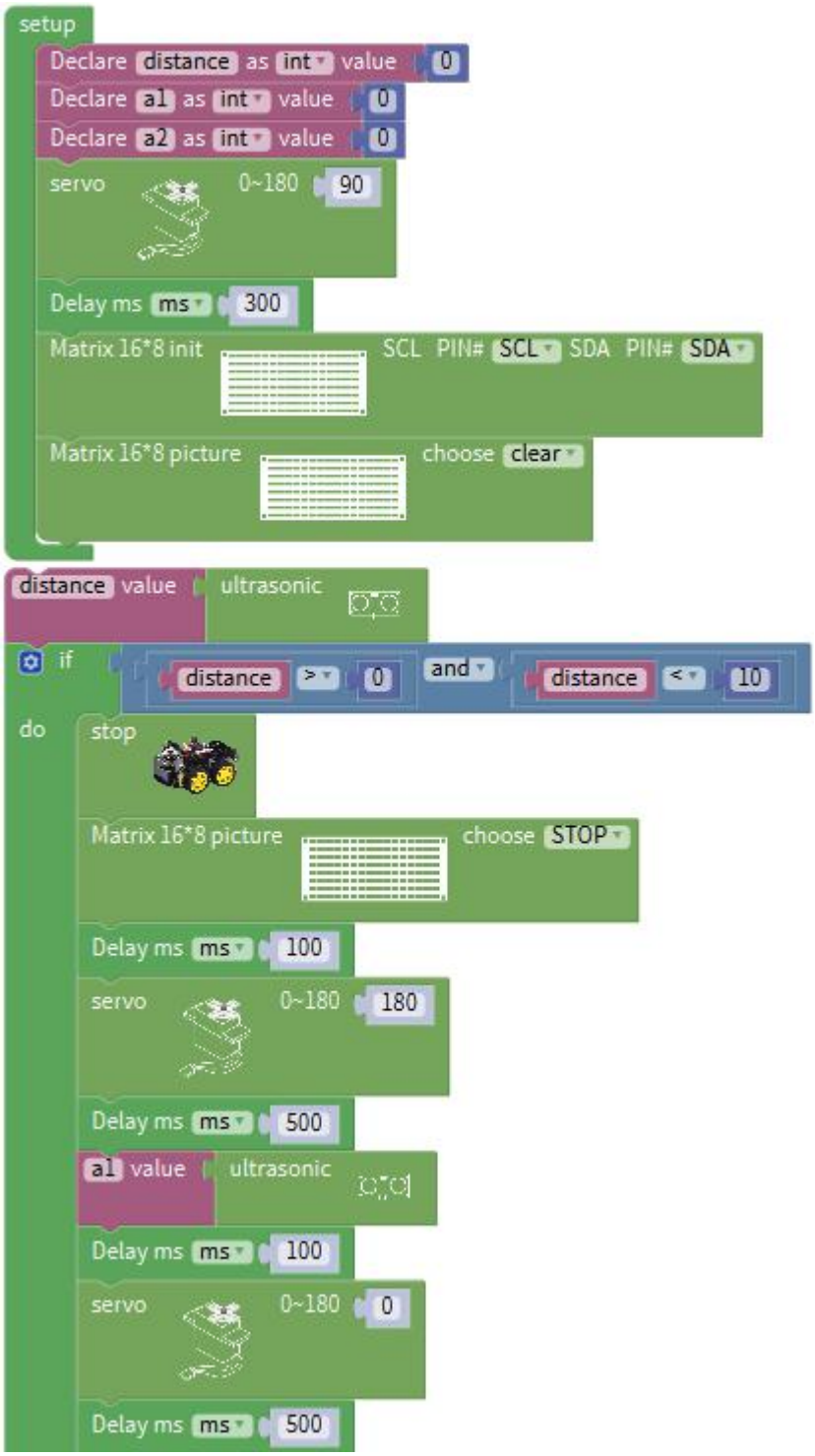
(38) Place them into



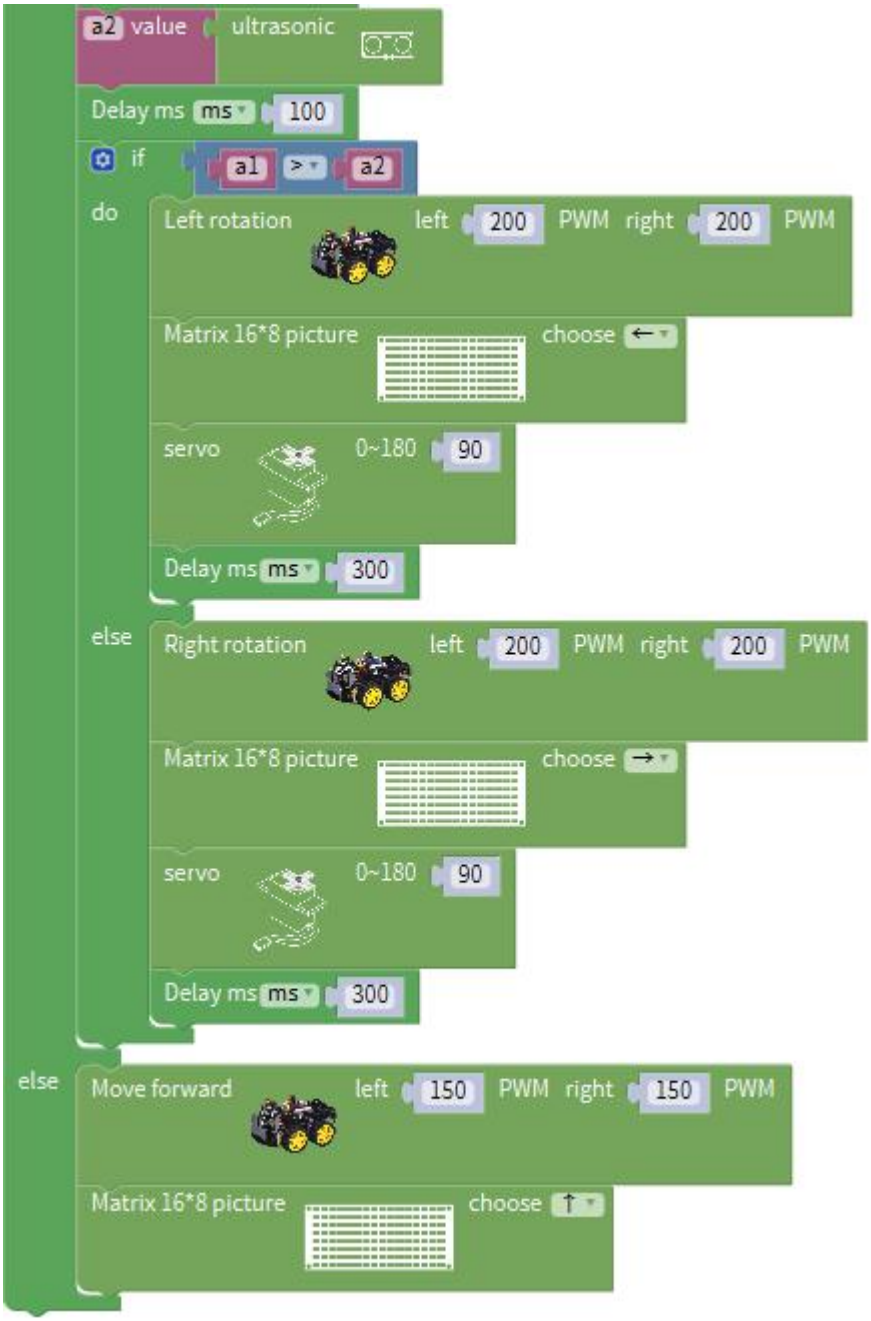
, and set to 150.



Complete Program :


setup
Declare distance as int value 0
Declare a1 as int value 0
Declare a2 as int value 0
servo 0~180 90
Delay ms ms 300
Matrix 16*8 init SCL PIN# SCL SDA PIN# SDA
Matrix 16*8 picture choose clear
distance value ultrasonic
if distance > 0 and distance < 10
do
stop
Matrix 16*8 picture choose STOP
Delay ms ms 100
servo 0~180 180
Delay ms ms 500
a1 value ultrasonic
Delay ms ms 100
servo 0~180 0
Delay ms ms 500

Initialization
Set distance to integer 0
Set a1 to integer 0
Set a2 to integer 0
Set the angle of servo to 90
Delay in 300ms
Set to SCL and SDA for 8x16 dot matrix, SCL corresponds to A5 of V4.0 board, SDA to A4 of V4.0 board
8x16 dot matrix clears screen
Set the distance detected by ultrasonic sensor to variable distance
When the front distance detected by ultrasonic sensor is more than 0cm and less than 10cm, the code under do block will be executed.
4wd smart car stops
8x16 dot matrix shows "↑"
Delay in 100ms
Servo rotates to 180°
Delay in 500ms
Set the left distance detected by ultrasonic sensor to a1
Delay in 100ms
Servo rotates to 0°
Delay in 500ms

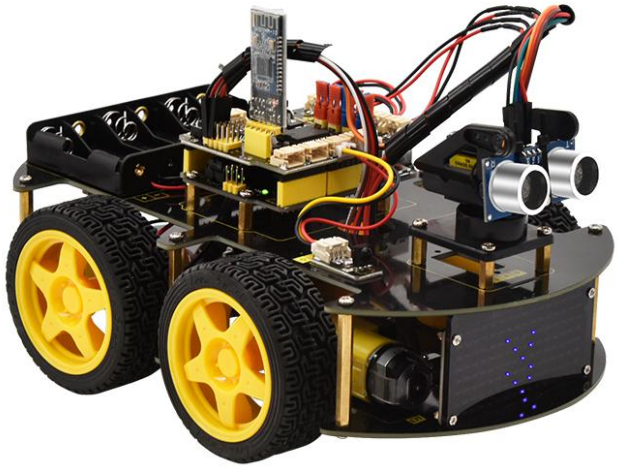

a2 value ultrasonic
Delay ms ms 100
if a1 > a2
do
Left rotation left 200 PWM right 200 PWM
Matrix 16*8 picture choose
servo 0~180 90
Delay ms ms 300
else
Right rotation left 200 PWM right 200 PWM
Matrix 16*8 picture choose
servo 0~180 90
Delay ms ms 300
else
Move forward left 150 PWM right 150 PWM
Matrix 16*8 picture choose

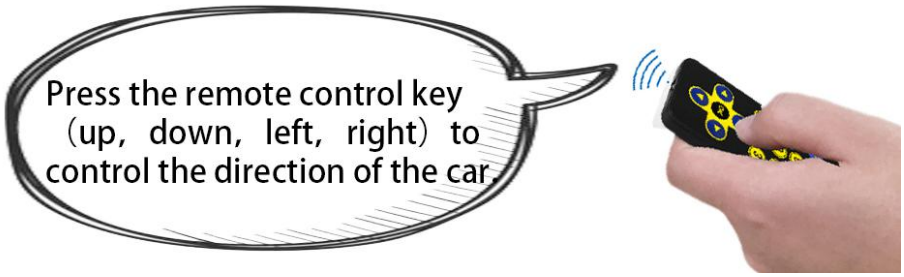
Set the right distance detected by ultrasonic sensor to a2
Delay in 100ms
When，the code under do block will be executed.
4wd smart car rotates to left at the speed of PWM200
8x16 dot matrix shows "←"
Servo rotates to 90°
Delay in 300ms
When a1 is not more than a2, the code under else block will be executed.
When a1>a2, the instruction under else will be executed
4wd smart car rotates to right at the speed of PWM200
8x16 dot matrix shows "→"
Servo rotates to 90°
Delay in 300ms
When the front distance detected by ultrasonic sensor is not more than 0cm and less than 10cm, the code under else block will be executed
4wd smart car moves forward at the speed of PWM150
8x16 dot matrix shows "↑"

(5) Test Result

After uploading the code on the keyestudio V4.0 board, wire according to connection diagram. After the DIP switch is dialed to the right end, the smart car can automatically avoid obstacles.

Project 13: IR Remote Control Robot





(1) Description

We combine the hardware knowledge -- sensors, motor drive, and IR

receiver, to build an infrared remote control robot car!

In the IR receiver section, we've listed out each key value of remote control. In this circuit design, we can set the key value in the code to navigate the robot car movement. The corresponding state pattern is displayed on the 8X16 LED matrix.

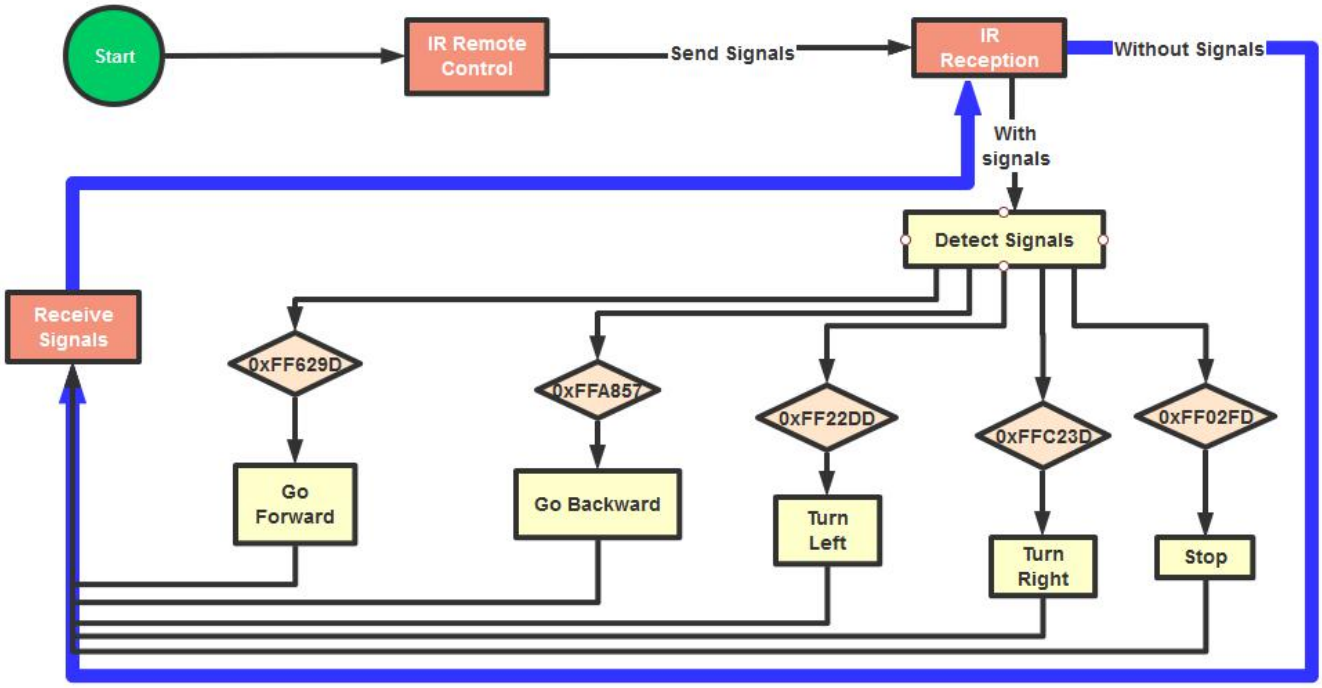
(2) Flow Chart

The specific logic of infrared remote control robot car is shown below:

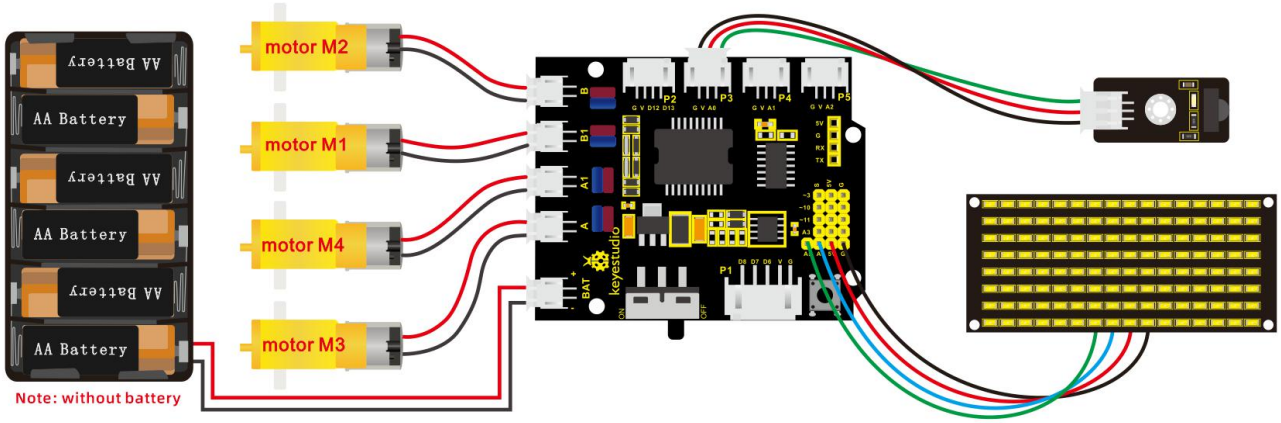
Initial setup	8X16 LED matrix Clear	
Remote control	Key Value	Key state
	FF629D	Go front（PWM set to 100）
		8*8 LED matrix shows front icon
	FFA857	Back（PWM set to 100）
		8*8 LED matrix shows back icon
	FF22DD	Rotate to left（PWM set to 200）
		8X16 LED matrix shows leftward icon

	FFC23D	Rotate to right（PWM set to 200）
		8X16 LED matrix shows rightward icon
	FF02FD	Stop
		8X16 LED matrix shows “STOP”

Based on the circuit design, we can start building our own remote control robot.



(3) Connection Diagram



(4) Test Code :

The program will be generated if you find the following file and drag it into Mixly software.

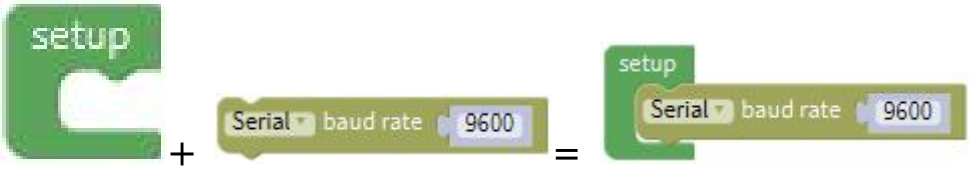
File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_13_IR Remote Control Robot	lesson_13_IR Remote Control Robot

You could edit code step by step as follows:

(1) Click “Control” to get block 

(2) Enter “Serial Port” to move block 

into 



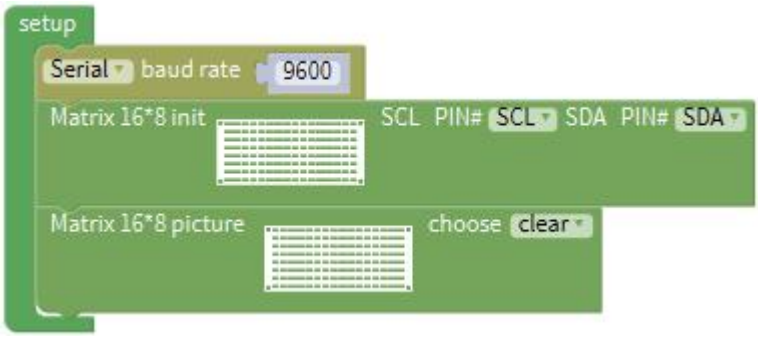
(3) Click " robot " → " 4wd-SmartCar " → 

and 

.

(4) Leave them into block 

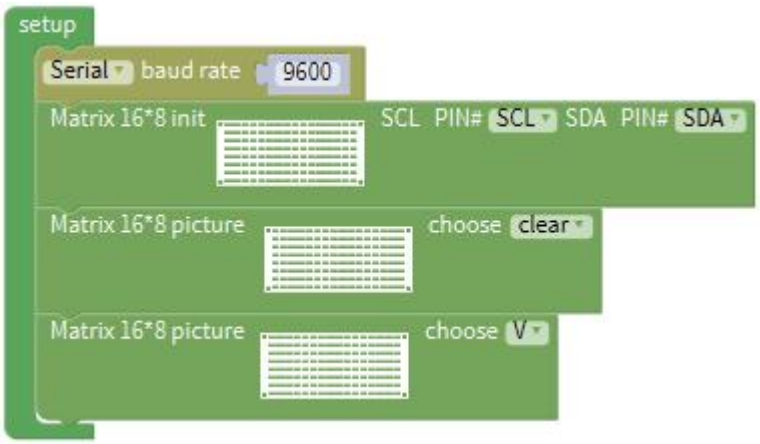
, click "↑" to select"clear".



(5) Copy 

once and place it into 

block, and click"clear"to choose"V".



(6) Enter “robot”→“4wd-SmartCar”→ 

,

Click “ Serial Port ” to move block 

into 

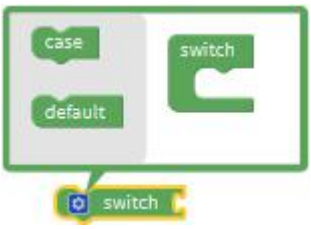
block,

(7) Click “ Variables ” to drag out block 

into block 

.

(8) Click “ Control ” to move out block 

, tap 
, 

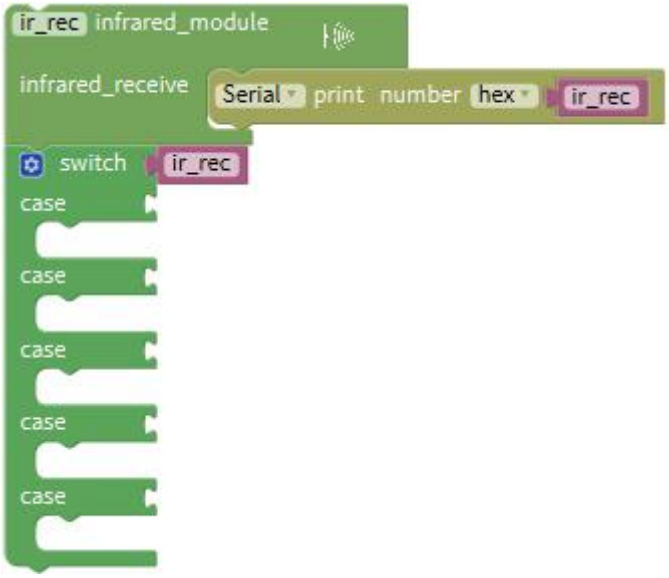
appears, drag 

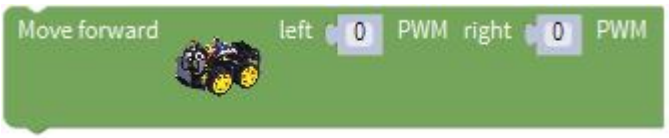
into block 

for 5 times , the

block is produced.

(9) Enter"Variables"to drag block  into"switch"block.

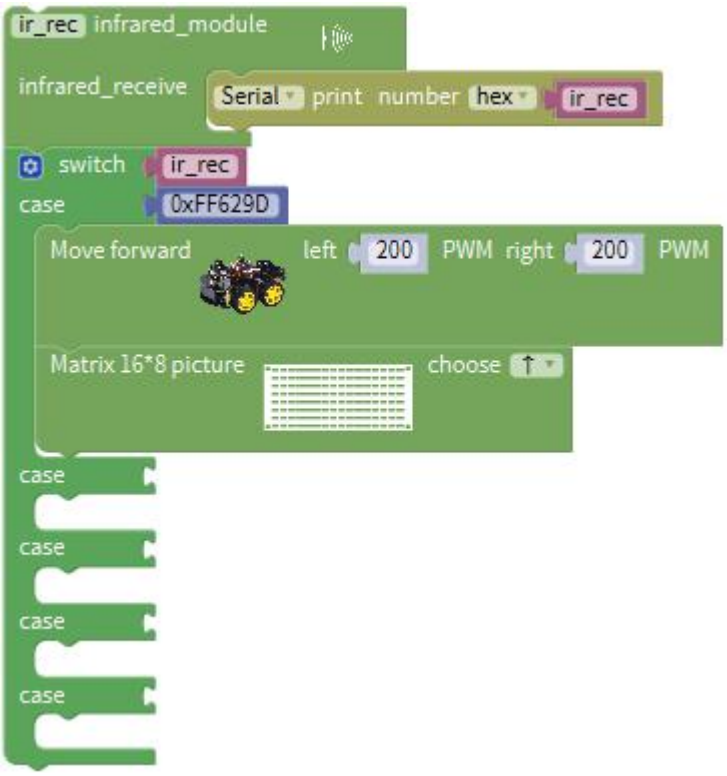


(10) Click"Math" to move block  into the first case, change a into 0xFF629D ; then go to "robot" → " 4wd-SmartCar " → 

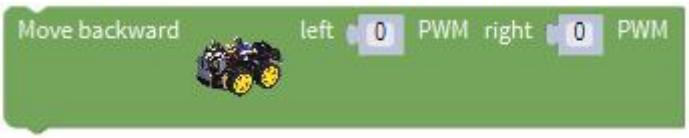
and 

.

Set to 200 and edit the code string as follows:



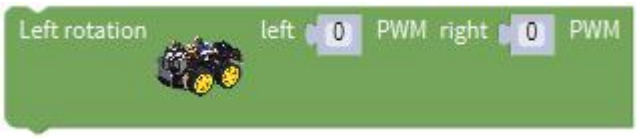
(11) Click "Math" to drag out block 0 into the second case, and change a into 0xFFA857 ;

(12) Go to " 4wd-SmartCar " to move out block 

and 

(13) Set to 200 and click "↑" to select "↓"

(14) Finish the code string as follows:

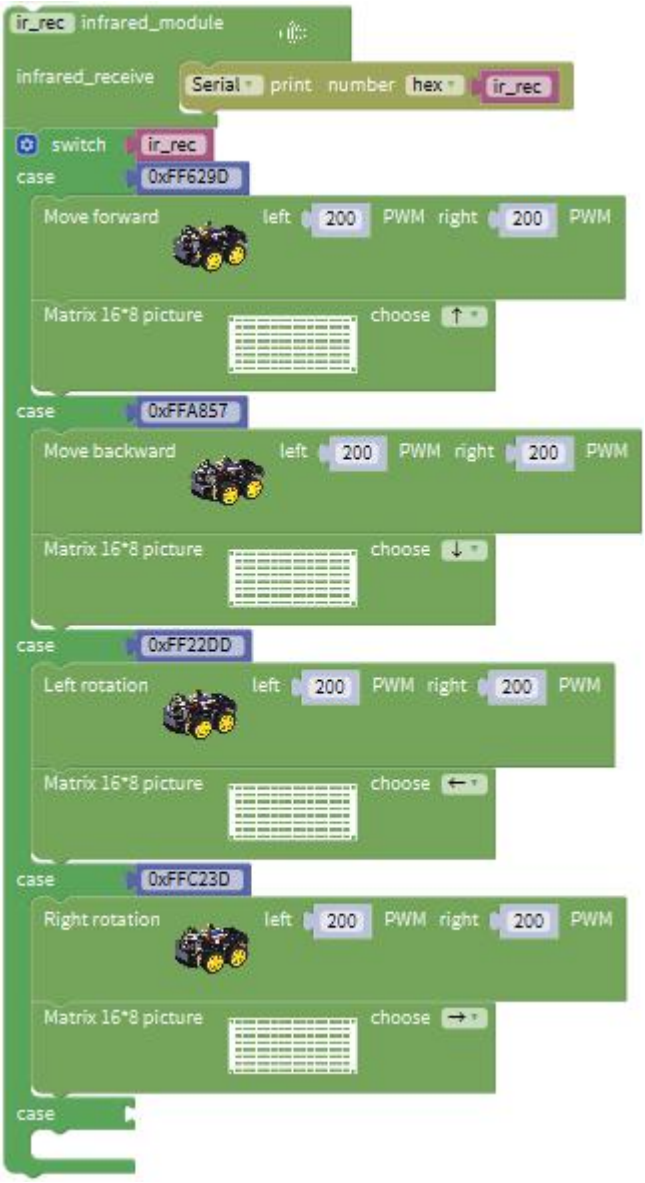
(15) Click "Math" to drag block  into the third case, and change a into 0xFF22DD ; enter "Robot" → "4wd-SmartCar" →  and  .

(16) Place them into the third case, set to 200 and click "↑" to choose "←"

(17) Click "Math" to move block  into the fourth case, and change a into 0xFFC23D ;

(18) Go to " robot " → " 4wd-SmartCar " →  and 

(19) Leave them into the fourth "case", and set to 200. Click"↑"to select "→".

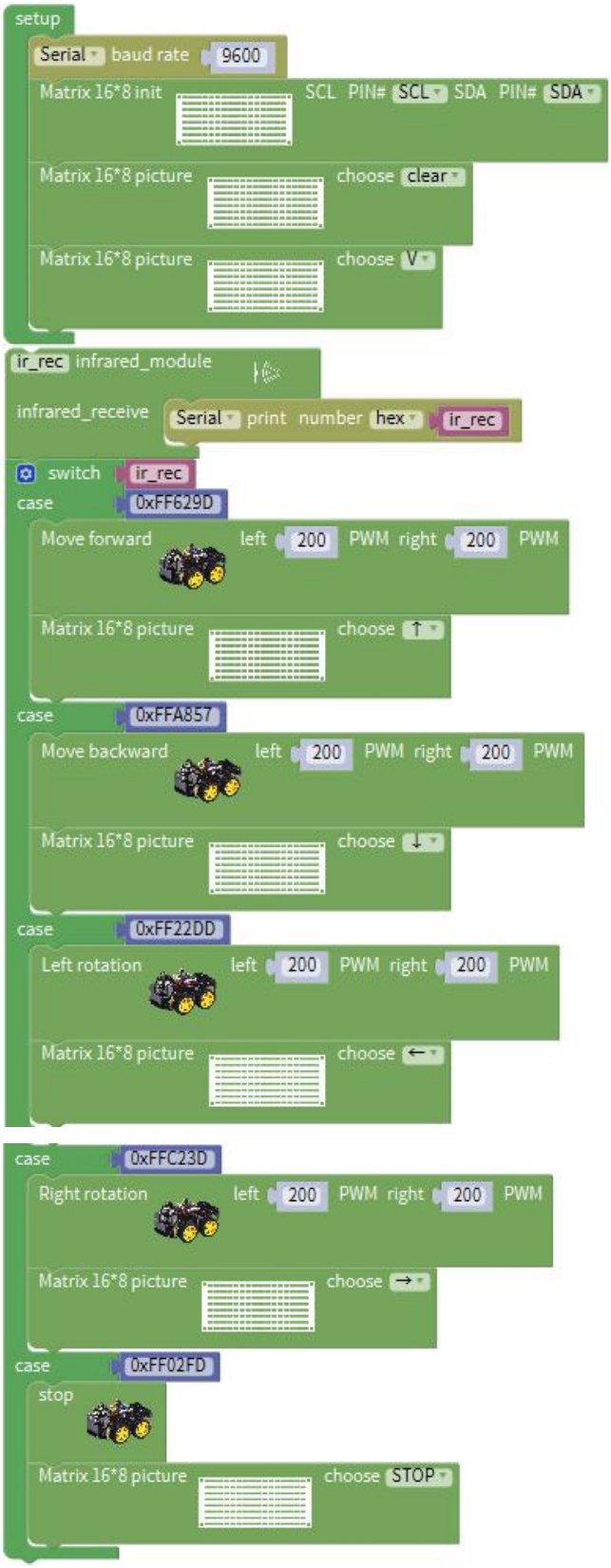


(20) Go to "Math" to move block 0 into the fifth "case", and alter a into 0xFF02FD ;

(21) Click " robot " → " 4wd-SmartCar " → stop and Matrix 16*8 picture choose ↑

(22) Keep them into the fifth "case"block, and click "↑" to set "STOP" .

Complete Program :


setup
Serial baud rate 9600
Matrix 16*8 init SCL PIN# SCL SDA PIN# SDA
Matrix 16*8 picture choose clear
Matrix 16*8 picture choose V
ir_rec infrared_module
infrared_receive Serial print number hex ir_rec
switch ir_rec
case 0xFF629D
Move forward left 200 PWM right 200 PWM
Matrix 16*8 picture choose ↑
case 0xFFA857
Move backward left 200 PWM right 200 PWM
Matrix 16*8 picture choose ↓
case 0xFF22DD
Left rotation left 200 PWM right 200 PWM
Matrix 16*8 picture choose ←
case 0xFFC23D
Right rotation left 200 PWM right 200 PWM
Matrix 16*8 picture choose →
case 0xFF02FD
stop
Matrix 16*8 picture choose STOP

Initialization
Set baud rate to 9600

Set to SCL and SDA for 8x16 dot matrix, SCL corresponds to A5 of V4.0 board, SDA to A4 of V4.0 board
Dot matrix 8x16 clears the screen
8x16 dot matrix shows "V"

Serial port prints the received IR signals in hexadecimal

When ir_rec=0xFF629D, the code under case block will be executed
4wd smart car moves forward at the speed of PWM200
8x16 dot matrix shows "↑"
When ir_rec=0xFFA857, the code under case block will be executed
4wd smart car moves backward at the speed of PWM200
8x16 dot matrix shows "↓"
When ir_rec=0xFF22DD, the code under case block will be executed
4wd smart car rotates to left at the speed of PWM200
8x16 dot matrix shows "←"
When ir_rec=0xFFC23D, the code under case block will be executed
4wd smart car rotates to right at the speed of PWM200
8x16 dot matrix shows "→"
When ir_rec=0xFF02FD, the code under case block will be executed.
4wd smart car stops
8x16 dot matrix shows "STOP"

(5) Test Result :

After uploading the code successfully on the keyestudio V4.0 board, wire according to the connection diagram, after DIP switch is dialed to the right end, we can use the infrared remote control to control the smart car movement. At the same time, the 8X16 LED light board displays the corresponding state pattern.

Project 14: Bluetooth Remote Control

(1) Description

We've learned the basic knowledge of Bluetooth, in this lesson, we will make a Bluetooth remote smart car. In the experiment, we default the HM-10 Bluetooth module as a Slave and the cellphone as a Host.

keyes BT car is an APP rolled out by keyestudio team. You could control the robot car by it readily.

Special Note: Before uploading the test code, you need to remove the Bluetooth module. Otherwise the test code will fail to upload. After the code is uploaded successful, then reconnect the Bluetooth module.

The program will be generated if you find the following file and drag it into

Mixly software.

File Type	Route	File Name
MIX File	../tutorialforMixly/MixlyCode/lesson_14_Bluetooth Remote Control	lesson_14.1_Bluetooth Reads Data

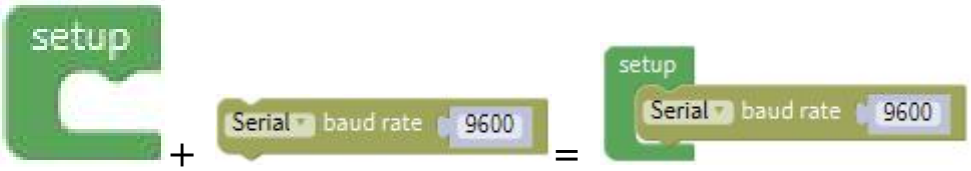
You could edit code step by step as follows:

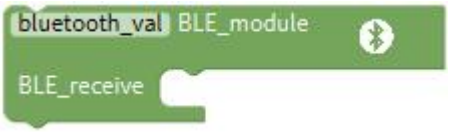
(1) Go to“Control”Module to find out block 

.

(2) Enter“Serial Port”to move block 

into 



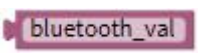
(3) Enter “robot”→ “4wd-SmartCar” → 

.

(4) Click “ Serial Port ” to find out block 

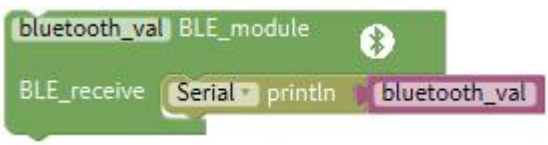
and place it into block 

;

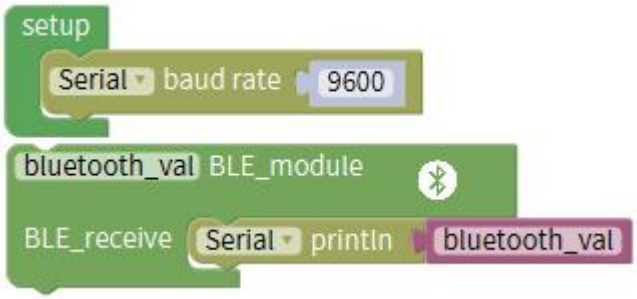
(5) Click “Variables”to move out block 

into block 

.



Complete Program :



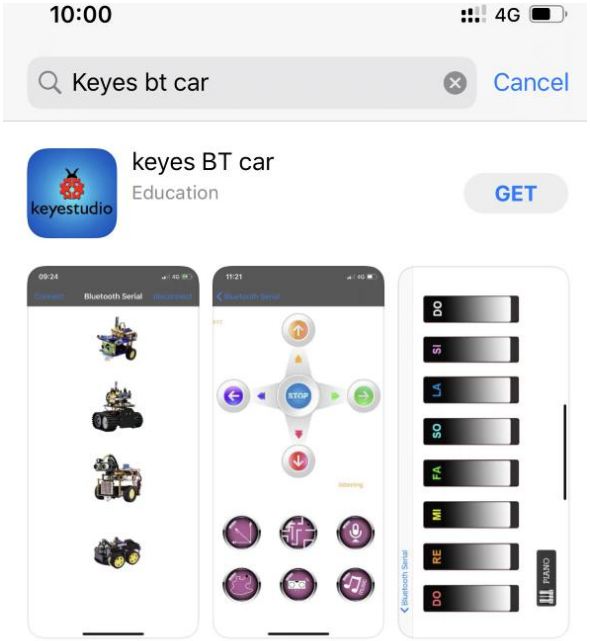
Initialization
Set baud rate to 9600
The Bluetooth data received by BLE module is printed on serial monitor

Upload test code on V4.0 development board and insert the Bluetooth module. Then we need to download APP.

For iOS system

Note: Allow APP to access "location" in settings of your cellphone when connecting to Bluetooth module. Otherwise, Bluetooth may not be connected.

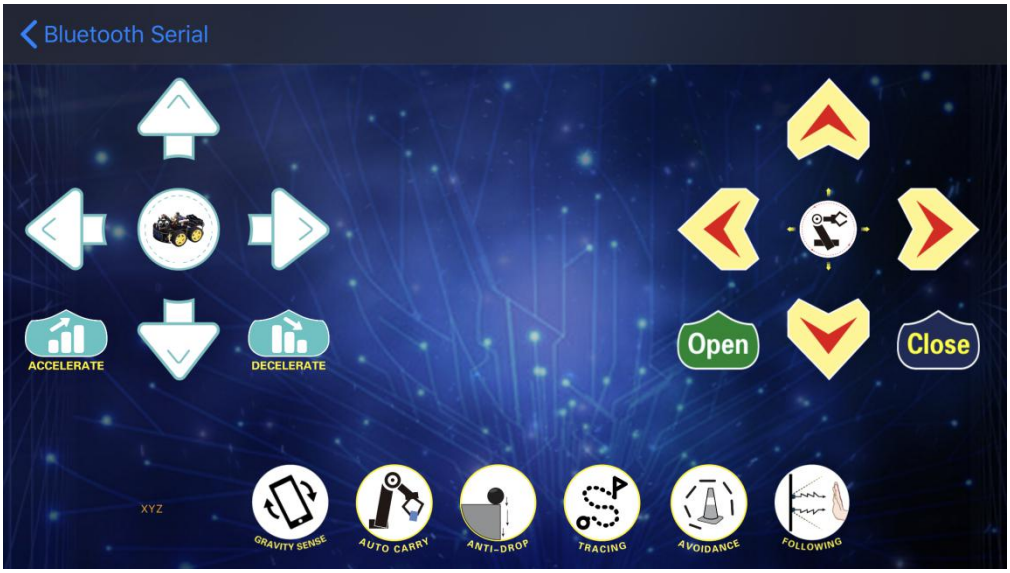
Search keyes BT car in App store



After installation, enter its interface.

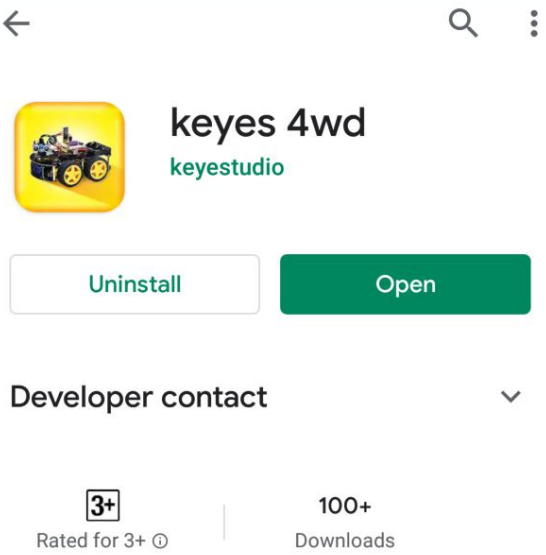
Click “Connect”to search and pair Bluetooth. After connecting well, click

to enter the main page of 4WD smart car.

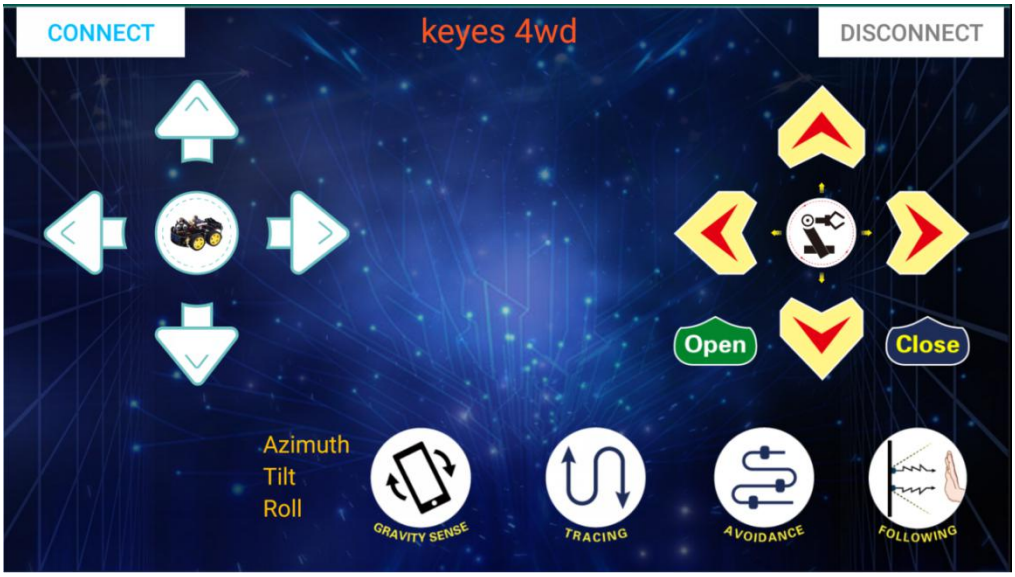


For Android System

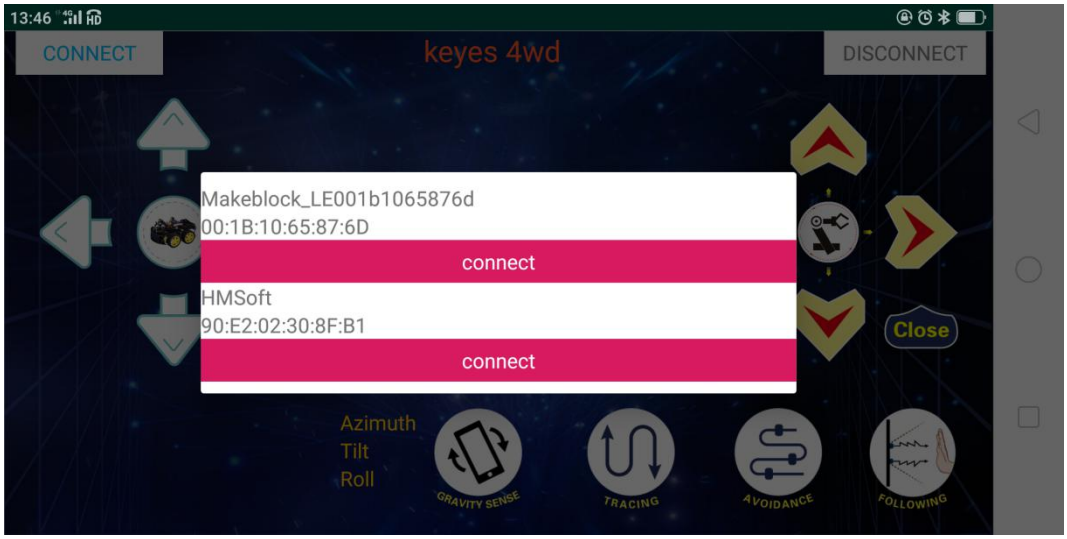
Enter Google play store to search keyes 4wd



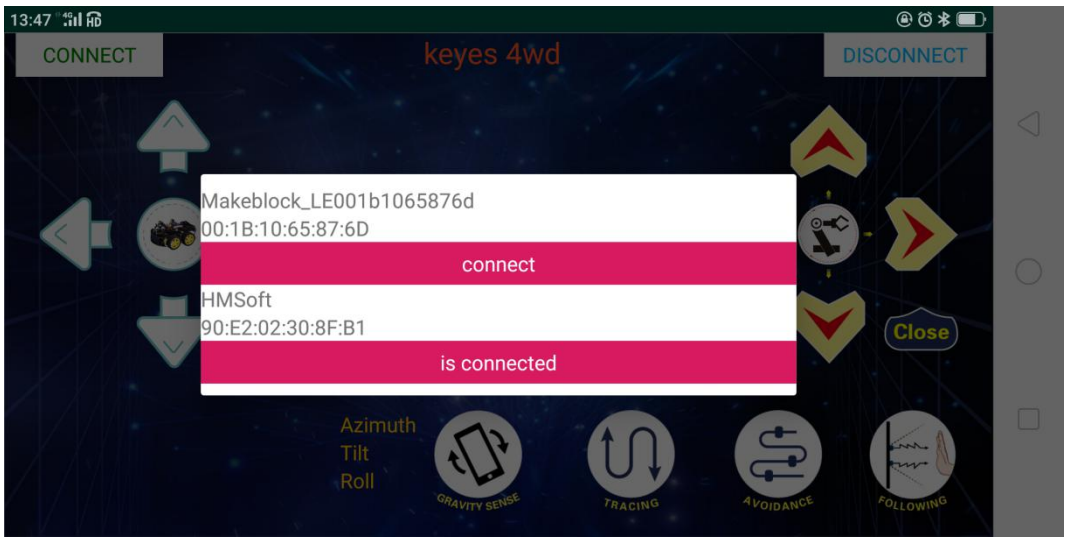
Its interface is shown below:



Click on APP CONNECT icon, searching the Bluetooth.



Click to connect the Bluetooth. HMSoft connected, Bluetooth LED will turn on normally.



After successful connection, press the button of the Bluetooth APP, and the corresponding characters are displayed in the serial monitor. For more details, as shown below.

Below is app of 4WD robot car interface and we have listed out what function of each key does

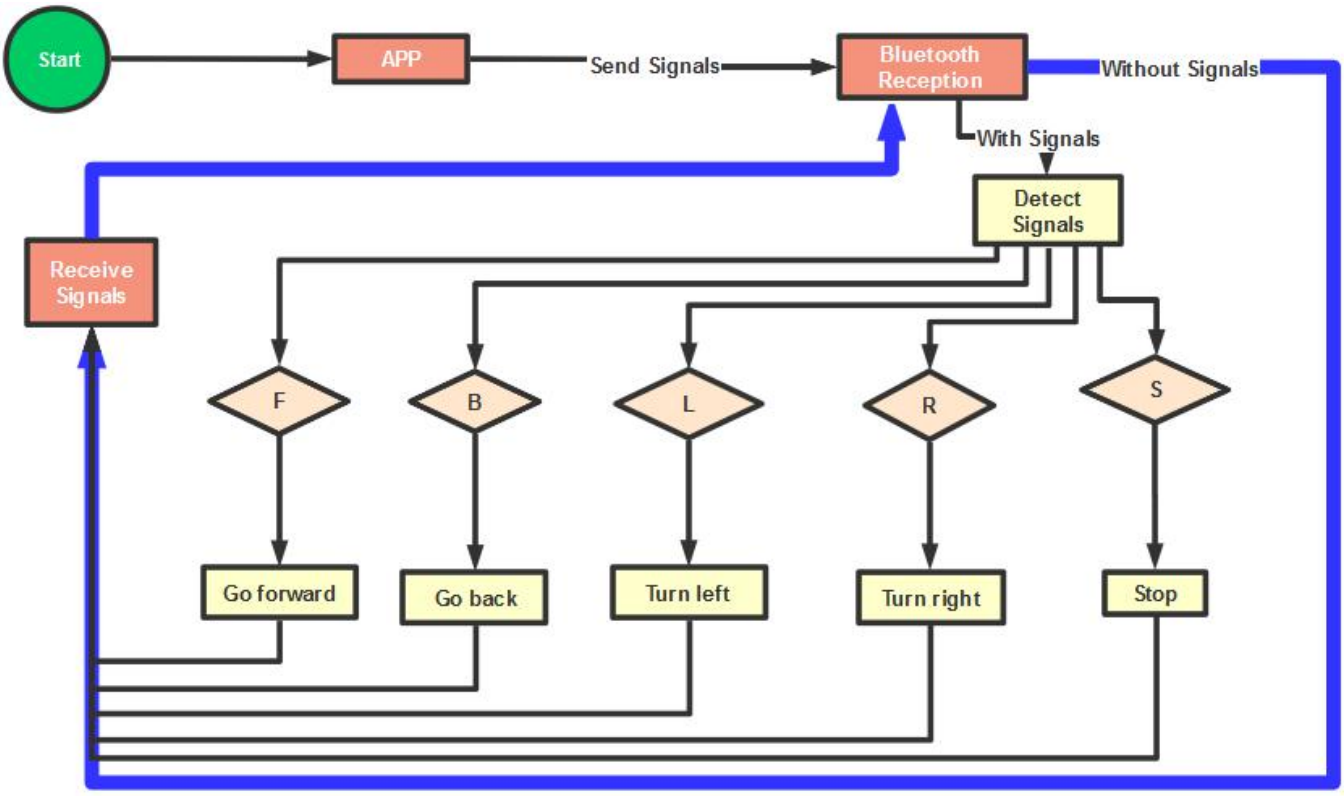
Key	Function	
CONNECT	match with connection HM-10 Bluetooth module	
DISCONNECT	disconnect Bluetooth	
	Control character	Function
	Press: F	Press the button, robot car

	Release: S	goes front; release to stop
	Press: L Release: S	Press the button, robot car turns left; release to stop
	Press: R Release: S	Press the button, robot car turns right; release to stop
	Press: B Release: S	Press the button, robot car goes back; release to stop
	Click to start the mobile gravity sensing; click again to exit	
	Click to send “X”, then click “S”	Start line tracking function; click Stop to exit
	Click to send “Y”, then click “S”	Start ultrasonic avoiding function; click Stop to exit
	Click to send “U”, then click “S”	Start Ultrasonic follow function; click Stop to exit

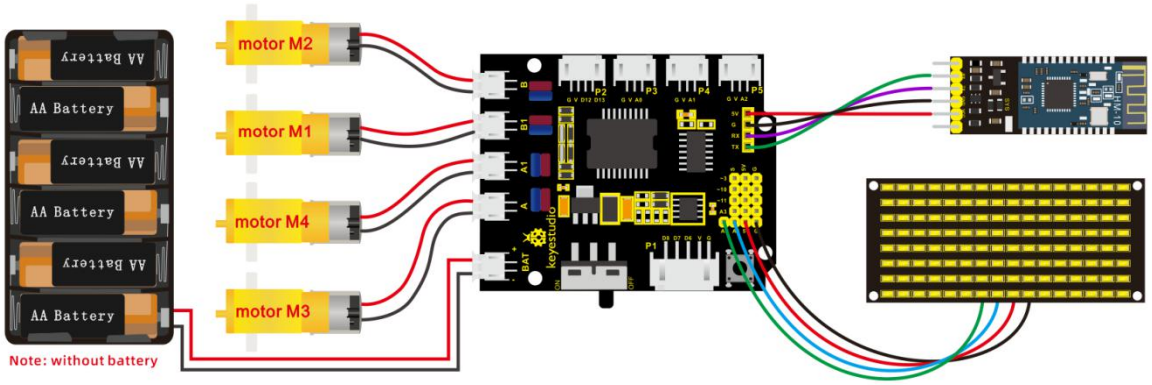
(2) Flow Chart

After designing the smart car according to the previous ideas, we need to start making smart cars according to the design ideas. We need to design the corresponding wiring, test code, and then wire the upload code to run,

to ensure that the smart car can achieve the desired function.



(3) Hook-up Diagram



(4) Test Code

The program will be generated if you find the following file and drag it into Mixly software.

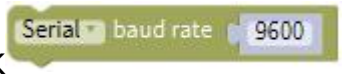
File	Route	File Name

Type		
MIX File	../tutorial for Mixly/Mixly Code/lesson_14_Bluetooth Remote Control	lesson_14.2_Bluetooth Control Smart Car

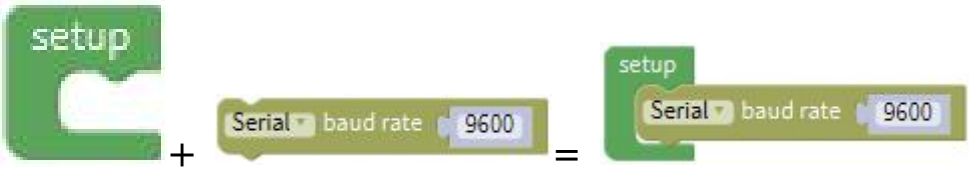
You could edit code step by step as follows:

(1) Click “Control” to find out block 

.

(2) Enter “Serial Port” to move block 

into 

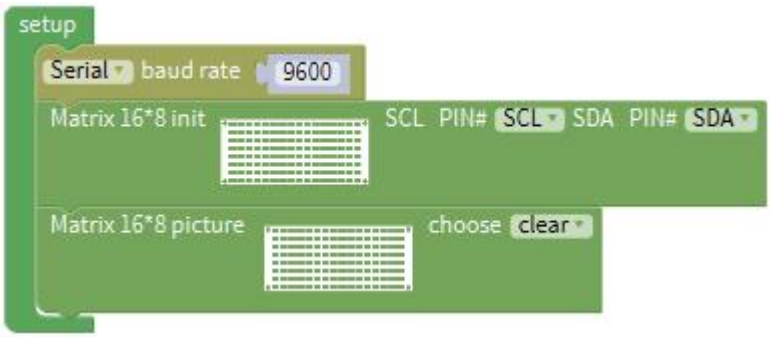


(3) Click “ Robot ” → “ 4wd-SmartCar ” → 

and 

(4) Leave them into block 

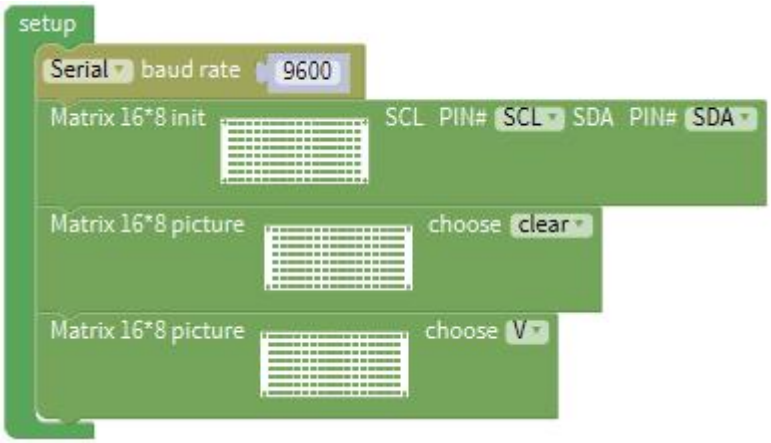
, and click“↑”to choose“clear”.



(5) Copy block 

once and keep it into 

, then click“clear”to select“V”.



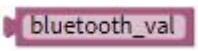
(6) Replicate “robot”→ “4wd-SmartCar”→ 

.

(7) Enter “ Serial Port ” to move block 

into block 

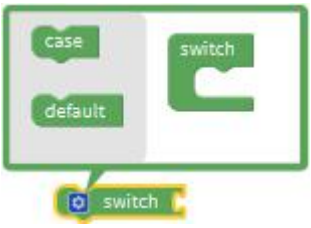
;

Go to “Variables” to move out block 

into 

block.

(8) Click “ Control ” to move out block 

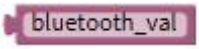
, tap 
, 

appears, drag 

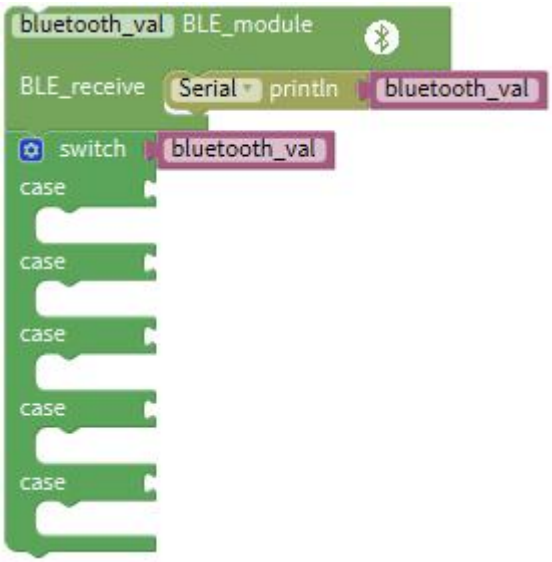
into block 

for 5 times , the

block is produced

(9) Go to “Variables” to drag block 

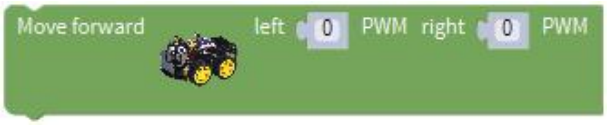
into“switch”block.

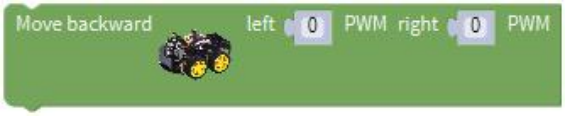


(5) Click “Text” to move out 

and copy it for 4 times. Respectively set to F, B, L, R and S.

(6) Enter “4wd-SmartCar”to move out 



,


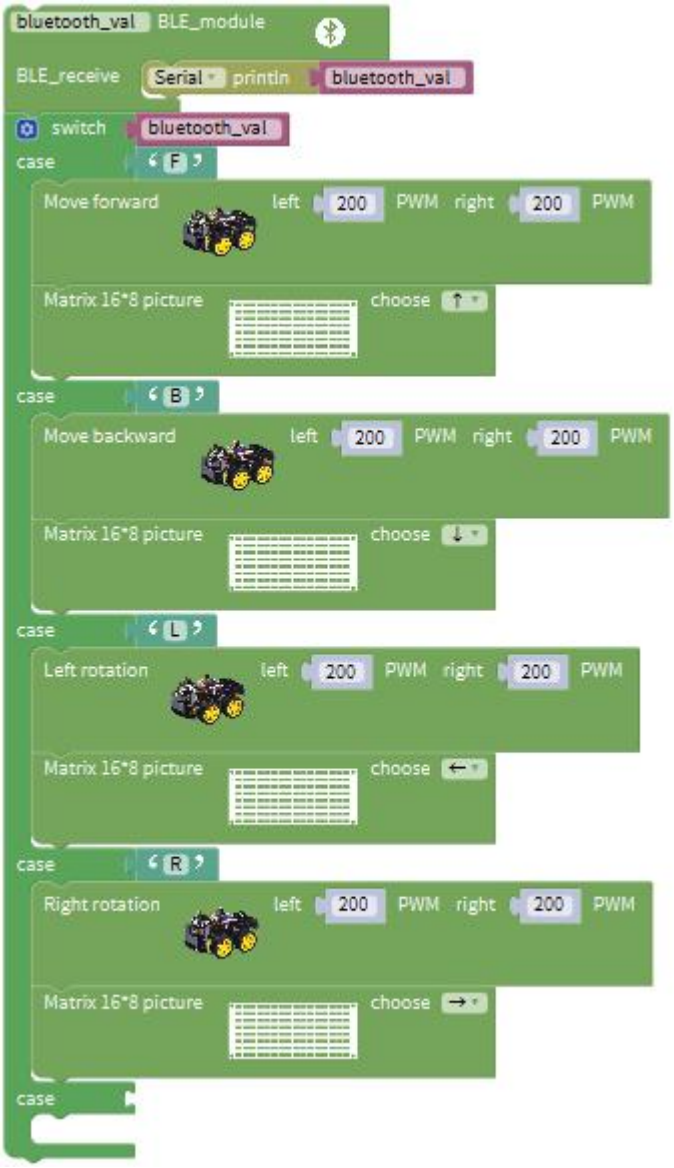
,

and

. Set to 200.

(7) Replicate  block for four times and separately select "↑", "↓", "←", "→"and STOP.

(8) Place the above blocks into "case" block and edit the whole code string as follows:



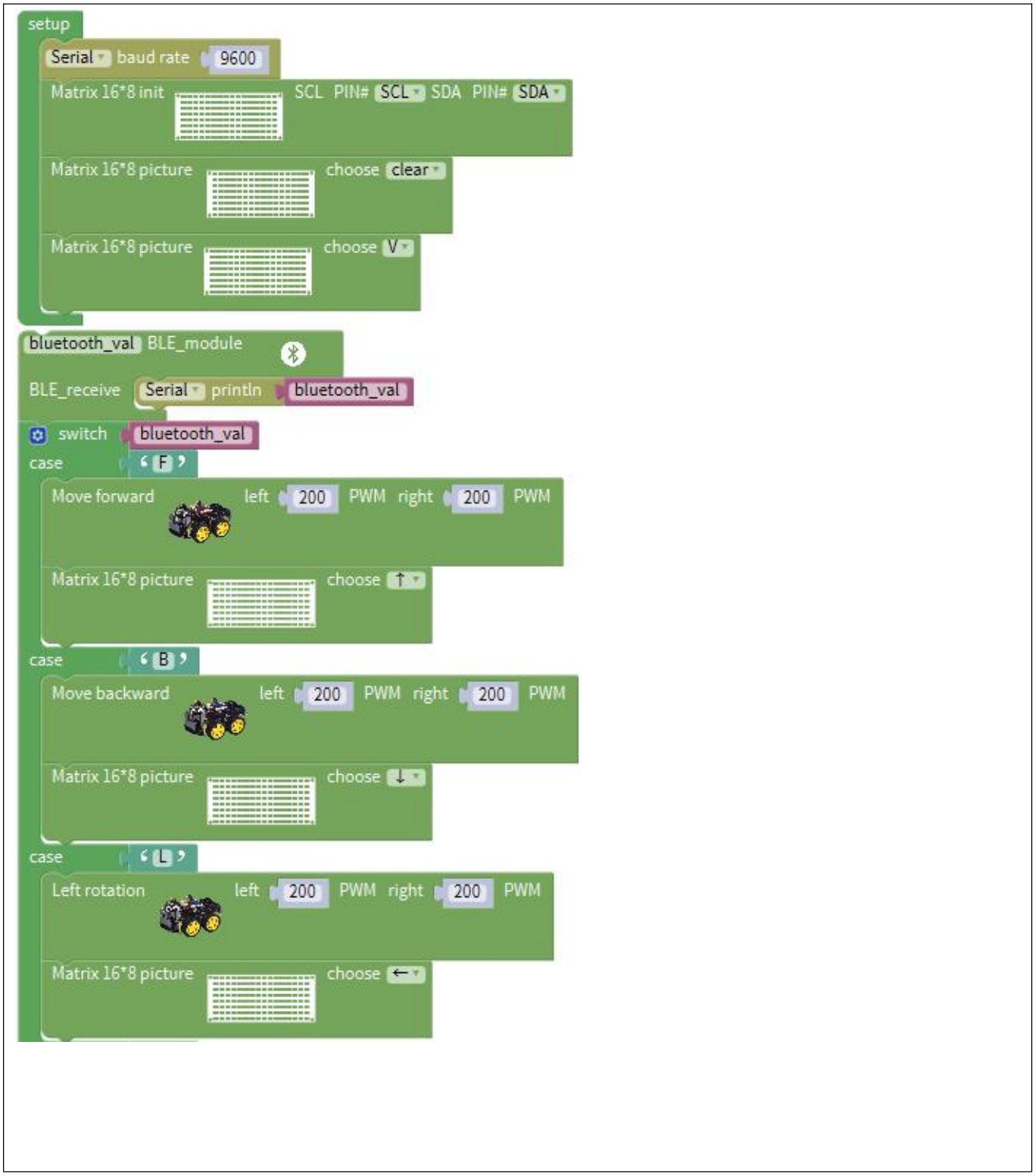
(14) Click“Text”to move  into the fifth case block ;

(15) Go to “ robot ” → “ 4wd-SmartCar ” →  and 

(16) Place them into the fifth case block, click“↑”to select“STOP”.


bluetooth_val BLE_module
BLE_receive Serial println bluetooth_val
switch bluetooth_val
case 'F'
Move forward left 200 PWM right 200 PWM
Matrix 16*8 picture choose ↑
case 'B'
Move backward left 200 PWM right 200 PWM
Matrix 16*8 picture choose ↓
case 'L'
Left rotation left 200 PWM right 200 PWM
Matrix 16*8 picture choose ←
case 'R'
Right rotation left 200 PWM right 200 PWM
Matrix 16*8 picture choose →
case 'S'
stop
Matrix 16*8 picture choose STOP

Complete Program :



Initialization
Set baud rate to 9600
Set to SCL and SDA for 8x16 dot matrix, SCL corresponds to A5 of V4.0 board, SDA to A4 of V4.0 boardclear the screen
8x16 dot matrix shows "V"
The received data is displayed on serial port.

When bluetooth_val is F, 4WD car moves forward
4WD car moves forward at the speed of PWM200
8x16 dot matrix shows "↑"
When bluetooth_val is B, 4WD car moves backward
4WD car moves backward at the speed of PWM200
8x16 dot matrix shows "↓"
When bluetooth_val is L, 4WD car rotates to left
4wd car rotates to left at the speed of PWM250
8x16 dot matrix shows "←"

When bluetooth_val is R, 4WD car rotates to right
4WD car rotates to right at the speed of PWM250
Dot matrix shows "→"
When bluetooth_val is S, 4WD car stops
4wd car stops
Dot matrix shows "STOP"

(5) Test Result

Upload the code on the keyestudio V4.0 board successfully. Stack the expansion board on the keyestudio V4.0 board and wire it according to the connection diagram. After power-on, the DIP switch will be dialed to the "ON" end. After connecting Bluetooth successfully, we can use the mobile APP to control the smart car to move.

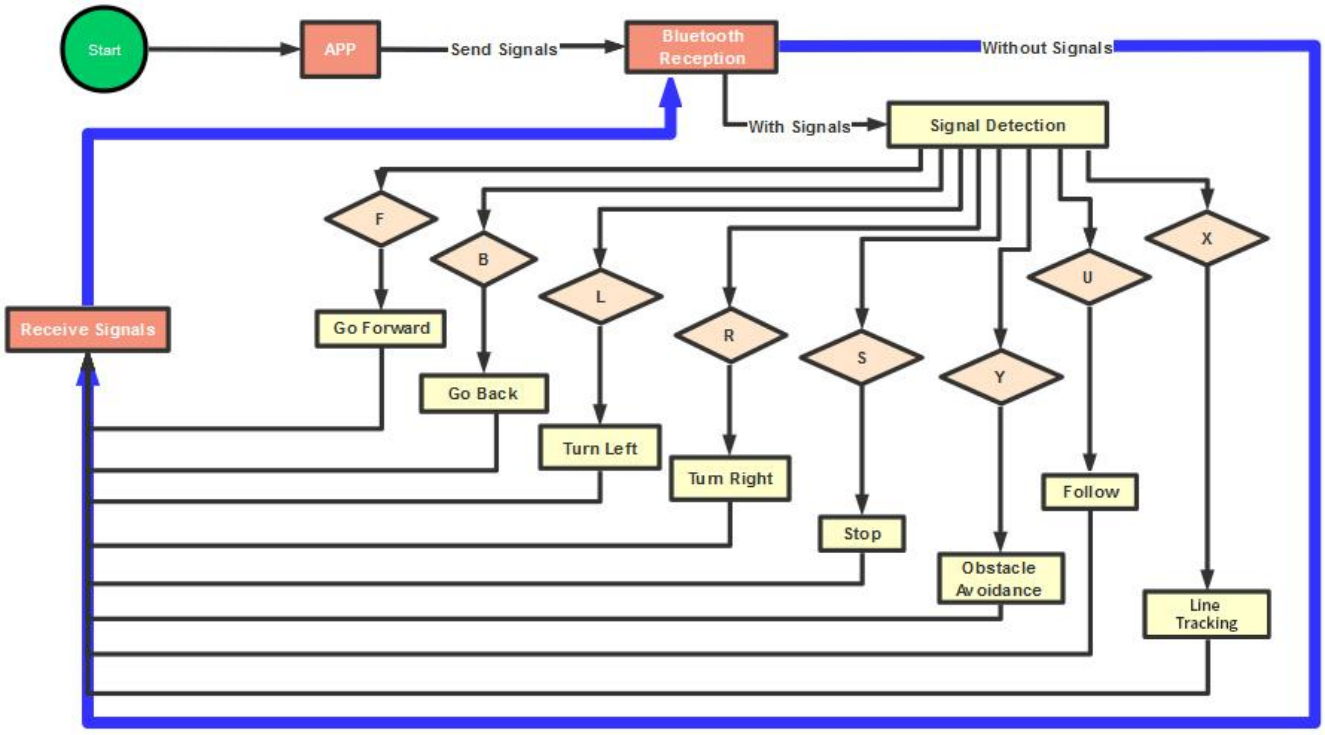
Project 15: Multi-purpose Bluetooth Robot

(1) Description

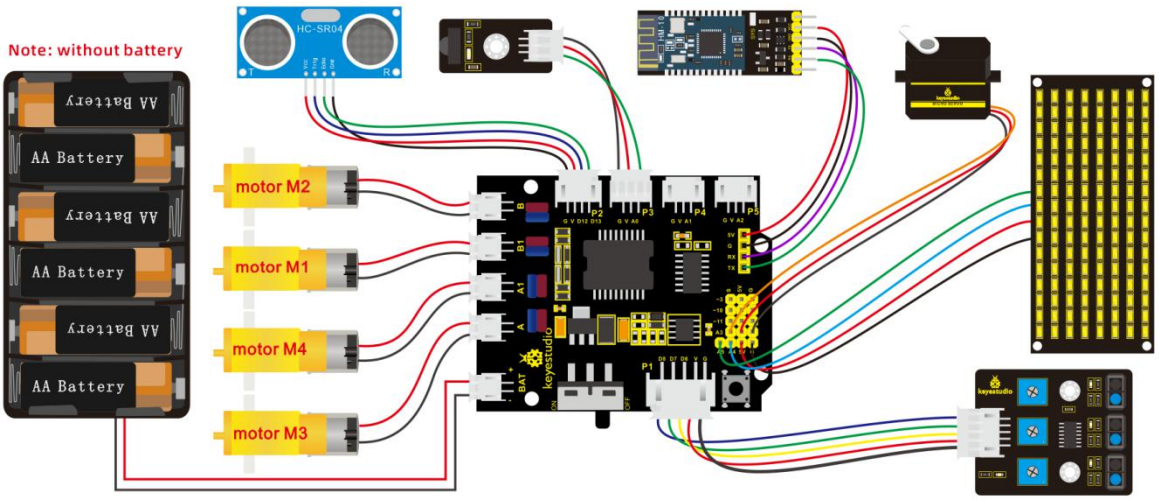
In previous projects, the robot car only performs single function, however, in this lesson, we integrate all of function to control smart car via Bluetooth control.

Here is a simple flow chart of multi-purpose robot car as for your reference.

(2) Flow Chart



(3) Hook-up Diagram

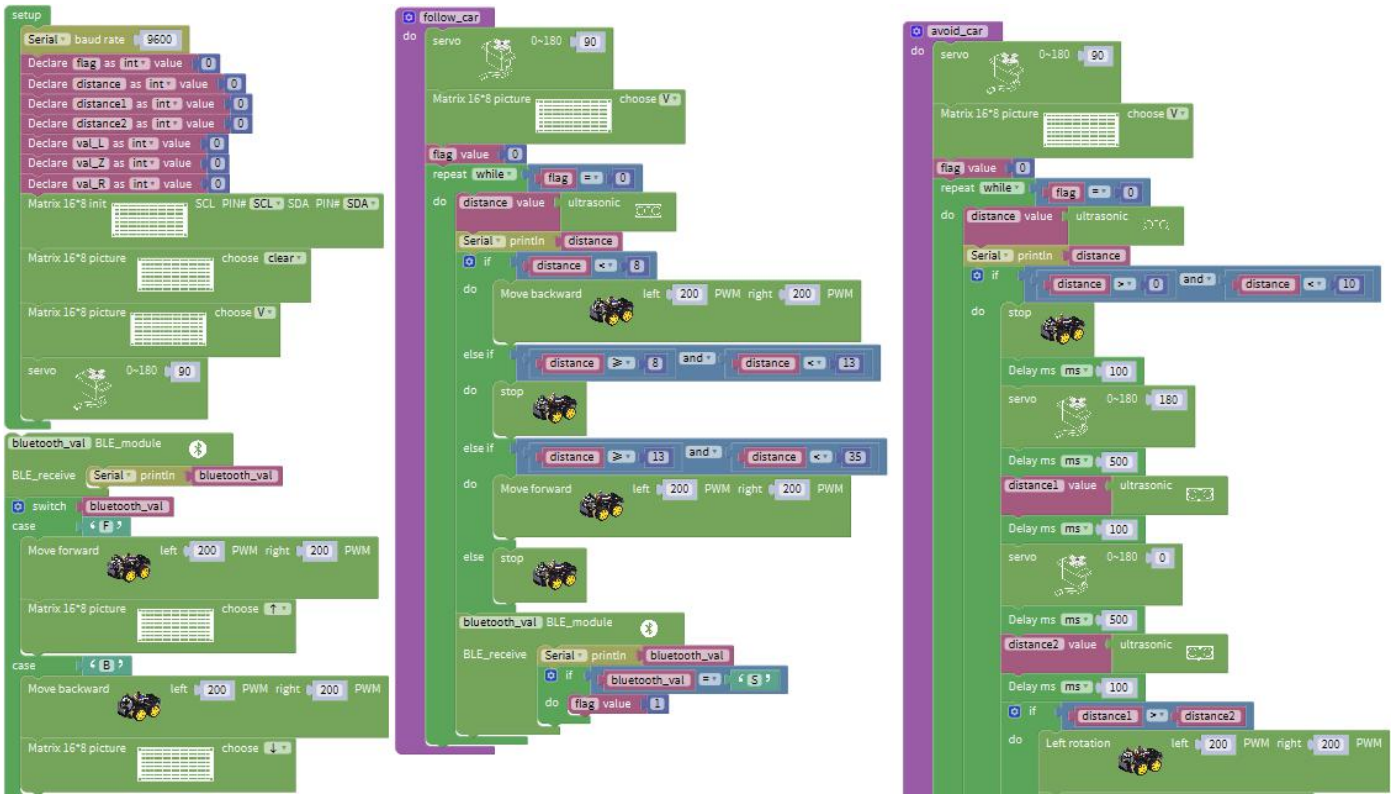


(4) Test Code

The program will be generated if you find the following file and drag it into Mixly software.

File Type	Route	File Name
MIX File	../tutorial for Mixly/Mixly Code/lesson_15_Multi-purpose Bluetooth Robot	lesson_15_Multi-purpose Bluetooth Robot

Complete Program :



(5) Test Result

Upload the code on the keyestudio V4.0 board. Stack the expansion board on it and wire them up according to the connection diagram. Plug in power and turn on switch on robot car. After connecting to Bluetooth successfully, we could control the smart car to move via App.

8. Resources

Wiki page: https://wiki.keyestudio.com/Main_Page

Official website: https://keyestudio.com/

Assembly Video Link: http://video.keyestudio.com/ks0470/