OSOYOO

Made in China
SKU:2022005100

OSOYOO BUILDING BLOCK ROBOT CAR KIT

Wifi Control | Ultrasonic Detection | Obstacle Avoidance | Line Tracking | Robot gripper

Learn STEM through hand-on projects

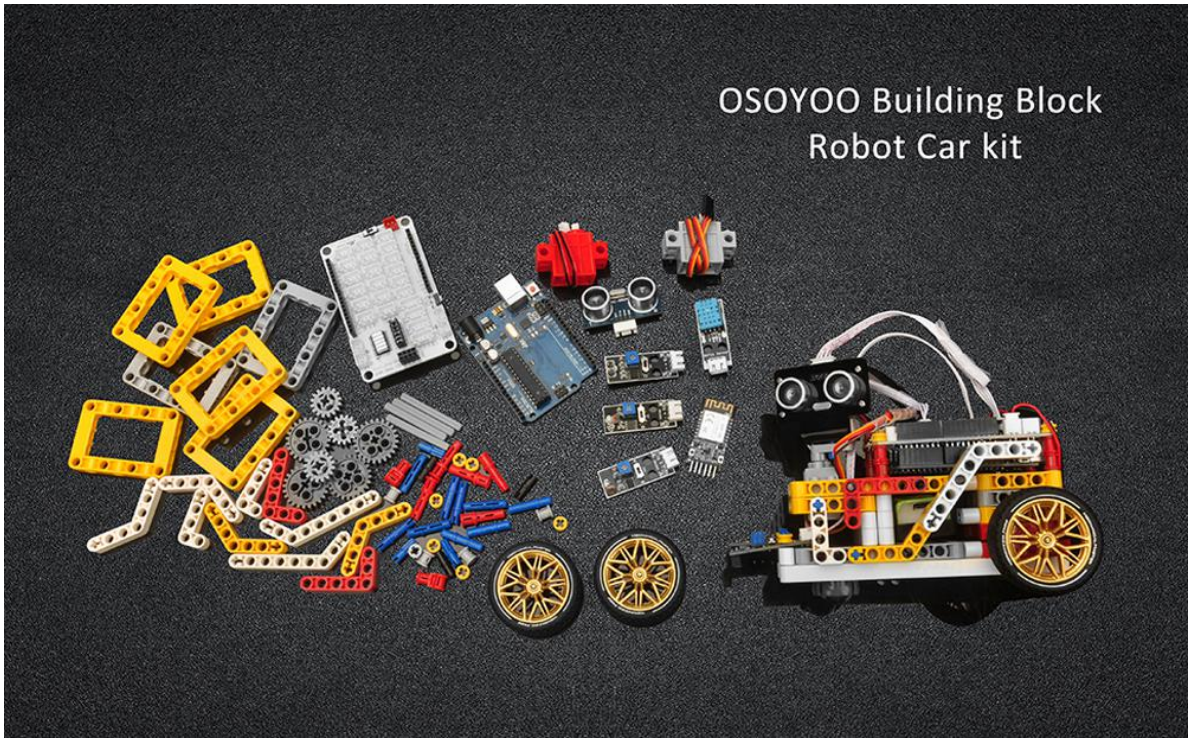
Increase visual perception and intelligence

Develop logical thinking and creativity

8+ AGES

WARNING: CHOKING HAZARD.
It contains small prats .
Not for children under 3 years.

Preface



About this kit

- 400+ parts includes bricks, gears, axis and 1 piece Controller board, expansion board, ultrasonic sensor, wifi module, servo motor, humidity and temperature, battery, 2pcs tracking sensor modules, Photoresistor, motors, and wheels
- The building Robot Car Kit designed to provide kids through DIY different Robot car to interesting and challenging building experience, robotic movement and mechanical structure
- The building Robot Car Kit helps to enhance kids' imagination and creativity, hand-eye coordination, self-confidence and promotes the importance of teamwork & collaboration

- This kit is based on OSOYOO basic board compatible with UNO. This is a an open source electronics archetype platform for electronics engineers, hobbyist, designers or anyone interested in creating interactive electronics projects
- This building Robot car kit has provide graphical programming (for basic projects) and IDE (for IOT projects). It's really suitable for teenage or adults to learn basic program language
- This learning kit provides 7 sample lessons including basic movement, following light, line tracking, obstacle avoidance, robot gripper, wifi remote control and IOT project. These tutorials teach how to build, code, and run the robot car
- OSOYOO Expansion board and Sensor modules are designed with XH2. 54 ports to avoid the many potential errors novice users encounter with similar products

Table of Contents

Preparation before Class 1: Graphical programming – mBlock

mBlock is designed for the Science, Technology, Engineering, Arts and Mathematics (STEAM) education. Inspired by Scratch 3.0, it supports both graphical and textual programming languages.

Software versions

Currently, the following versions are available:

- For PCs (software required): https://www.mblock.cc/en-us/download
- Web version (no software required): https://ide.mblock.cc/
- For Android and iOS: Search for **mBlock** on any app store to download it

By signing in to mBlock 5, you can have your projects automatically stored in the cloud. Try synchronizing your projects across devices (such as PCs and mobile devices) now!

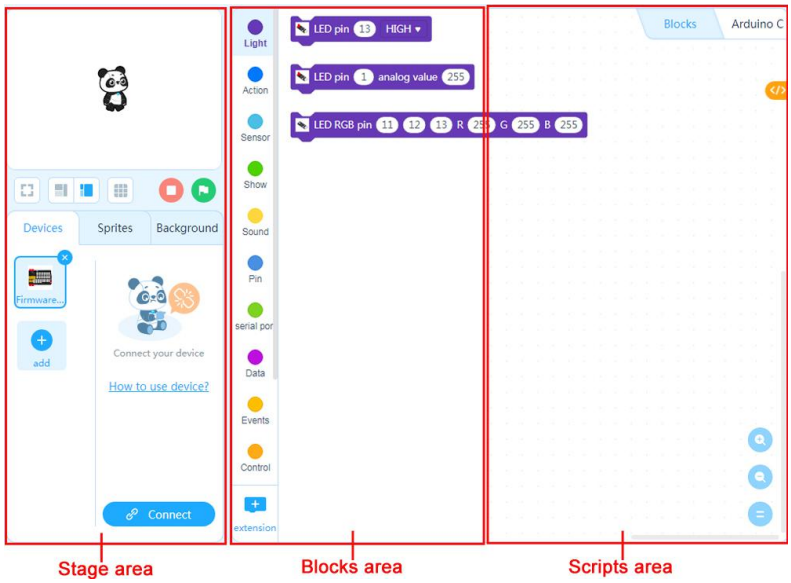
Get to Know the UIToolbar

Toolbar

1. Language: Click it to change the UI language.
2. File: Click it to create, open, save a project, import a project from your PC, or export your project to PC.
3. Edit: Click it to turn on/off the stage turbo mode or hide/unhide the stage.
4. Title: Click it to set or change the title of the current project.

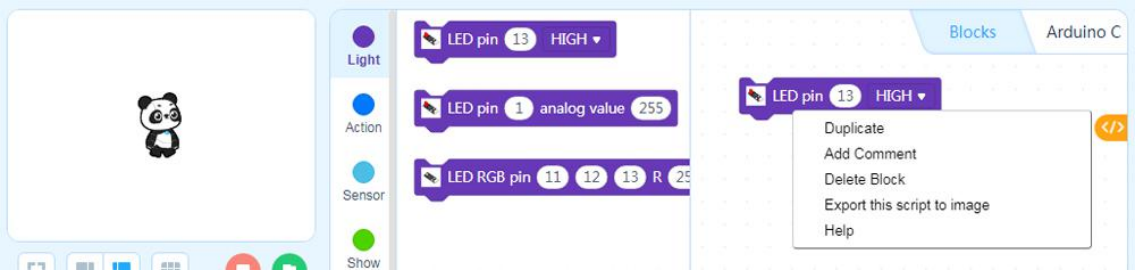
5. Save: Click it to save the current project to My Projects.
6. Publish: Cick it to publish the current project to the mBlock community.
7. Courses: Click it to visit the website: https://education.makeblock.com/resource/
8. Tutorials: Click it to view online user guide and example programs.
9. Feedback: Click it to give your feedback for mblock.
10. More: Click it to see more functions about Check For Updates, About mBlock, About version 5.4.0, Serial driver quick install, Set as default editor, Cooperation
11. Sign up/Sign in: Click it to sign up an mBlock account or sign in. After signing in to mBlock 5, you can click it to view your projects, profile, account center, and cloud service authentication code, or click to sign out.
12. Python Editor: Click it to enter mBlock-Python Editor.

Editing area

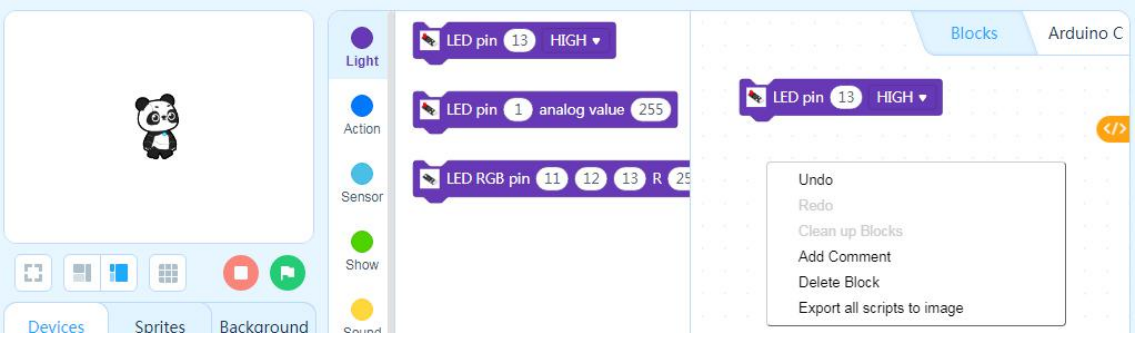


- **Stage area**: You can present your designs, connect devices, and set your sprites and backdrops here.
- **Blocks area**: You can find the blocks you need by category and color here.

- **Scripts area**: You can compile your program by dragging blocks to this area.



When you drag a block in Scripts area, and right click on the block, you will can do these operation for the block as the above picture



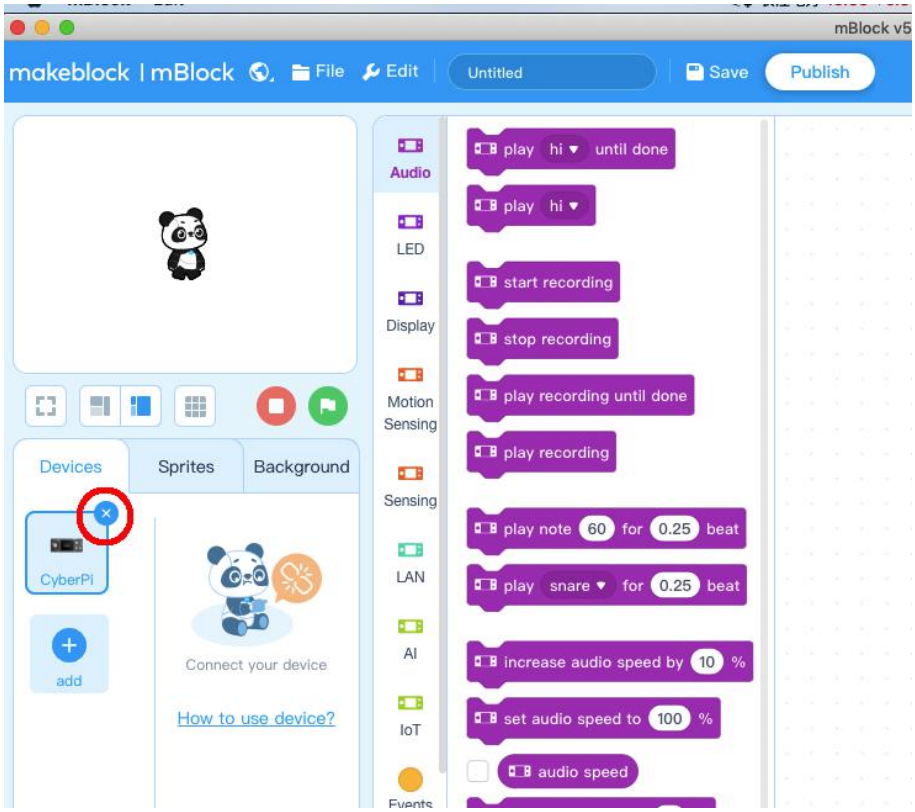
When you drag a block in Scripts area, and right click on blank, you will can do these operation for the block as the above picture

Connect OSOYOO basic board

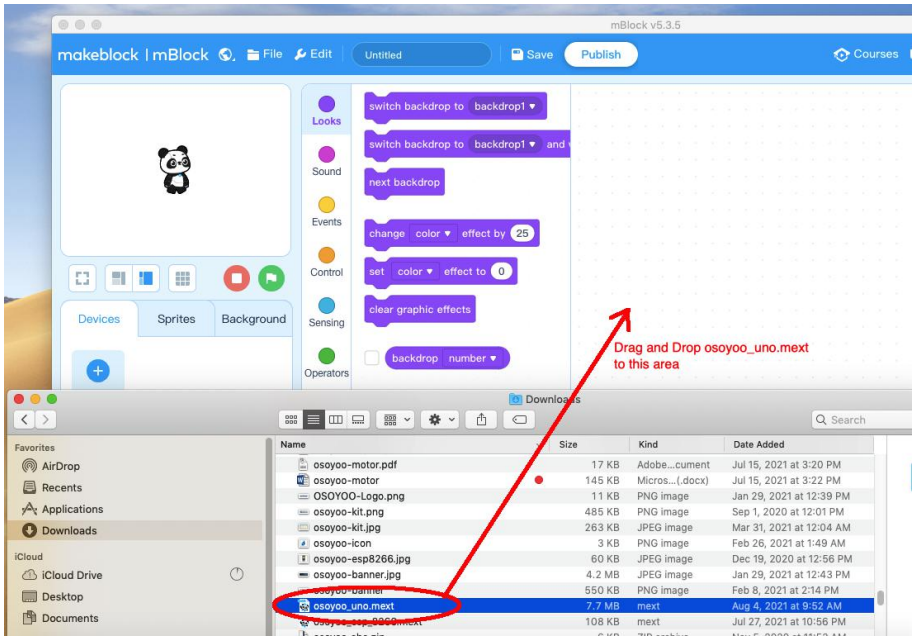
Step1) Download OSOYOO_UNO.mext device file from

https://osoyoo.com/driver/mblock/osoyoo_uno.mext

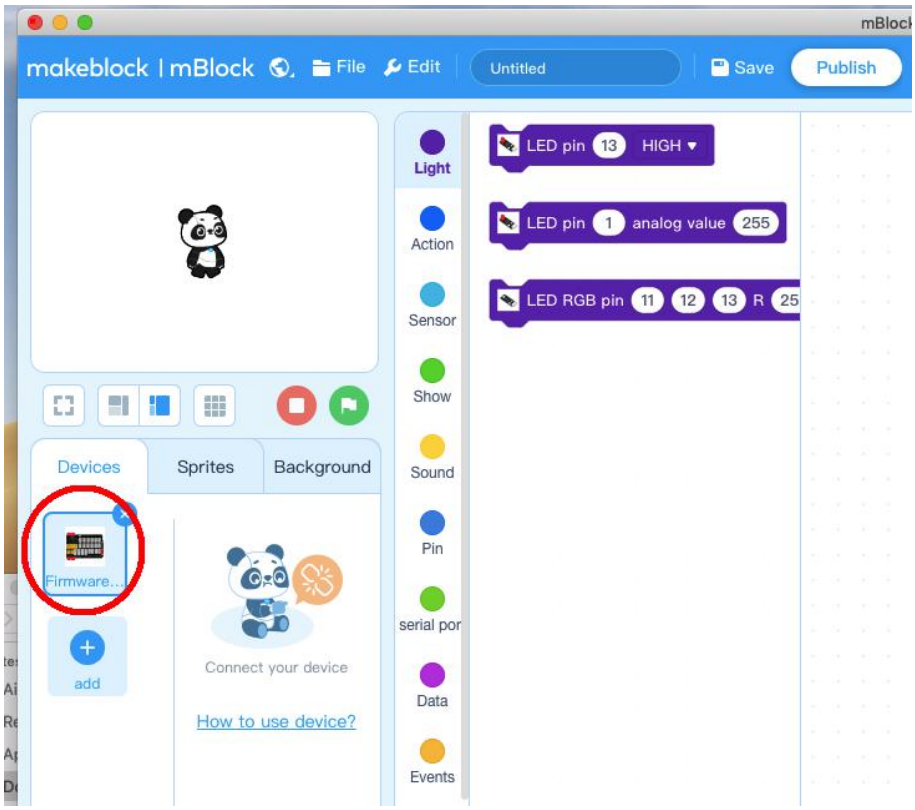
Step2) Run the mBlock PC software by double click the lovely Panda icon. you will see mBlock UI as following picture. Please delete the default device **CyberPi** by click the cross in the red circle.



Step3) Drag and Drop osoyoo_uno_mext file(downloaded in Step 1) to mBlock software as following:



Now you will see a new device firmware in mBlock, see following picture:



Now mBlock software and OSOYOO_UNO device firmware have been successfully installed in our PC!

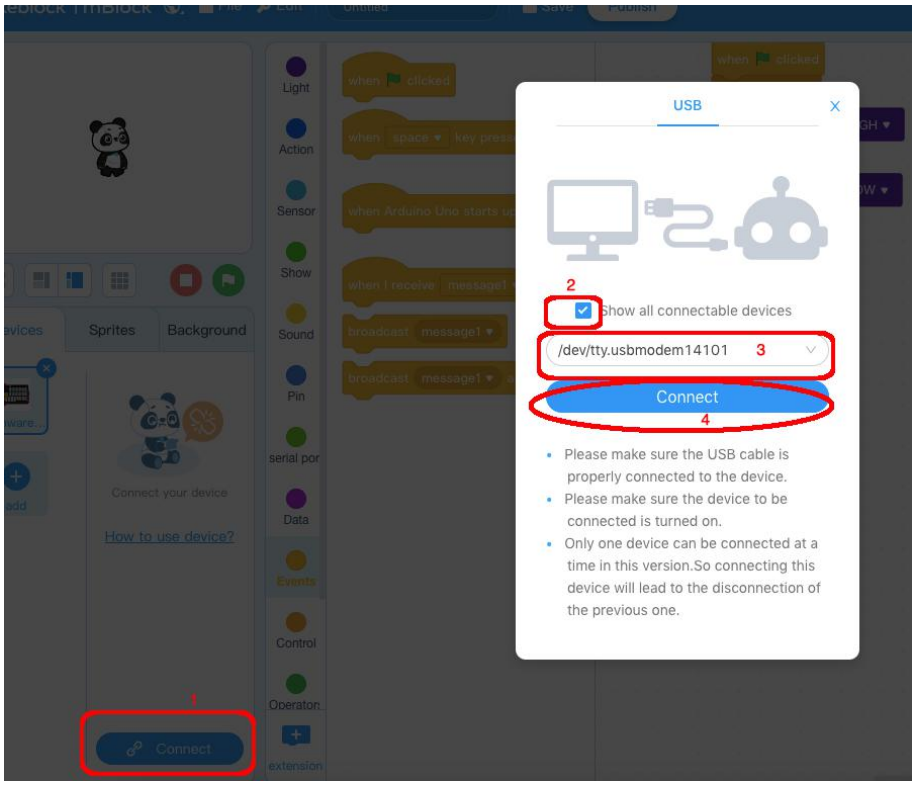
Upload program to OSOYOO Basic board

1) You need connect your board to your PC with a USB cable first. Then click the Connect button in the bottom of the mBlock software, you will see a USB window pop up,

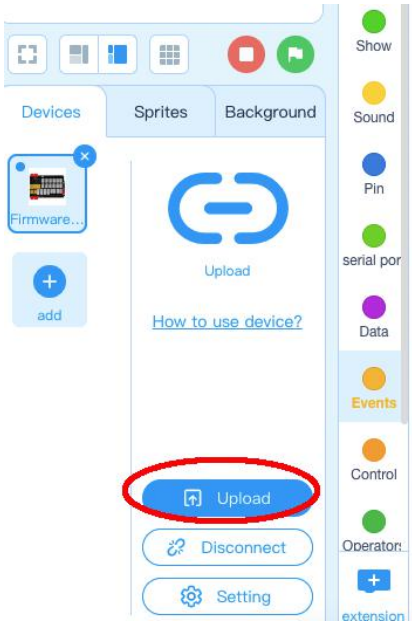
2) Select Show all connectable device check box , then a device drop-down menu will show up,

3) Select your port from device drop-down menu

4) Click Connect button to connect your PC to your board



5)After you PC is connected to your board, please click Upload button in the bottom of your software, then the code will be uploaded to your board:



For more information, please click user guide to learn more or visit the link:

https://www.yuque.com/makeblock-help-center-en/mblock-5

Preparation before Class 2:
Download and Install Arduino IDE

Introduction

This lesson will walk you through downloading, installing, and testing the IDE (short for Arduino Integrated Development Environment).Before you jump to the page for your operating system, make sure you've got every thing prepared.

Preparation:

- A computer (Windows, Mac, or Linux)
- An Arduino-compatible microcontroller(Osoyoo NodeMCU here)
- A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer (check out this USB buying guide if you're not sure which cable to get).

Extended Reading：

If you are new to Arduino general, check below links and you will be more familiar with it:

- What is Arduino and why we want to use it

- Introduction of Arduino IDE

If you're ready to get started, click on the link in the column on the left that matches up with your operating system, or you can jump to your operating system here.

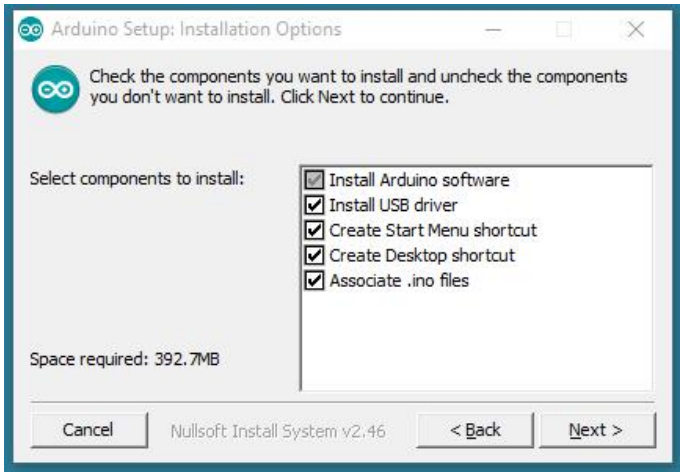
- Windows
- Mac
- Linux

INSTALL THE SOFTWARE (IDE) ON WINDOWS PC

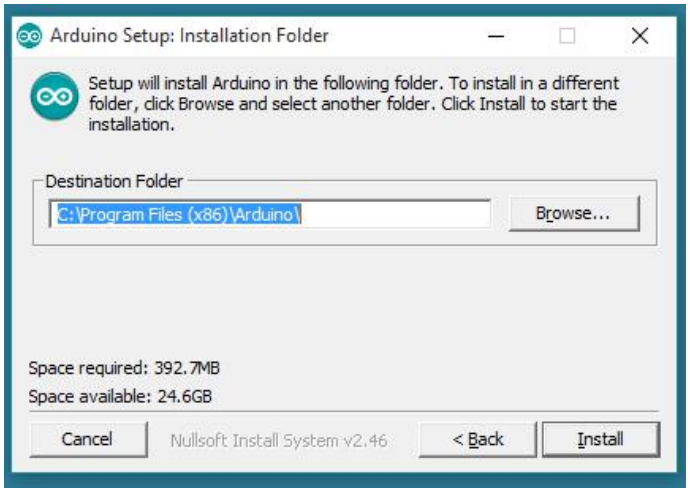
Download the Software (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

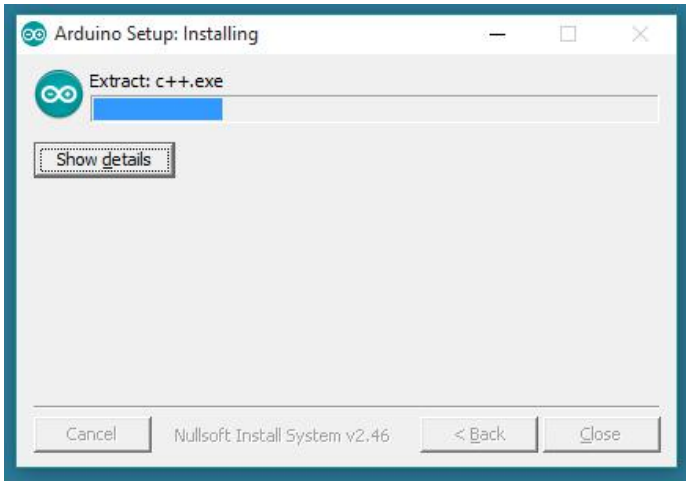
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.



Choose the components to install



Choose the installation directory (we suggest to keep the default one)



The process will extract and install all the required files to execute properly the Software (IDE)

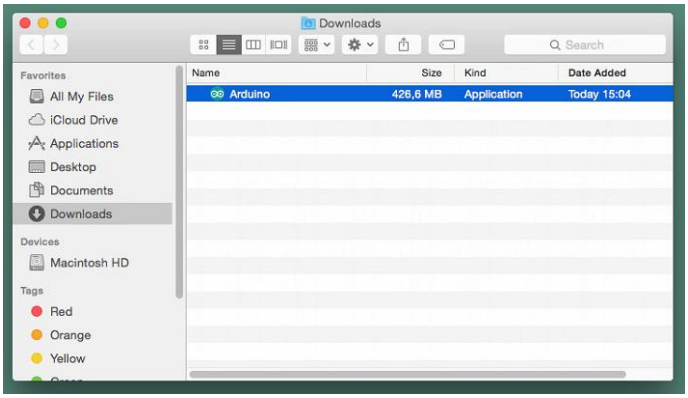
Proceed with board specific instructions

When the Software (IDE) is properly installed you can go back to the Getting Started Home and choose your board from the list on the right of the page.

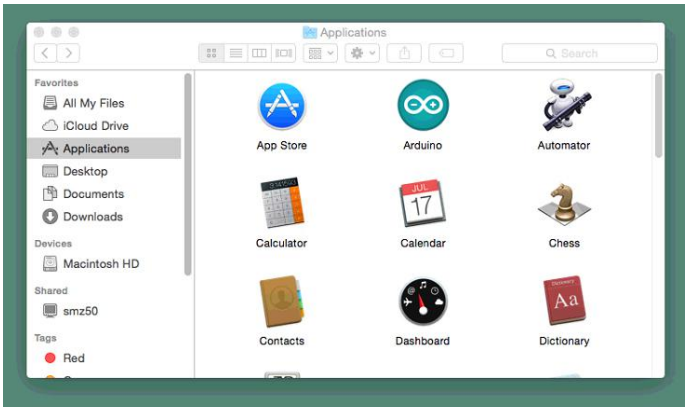
INSTALL THE SOFTWARE (IDE) ON MAC OS X

Download the Software (IDE)

Get the latest version from the download page. The file is in Zip format; if you use Safari it will be automatically expanded. If you use a different browser you may need to extract it manually.



Copy the application into the Applications folder (or elsewhere on your computer).



Proceed with board specific instructions

When the Software (IDE) is properly installed you can go back to the Getting Started Home and choose your board from the list on the right of the page.

INSTALL THE SOFTWARE (IDE) ON ON LINUX

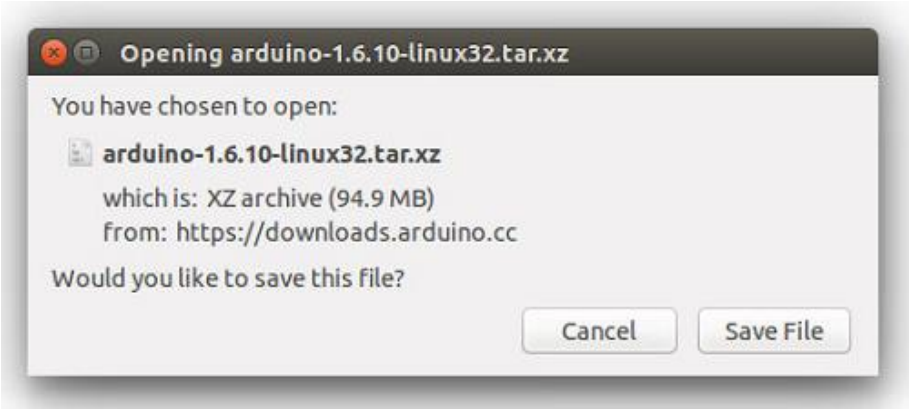
Quick Start

The Linux build of the Software (IDE) is now a package that doesn't require ansy specific procedure for the various distributions availabe of Linux. The only relevant information is the 32 or 64 bit version of the OS.

Download the Software (IDE)

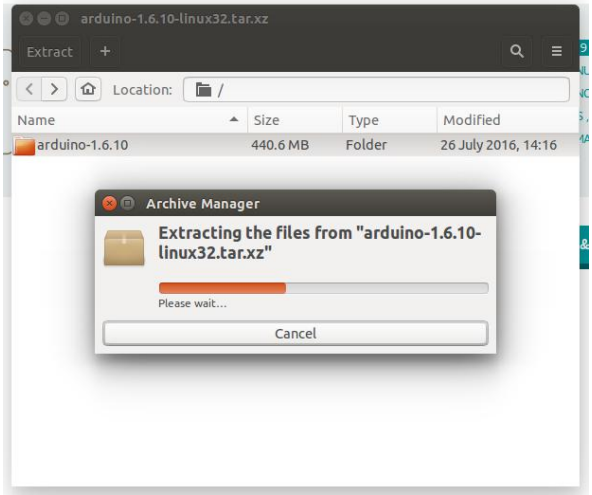
Get the latest version from the download page. You can choose between the 32, 64 and ARM versions. It is very important that you choose the right version for your

Linux distro. Clicking on the chosen version brings you to the donation page and then you can either open or save the file. Please save it on your computer.



Extract the package

The file is compressed and you have to extract it in a suitable folder, remembering that it will be executed from there.

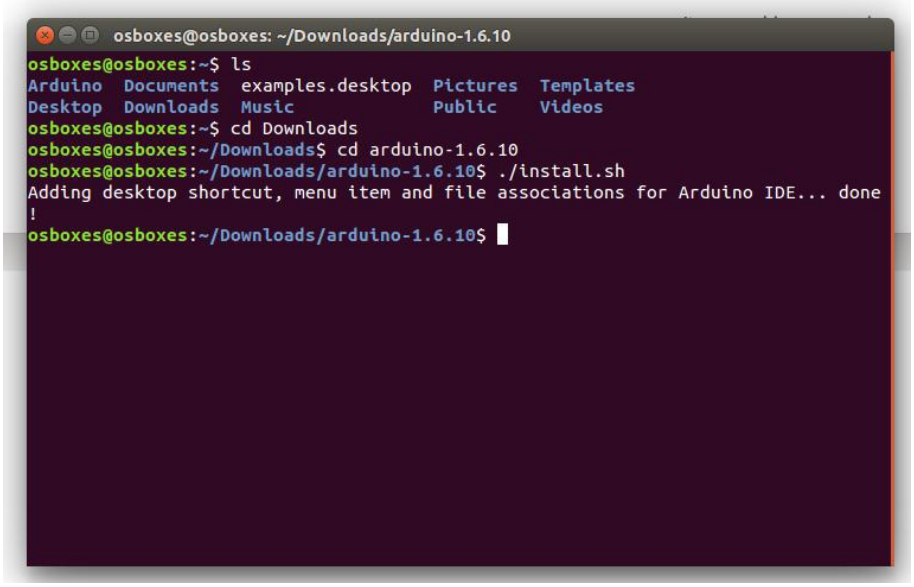


Run the install script

Open the **arduino-1.6.x** folder just created by the extraction process and spot the **install.sh** file. Right click on it and choose **Run in Terminal** from the contextual menu. The installation process will quickly end and you should find a new icon on your desktop.

If you don’t find the option to run the script from the contextual menu, you have to open a Terminal window and move into the **arduino-1.6.x** folder. Type the

command **./install.sh** and wait for the process to finish. You should find a new icon on your desktop.



Proceed with board specific instructions

When the Software (IDE) is properly installed you can go back to the Getting Started Home and choose your board from the list on the right of the page.

Please Read…

It might hapen that when you upload a sketch – after you have selected your board and serial port -, you get an errorError opening serial port … If you get this error, you need to set serial port permission.

Open Terminal and type:

ls -l /dev/ttyACM*

you will get something like:

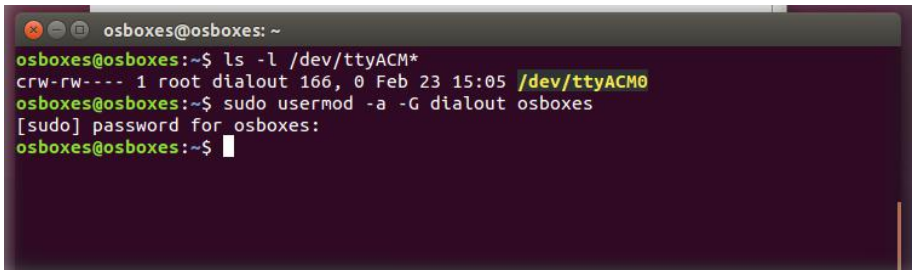
crw-rw---- 1 root dialout 188, 0 5 apr 23.01 ttyACM0

The “0” at the end of ACM might be a different number, or multiple entries might be returned. The data we need is “dialout” (is the group owner of the file).

Now we just need to add our user to the group:

sudo usermod -a -G dialout

where is your linux user name. **You will need to log out and log in again for this change to take effect.**



This is the procedure to access the serial port from the Software (IDE) if you get an error

After this procedure, you should be able to proceed normally and upload the sketch to your board or use the Serial Monitor.

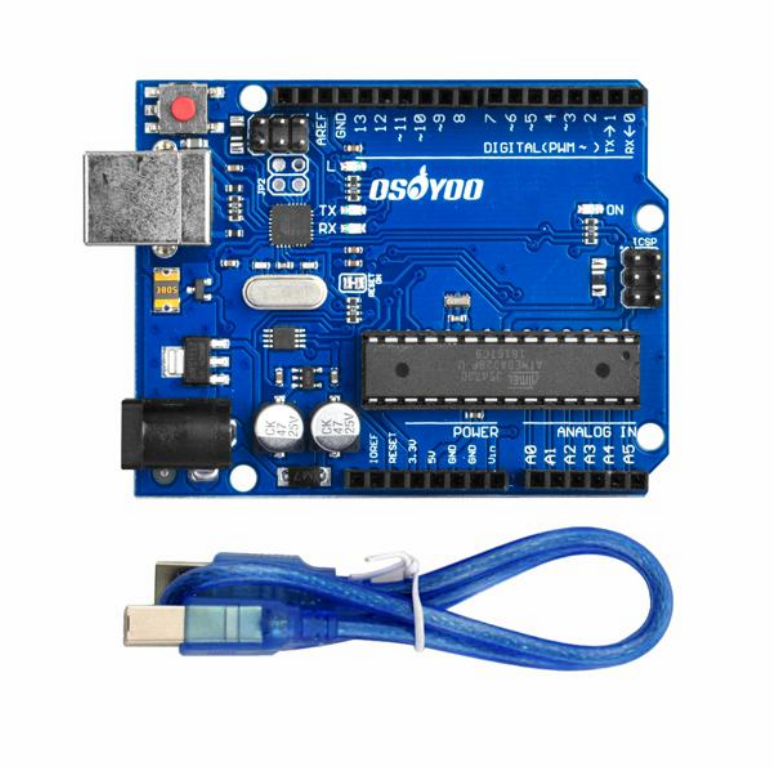
This paper is from:https://www.arduino.cc/en/Guide/HomePage,more info please click:https://www.arduino.cc/en/Guide/Environment

Preparation before Class 3: OSOYOO basic board for Arduino UNO

CONTENT

4. Schematics
5. Documentations
    1. Programming
    2. Warnings
    3. Differences with other boards
    4. Power
    5. Memory
    6. Input and Output
    7. Communication
    8. Automatic (Software) Reset
    9. Revisions

“The UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.” ————————- www.Arduino.cc



Overview

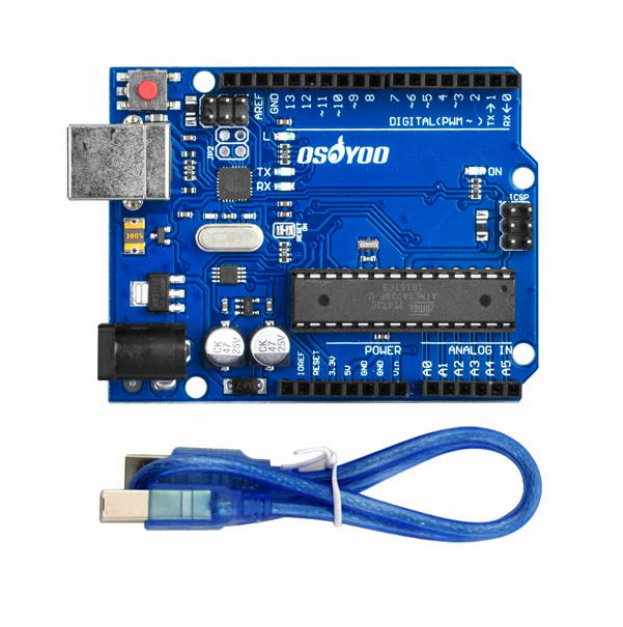
The Osoyoo basic Board is fully compatible with Arduino UNO rev.3, it is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your board without worring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

“Uno” means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The board is the first

in a series of Arduino boards, and the reference model for Arduino platform; for an extensive list of current, past or outdated boards see the index of boards.

Note:

This is an Arduino Compatible board. It is NOT an original Arduino board, but is similar. None of the Arduino Uno R3 boards sold on the internet at this price are original, they are all copies. This is perfectly legal, seeing that the whole Arduino ecosystem is open source! Please note this board is manufactred by Osoyoo! We do have control of the brand and quality of components used! We have also carefully selected suppliers that consistently supply quality products. We strictly control the quality of the products before leaving the factory. The excellent after-sales service and professional technical support will ensure you have a good time with Osoyoo UNO Board

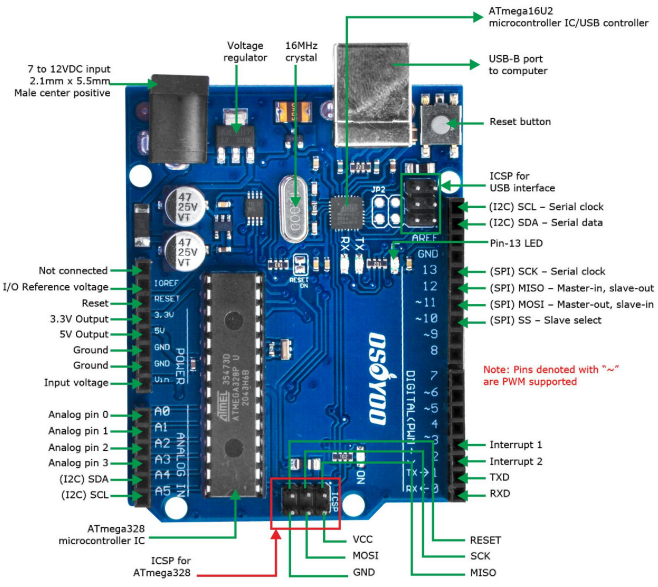


Specifications of Osoyoo UNO

- Microcontroller: ATmega328P-PU
- Operating Voltage: 5V

- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

Pinout of Osoyoo Basic Board



Schematics

Arduino Uno is open-source hardware! You can build your own board using the following files from the Arduino official site:

- Arduino_Uno_Rev3 Eagle files
- Arduino_Uno_Rev3 schematic PDF file
- Arduino_Uno_Rev3 DXF file

Documentations

PROGRAMMING

The OSOYOO Basic board can be programmed with the (Arduino Software (IDE)). Select “Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the board comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then rese ing the 8U2.

- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

WARNINGS

The board has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

DIFFERENCES WITH OTHER BOARDS

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

POWER

The board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- Vin. The input voltage to the board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 – 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

MEMORY

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

INPUT AND OUTPUT

See the mapping between pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.

PIN MAPPING ATmega328P

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(),digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

COMMUNICATION

The board has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on

the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which

it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

REVISIONS

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

  The Osoyoo basic Board is 100% Software and Hardware compatible with Arduino UNO Board, you can get more info from www.arduino.cc.

  Thanks for their efforts, it's easier for us to learn Arduino！

Preparation before Class 4:
OSOYOO Magic I/O Shield for Arduino

Arduino Uno is the most popular Arduino board, however it is often frustrating when your UNO board needs connect many sensors and actuators with jumper wires . In worst cases, wrong connection of wires can burn Arduino board , cause circuit short-cut fire and make your project become a nightmare.

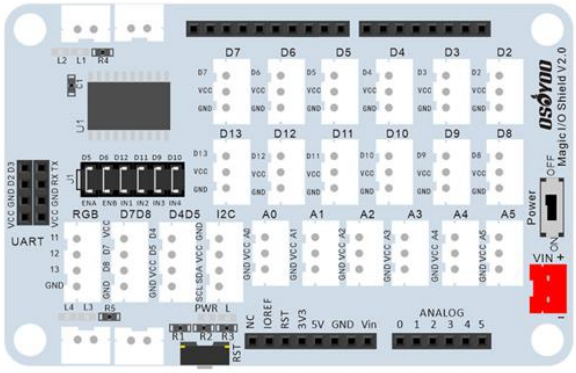
The purpose of creating the OSOYOO Magic I/O Shield for Arduino is to help people, especially beginners to make wire connection simple,convenient and safe.

With the rich Plug and Play(PnP) connectors on the base board, you can connect standardized OSOYOO sensor/actuator modules with the Arduino Uno through this shield conveniently!

This OSOYOO Magic I/O Shield fits standard Arduino UNO compatible board. It is a connection bridge between the electronic block module and UNO board. It is has a build-in motor driver similar to L293/L298, which can provide adjustable power to inductive loads such as relays, solenoids, DC and stepping motors. It can drive two channels of DC motors with your Arduino board, controlling the speed and direction of each one independently.

It extends SPI port, UART, I2C, PWM and analog INPUT pins for Arduino board, so DIY enthusiasts, interactive designers can quickly attach the modules to the board for Arduino and accelerate project development progress. An intelligent power regulator allows you to connect a battery and USB cable simultaneously, and an

on-board voltage switch makes it easy to change the output voltage, making 3.3V or 5V sensors compatible. With the plug-and-play interfaces, this board simplifies all your needs for an autonomous robot such as line tracer, obstacle avoidance, maze solver etc., The shield can be interfaced with 2 motors, 12 digital sensors, 6 analog sensors, a line array sensor and an ultrasonic sensor (4 pin and 3 pin).



FEATURES

– Compatible for Arduino UNO R3, Leonardo R3, Mega2560 R3

– All sockets onboard are XH 2.54mm pitch.

– Motor socket number: 4

– I2C socket number: 1

– RGB socket number: 1

– 4PIN socket number: 4

– Power socket number: 1(DC 5V~12V)

– Software Uart interface number: 1

– Hardware Uart interface number: 1

– Analog input socket number: 6(A0, A1, A2, A3,A4,A5)

– Digital pin socket number：12（D2~D13）

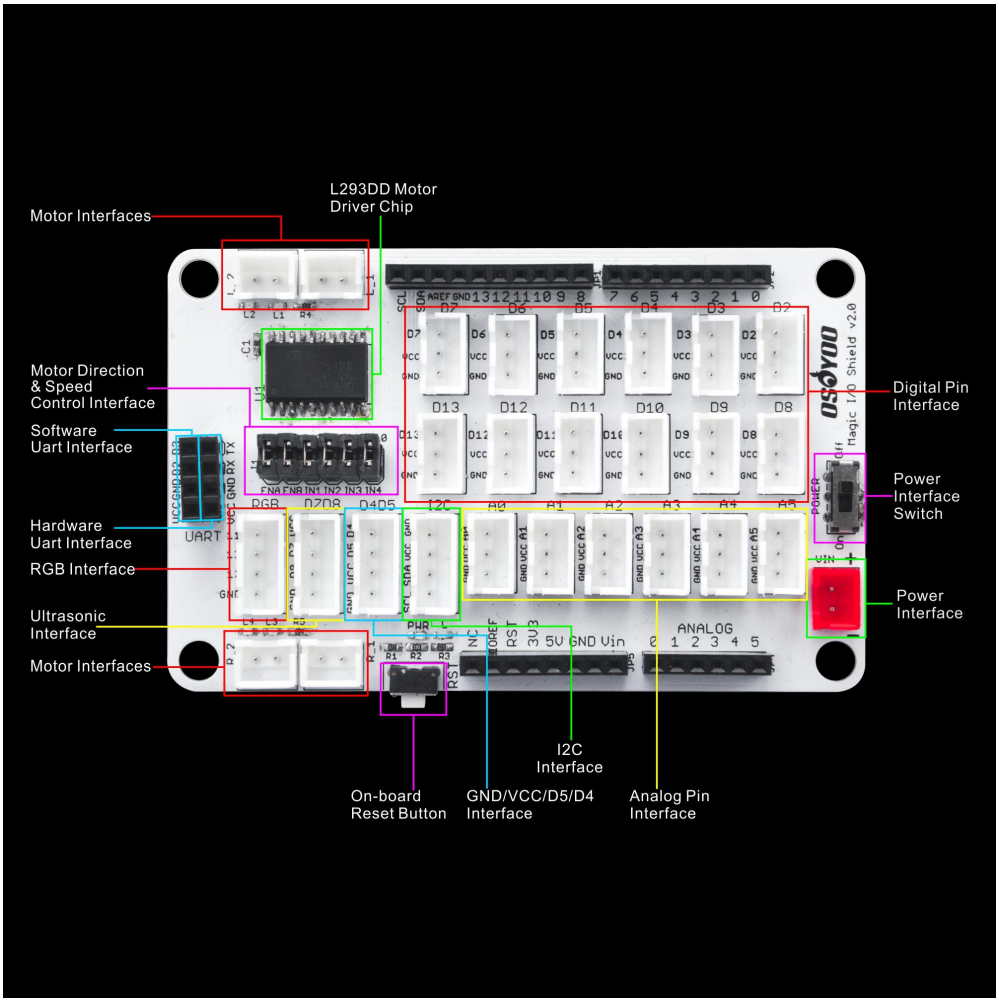
– Motor Driver socket: for OPEN-SMART car chassis

– L LED for D13 pin

-3.3V/5V Operating Voltage Switch

– On-board reset button, power indicator led.

WHAT'S ON BOARD



1. Power Interface

2. Power indicator led

3. L LED for D13 pin

4. 3.3V/5V Operating Voltage Switch

5. On-board reset button

6. Digital Pin Interface

7. Analog Pin Interface

8. I2C Interface

9. Software Uart Interface

10. Hardware Uart Interface

11. L293DD Motor Driver Chip

12. 4 Motor Interfaces

13. Motor Direction Control Interface

14. Motor Speed Control Interface

Easy to plug OPEN-SMART products with XH2.54 socket to this shield and do not need to pay attention to the connection. So it is great for Training institutions and visual programming education.

TECHNICAL DETAILS

Dimensions:82mm x56mm x20mm

Weight:G.W 32g

Lesson 1: Basic Car

OBJECTIVE

In this lesson, we will show how to assemble a simple robot car with OSOYOO Building Blocks and use mBlock to program its movement

.

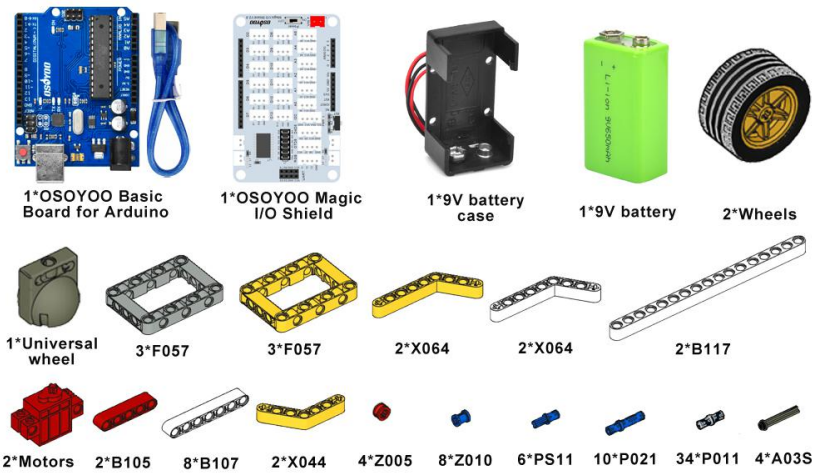
PARTS & DEVICES

Please prepare the following parts to complete this project

NOTE:

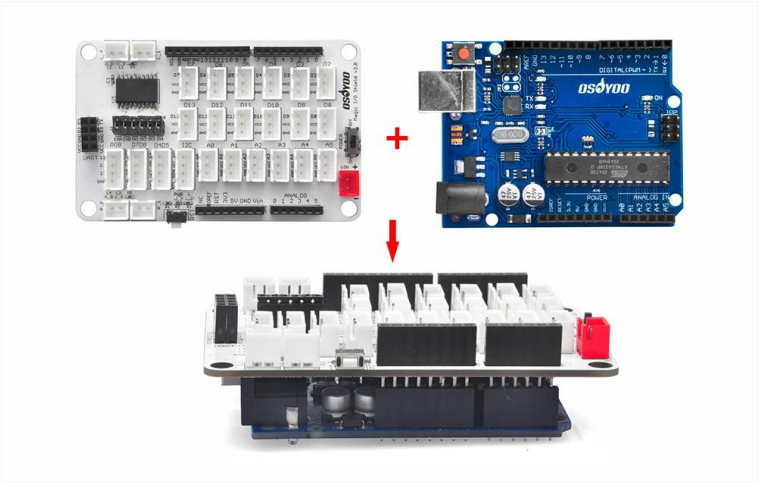
1.the color of the building block is subject to the actual product, which does not affect the use.

2. ALL OSOYOO PRODUCTS FOR ARDUINO ARE THIRD PARTY BOARD WHICH IS FULLY COMPATITABLE WITH ARDUINO

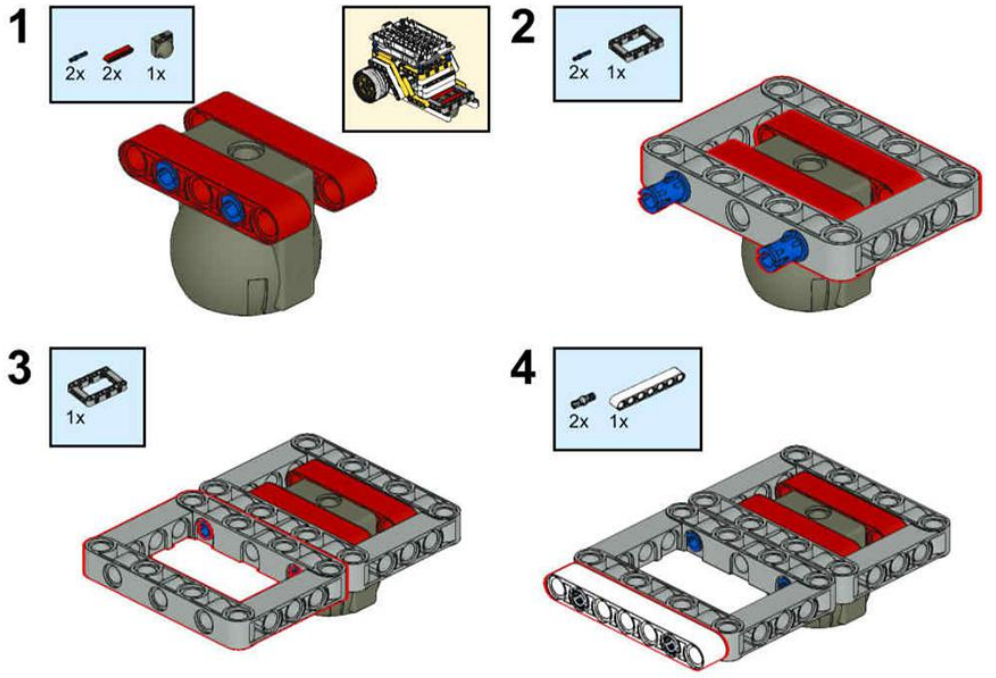


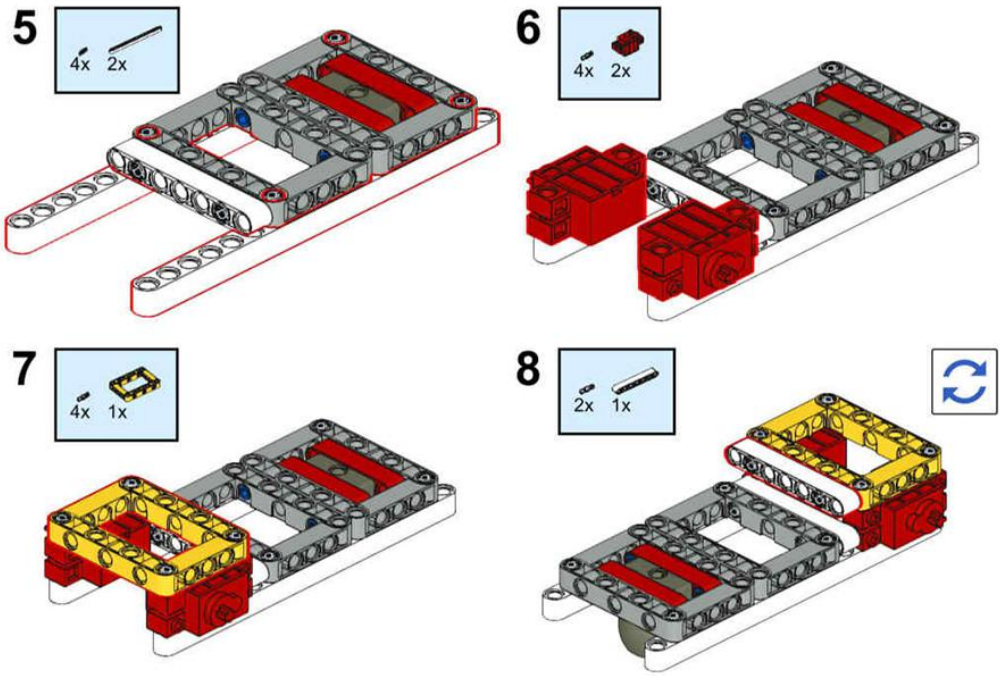
HOW TO MAKE

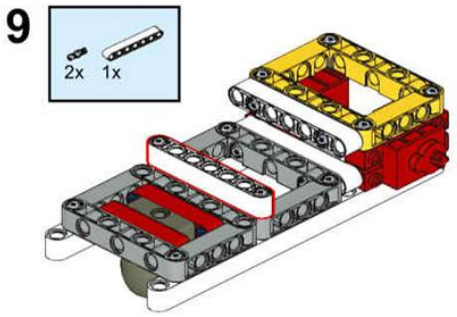
Before you build the robot with blocks, please install OSOYOO basic board for Arduino under OSOYOO Magic I/O shield as following (*Attention please : the pins of I/O shield is* aligned with the port of basic the board firstly, then press the shield tightly on the board).

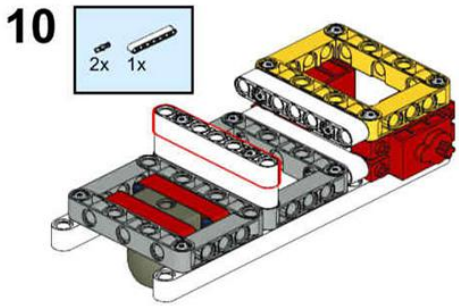


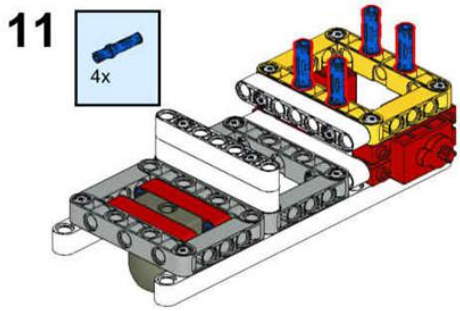
Please follow the building steps to build this robot car, If you want to get clear PDF building steps, please download from https://osoyoo.com/picture/Building_Robot_Car/lesson1/lesson1.pdf

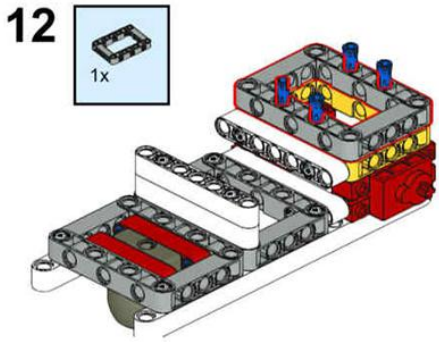


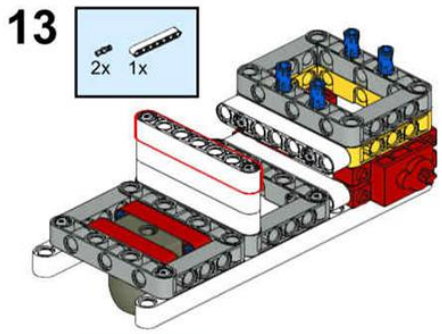


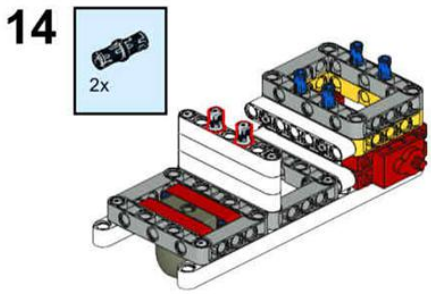

9
2x
1x

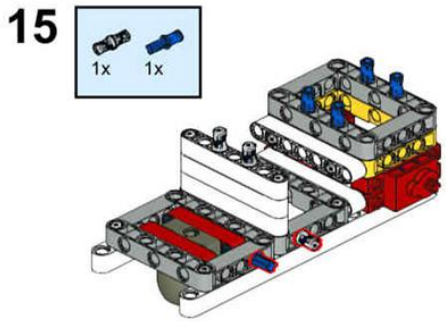

10
2x
1x

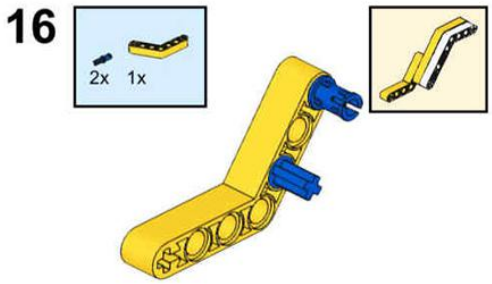

11
4x


12
1x

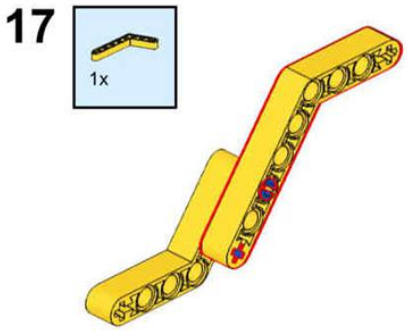

13
2x
1x


14
2x

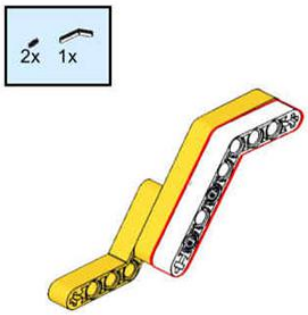

15
1x
1x


16
2x
1x

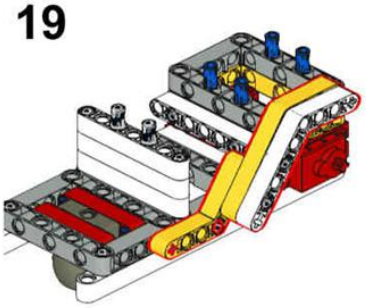
17



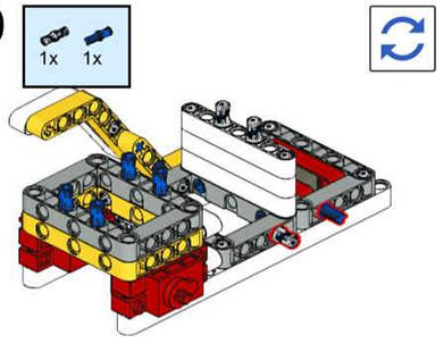
18



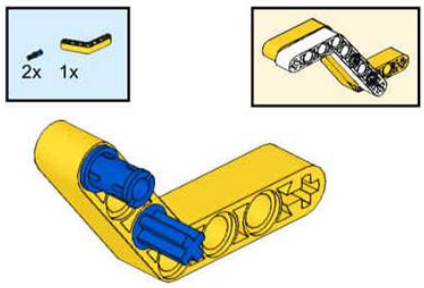
19



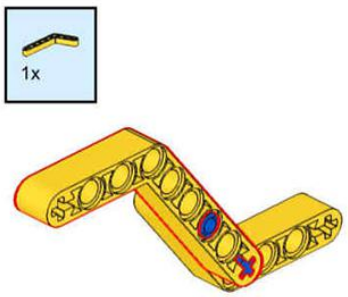
20



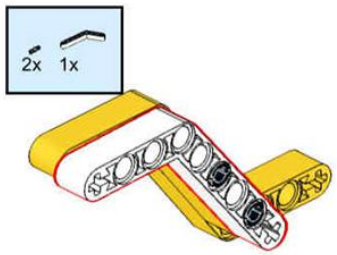
21



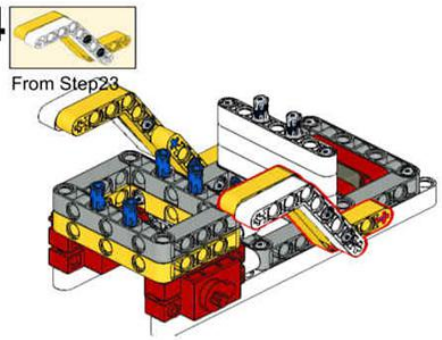
22

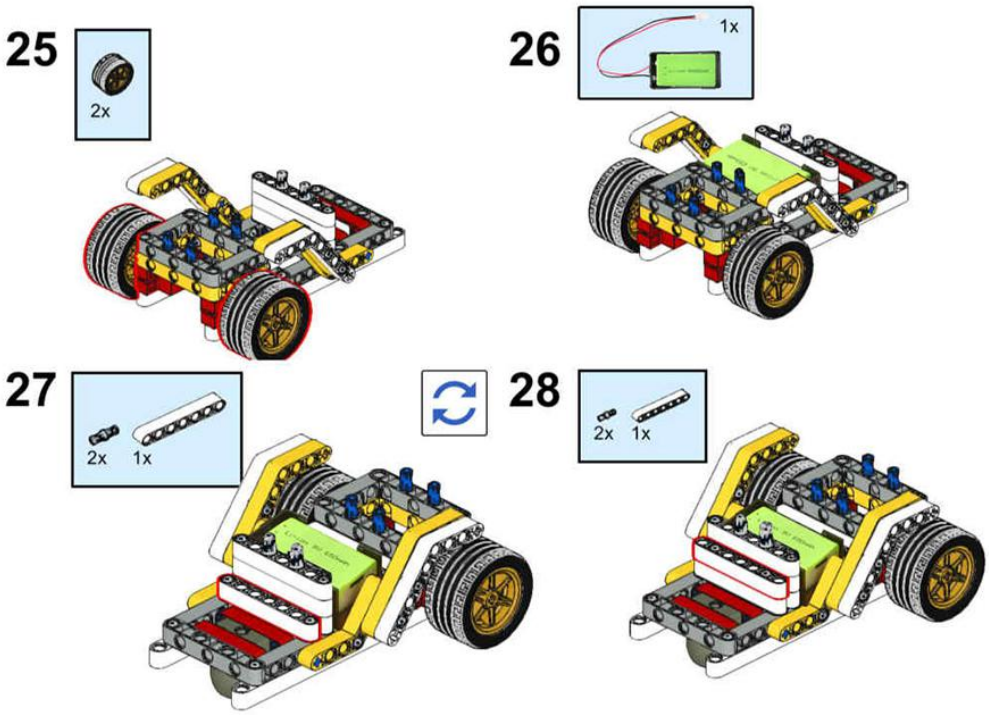


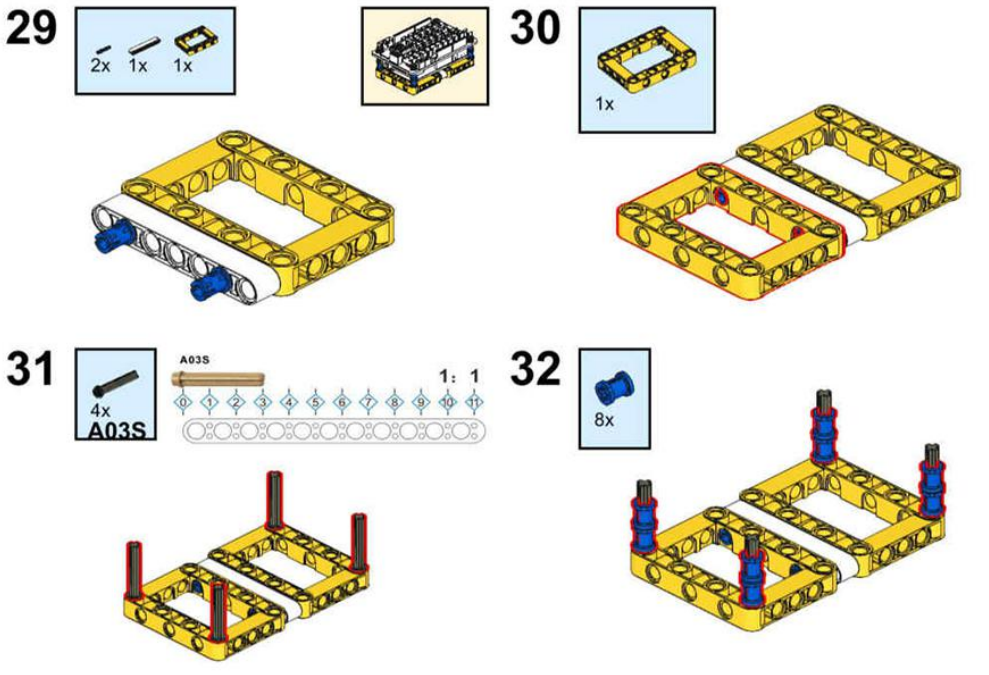
23

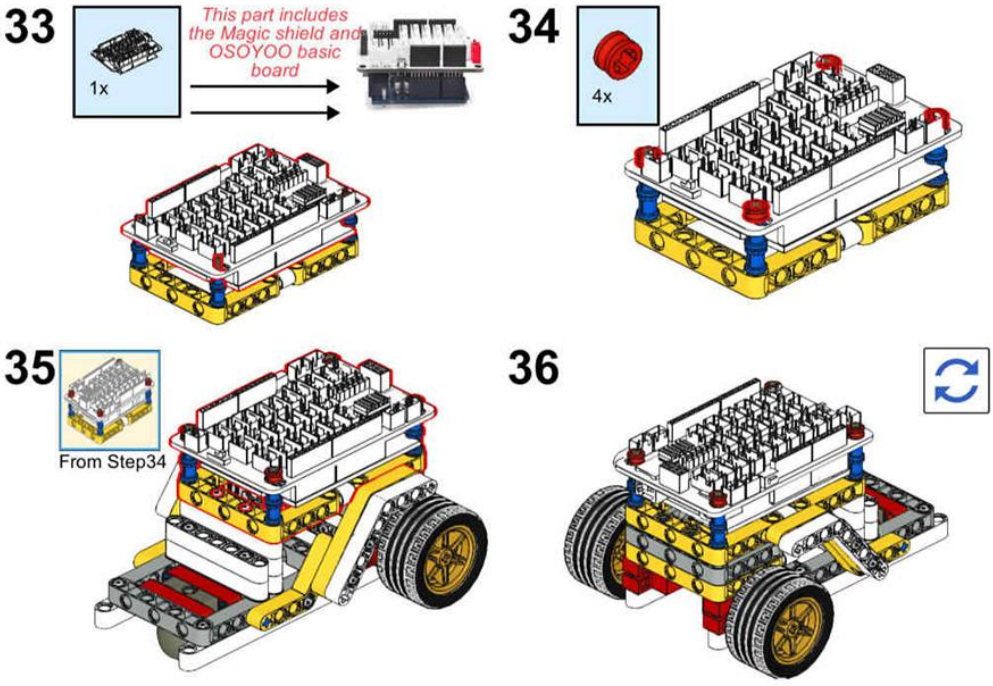


24



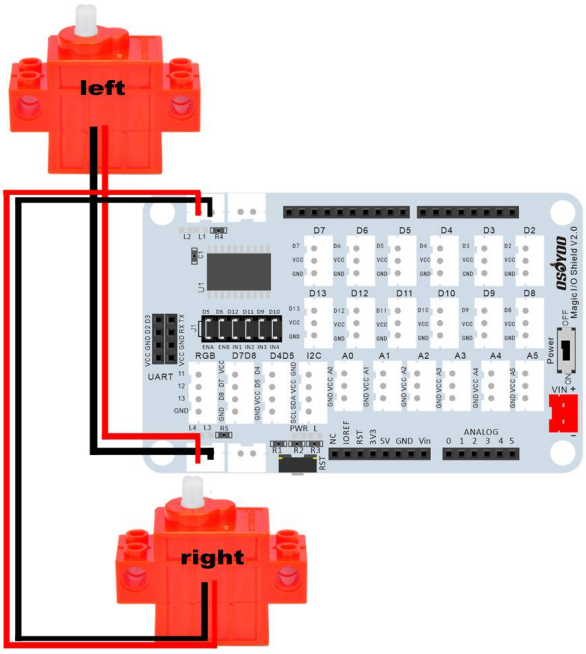

25
2x
26
1x
27
2x 1x
28
2x 1x


29
2x 1x 1x
30
1x
31
4x
A03S
A03S
1: 1
32
8x

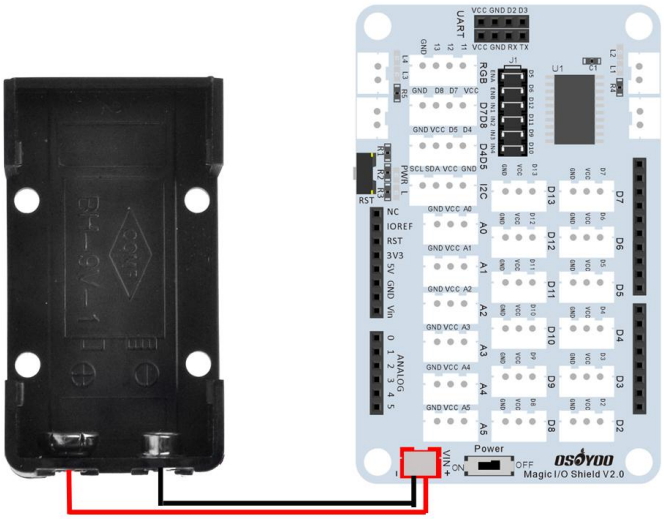


CIRCUIT CONNECTION

Connect Left wheel motor to R1 or R2 port of Magic I/O shield; Connect Right wheel motor to L1 or L2 port as per following picture (Attention please: there are six jumper caps on ENA/ENB/IN1/IN2/IN3/IN4)



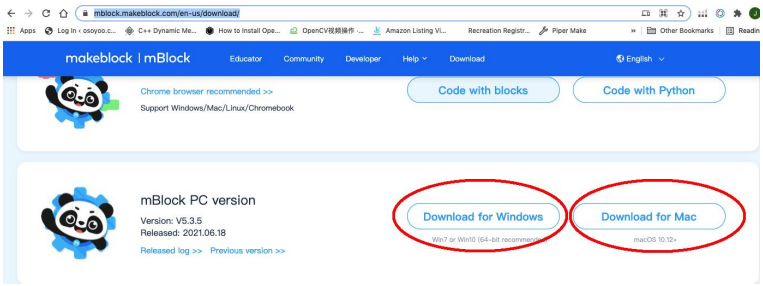
Connect 9V battery case to power port of Magic I/O shield as following:



HOW TO CODE

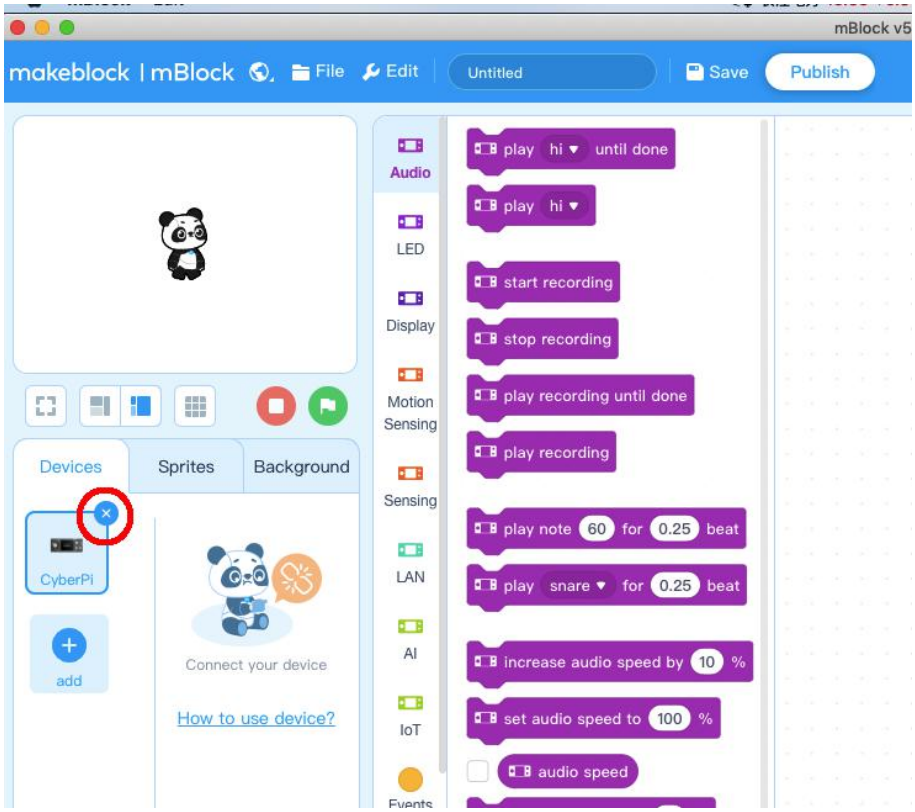
Note: In this kit, we use mblock as programming tool, if you want to learn more about mblock, please visit preparation before class 1: https://osoyoo.com/2021/12/01/preparation-before-class-1

Step 1) Download mBlock PC from https://mblock.makeblock.com/en-us/download/, select the download file as per your computer OS type:

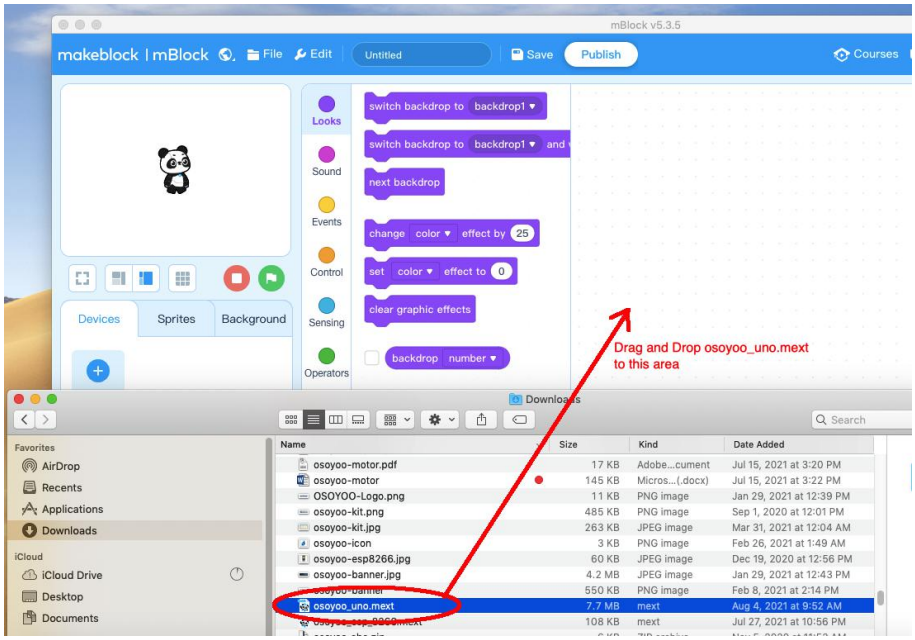


Step 2) Download OSOYOO_UNO.mext device file from https://osoyoo.com/driver/mblock/osoyoo_uno.mext

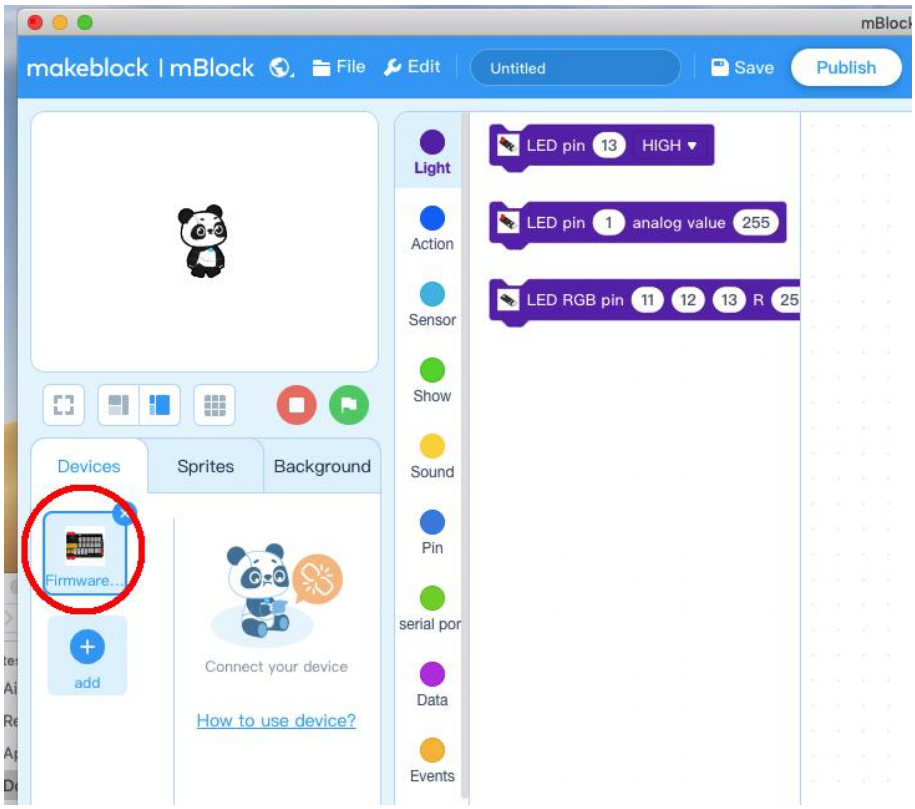
Step 3) Run the mBlock PC software by double click the lovely Panda icon. you will see mBlock UI as following picture. Please delete the default device **CyberPi** by click the cross in the red circle.



Step 4) Drag and Drop osoyoo_uno_mext file(downloaded in Step 2) to mBlock software as following:



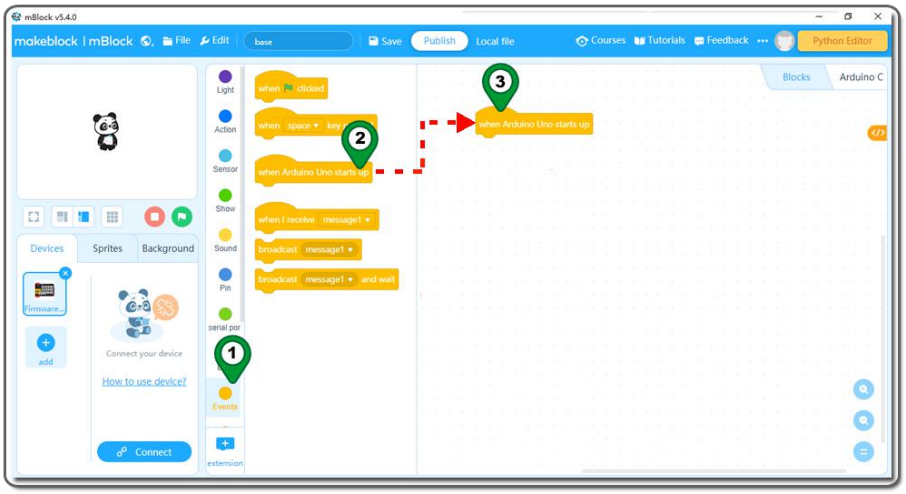
Now you will see a new device firmware in mBlock, see following picture:



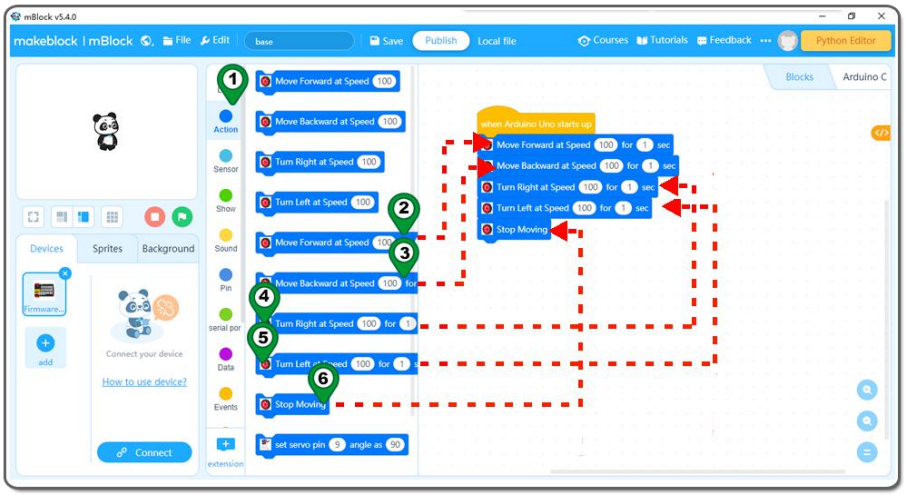
Now mBlock software and OSOYOO_UNO device firmware have been successfully installed in our PC!

Now we will show you how to use blocks to turn above idea into reality.

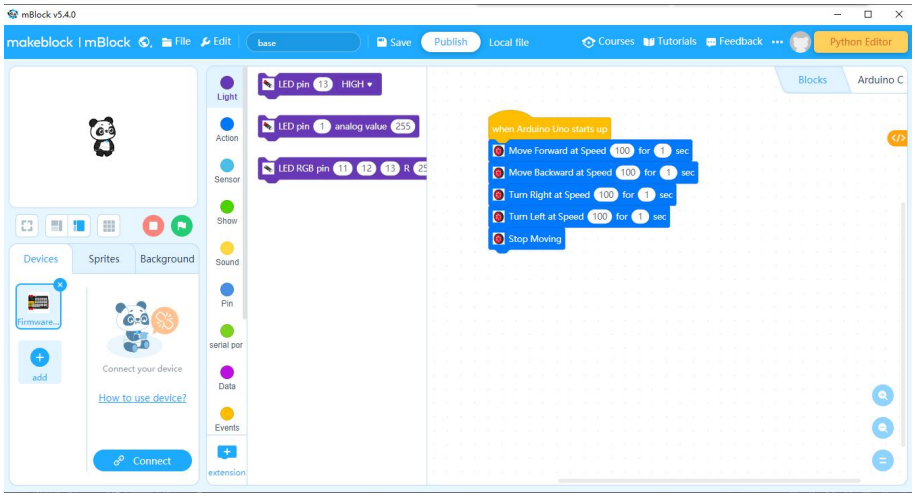
Step 5): Click **Events**,add **when Arduino Uno starts up** block to coding area:



Step 6): Click **Action**,add 5 action blocks below **when Arduino Uno starts up** block .This part is to confirm the speed of the operation.If you want change the speed ,you can change this value .On here we writed 100 as following picture;



Now we have completed the block programming.The final blocks look like following:



Step 7) Upload the program to OSOYOO basic board

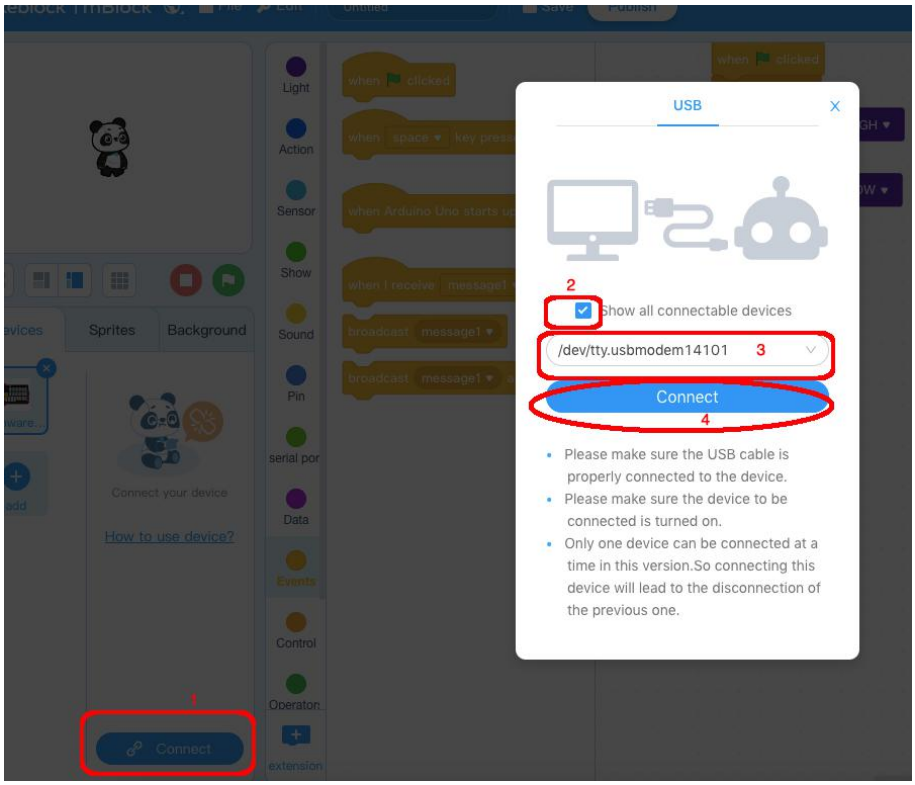
1) Please connect your OSOYOO basic board to your PC with USB cable firstly.

Then click the Connect button in the bottom of the mBlock software, you will see a USB window pop up,

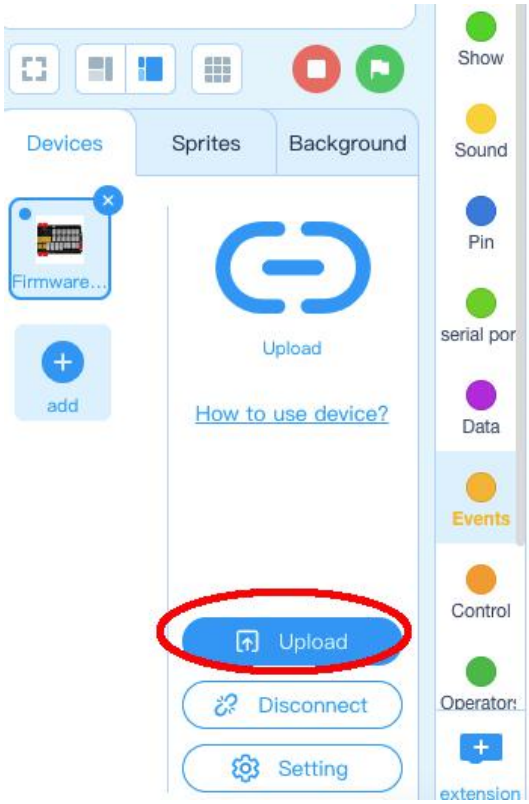
2) select **Show all connectable device** check box , then a device drop-down menu will show up,

3) select your port from device drop-down menu

4) click **Connect** button to connect your PC to OSOYOO basic board.



5) After you PC is connected toOSOYOO basic board, please click Upload button in the bottom of your software, then the code will be uploaded to OSOYOO basic board:



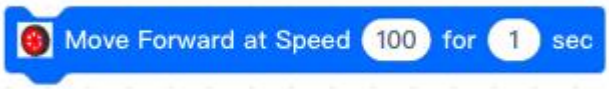
The program code blocks are quite straight forward, so we just make brief explanations as following:

i) The first yellow block

Above block is dragged from Events category. It tell the system that the blue blocks following it will start when the Arduino UNO board power is turned on.

In other projects, you can see other yellow blocks from Events category. Their purpose are the same, define when should the following program blocks be executed.

ii)The blue blocks following the yellow event block



These blue blocks are from Action category which means the block defined some kind of actuator action.

Take above **Move Forward at Speed 100 for 1 sec** block example, it tells the car to move forward at speed 100 for one second then stop.

The speed value can be **0** (stop) to **255** (maximum speed). You can change the speed and rotation time in the code block.

Base on same logic, you can easily understand the meaning of other 4 blue blocks which make the car move backward, left turn, right turn and stop.

iii)Full program block explanation

Now we can review the whole blocks functionality in this lesson:

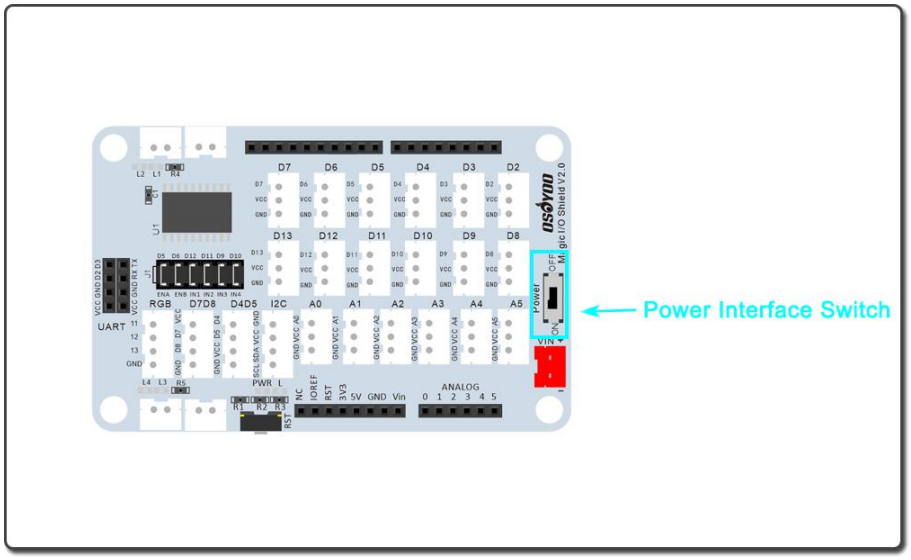
So in plain English, the program blocks in this lesson will ask the car to move forward for 1 second, then move backward for one second, then turn right for one second, then turn left for one second and finally stop. The program will start immediately after Arduino power is on.

Note:

If you want to use Arduino IDE to compile the program, here is the Arduino source code download link: https://osoyoo.com/driver/miniblock/basic-car/basic-car.zip

HOW TO PLAY

Disconnect Arduino from PC, put a 9V battery into battery pox**(make sure polar direction is correct, otherwise it can destroy your device and cause fire hazard).** Put the car on the ground, wave the Motor Power switch on the OSOYOO Magic I/O Shield V2.0, the car should go forward 1 seconds, then go backward 1 seconds, then right turn for 1 seconds, then left turn for 1 seconds, then stop.

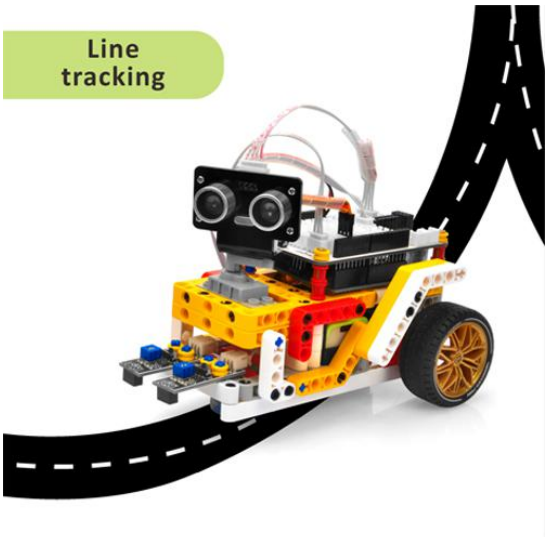


If the car does not move as per above mentioned result, you should check your wire connection, battery voltage(must over 7.2v).

Lesson2: Line Tracking Car

OBJECTIVE

In this project, we use two IR tracking sensors to design a simple line tracking Robot car. A Line tracking Robot, as the name suggests, is an auto-driving vehicle which follows a black track line printed in white ground. Tracking sensors in the robot keeps checking if the current car location is out of the track. If yes, then car will make a negative movement to draw the robot car back to track.



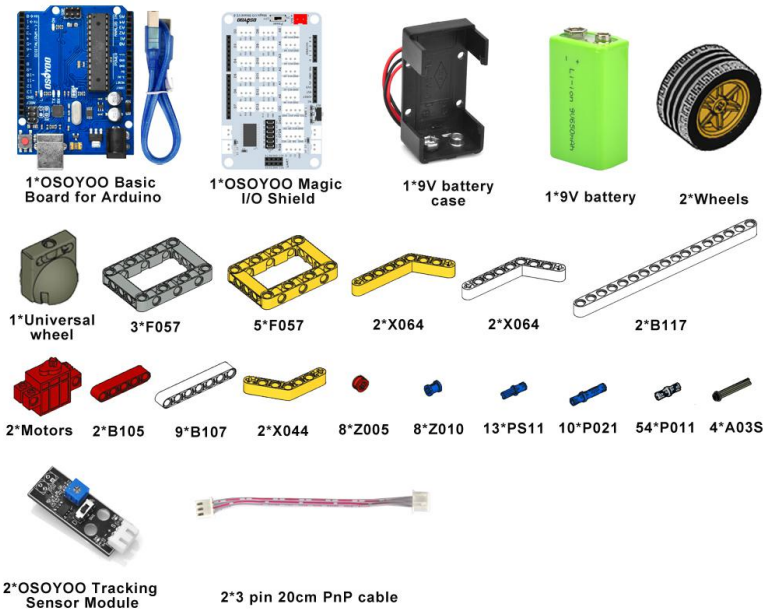
PARTS & DEVICES

Please prepare the following parts to complete this project

NOTE:

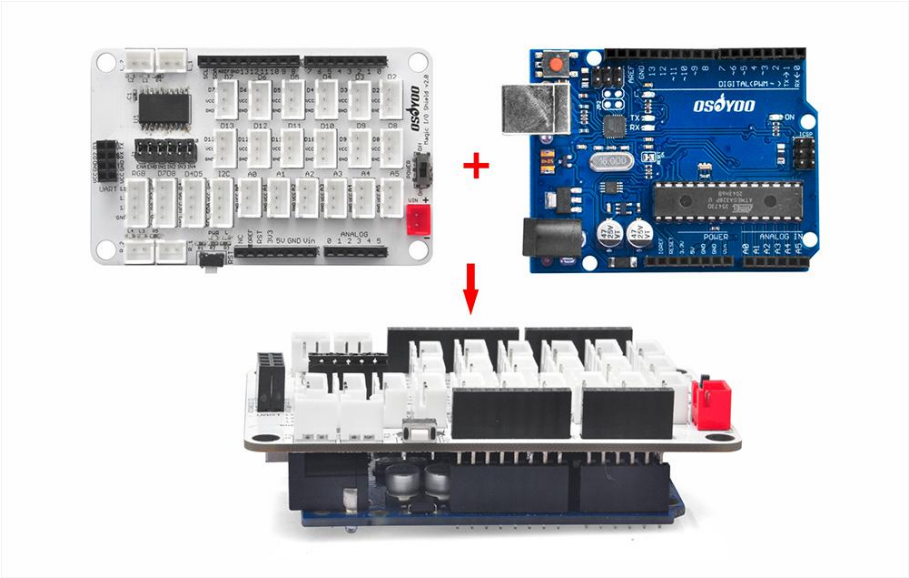
1.the color of the building block is subject to the actual product, which does not affect the use.

2. ALL OSOYOO PRODUCTS FOR ARDUINO ARE THIRD PARTY BOARD WHICH IS FULLY COMPATIBLE WITH ARDUINO



HOW TO MAKE

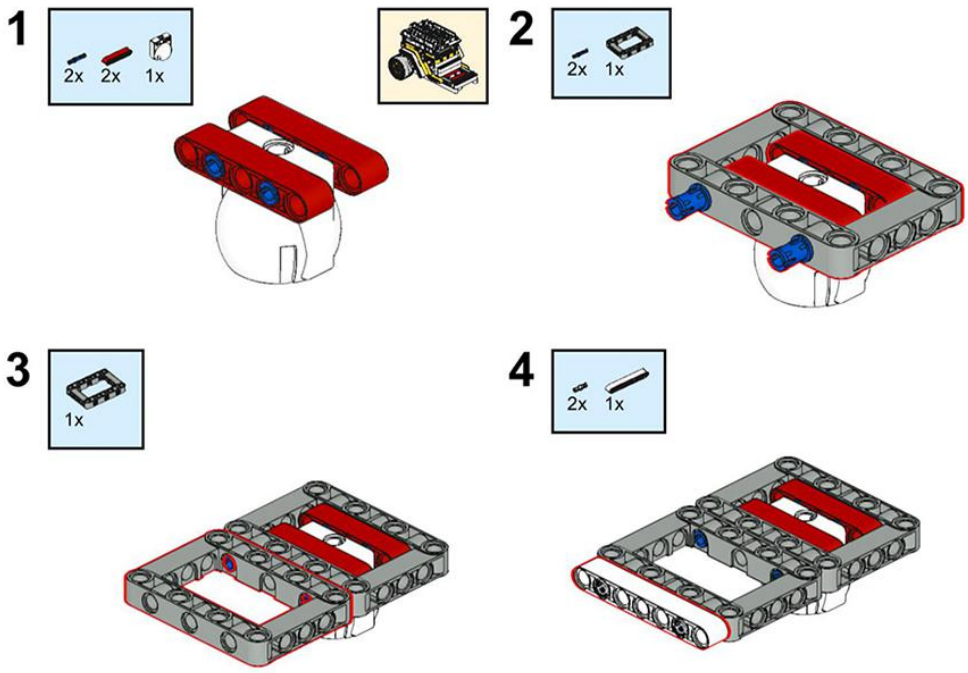
Before you build the robot with blocks, please install OSOYOO basic board for Arduino under OSOYOO Magic I/O shield as following *(Attention please : the pins of I/O shield is aligned with the port of basic the board firstly, then press the shield tightly on the board*.)

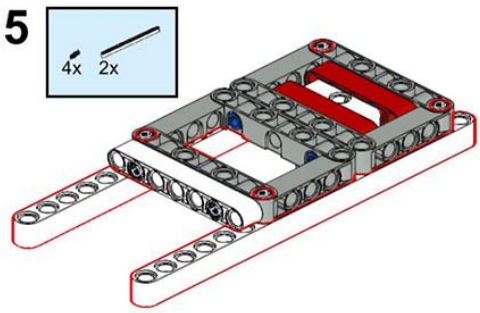


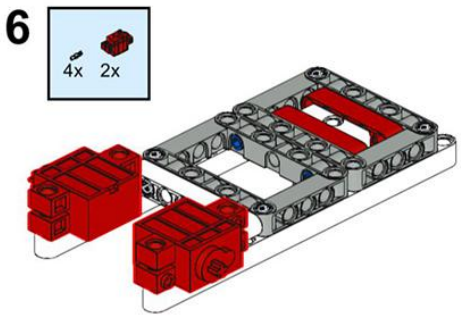
Please follow the building steps to build this robot car, If you want to get clear PDF building steps, please download

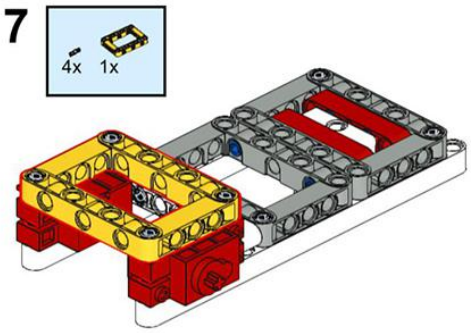
from https://osoyoo.com/picture/Building_Robot_Car/lesson2/LESSON2.pdf

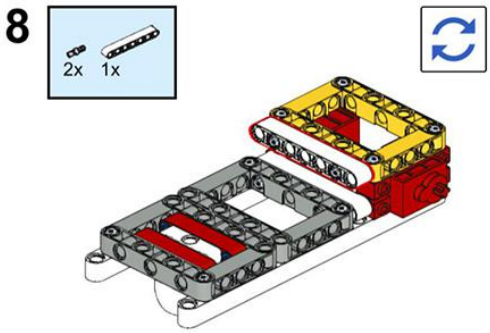
Note: If you have built [the robot car for lesson1](), please skip to the step35 in this PDF.

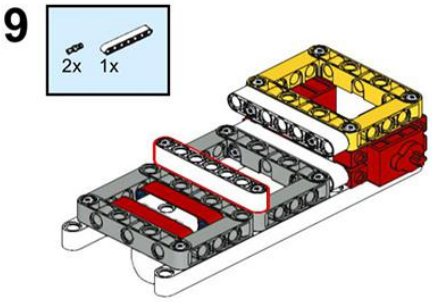


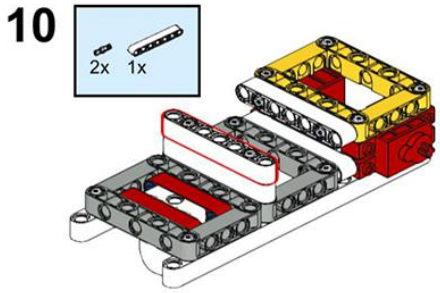

5
4x
2x

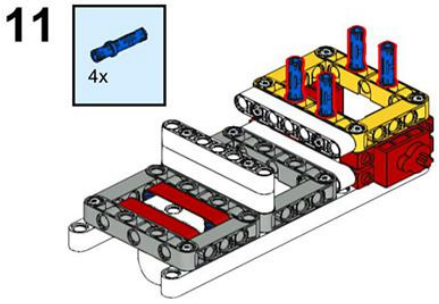

6
4x
2x

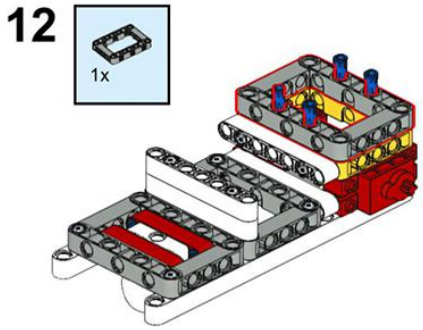

7
4x
1x


8
2x
1x

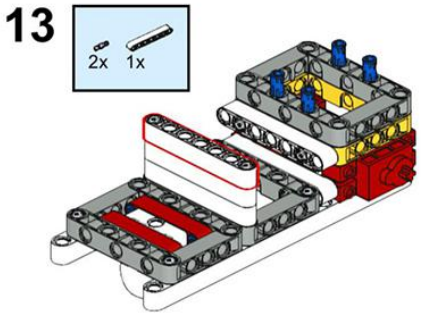

9
2x
1x


10
2x
1x

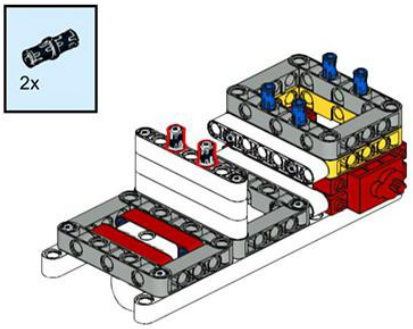

11
4x


12
1x

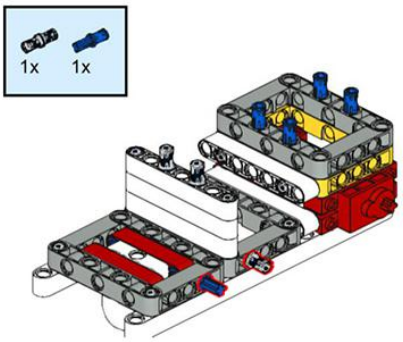
13



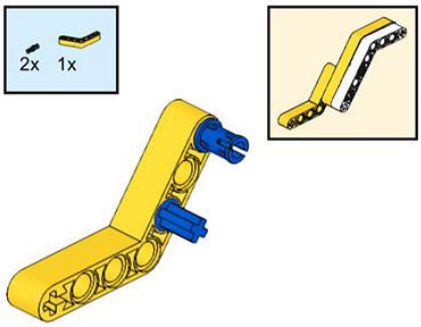
14



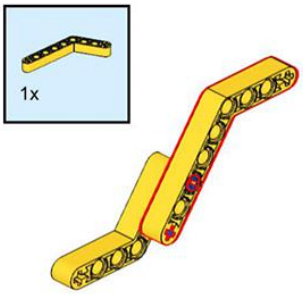
15



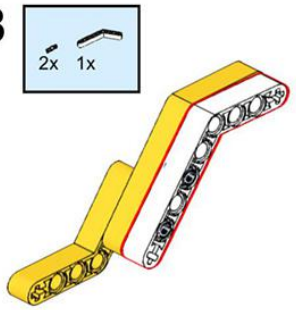
16



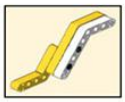
17



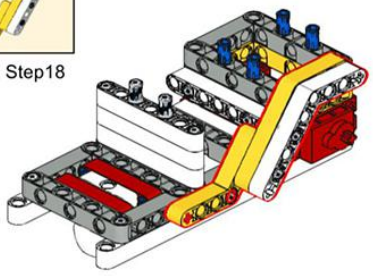
18



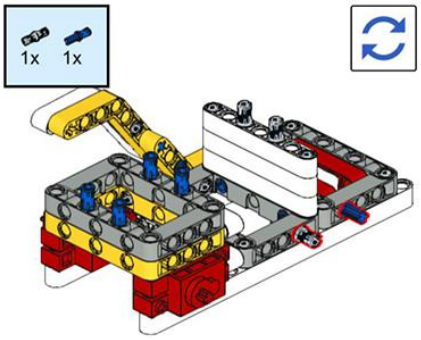
19

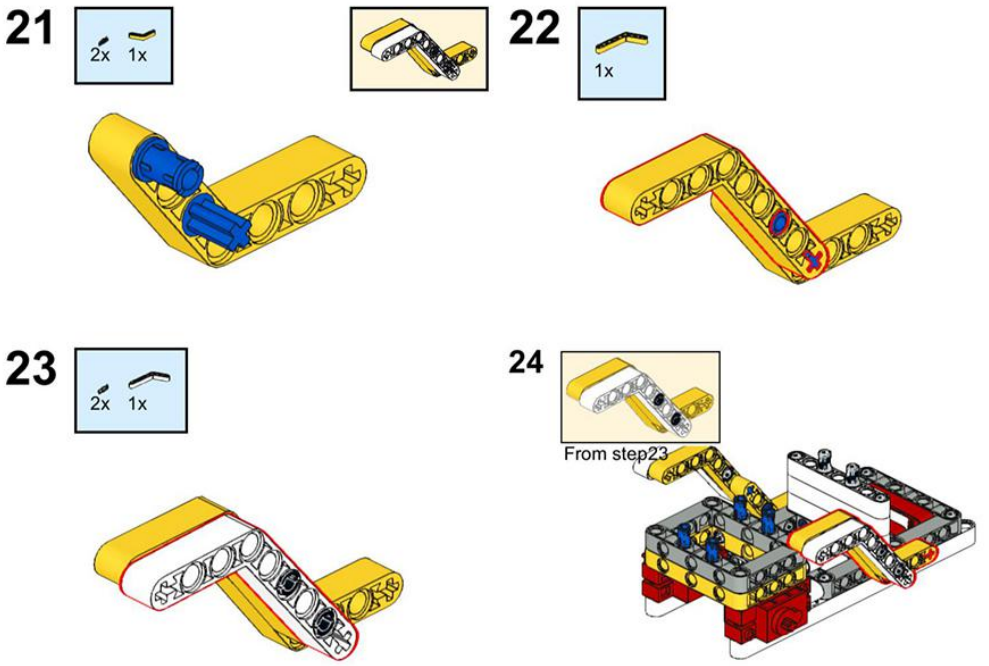


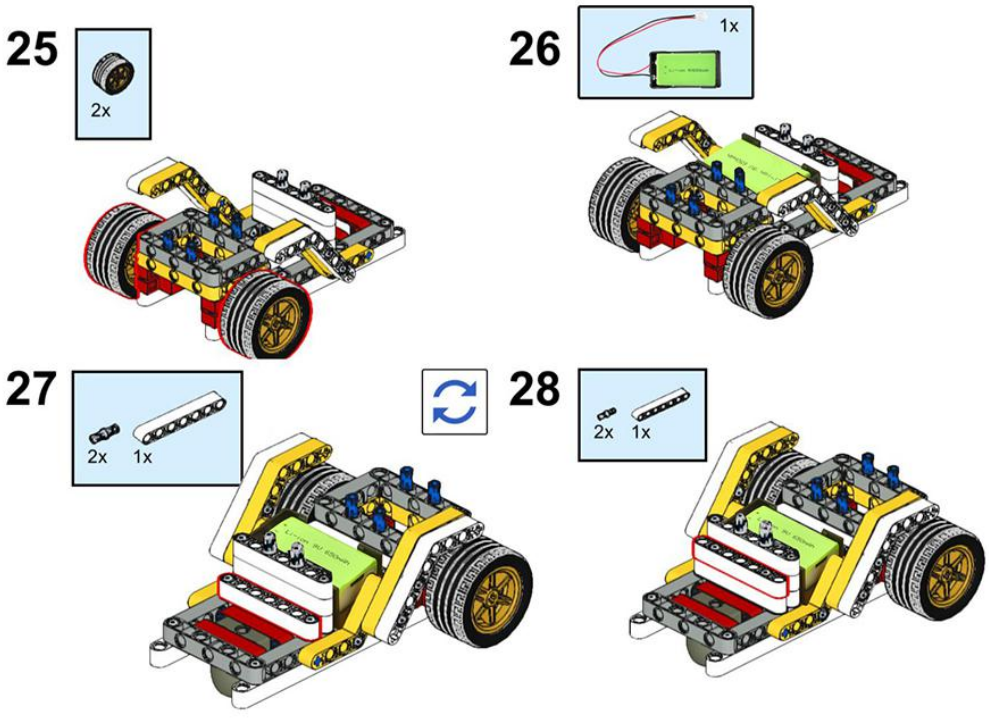
From Step18

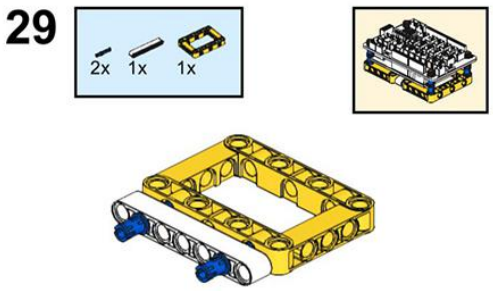


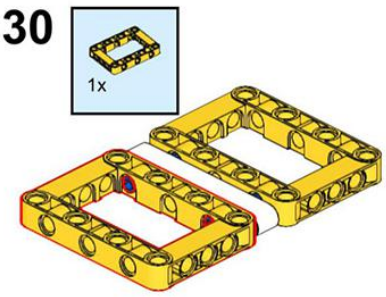
20

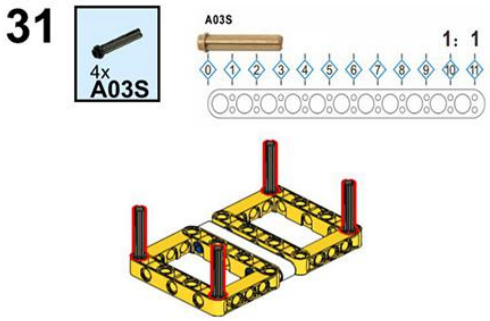


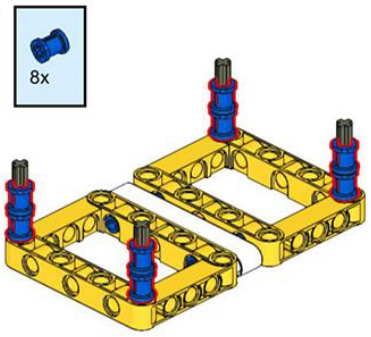

21
2x
1x
22
1x
23
2x
1x
24
From step23


25
2x
26
1x
27
2x
1x
28
2x
1x

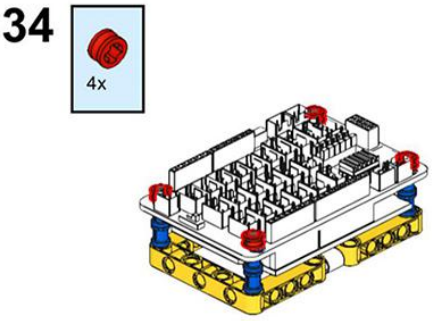

29
2x
1x
1x

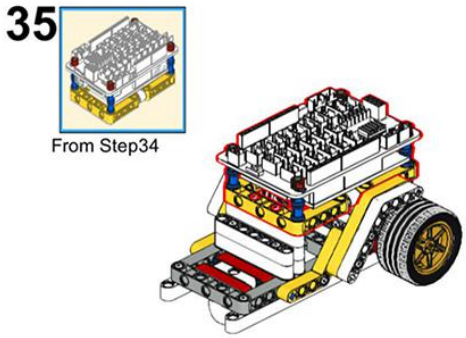

30
1x

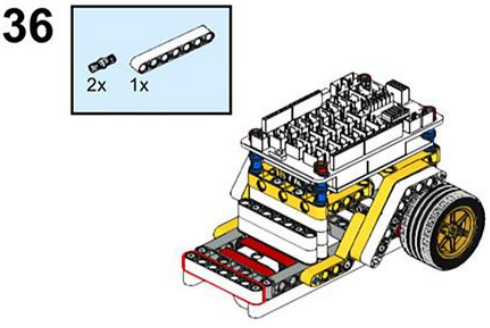

31
4x
A03S
A03S
1: 1
0
1
2
3
4
5
6
7
8
9
10
11

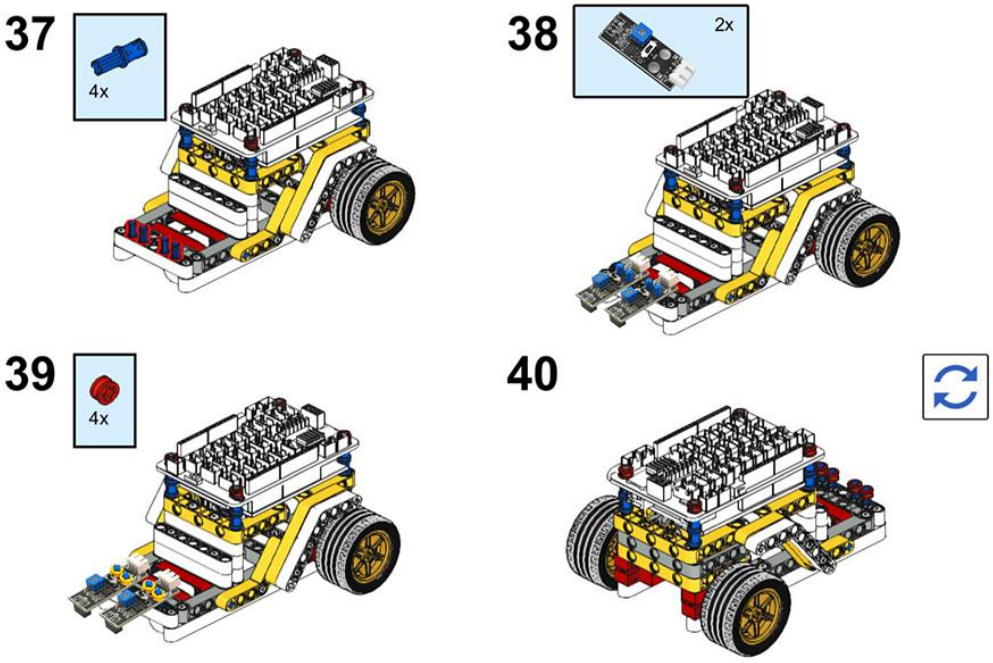

32
8x


33
1x
This part includes
magic shield and
OSOYOO Basic board


34
4x

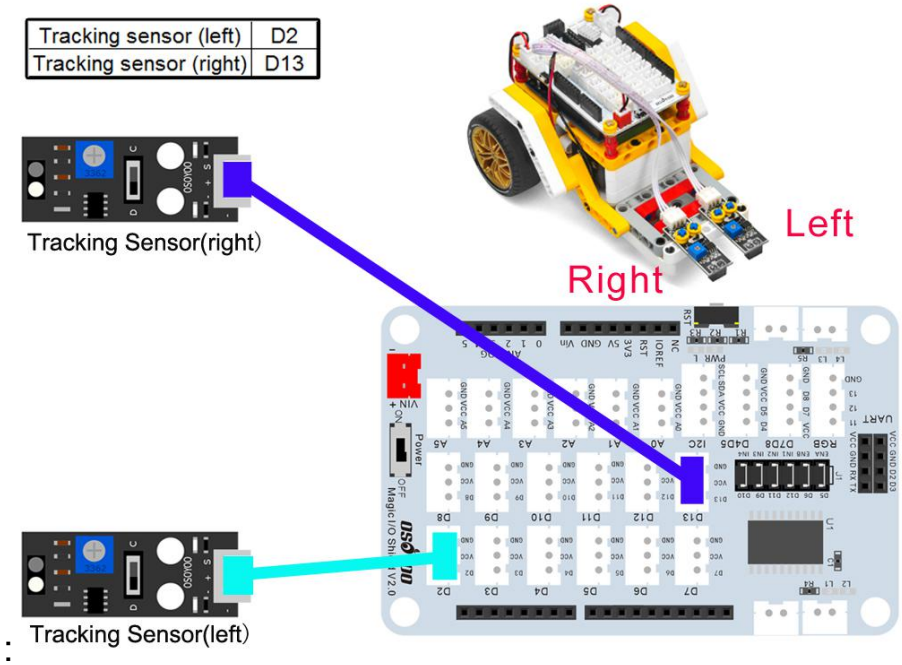

35
From Step34


36
2x
1x



CIRCUIT CONNECTION

Please connect motors and 9V battery case as lesson1.Then connect left tracking sensor to D2 of the Magic I/O shiel, right to D13 port with 3-pin PNP cables as below (Attention please: there are six jumper caps on ENA/ENB/IN1/IN2/IN3/IN4)

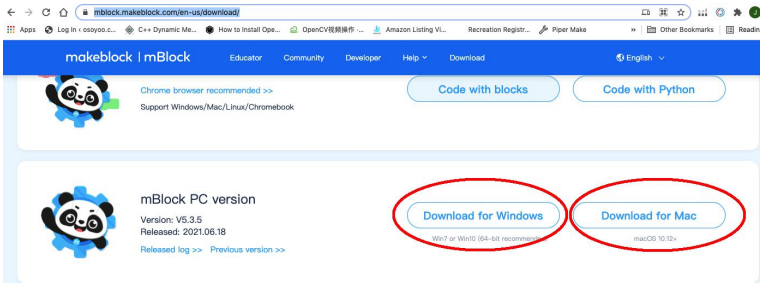


HOW TO CODE

Note: In this kit, we use mblock as programming tool, if you want to learn more about mblock, please visit preparation before class

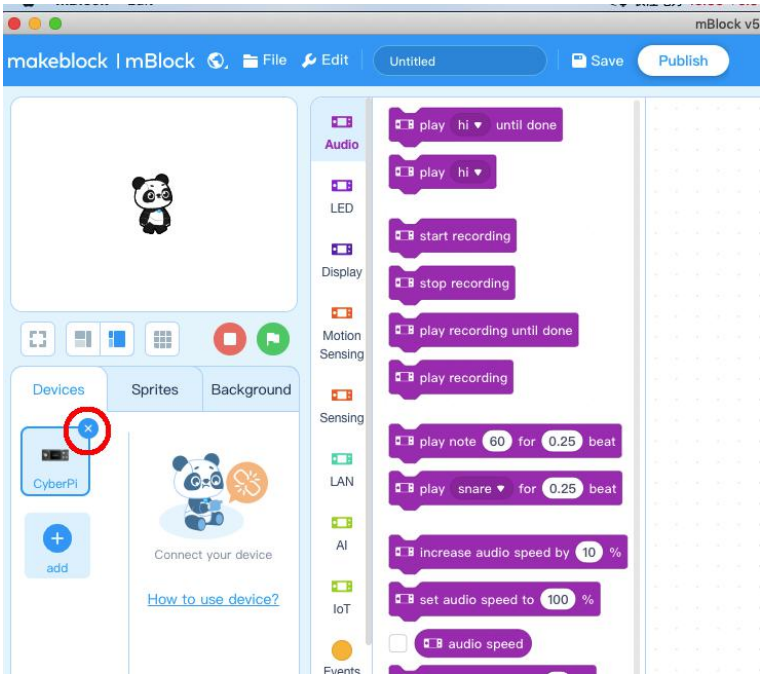
1: https://osoyoo.com/2021/12/01/preparation-before-class-1

Step 1) Download **mBlock PC version** from https://mblock.makeblock.com/en-us/download/, select the download file as per your computer OS type (Please don't use the browser version):

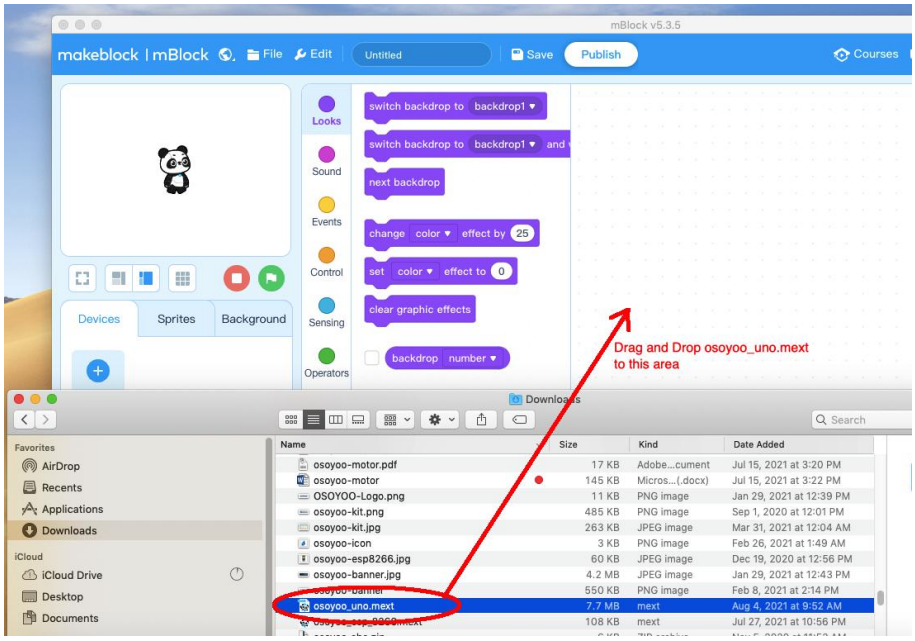


Step 2) Download OSOYOO_UNO.mext device file from https://osoyoo.com/driver/mblock/osoyoo_uno.mext

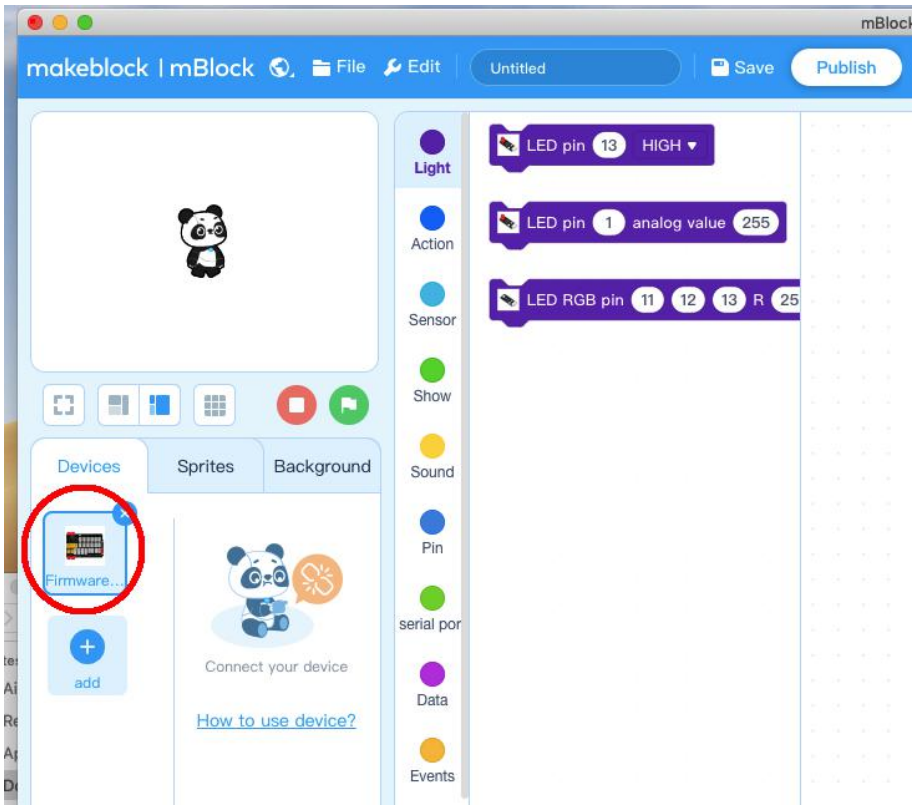
Step 3) Run the mBlock PC software by double click the lovely Panda icon. you will see mBlock UI as following picture. Please delete the default device **CyberPi** by click the cross in the red circle.



Step 4) Drag and Drop osoyoo_uno_mext file(downloaded in Step 2) to mBlock software as following:



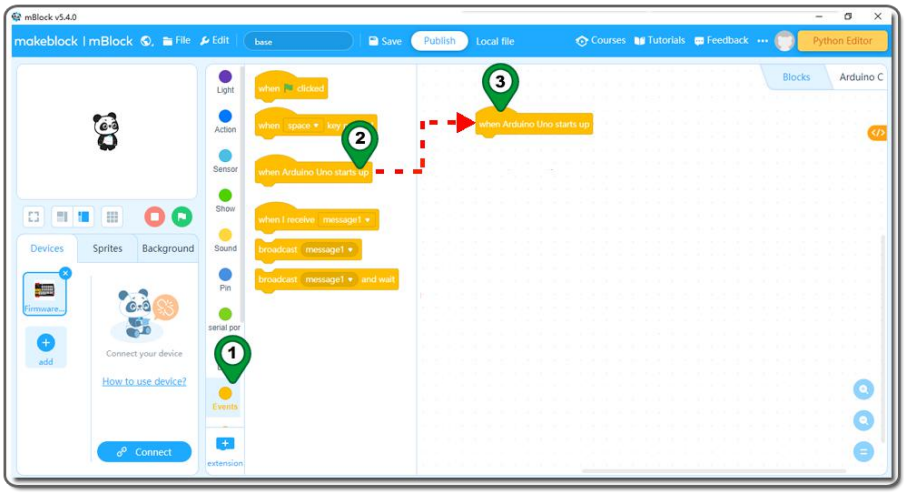
Now you will see a new device firmware in mBlock, see following picture:



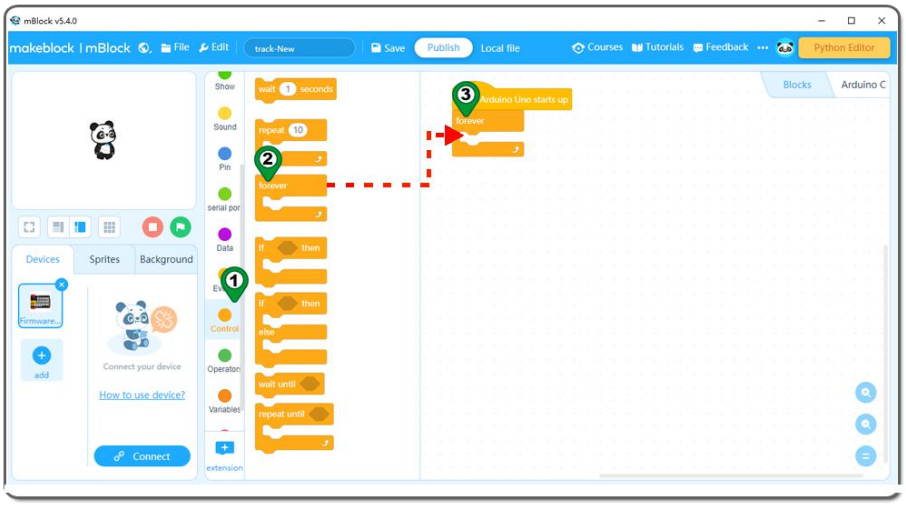
Now mBlock software and OSOYOO_UNO device firmware have been successfully installed in our PC!

Now we will show you how to use blocks to turn above idea into reality.

Step 5: Click **Events**, add **when Arduino Uno starts up** block to the top:



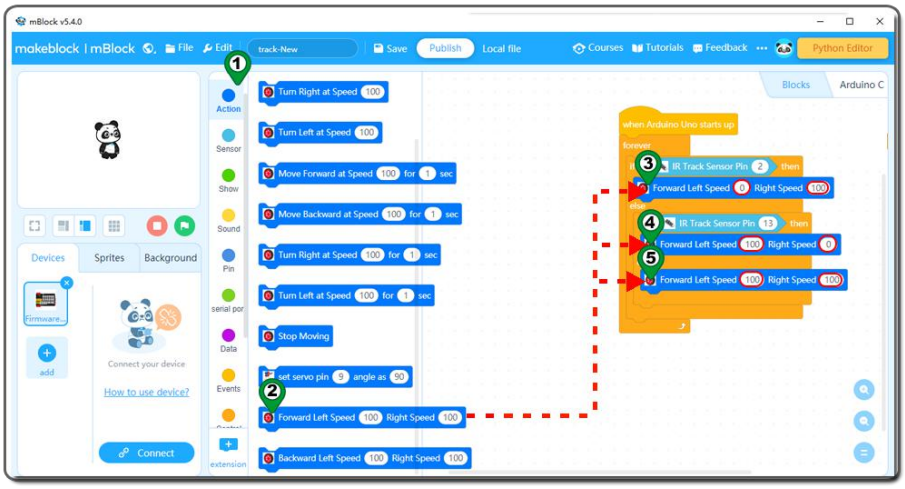
Step 6: Click **Control**, then Drag and drop **Forever** block to programming area as following:



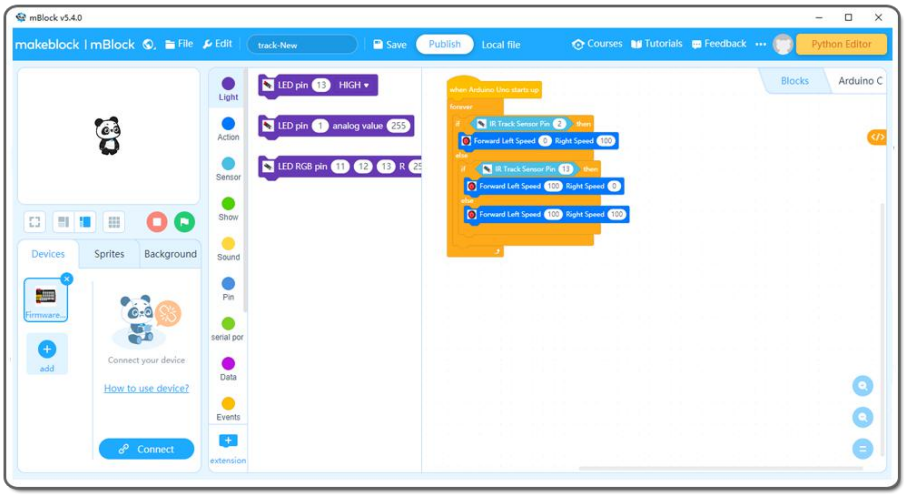
Step 7: Click **Control**, add an **if else** block inside **forever** loop, then add another **if else** block inside the **else** area of first **if else** block:

Step 8: Click **Sensor**, add 2 pcs of IR Tracking Sensor Pin 2 blocks inside **if** condition area, change 2nd Pin number from 2 to 13 as following:

Step 9: Click **Action** category, add 3 pcs **Barkward Left Speed 100 Right Speed 100** block inside **if else** blocks, change first **Right** Speed from 100 to 0, change 2nd **Left** Speed from 100 to 0 :



Now we have completed the block programming.The final blocks look like following:



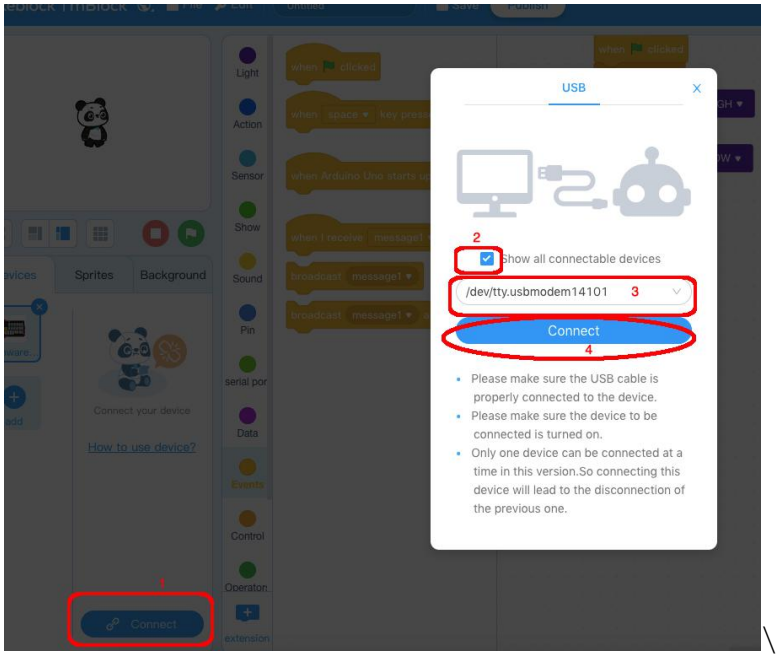
Step 10) Upload the program to OSOYOO basic board

1) Please connect your OSOYOO basic board to your PC with USB cable firstly. Then click the Connect button in the bottom of the mBlock software, you will see a USB window pop up,

2) select **Show all connectable device** check box , then a device drop-down menu will show up,

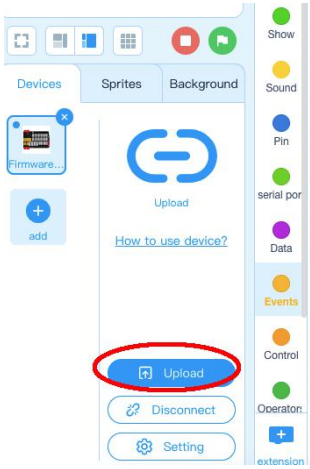
3) select your port from device drop-down menu

4) click **Connect** button to connect your PC to OSOYOO basic board.



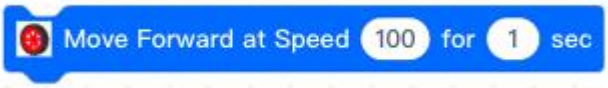
\

5)After you PC is connected to OSOYOO basic board, please click Upload button in the bottom of your software, then the code will be uploaded to OSOYOO basic board:



PROGRAM EXPLANATION

In Lesson 1 , we have learned an Yellow Event Program block and Some Blue Action blocks



In this lesson, we will see some new program blocks from Control Category

i) Forever Loop

This block like a crocodile head which has some inside blocks inside its mouse. These inside blocks will be executed in order, after the last inside block is executed, it will go back to the first inside block and repeat the procedure again and again like a loop

ii) A blue IR tracking sensor block from Sensor category

Above block is a hexagon block which means it will return True or False value to Control blocks

Take above block as example, this block will read tracking sensor in D2 pin(left sensor), if the sensor detects black, it will return TRUE, if detect white, it will return False

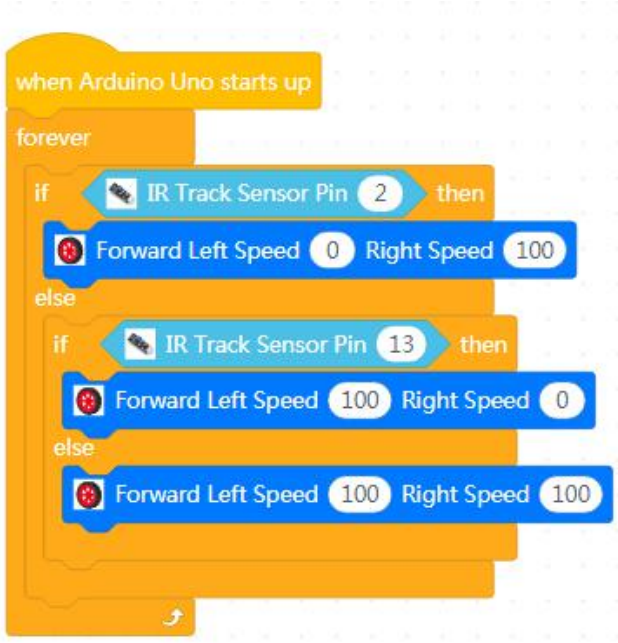
iii) If Else Block in Control category

Above block has two mouses, it also has a condition block in the top. When condition block returns True, it will execute the blocks inside the if area(upper mouse), if condition block returns False, it will execute the blocks inside the else area(lower mouse).

iv)Another action block which can control the left wheel and right wheel speed

above block can control the left wheel and right wheel running at different speed. In above example, left speed is 0 and right speed is 100. This will make the car make a left turn.

v) Now we can review the whole blocks functionality in this lesson:



Above program blocks are running as a forever loop which means it will never stop unless you turn off the power.

The program will first test left sensor (D2 pin) and see if left sensor detects any black line, if yes, it will make a left turn.

If left sensor detects white , then it will detect right sensor(D13), if right sensor returns True, it means the black line is in the right side and the car will make a right turn.

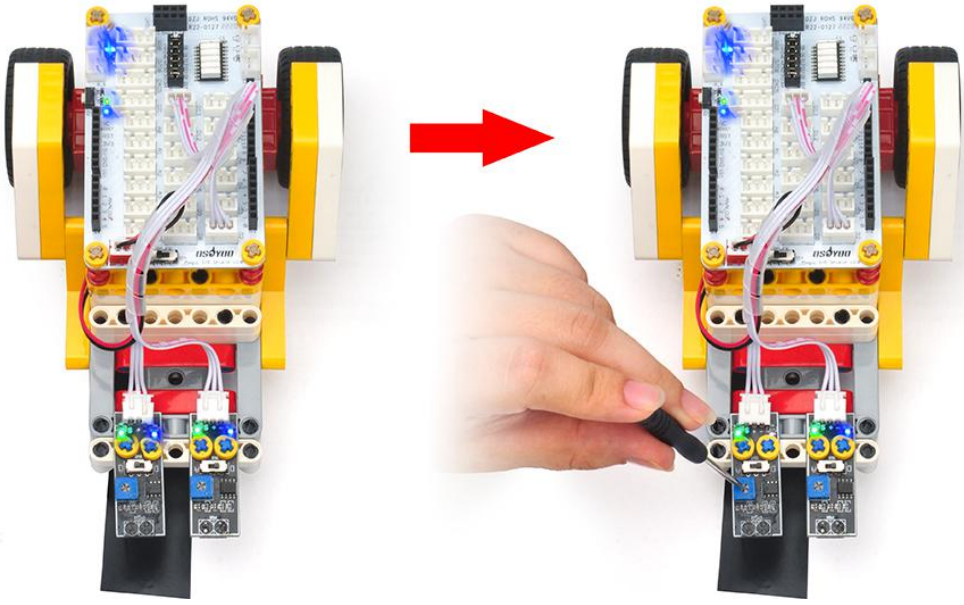
If both sensors don't detect black line, it means the black line is in the middle, the car will move forward.

HOW TO PLAY

Disconnect Arduino from PC, put a 9V battery into battery pox**(make sure polar direction is correct, otherwise it can destroy your device and cause fire hazard).**

Adjust Tracking Sensor Sensitivity:

Turn on and hold the car and adjust the sensitivity screw on the tracking sensor with a screwdriver until you get the best sensitivity status: the signal indicate LED light will turn on when sensor is above white ground, and the signal LED will turn off when the sensor is above black track.



Prepare a black track (the width of the black track is more than 30mm and less than 60mm) in white ground. Please note, the turning angle((bending curve) of track can’t be two sharp(larger than 90 degree). If the turning is too sharp, the car will move out of the track.Turn on the car and put the middle of tracking sensor module facing over black track, and then the car will move along the black track.

Lesson3 : Light follower

OBJECTIVE

In this project, we use two photoresistor sensors to design a simple Light follower Robot car.Using a flashlight to shine on the Photo resistor modules, the robot car will follow the light to move forward, turn light or turn left just like a cute cat play with the light.The Photoresistor modules consist of sensors at the front of the car; you can program it to follow the stronger light.

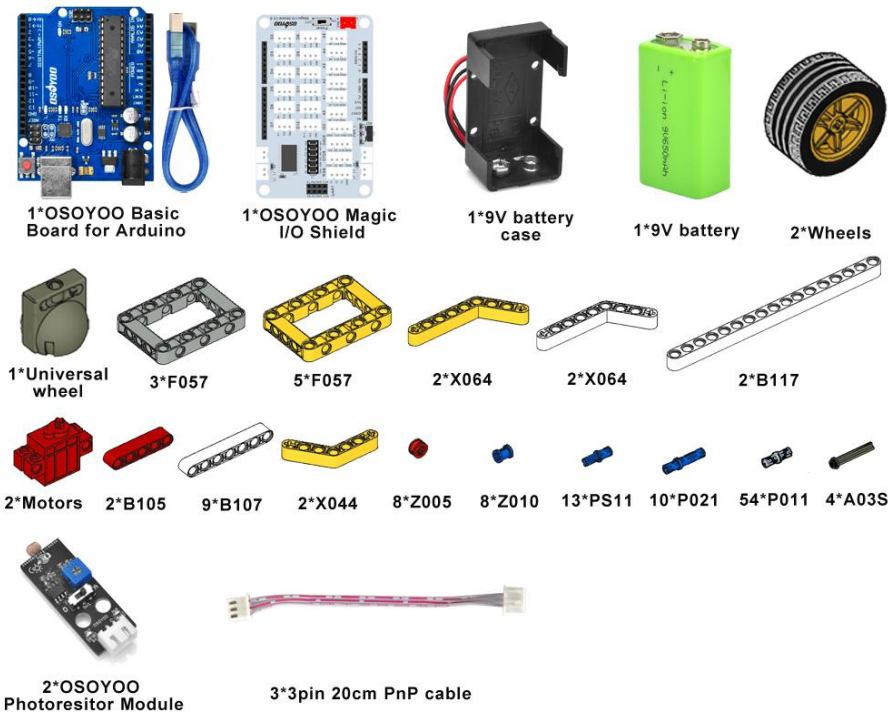
PARTS & DEVICES

Please prepare the following parts to complete this project

NOTE:

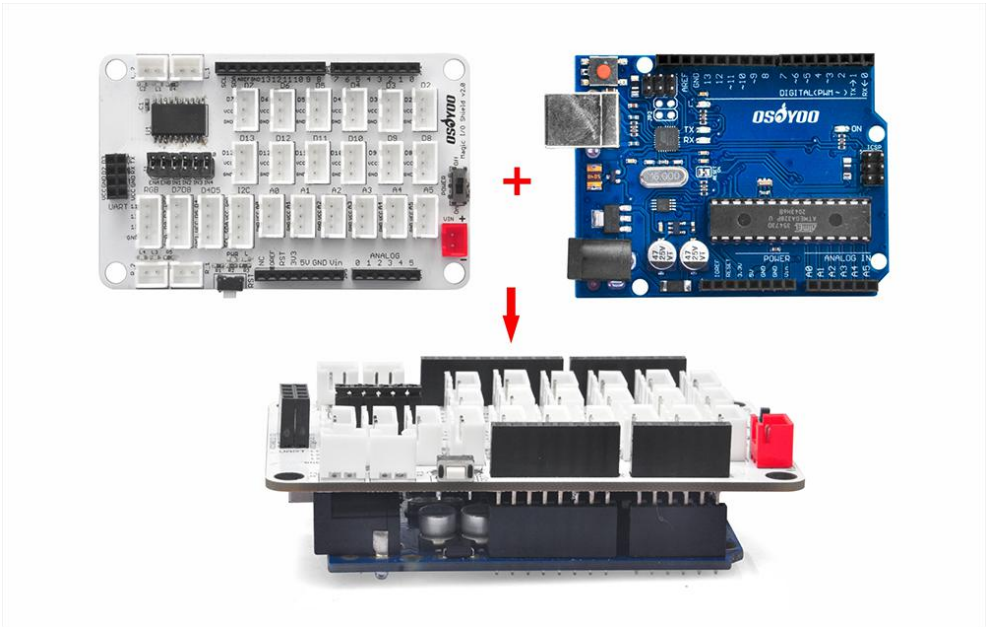
1.the color of the building block is subject to the actual product, which does not affect the use.

2. ALL OSOYOO PRODUCTS FOR ARDUINO ARE THIRD PARTY BOARD WHICH IS FULLY COMPATIBLE WITH ARDUINO



HOW TO MAKE

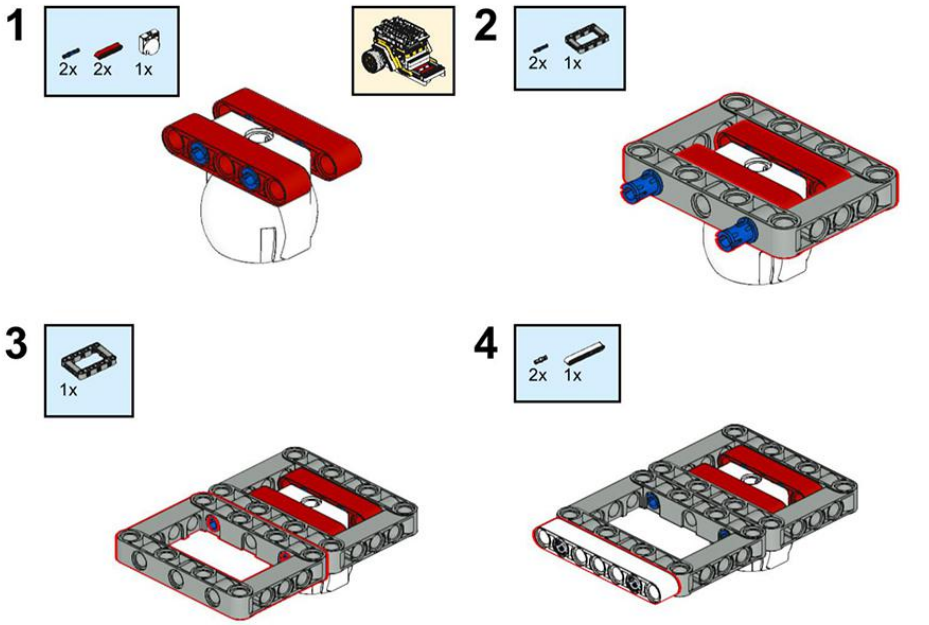
Before you build the robot with blocks, please install OSOYOO basic board for Arduino under OSOYOO Magic I/O shield as following *(Attention please : the pins of I/O shield is* aligned with the port of basic the board firstly, then press the shield tightly on the board).

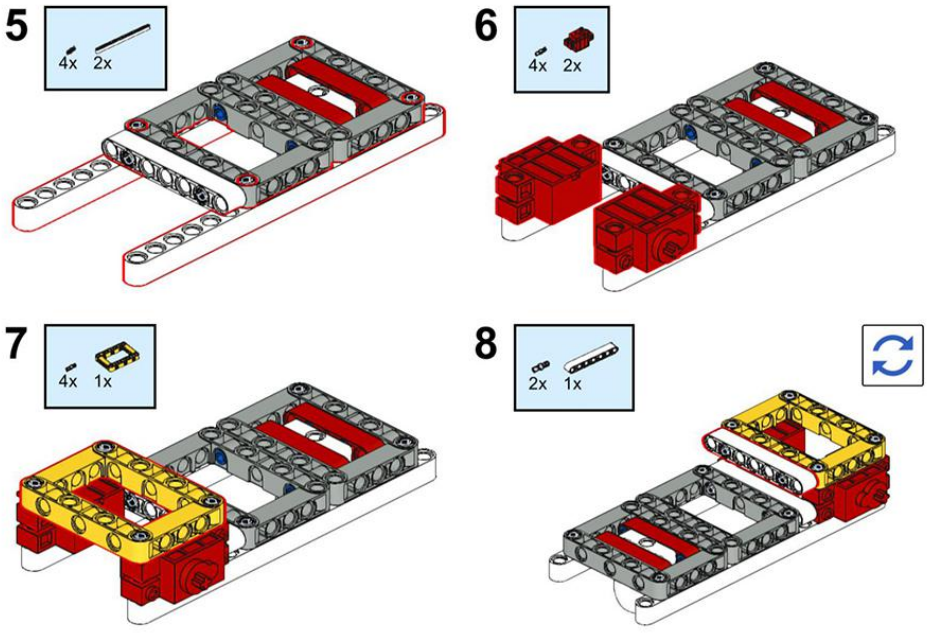


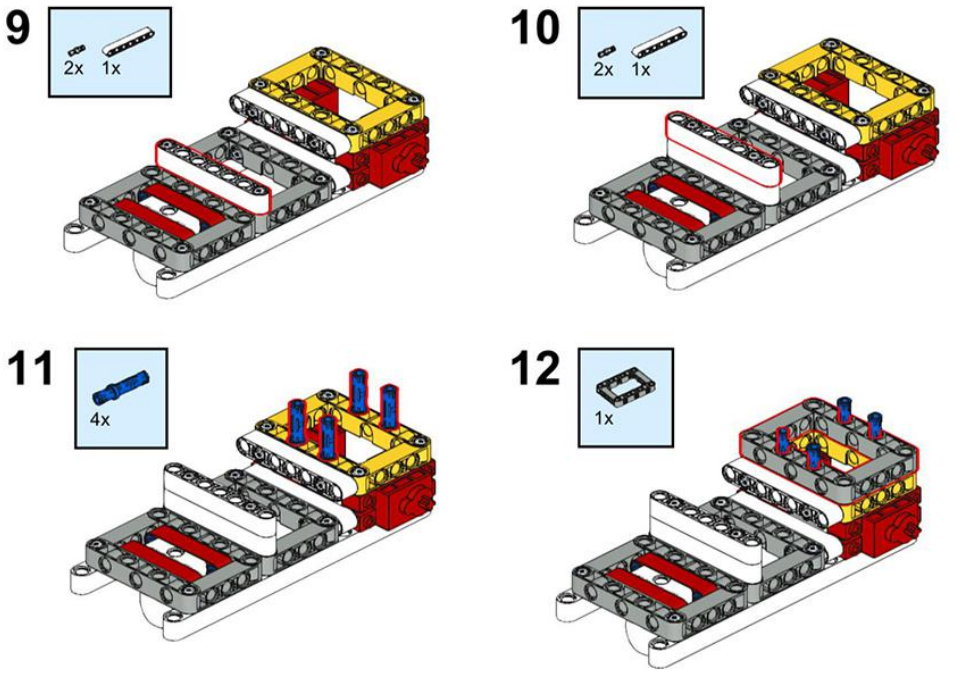
Please follow the building steps to build this robot car, If you want to get clear PDF building steps, please download

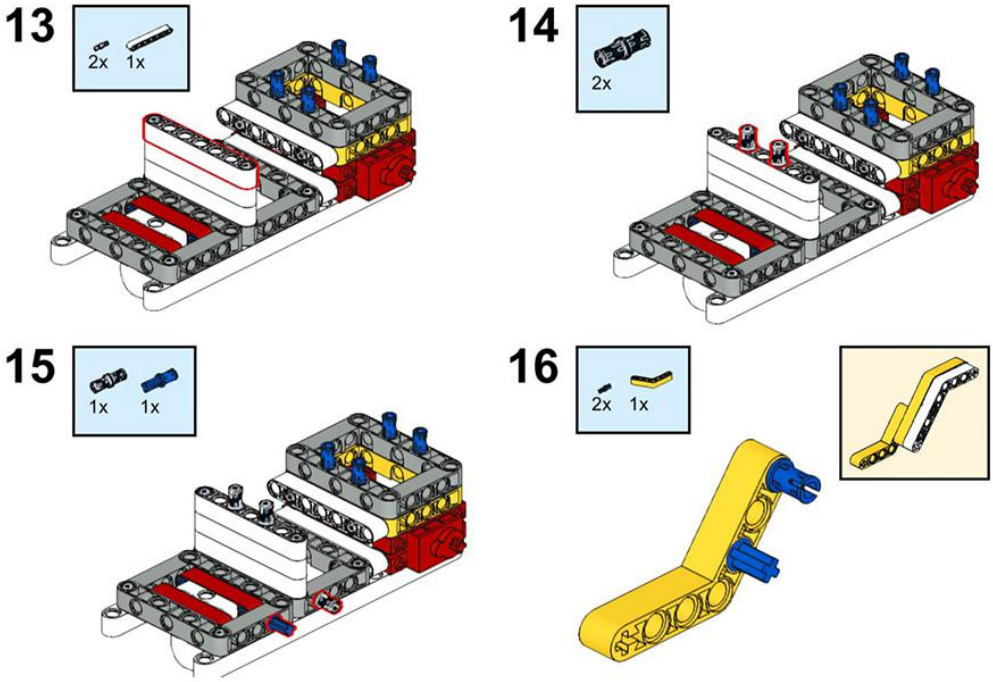
from https://osoyoo.com/picture/Building_Robot_Car/lesson3/LESSON3.pdf

Note: If you have built the robot car for lesson1, please skip to the step35 in this PDF.

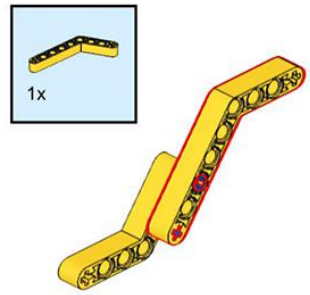




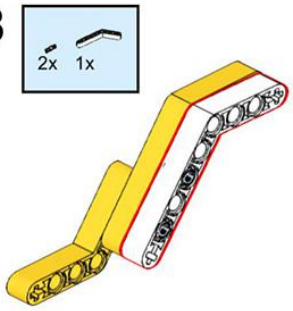

9
2x 1x
10
2x 1x
11
4x
12
1x


13
2x 1x
14
2x
15
1x 1x
16
2x 1x

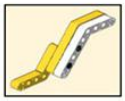
17



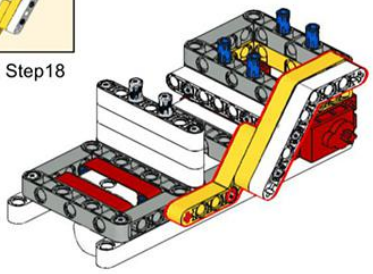
18



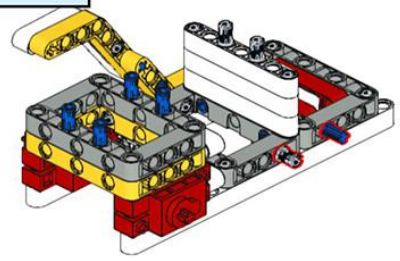
19



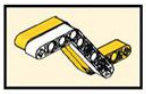
From Step18

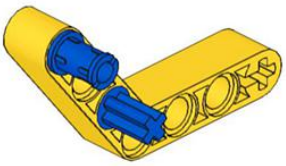


20

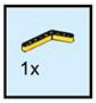


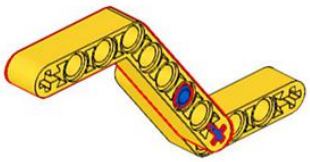
21



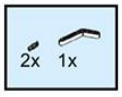


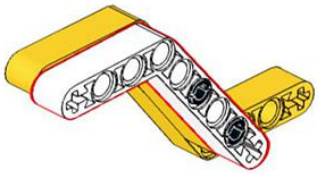
22



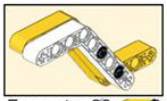


23

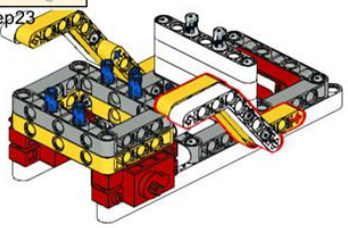




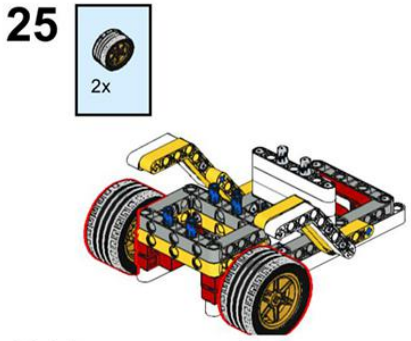
24



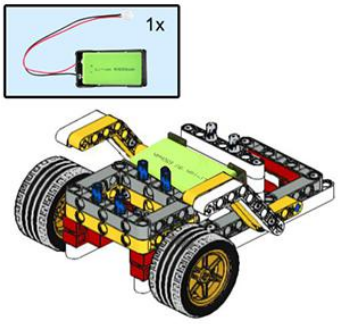
From step23



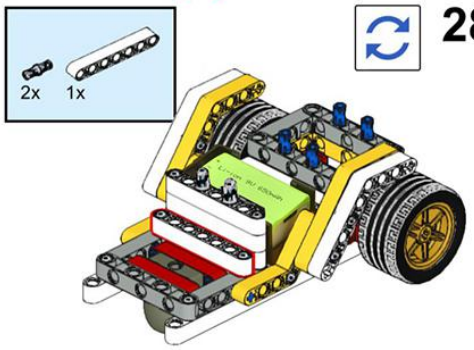
25



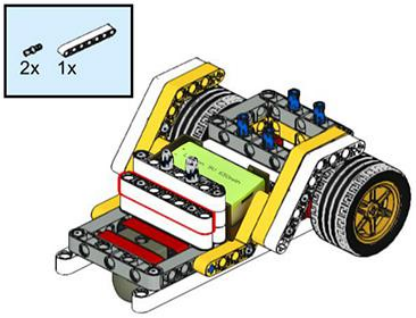
26



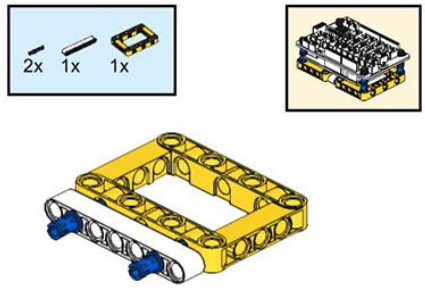
27



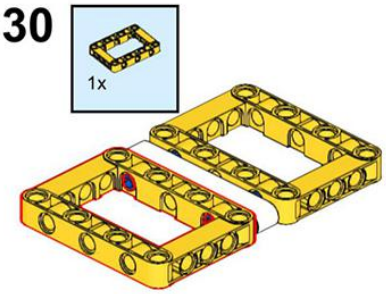
28



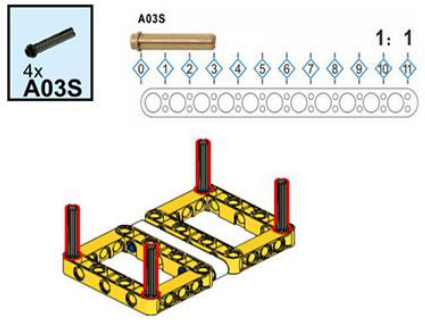
29



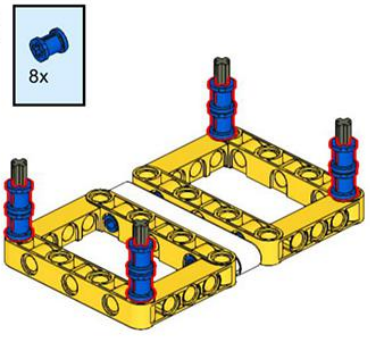
30

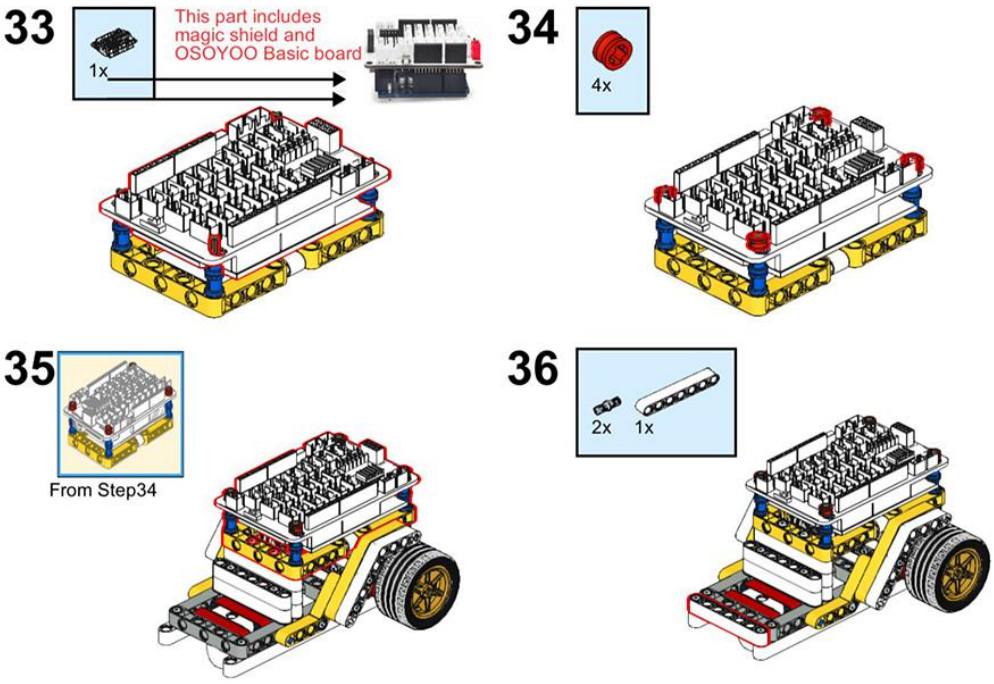


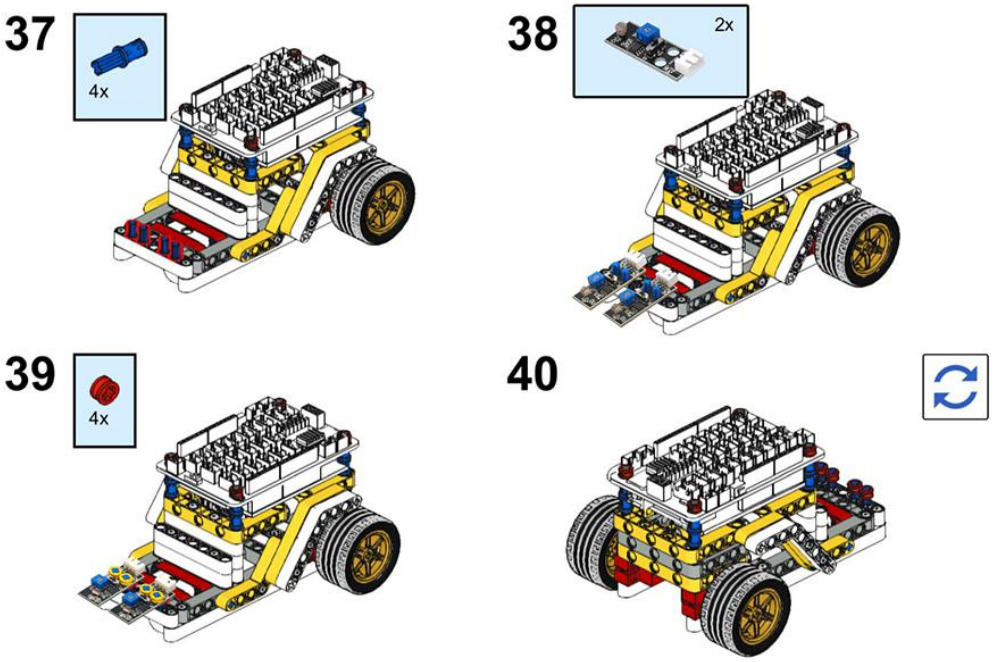
31



32



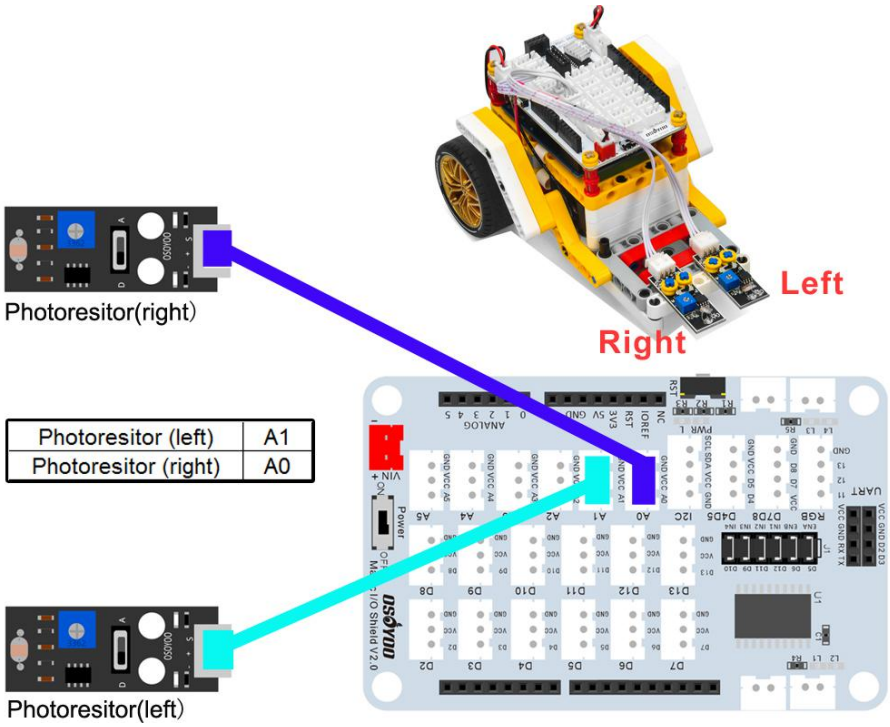




CIRCUIT CONNECTION

Please connect motors and 9V battery case as lesson1.Then connect left

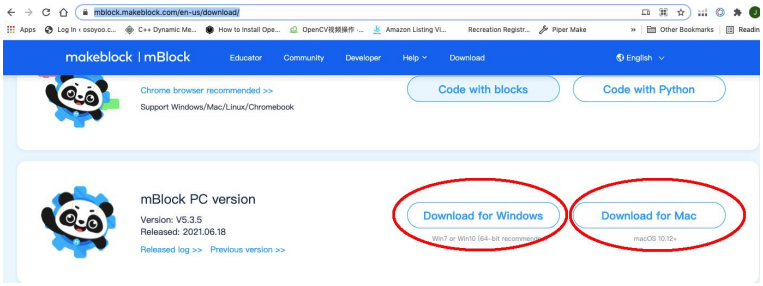
photoresistor to A1 of the Magic I/O shield, right to A0 port with 3-pin PNP cables as below (Attention please: there are six jumper caps on ENA/ENB/IN1/IN2/IN3/IN4):



HOW TO CODE

Note: In this kit, we use mblock as programming tool, if you want to learn more about mblock, please visit preparation before class 1: https://osoyoo.com/2021/12/01/preparation-before-class-1

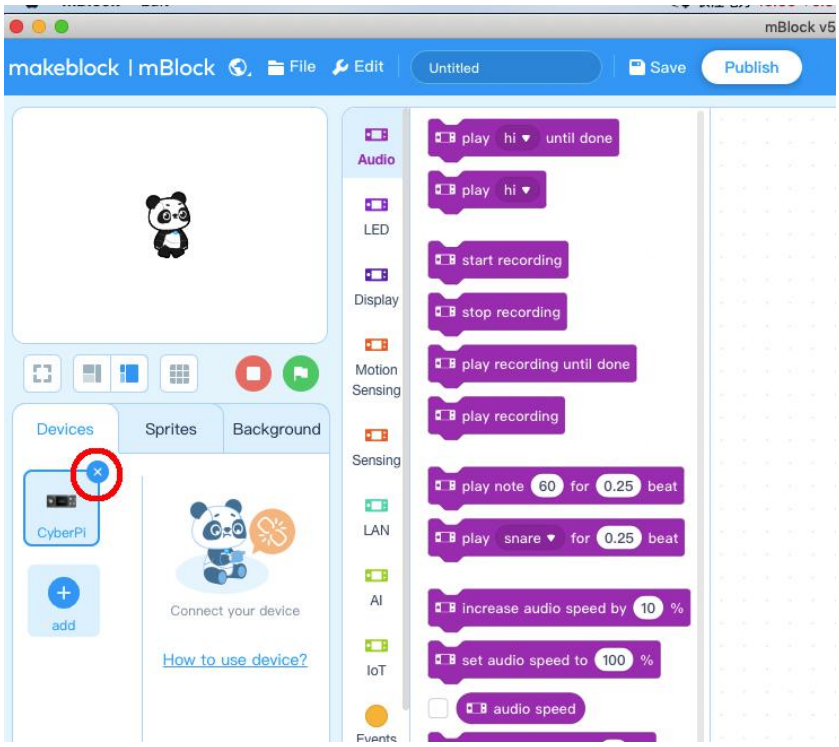
Step 1) Download mBlock PC version from https://mblock.makeblock.com/en-us/download/, select the download file as per your computer OS type (Please don't use the browser version):



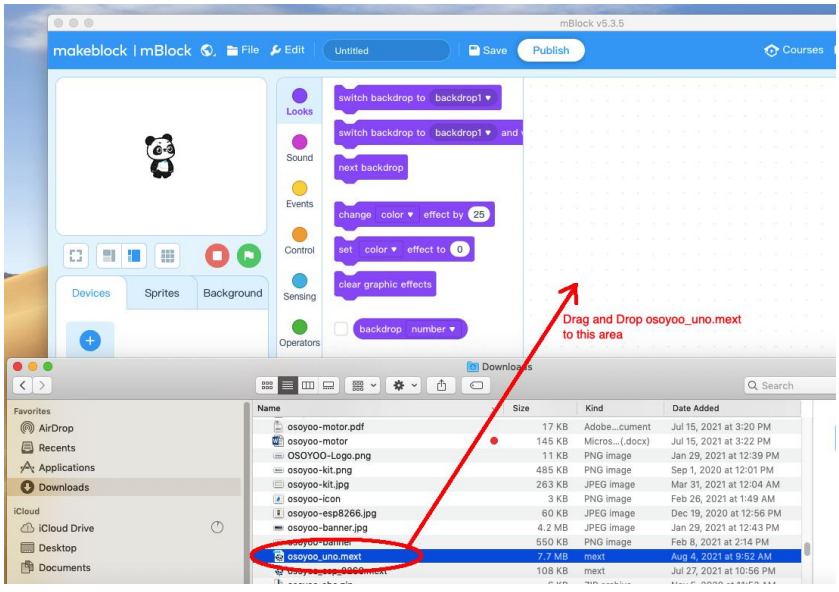
Step 2) Download OSOYOO_UNO.mext device file

from https://osoyoo.com/driver/mblock/osoyoo_uno.mext

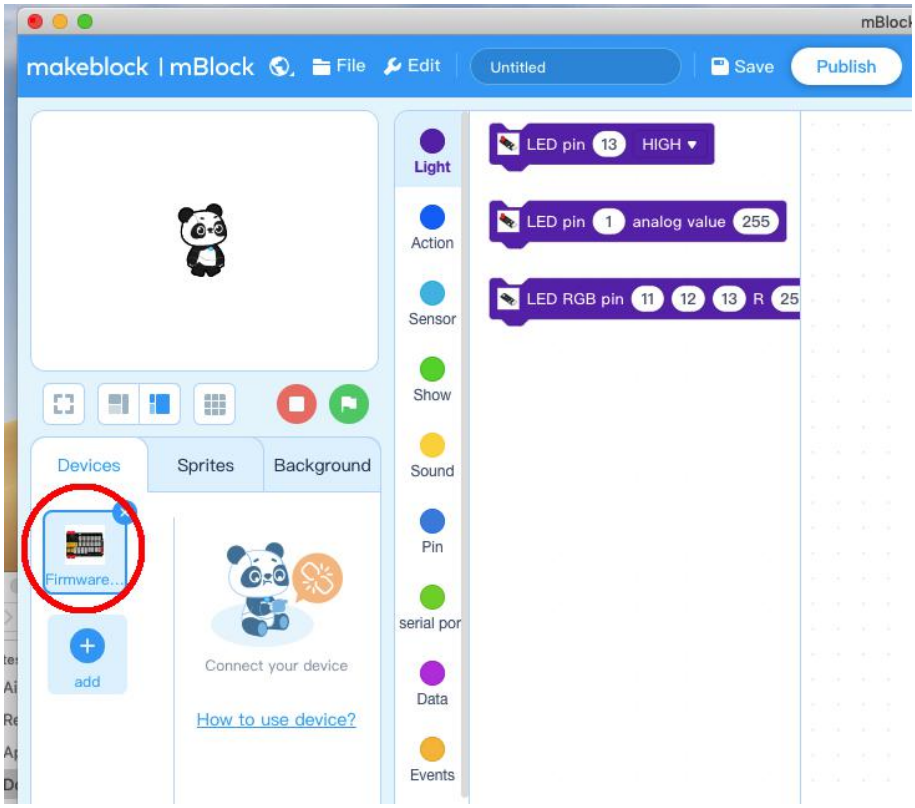
Step 3) Run the mBlock PC software by double click the lovely Panda icon. you will see mBlock UI as following picture. Please delete the default device **CyberPi** by click the cross in the red circle.



Step 4) Drag and Drop osoyoo_uno_mext file(downloaded in Step 2) to mBlock software as following:



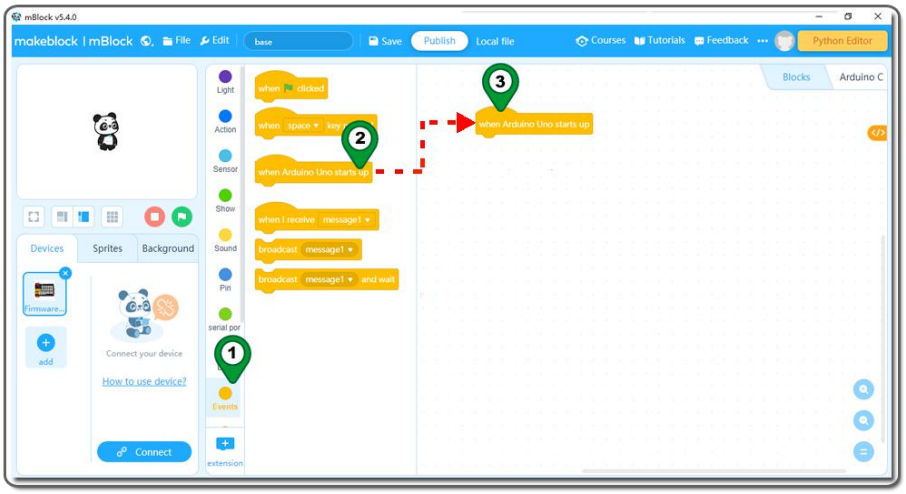
Now you will see a new device firmware in mBlock, see following picture:



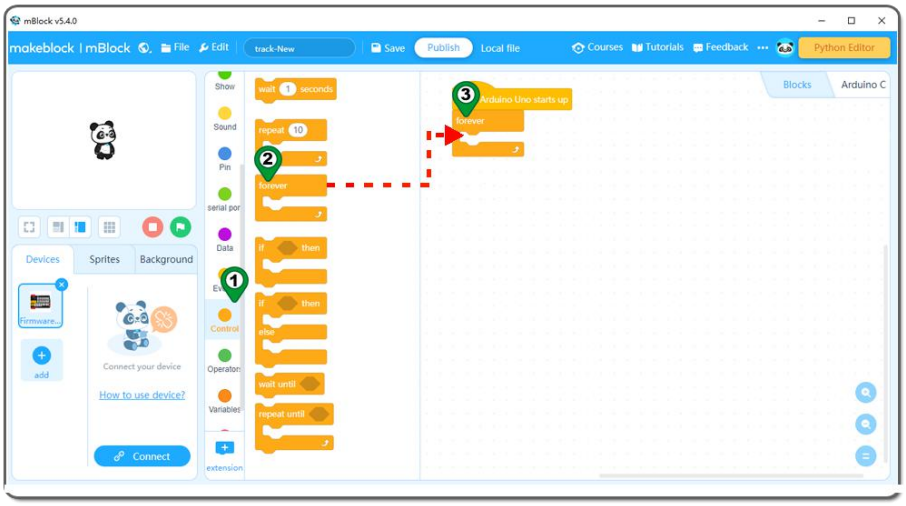
Now mBlock software and OSOYOO_UNO device firmware have been successfully installed in our PC!

Now we will show you how to use blocks to turn above idea into reality.

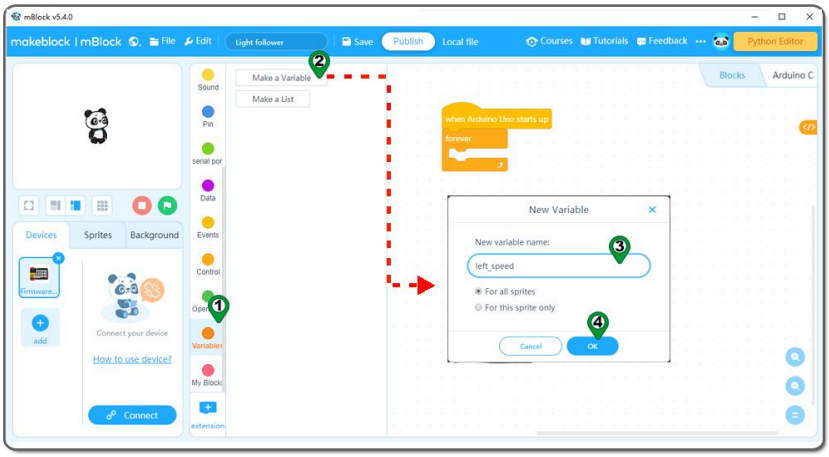
Step 5)Click Events, add when Arduino Uno starts up block to the top:



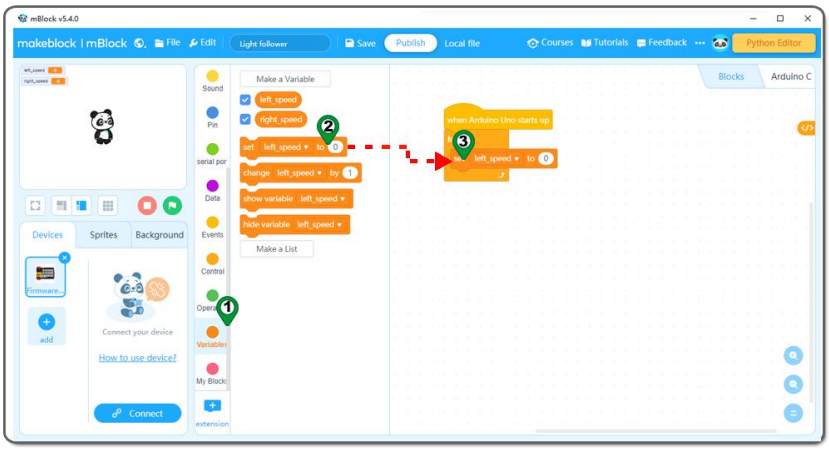
Step 6: Click **Control**, then Drag and drop **Forever** block to programming area as following:



Step7: Click **Variable** , then click **Make a Variable** ,A new dialog will pop up,write **left_speed** ,than click **OK**,you will have a new variable **left_speed**;Use the same method to create a **right_speed** variable;

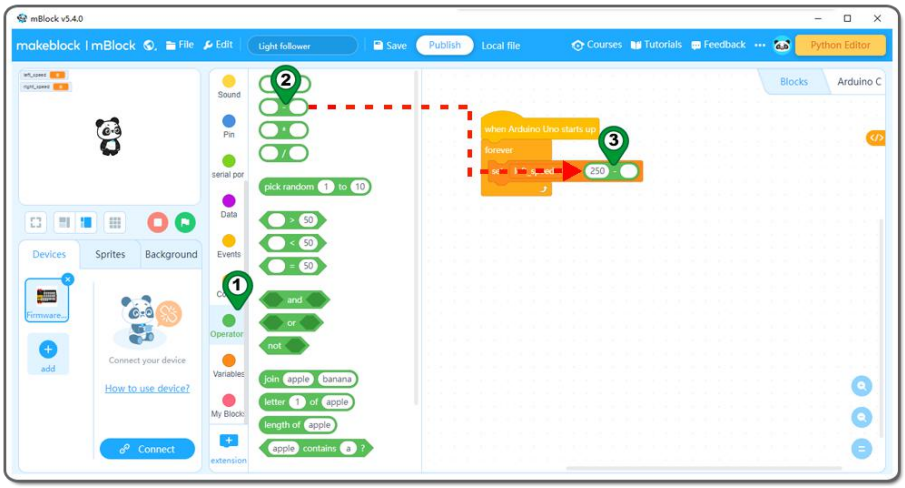


Step 8: Click **Variable** again, then Drag and drop **set left_speed to** block to programming area as following:

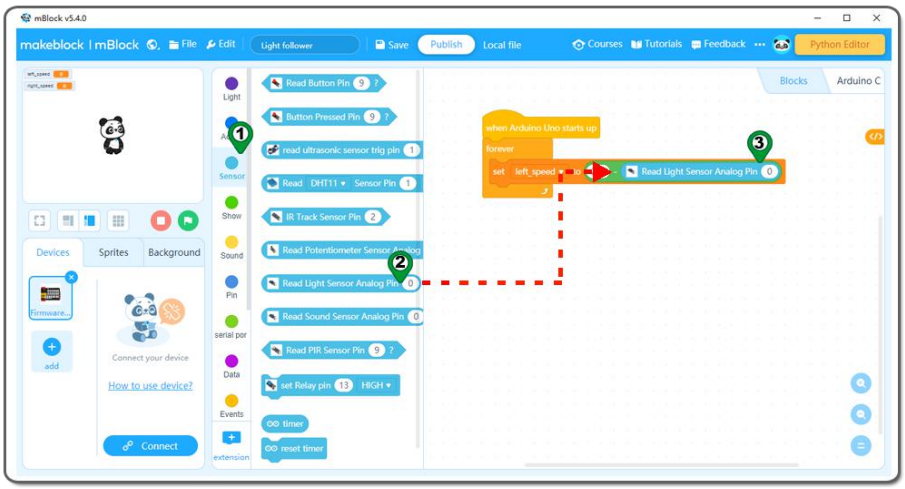


Step 9: Click **Operation** , then Drag and drop **0-0** block to programming

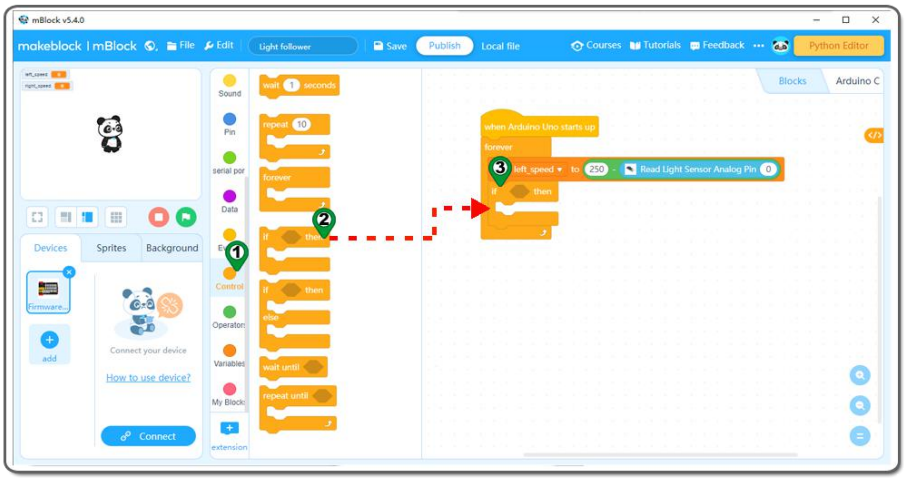
area ,write **250** to the first area as following；



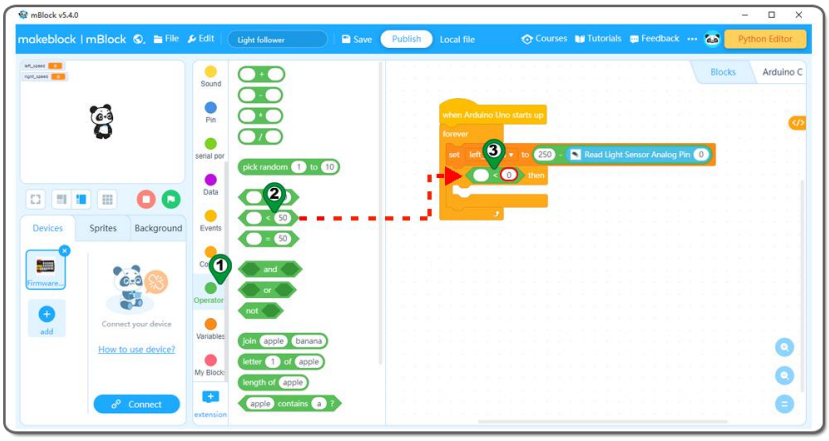
Step 10: Click **Sensor**,then Drag and drop **Read Light Sensor Analog Pin block** to programming area as following；



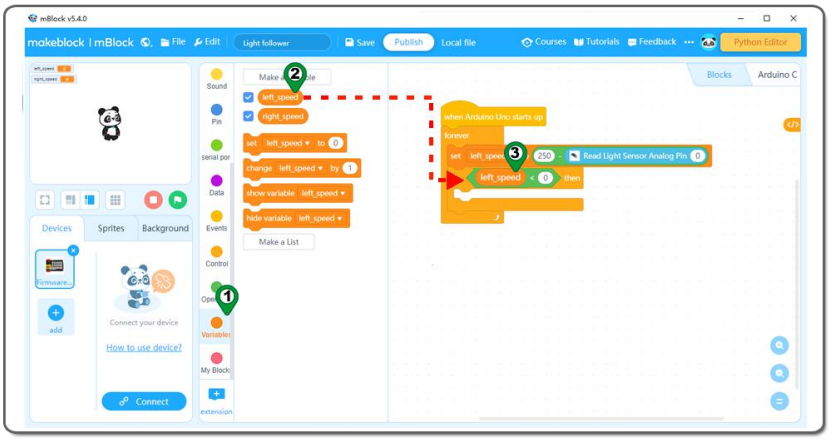
Step 11: Click **Control**, add an if then block inside forever loop as following :



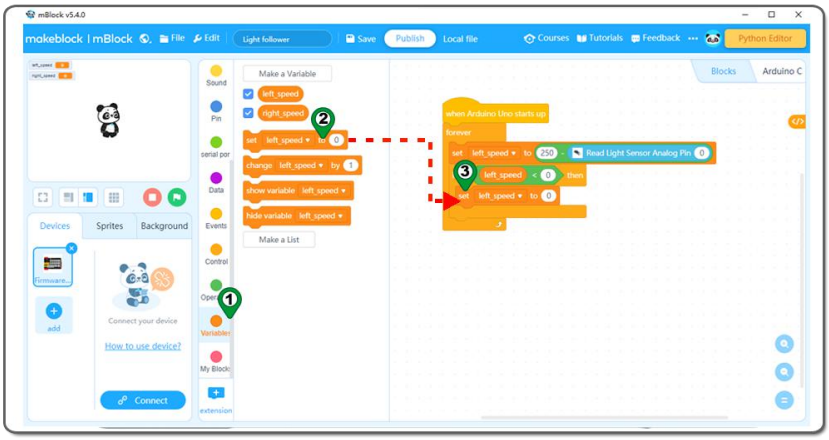
Step 12: Click **Operation** , then Drag and drop **0<0** block to programming area ,write 0 to the second area as following；



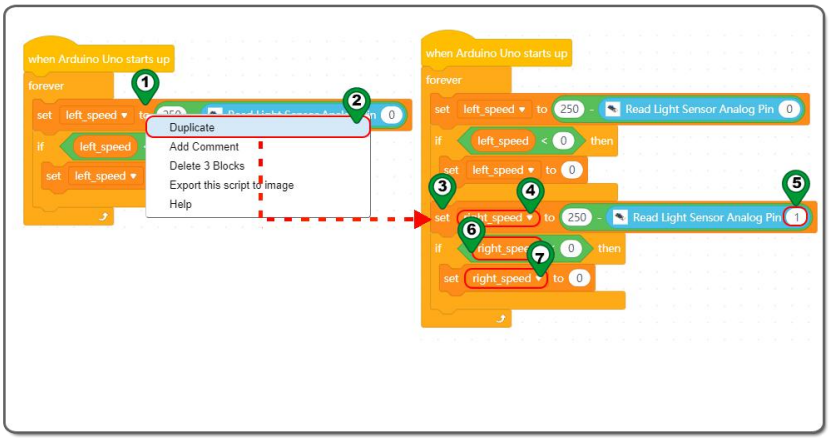
Step 13: Click **Variable** , then Drag and drop **left_speed** block to programming area as following:



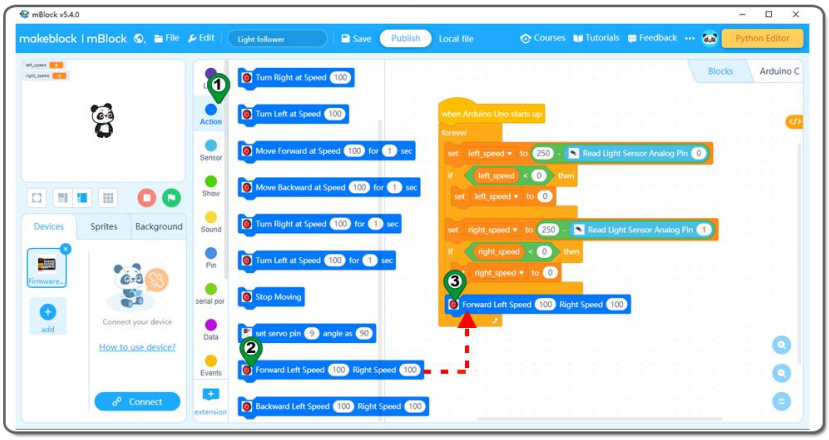
Step 14: Click **Variable** again, then Drag and drop **set left_speed to** block to programming area as following:



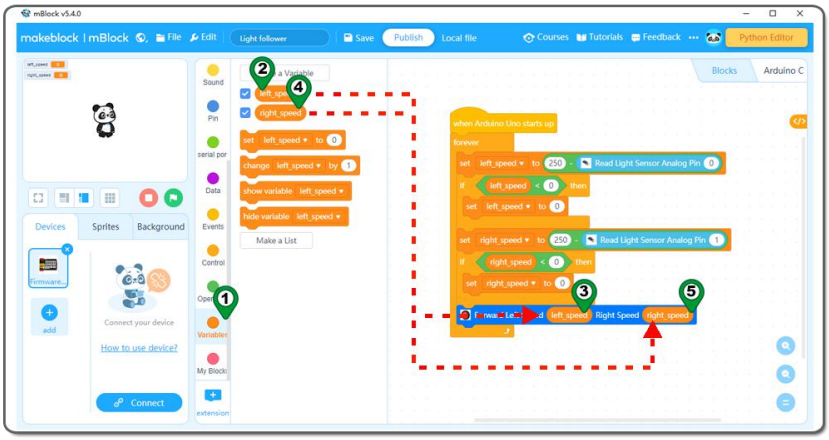
Step 15: Right Click **set left_speed to block ,** then Drag and drop these block behind **if then block ,**change the **left_speed** to **right_speed** as following:



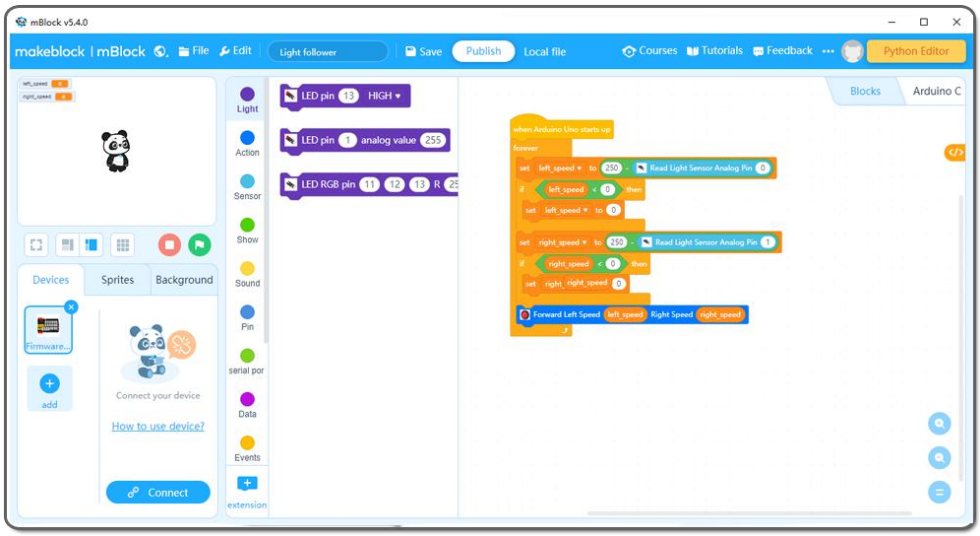
Step 16: Click **Action** again, then Drag and drop **Forward Left Speed 100 Right Speed 100** block to programming area as following:



Step17: Click **Variable ,** then **left_speed block** and **right_speed block** to programming area as following:



Now we have completed the block programming.]The final blocks look like following:



Now all the programming blocks have been completed! From above picture, the logic is pretty straight forward:

When Arduino is started, computer will enter a dead loop which will check the button status, when button is not pressed, the servo will stop at original position (in my servo 0 degree) and elevator is in the ground , when button is pressed, the servo arm will rotate from 0 degree position to 180 degree position and the elevator is lifted to the air.

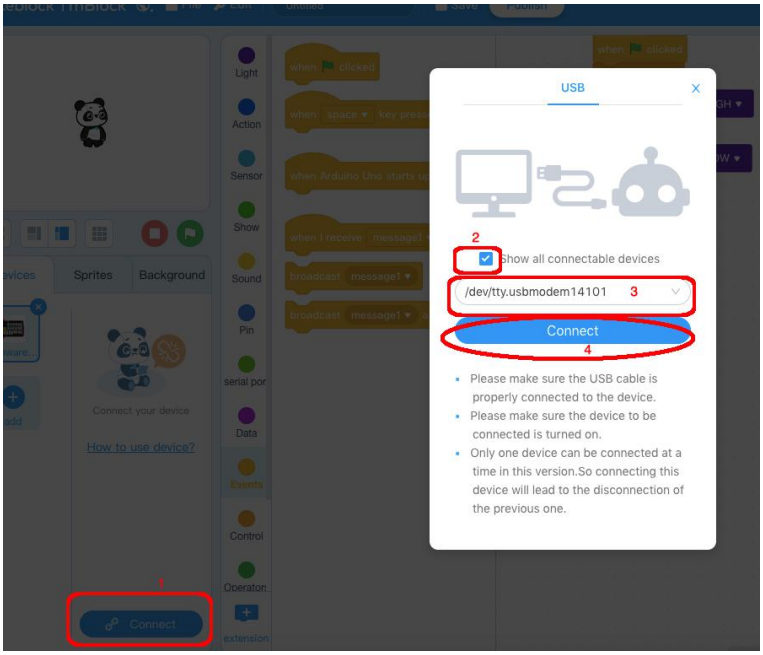
Step 18) Upload the program to OSOYOO basic board

1) Please connect your OSOYOO basic board to your PC with USB cable firstly. Then click the Connect button in the bottom of the mBlock software, you will see a USB window pop up,

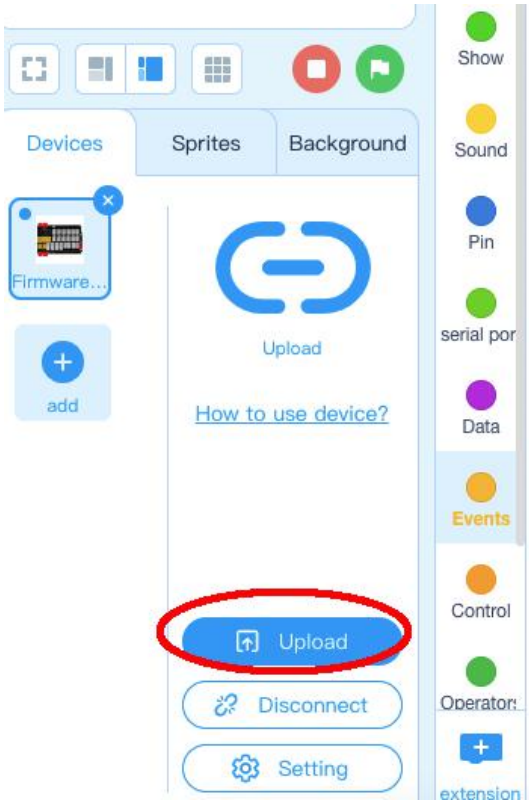
2) select **Show all connectable device** check box , then a device drop-down menu will show up,

3) select your port from device drop-down menu

4) click **Connect** button to connect your PC to OSOYOO basic board.



5)After you PC is connected to OSOYOO basic board, please click Upload button in the bottom of your software, then the code will be uploaded to OSOYOO basic board:



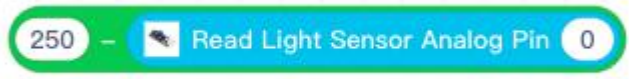
PROGRAM EXPLANATION

In previous Lesson 1 and Lesson 2, we have learned some basic control and action program blocks. In this lesson, we will learn some new program blocks:

i)Read Light Sensor from Analog Pin Block from Sensor category

This block is an oval shape block which means it returns numeric value between 0 to 255. The number zero in the right side means the sensor will read light value from A0 pin(left light sensor).

ii)Minus calculation block from Operator category



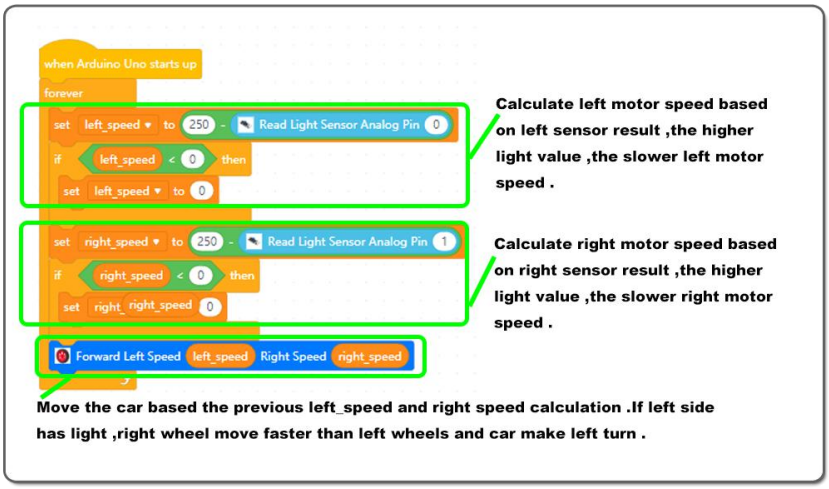
Above block calculates the result of 250 minus the value from A0 sensor .

iii)Variable Block and Set Variable to Block in Variable Category:

Variable Block is often defined to save a calculation result for later use.

Above block defined a variable block “left_speed”, then set its value to the calculation of 250 minus A0 light sensor return value. The light value is higher, the left_speed is smaller, so the car will make left turn.

Here is full program block explanation:



The working principle of Photoresistor is that the stronger light of the Photoresistor detecting, the lower value of Photoresistor reading. The value of the Photoresistor reading are more than 0.

In the program we set left motor speed as a variable of left_speed, right motor speed as a variable of right_speed. Speed motor is between 0-255.

The variable of left_speed equals to 250 minus the value of left Photoresistor(A0) reading, and the variable of right_speed equals to 250 minus the value of right Photoresistor(A1) reading. If the variables are less than 0, the program sets these variables equal to 0。

If the value of left sensor(A0) reading is less than 250, left motor will move forward;

If the value of left sensor(A0) reading is more than 250, left motor will stop;

If the value of right sensor(A1) reading is less than 250, right motor will move forward;

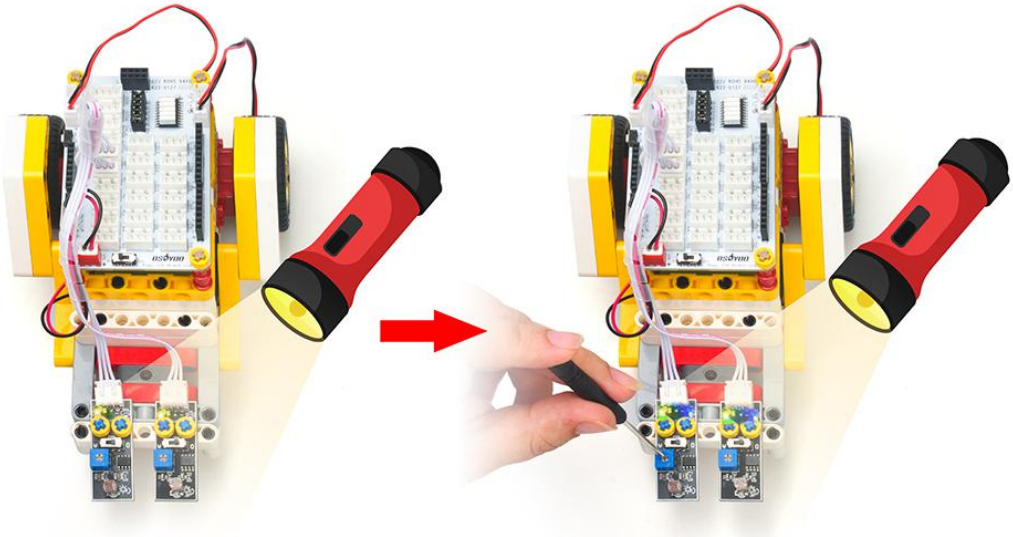
If the value of right sensor(A1) reading is more than 250, right motor will stop;

If the value of both sensor reading are less than 250, both motors move forward.

Upload followlight.ino sketch code to Arduino. Turn on the car.

Illuminate the front photoresistor sensor with a flashlight. Then the car will follow the flashlight to make movement.

If the torch shines on the photoresistor sensor, but the car does not move. Please adjust the adjuster as shown in the picture. Under normal circumstances, the yellow light and the blue light will be on at the same time when the torch is illuminated.



Lesson 4: Obstacle Avoidance Car

OBJECTIVE

In this project, we will use obstacle sensor(ultrasonic distance detector) to detect obstacles ,avoid collision and auto-drive.

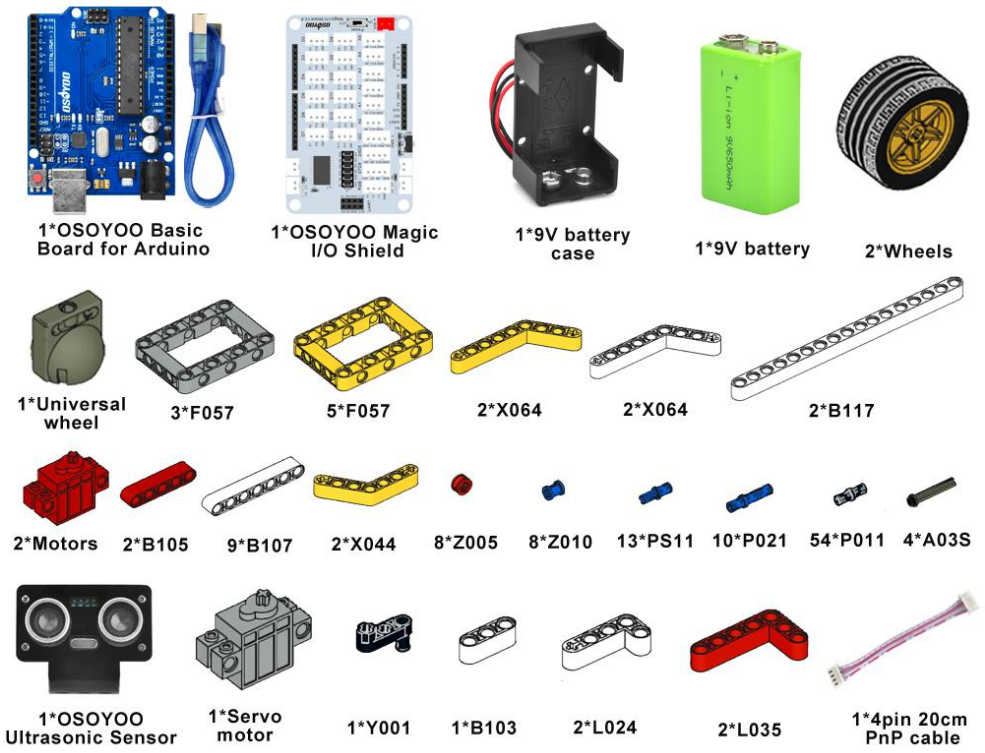
PARTS & DEVICES

Please prepare the following parts to complete this project

NOTE:

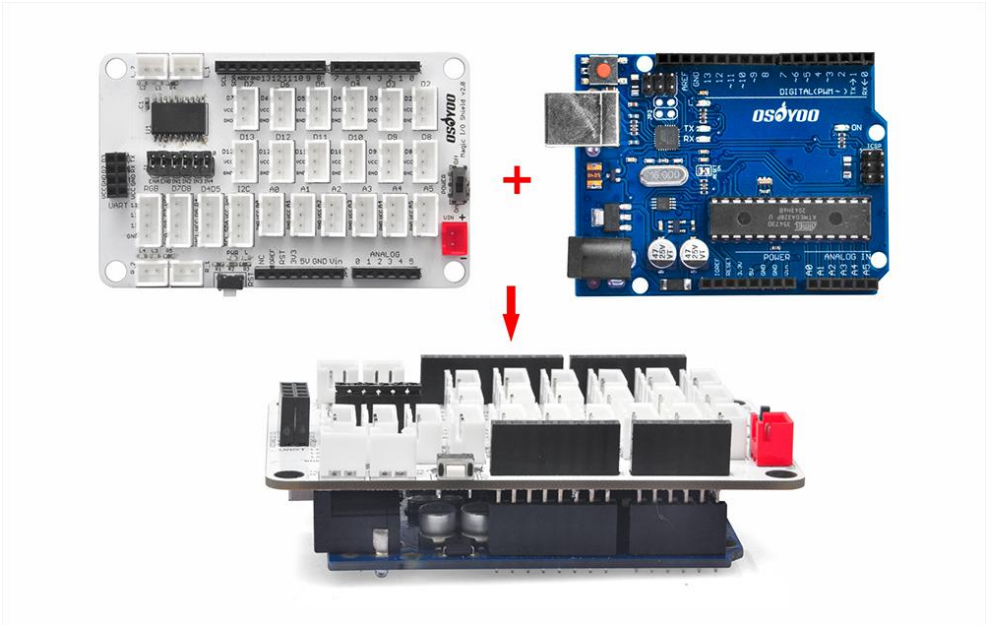
1.the color of the building block is subject to the actual product, which does not affect the use.

2. ALL OSOYOO PRODUCTS FOR ARDUINO ARE THIRD PARTY BOARD WHICH IS FULLY COMPATIBLE WITH ARDUINO



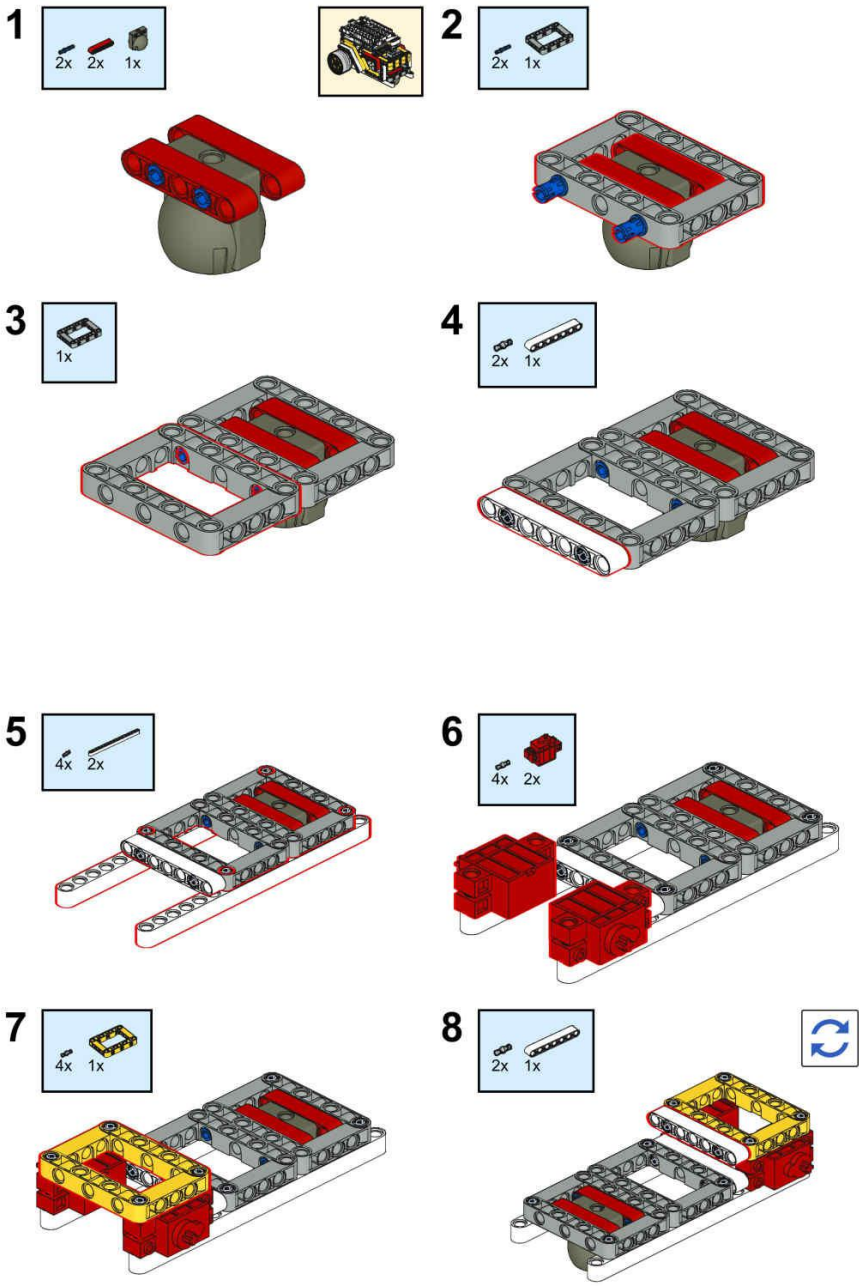
HOW TO MAKE

Before you build this robot car with blocks, please install OSOYOO basic board for Arduino under OSOYOO Magic I/O shield as following *(Attention please : the pins of I/O shield is* aligned with the port of basic the board firstly, then press the shield tightly on the board).

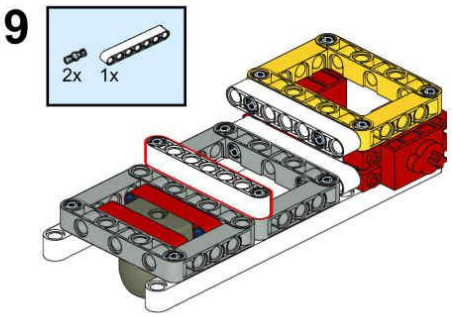


Please follow the building steps to build this robot car, If you want to get clear PDF building steps, please download from https://osoyoo.com/picture/Building_Robot_Car/lesson4/LESSON4.pdf

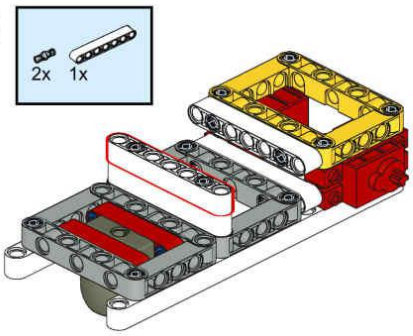
Note: If you have built the robot car for lesson1, please skip to the step35 in this PDF.



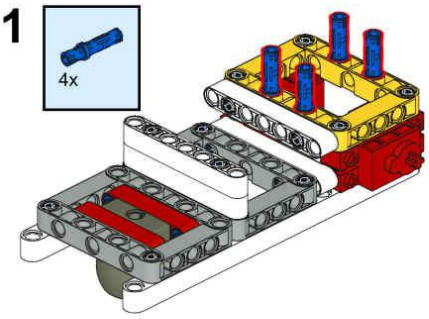
9



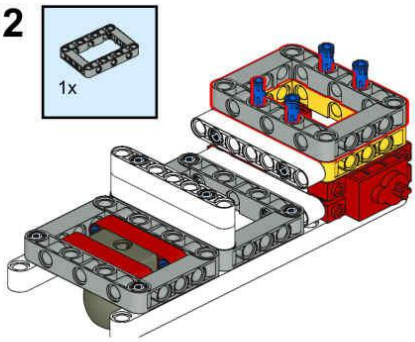
10



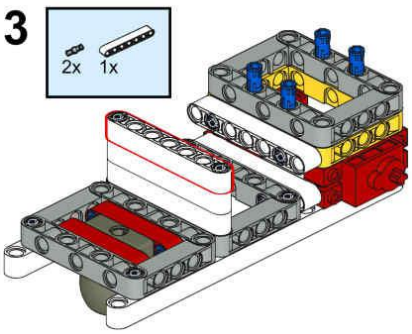
11



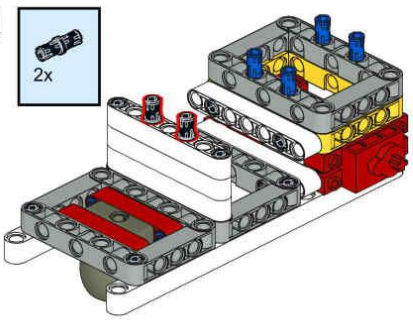
12



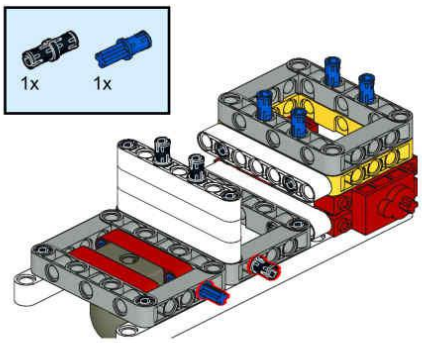
13



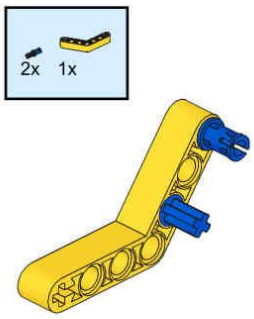
14

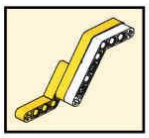


15

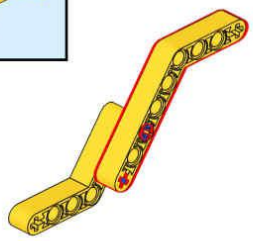


16

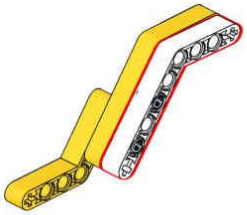




17

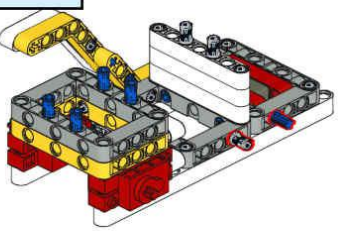


18

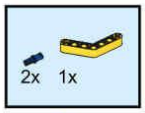


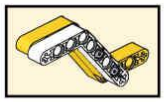
19

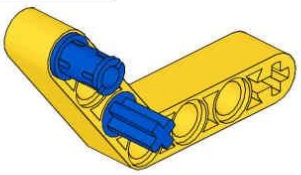
20



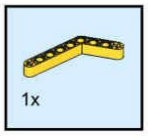
21

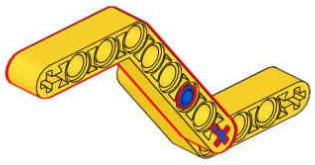




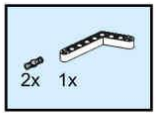


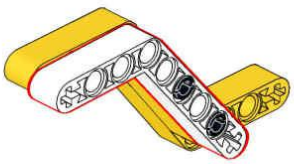
22



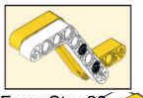


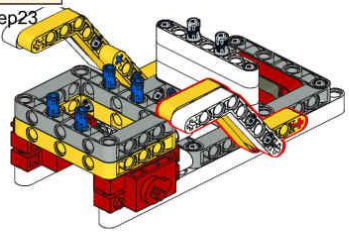
23

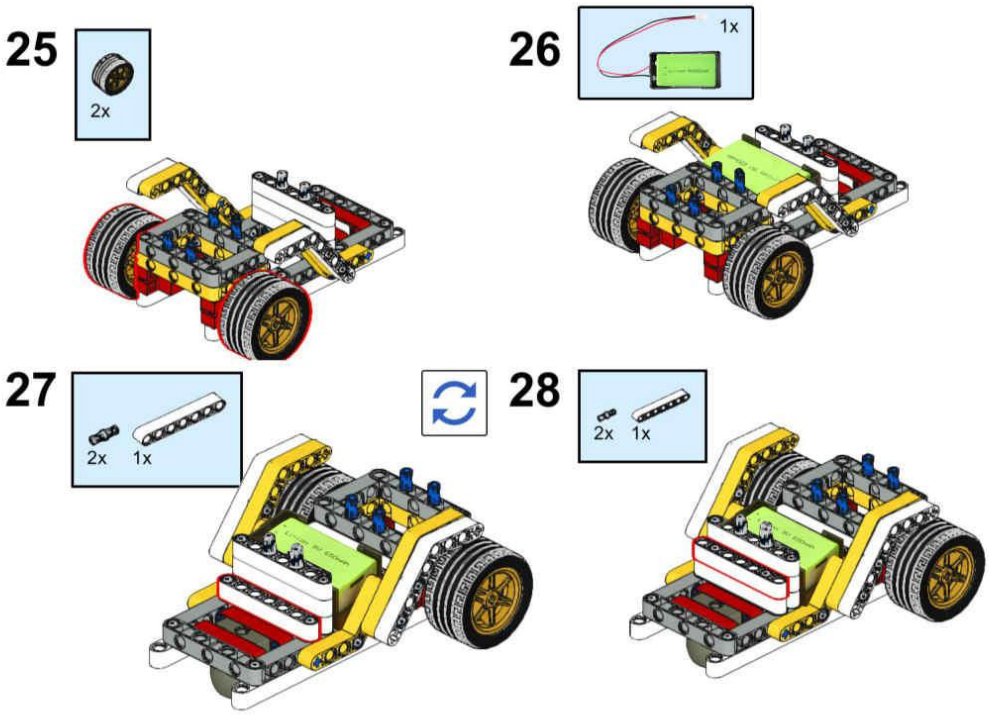


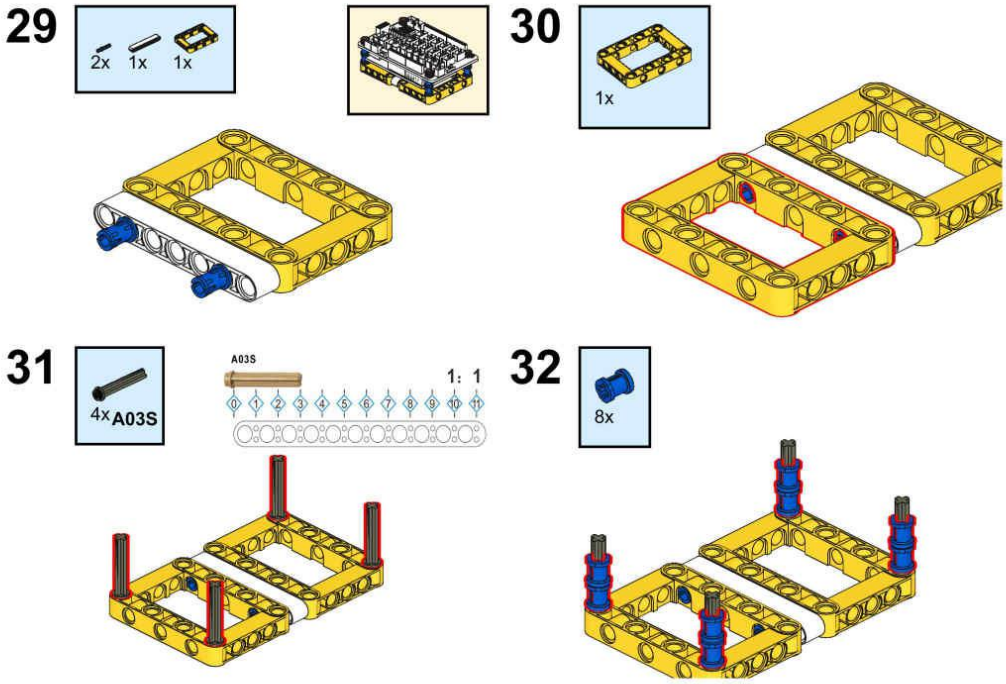


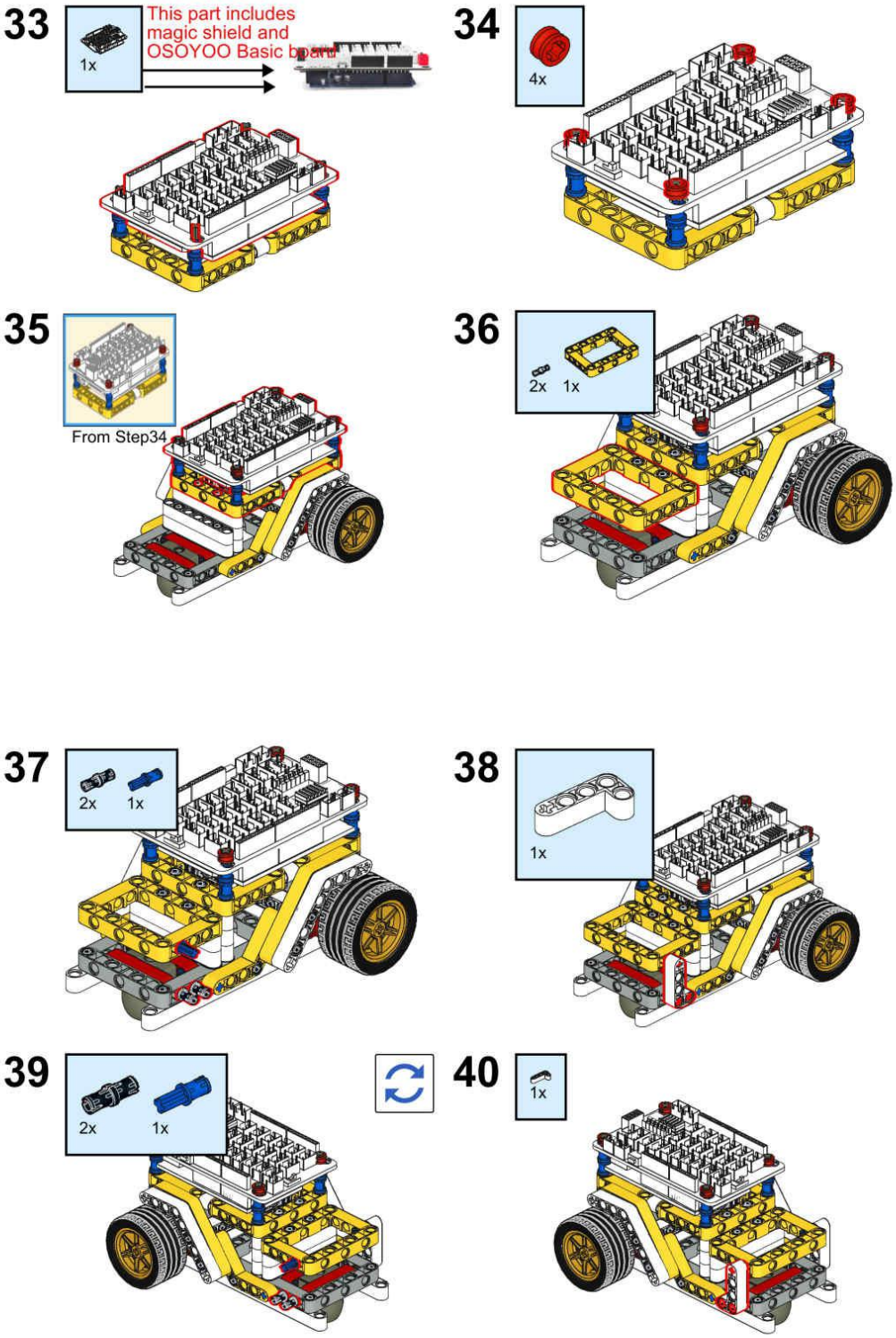
24



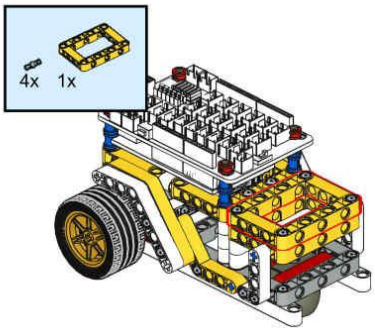



25
2x
26
1x
27
2x
1x
28
2x
1x

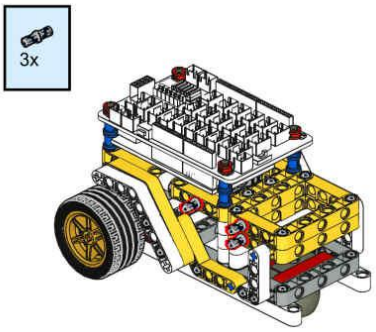

29
2x
1x
1x
30
1x
31
4x
A03S
A03S
1: 1
32
8x


33
1x
This part includes
magic shield and
OSOYOO Basic board
34
4x
35
From Step34
36
2x
1x
37
2x
1x
38
1x
39
2x
1x
40
1x

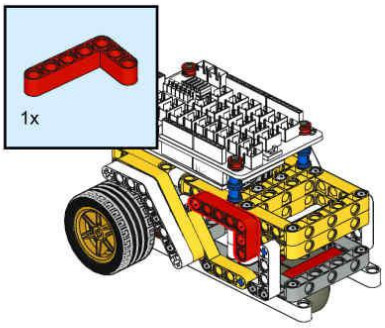
41



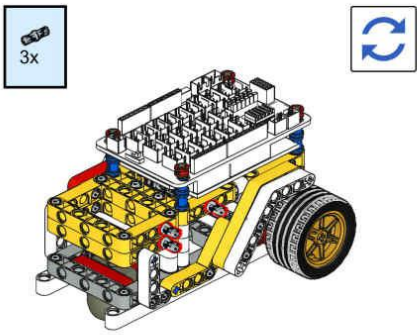
42



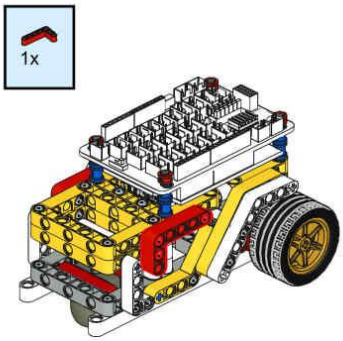
43



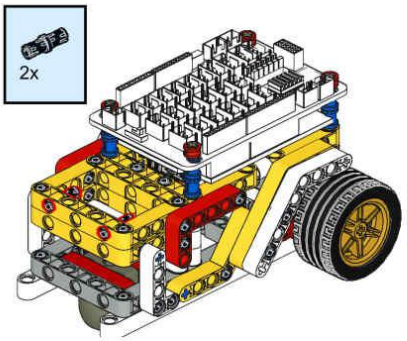
44



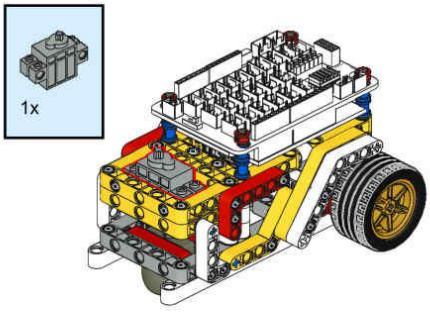
45



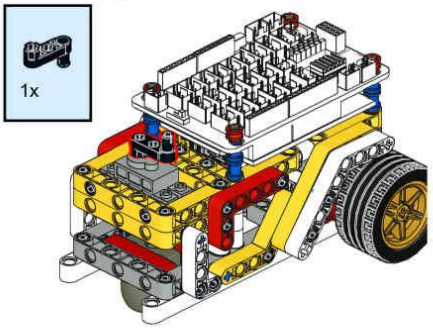
46

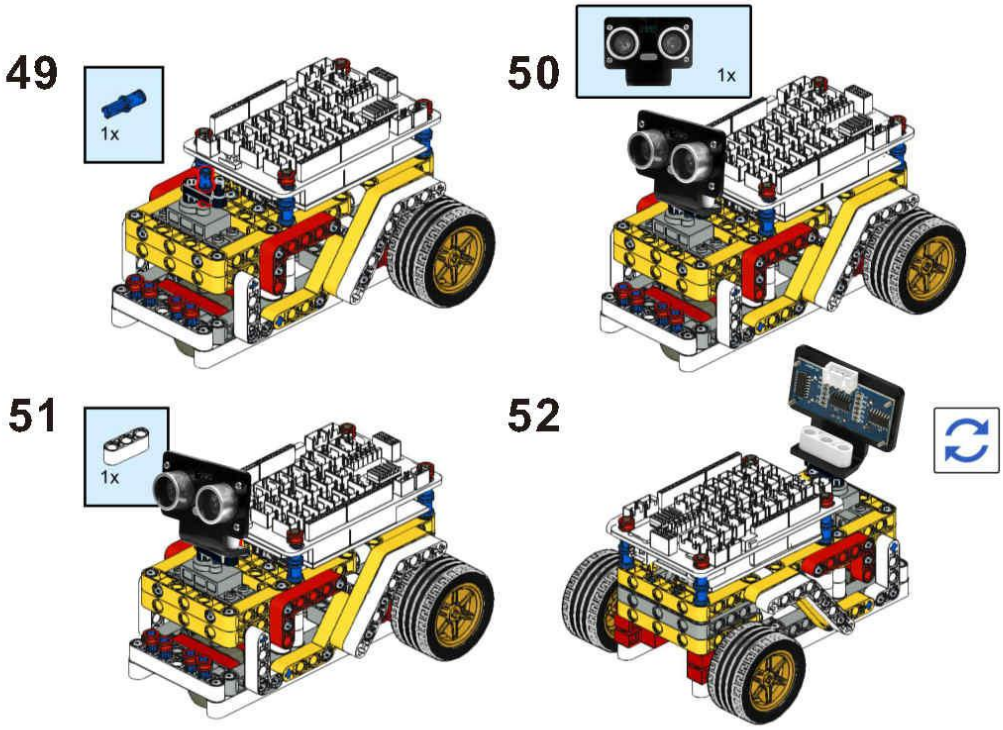


47



48

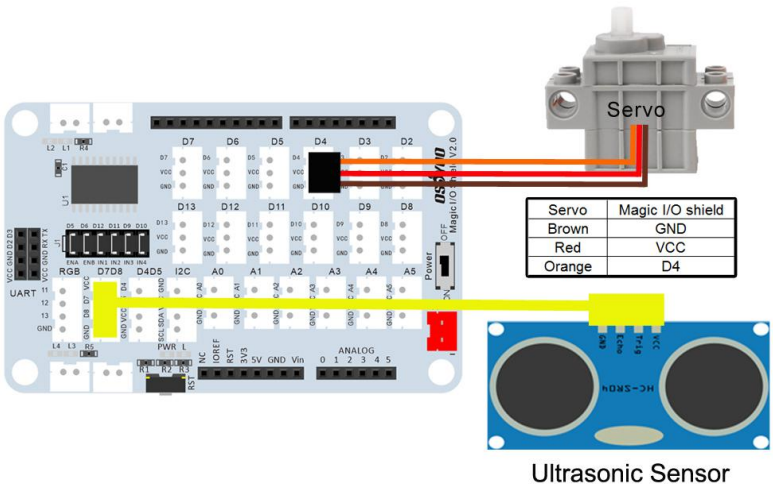




CIRCUIT CONNECTION

Please connect motors and 9V battery case as lesson1.

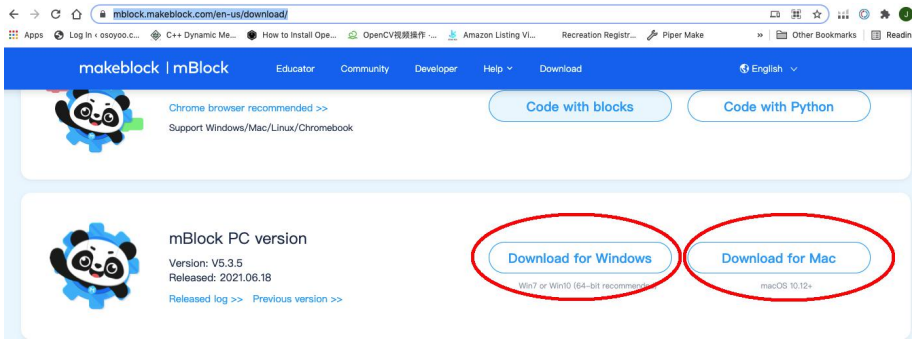
Then Connect SG90 servo motor to D4 port of OSOYOO Magic I/O shield, Ultrasonic module sensor to the D7D8 port of the Magic I/O shield with a 4-pin PNP cable as below:



HOW TO CODE

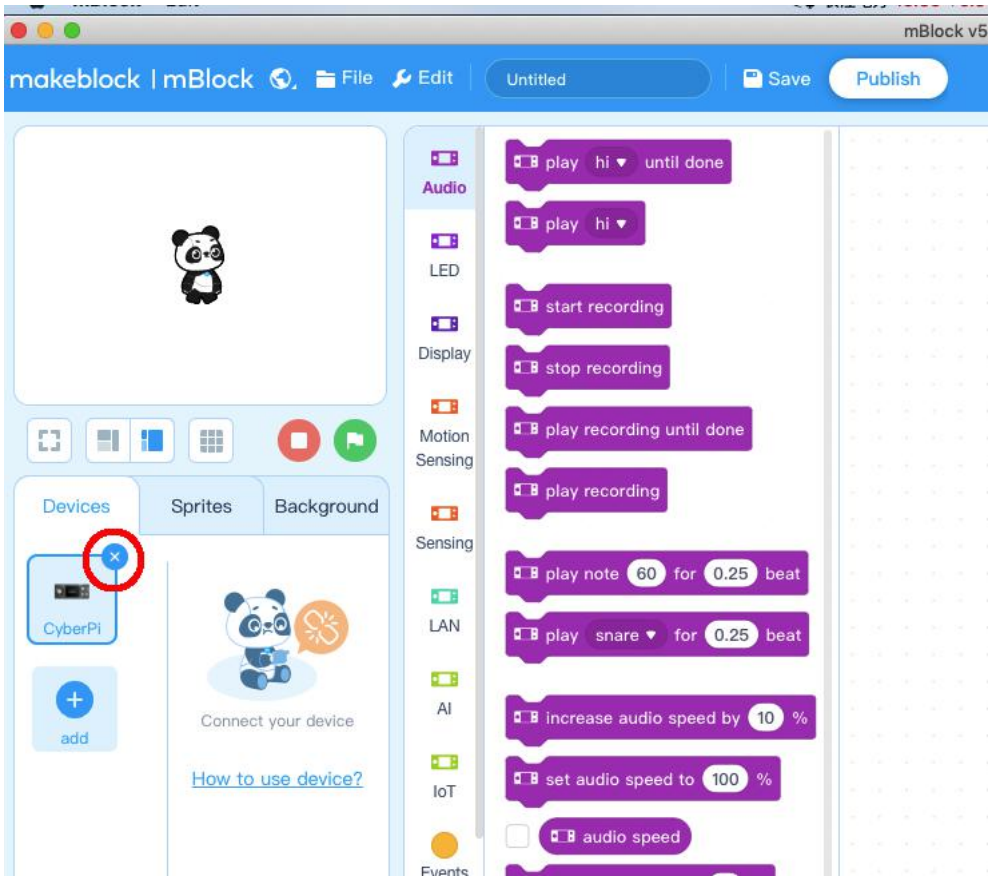
Note: In this kit, we use mblock as programming tool, if you want to learn more about mblock, please visit preparation before class 1: https://osoyoo.com/2021/12/01/preparation-before-class-1

Step 1) Download **mBlock PC version** from https://mblock.makeblock.com/en-us/download/, select the download file as per your computer OS type (Please don't use the browser version):

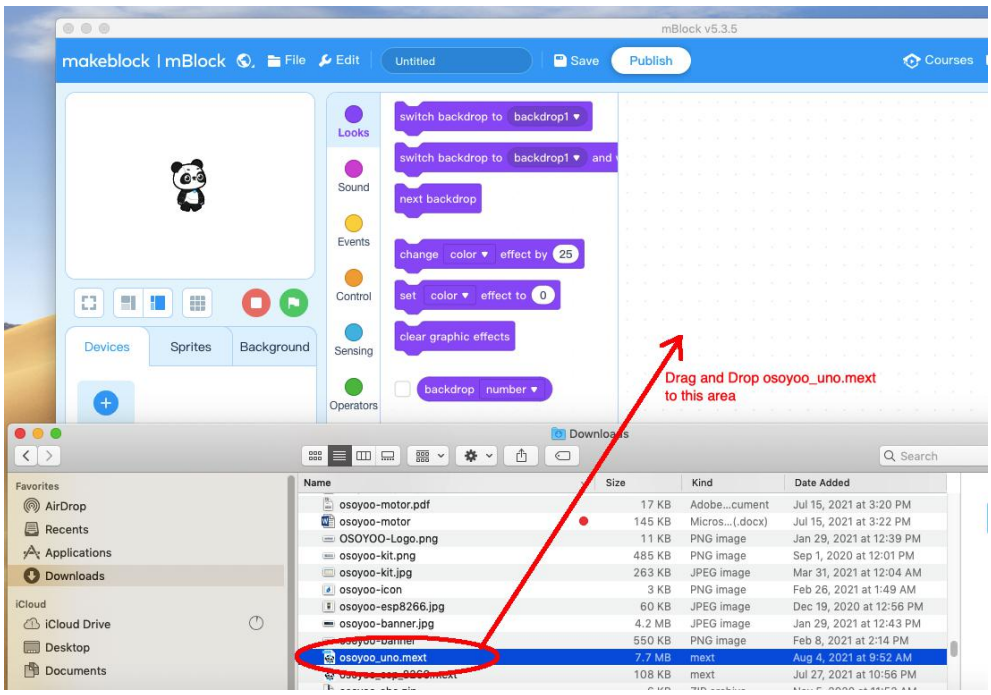


Step 2) Download OSOYOO_UNO.mext device file from https://osoyoo.com/driver/mblock/osoyoo_uno.mext

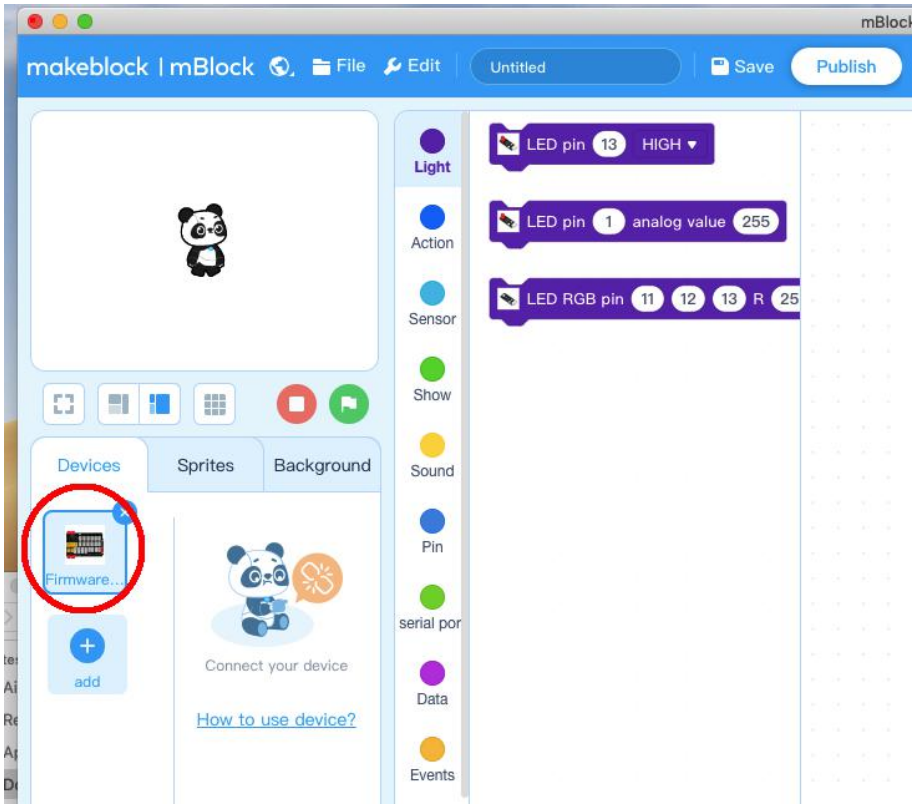
Step 3) Run the mBlock PC software by double click the lovely Panda icon. you will see mBlock UI as following picture. Please delete the default device **CyberPi** by click the cross in the red circle.



Step 4) Drag and Drop osoyoo_uno_mext file(downloaded in Step 2) to mBlock software as following:



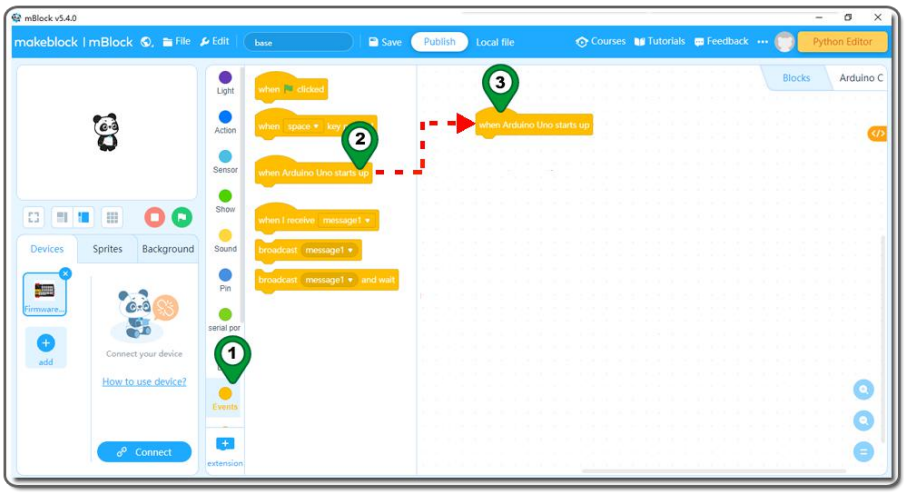
Now you will see a new device firmware in mBlock, see following picture:



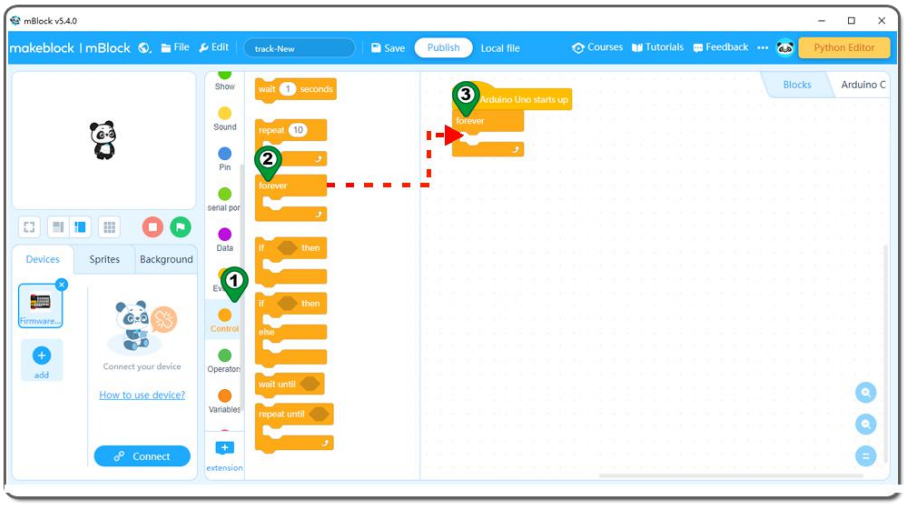
Now mBlock software and OSOYOO_UNO device firmware have been successfully installed in our PC!

Now we will show you how to use blocks to turn above idea into reality.

Step 5: Click **Events**, add **when Arduino Uno starts up** block to the top:

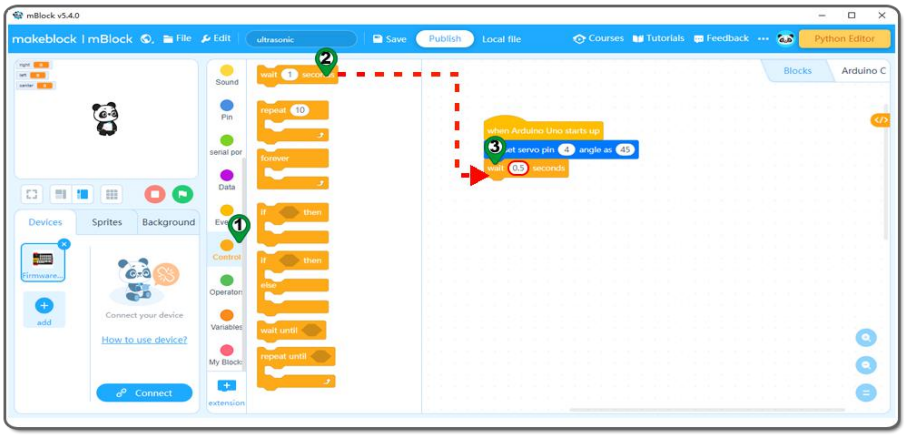


Step 6: Click **Control**, then Drag and drop **Forever** block to programming area as following:

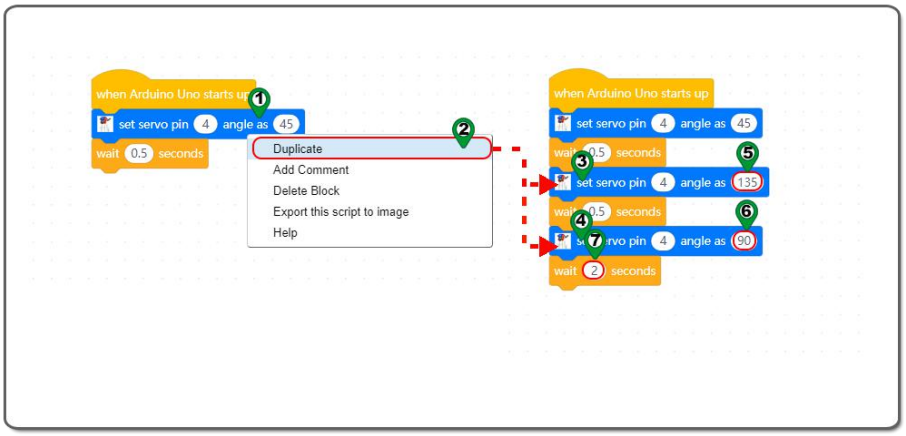


Step 7: Click **Action**, then Drag and drop **set servo pin angle as** block to programming area as following

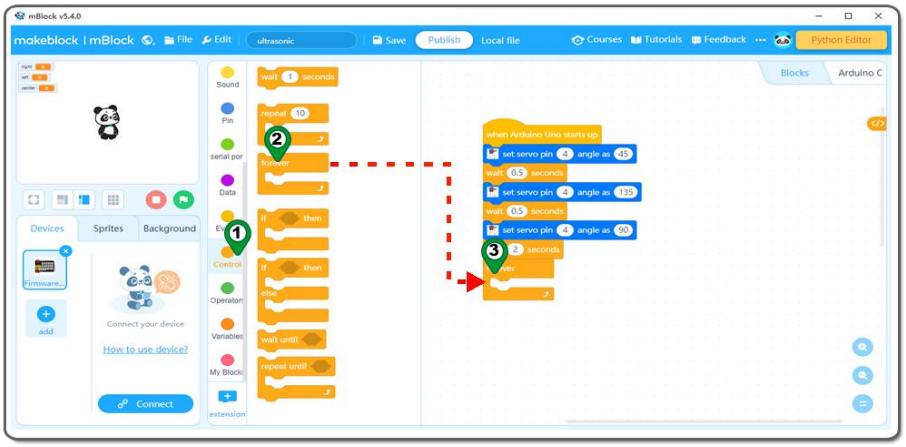
Step 8: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed 0.5 seconds as following:



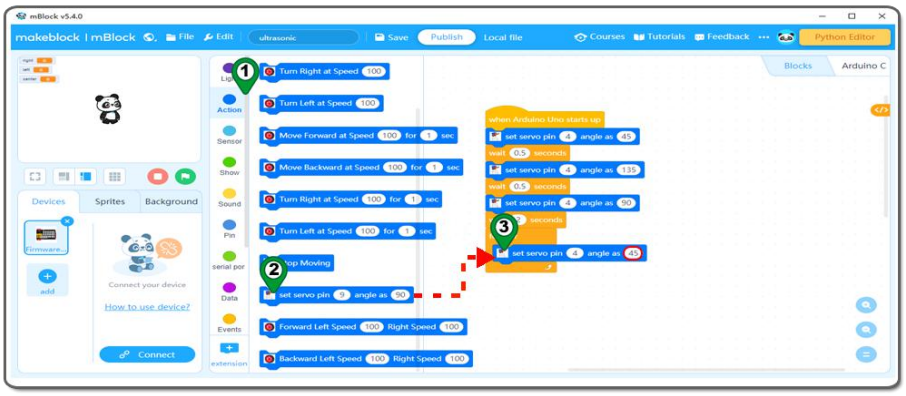
Step 9: Right click **set servo pin angle as** block ,Click duplicate from the digital write pin to block over . You will get a duplicated block ,and place it behind the **wait seconds** block .set the 45 to 135 and 90 value as following figure.



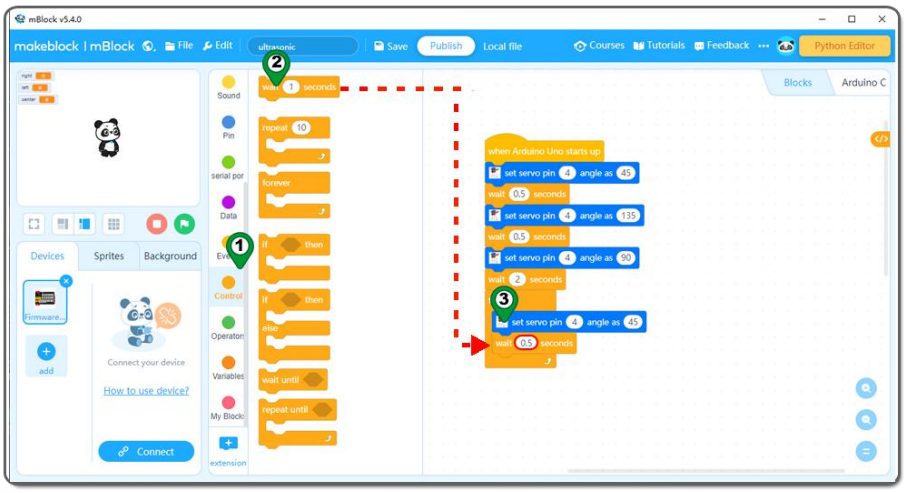
Step 10: Click **Control**, then Drag and drop **forever** block to programming area as following:



Step 11: Click **Action**, then Drag and drop **set servo pin angle as** block to programming area,change the 90 to 45 value as following;

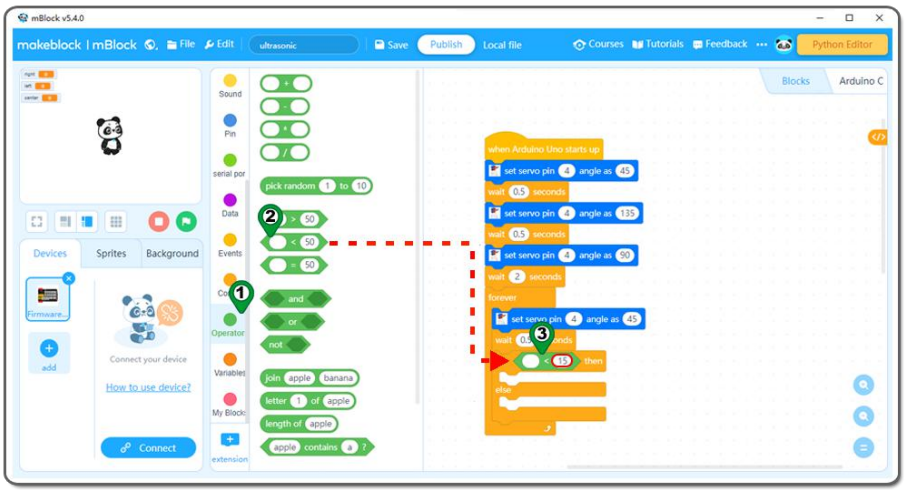


Step 12: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed 0.5 seconds as following:

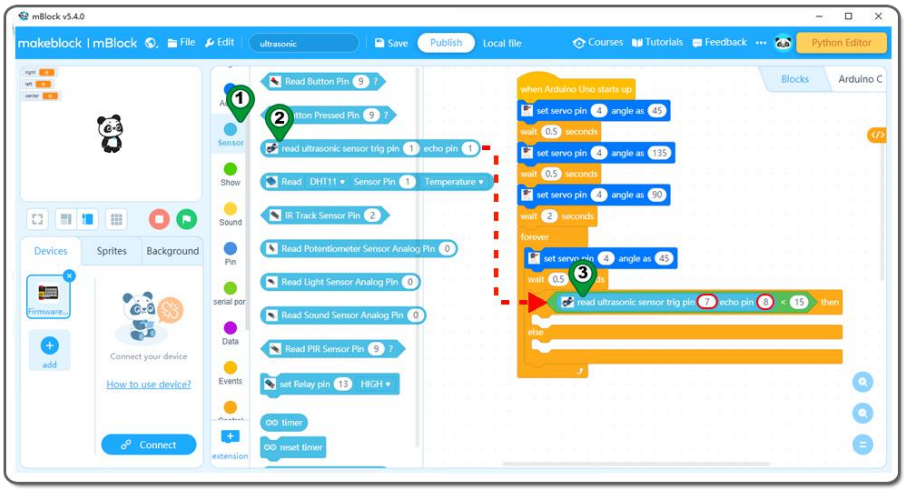


Step 13: Click **Control**, then Drag and drop **if then else** block to programming area as following:

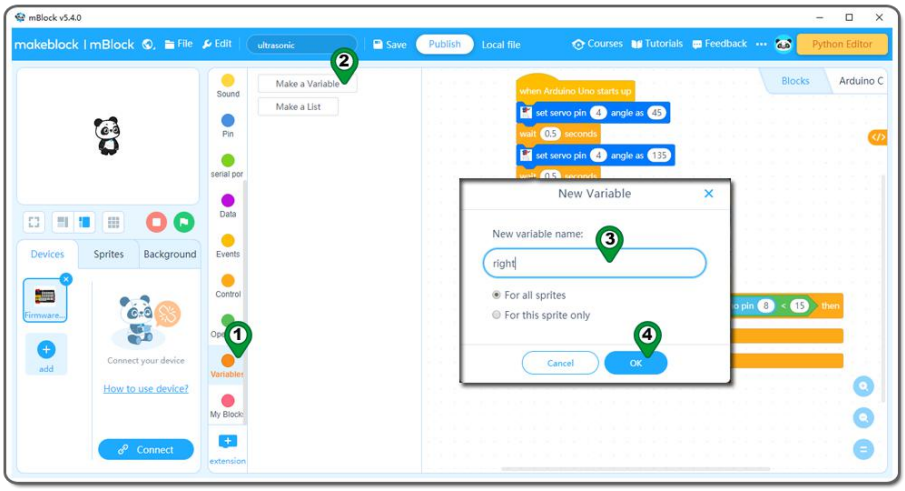
Step 14: Click **Operation**, then Drag and drop **0<0** block to programming area ,change the 0 to 15 as following:



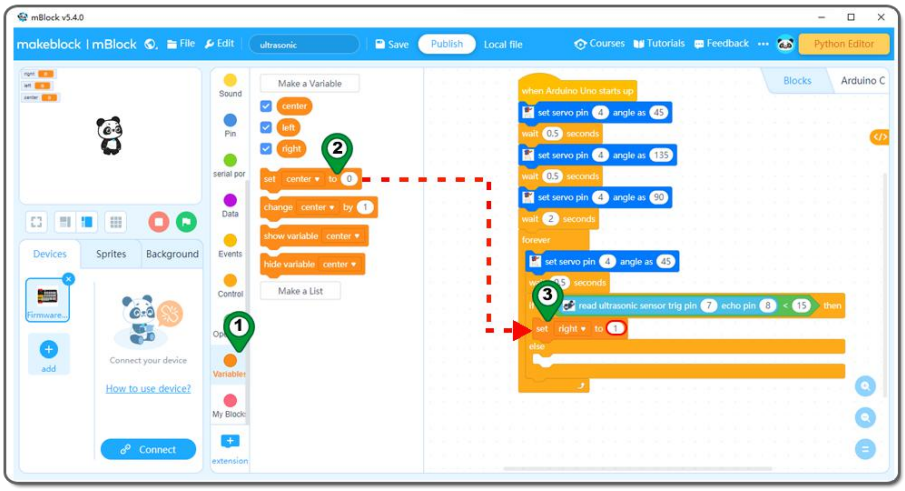
Step 15: Click **Sensor**, then Drag and drop **read ultrasonic sensor trig pin echo pin** block to programming area ,writed pin 7 and pin 8 as following;



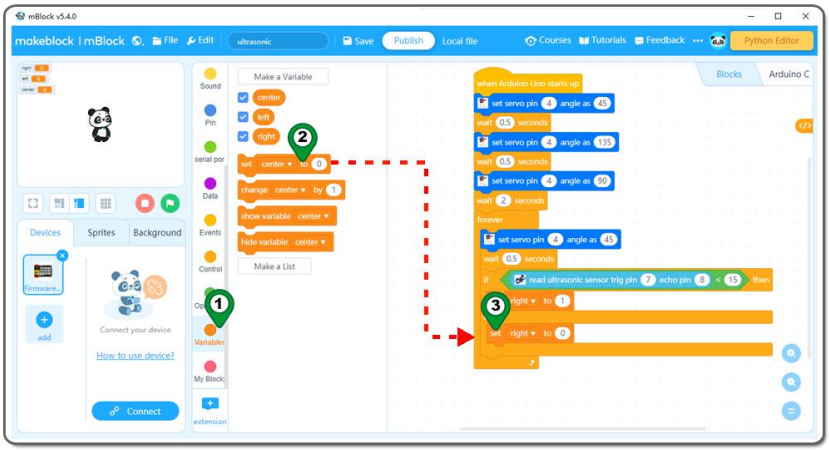
Step 16: Click **Variable** , then click the **make a Variable** block ,writed **right** in the dialog box blank as following. Create **left** and **center** variables in the same way;



Step 17: Click **Variable**, then Drag and drop **set center to** block to programming area ,change the **center** to **right** ,writed **0** to **1** value , as following:



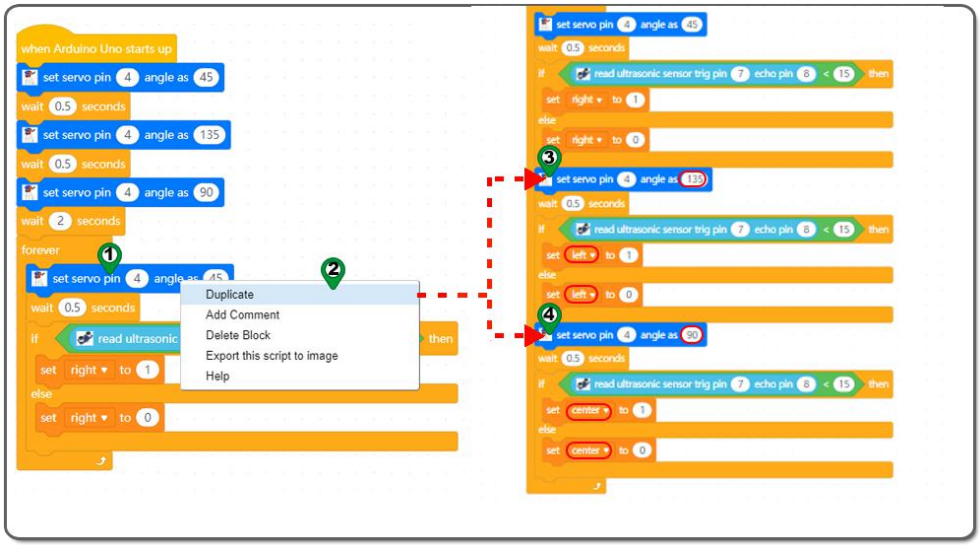
Step 18: Repeat the previous step,change the **1** to **0** as following:



Step 19: Right click **set servo pin angle as** block ,Click duplicate from the digital write pin to block over . You will get a duplicated block ,and place it behind the **wait**

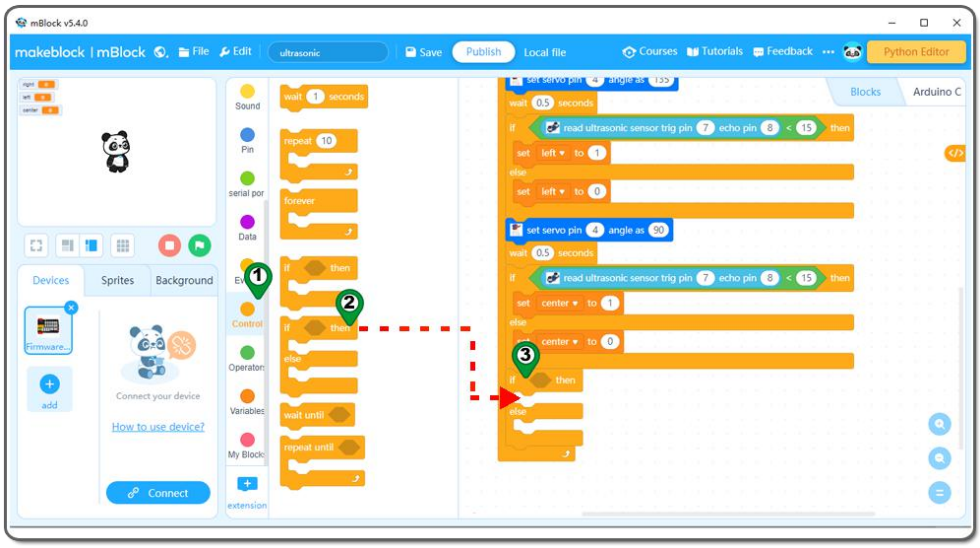
seconds block .set the **45** to **135** and **90** value,change

the **right** to **left** and **center** as following figure.



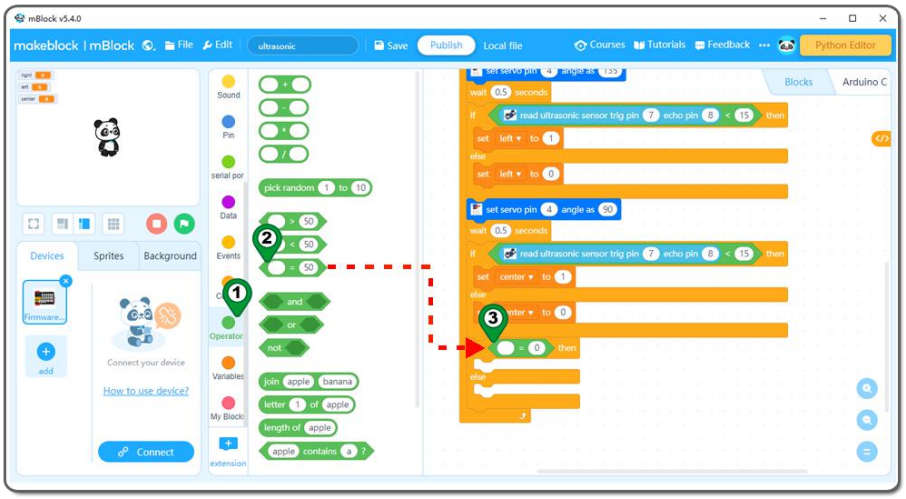
Step 20: Click **Control**, then Drag and drop **if then else** block to programming

area as following:

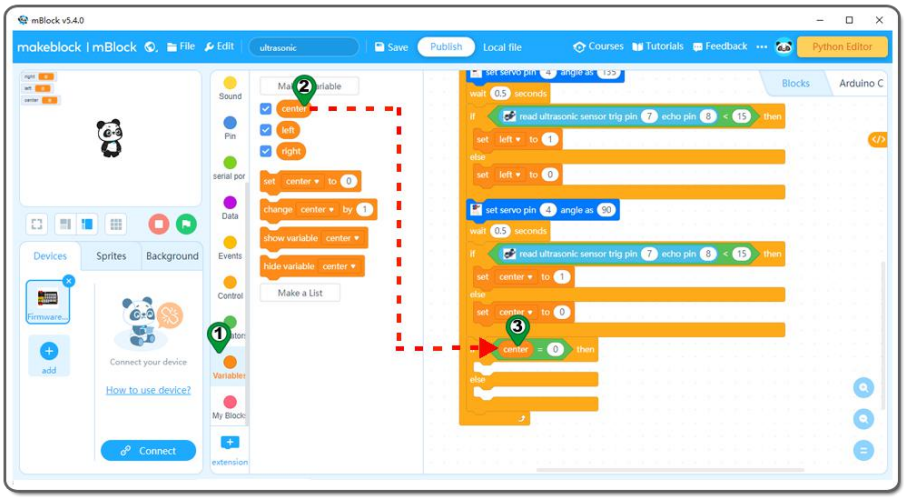


Step 21: Click **Operation**, then Drag and drop **0=0** block to programming

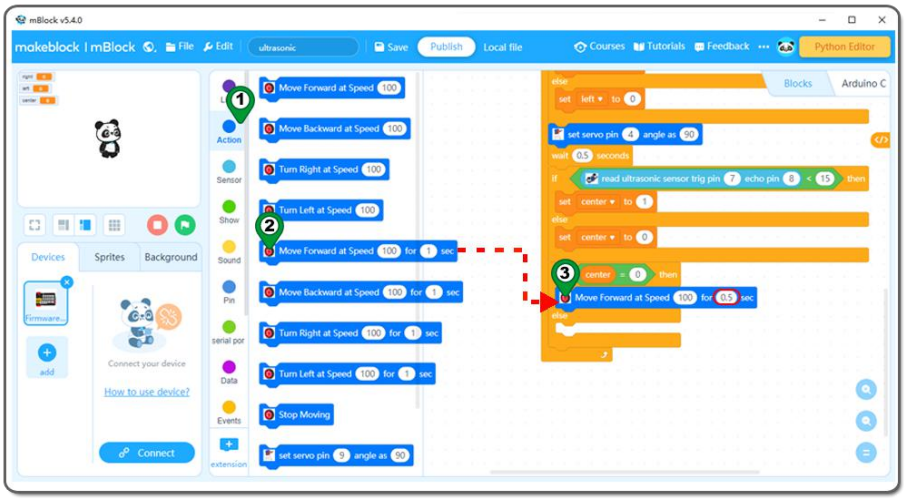
area ,change the 50 to 0 as following:



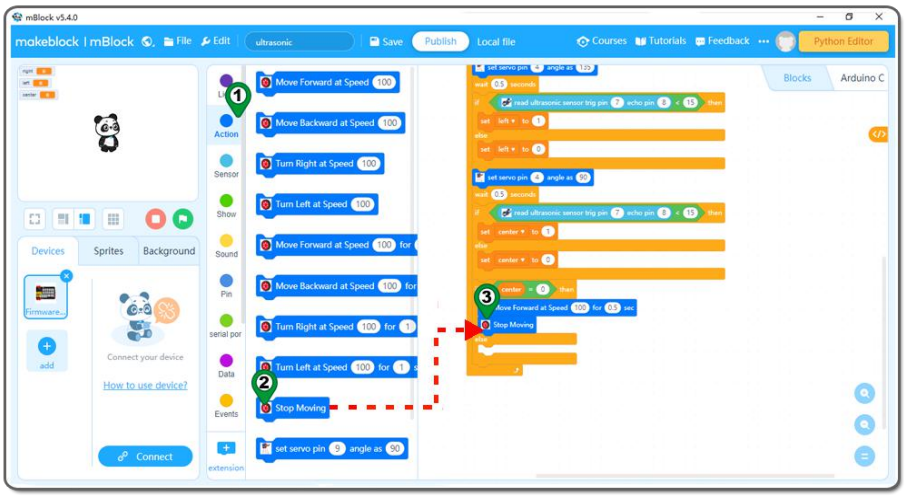
Step 22: Click **Variable**, then Drag and drop **center** block to programming area as following:



Step 23: Click **Action**, then Drag and drop **Move forward at speed for** block to programming area,change the **1 sec** to **0.5 sec** as following;

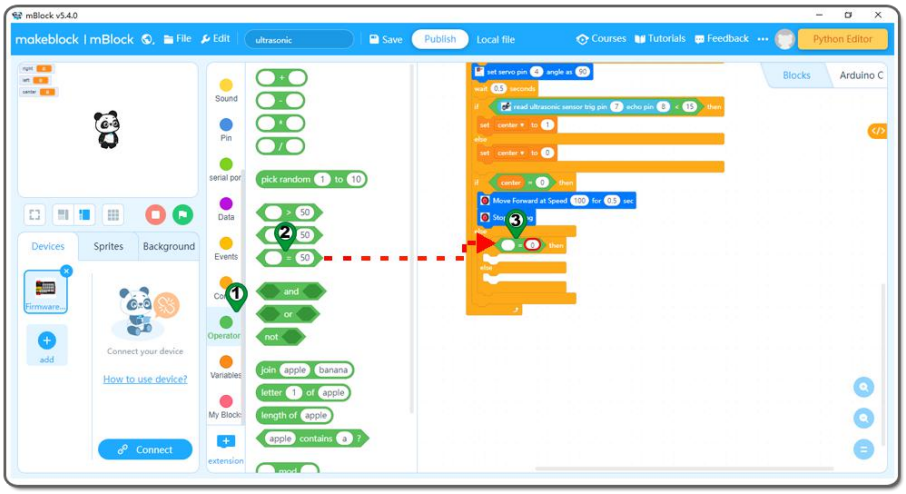


Step 24: Click **Action** again, then Drag and drop **Stop Moving** block to programming area as following;

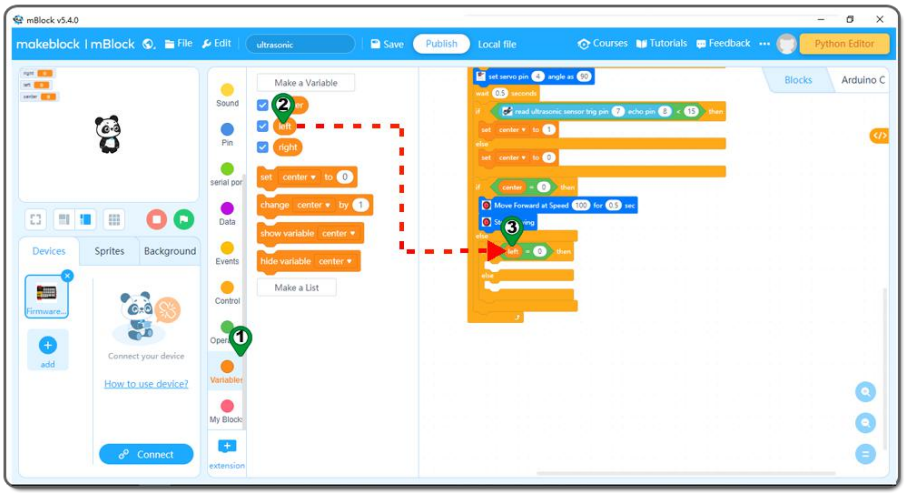


Step 25:Click **Control**, then Drag and drop **if then else** block to programming area as following.

Step 26: Click **Operation**, then Drag and drop **0=0** block to programming area ,change the **50** to **0** as following；



Step 27: Click **Variable**, then Drag and drop **left** block to programming area as following:

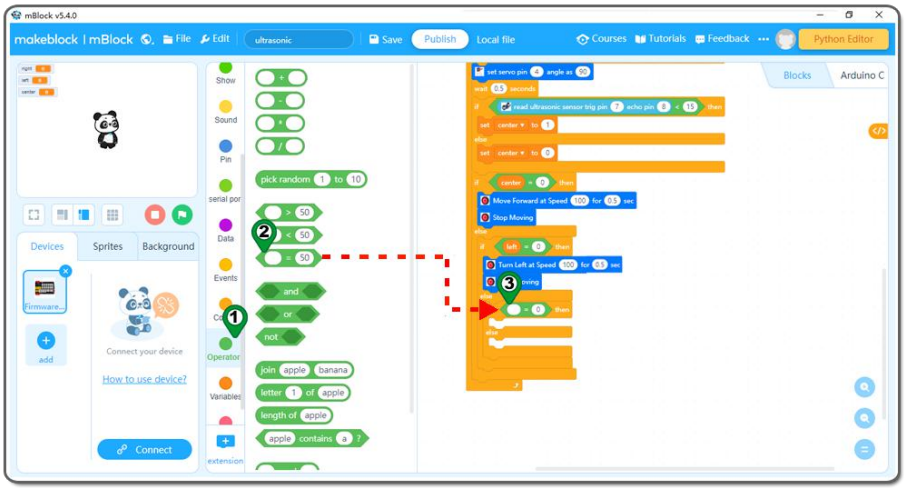


Step 28: Click **Action**, then Drag and drop **Turn Left at speed for** block to programming area,change the **1 sec** to **0.5 sec** as following;

Step 29: Click **Action** again, then Drag and drop **Stop Moving** block to programming area as following;

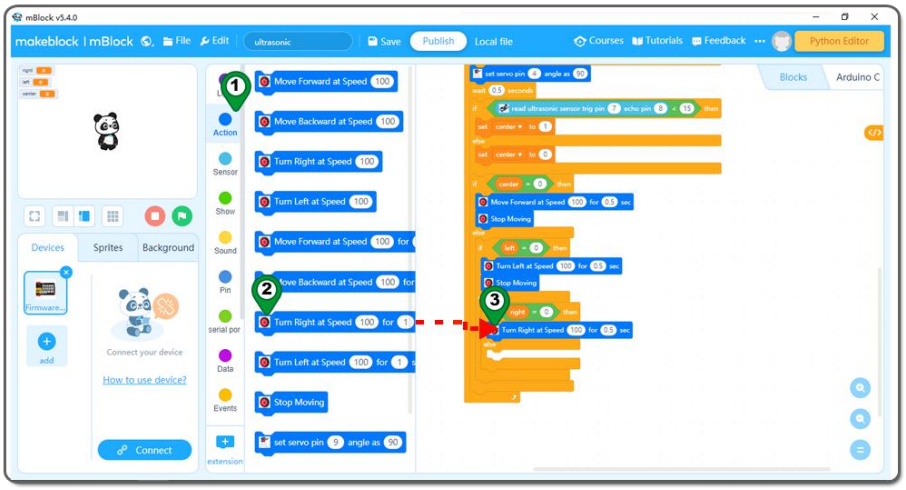
Step 30: Click Control, then Drag and drop **if then else** block to programming area as following.

Step 31: Click Operation, then Drag and drop **0=0** block to programming area ,change the **50** to **0** as following；



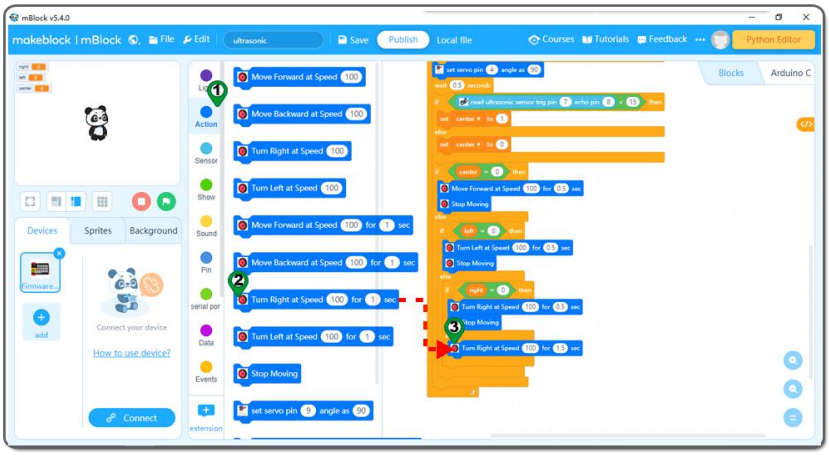
Step 32: Click **Variable**, then Drag and drop **right** block to programming area as following:

Step 33: Click **Action**, then Drag and drop **Turn Right at speed for** block to programming area,change the **1 sec** to **0.5 sec** as following;

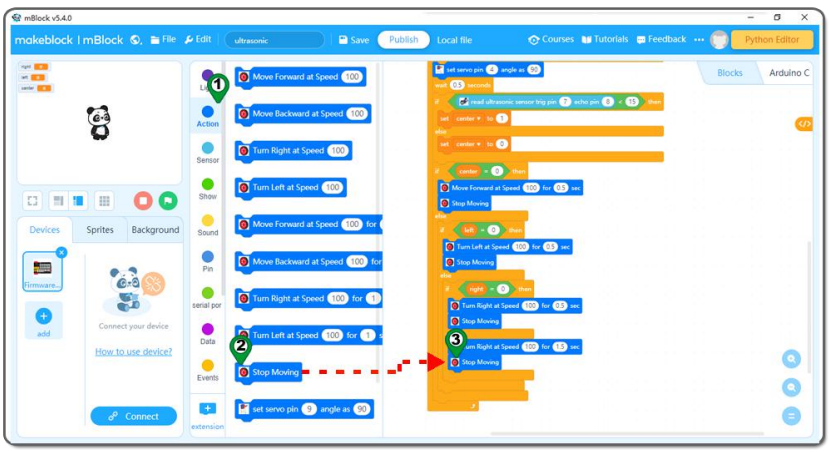


Step 34: Click **Action** again, then Drag and drop **Stop Moving** block to programming area as following;

Step 35: In **Action**, then Drag and drop **Turn Right at speed for** block to programming area,change the **1 sec** to **1.5 sec** as following;



Step 36: In **Action**, then Drag and drop **Stop Moving** block to programming area as following;



Now we have completed the block programming.The final blocks look like following:

Now all the programming blocks have been completed! From above picture, the logic is pretty straight forward:

When Arduino is started, computer will enter a dead loop which will check the button status, when button is not pressed, the servo will stop at original position (in my servo 0 degree) and elevator is in the ground , when button is pressed, the servo arm will rotate from 0 degree position to 180 degree position and the elevator is lifted to the air.

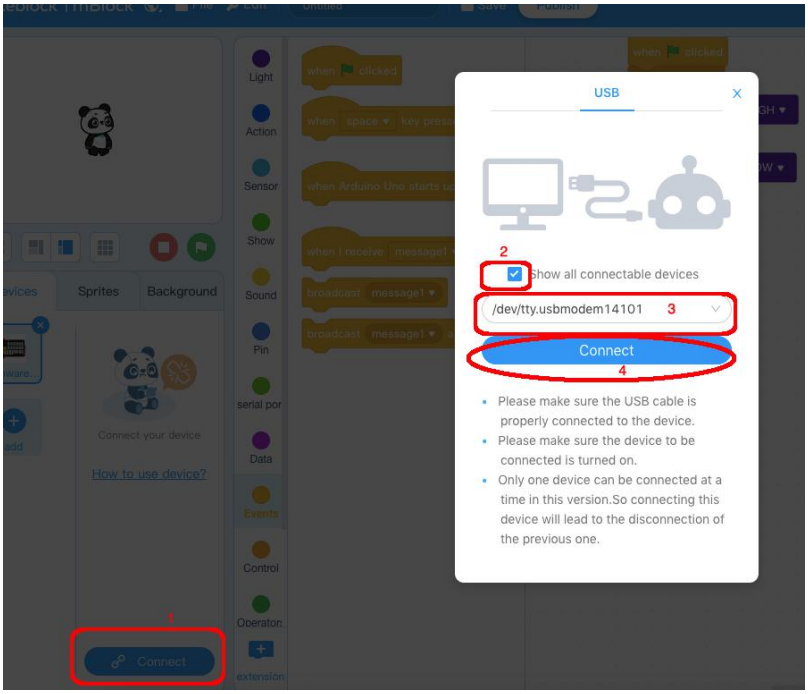
Step 37 Upload the program to OSOYOO basic board

1) Please connect your OSOYOO basic board to your PC with USB cable firstly. Then click the Connect button in the bottom of the mBlock software, you will see a USB window pop up,

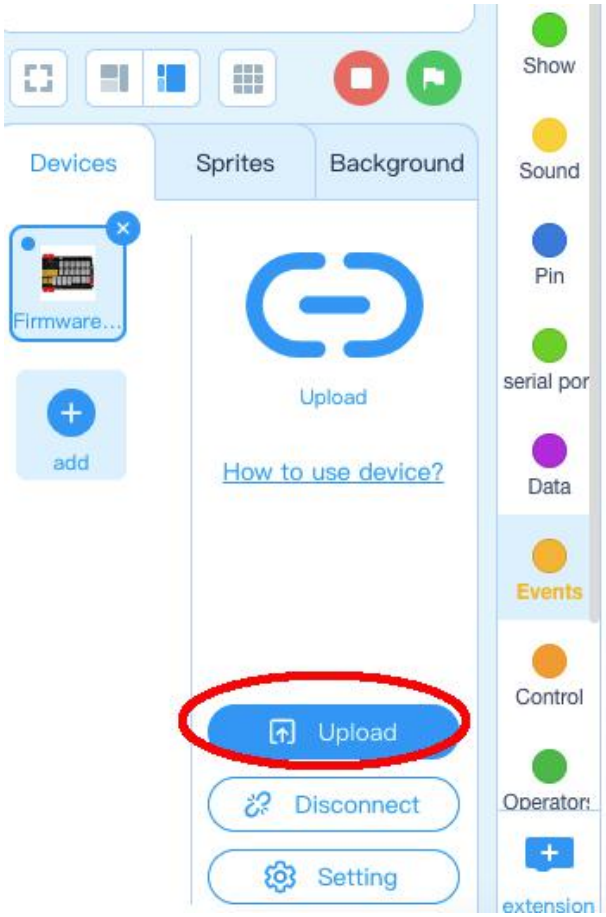
2) select **Show all connectable device** check box , then a device drop-down menu will show up,

3) select your port from device drop-down menu

4) click **Connect** button to connect your PC to OSOYOO basic board.



5)After you PC is connected to OSOYOO basic board, please click Upload button in the bottom of your software, then the code will be uploaded to OSOYOO basic board:



PROGRAM EXPLANATION

In this lesson, we will learn some new program block as following:

i)Set Servo angle block in Action category:

This block will control a servo in specific Digital port to a specific angle.

Take Above block as example, it will rotate servo in D4 pin to the angle of 90 degree which will make the Ultrasonic sensor facing front. In this case, the sensor will detect if the front side has obstacle.

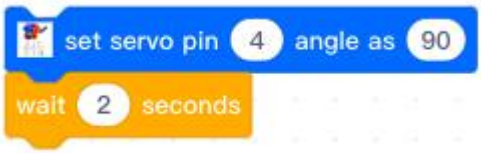
If you change the angle value from 90 to 45, the sensor will rotate 45 degree to the

left and ultrasonic sensor will test left side obstacles.

If you change the angle value from 90 to 135, the sensor will rotate 45 degree to the right and ultrasonic sensor will test right side obstacles.

ii)Yellow Wait Block from Control Category

This block will keep current status for a specific time period before execute next program block



In above example, servo in D4 rotates to 90 degree direction, then stops there for 2 seconds.

iii)Read Ultrasonic Sensor Block from Sensor Section

This block will read distance value from Ultrasonic Sensor

In the above example , the block will read obstacle distance value to an ultrasonic sensor whose trig pin in D7 and echo pin in D8.

If ultrasonic sensor detects obstacle, it will return distance value (integer).

iv)Compare block in Operator category

Compare block will return True or False based on compare value

In the above example, when Ultrasonic Sensor distance value less than the 15（threshold value), it will return True which means obstacle is detected

Otherwise it will return false which means no obstacle in this direction. You can

change the threshold to other value. The bigger the value, the more sensitive your car will act.

The whole programming logic :

First, program will make ultrasonic sensor to rotate from 45 degree to 135 degree and then stop at 90 degree for 2 seconds. This will give you a chance to make servo direction alignment. If sensor doesn't face front during the 2 seconds at 90 degree, you need to adjust the sensor location in servo and make it face to 90 degree.

Then the sensor enter into a forever loop which make the car run in obstacle avoidance mode.

The program rotate the servo to 45 degree, if sensor detect an obstacle within 15cm, it will set a variable **right** to 1, this means right side has obstacle. Otherwise variable **right** will be set to 0.

Then program rotates the servo to 135 degree and 90 degree to test left side obstacle and center direction obstacle. If left side has obstacle within 15 cm, variable **left** will sent to 1 , If center direction has obstacle within 15 cm, variable **center** will sent to 1. If no obstacle, then these vaiables are set to 0.

Now we have known the obstacle status in three variables **left**, **right** and **center**. The program will decide car movement based on these values.

If center direction has NO obstacle (**center**=0), then car move forward.

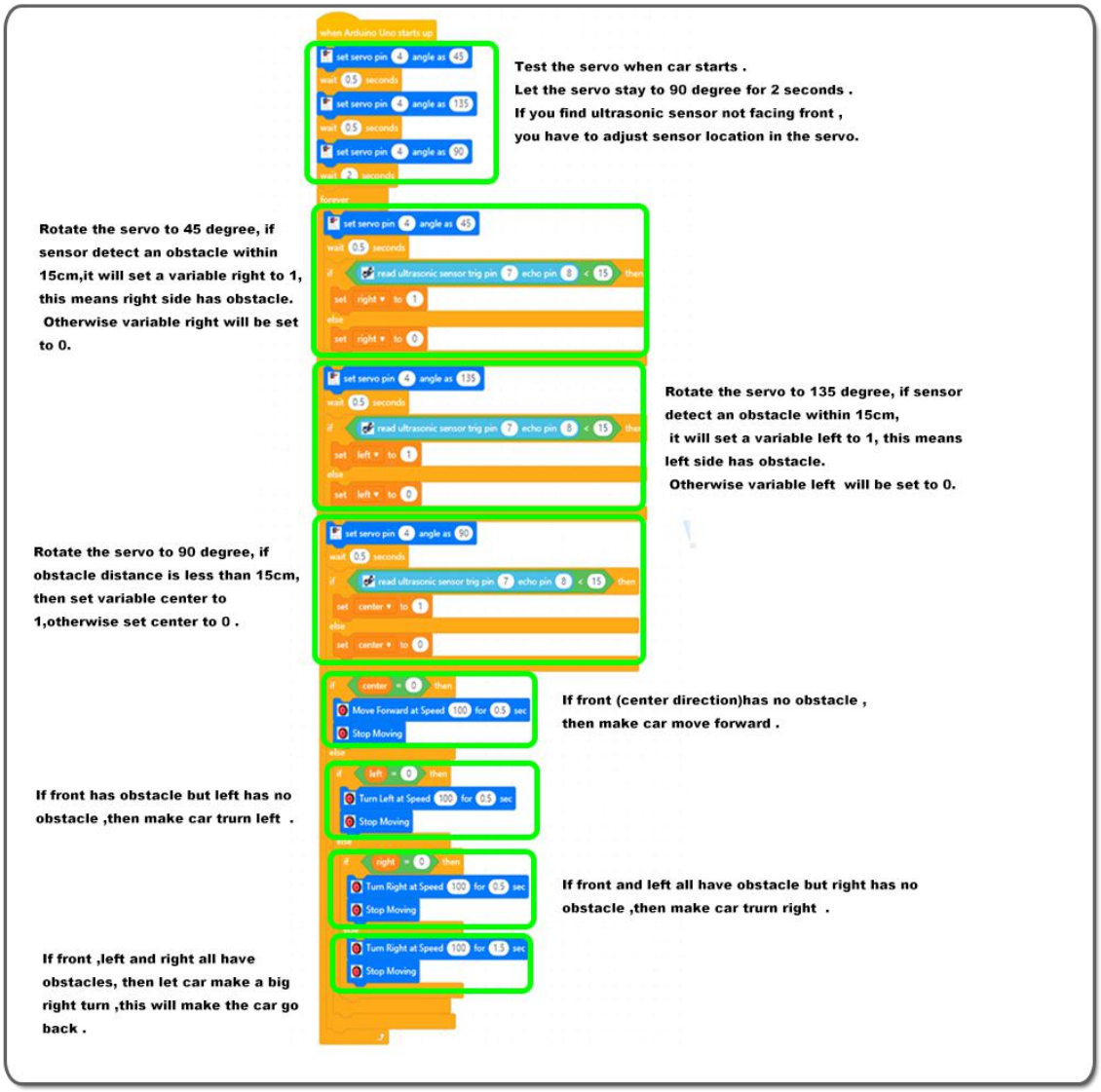
If center direction has obstacle but left side has NO obstacle (**center**=1 but **left**=0), then car turn left.

If center direction, left side all have obstacle but right side has NO obstacle

(**center**=1 and **left**=1 but **right**=0), then car turn right.

If center direction, left and right side all have obstacle (**center**=1 and **left**=1 **right**=1), then car turn right for long time, this will make the car turn back.

Following picture shows above programming logic in the program blocks.

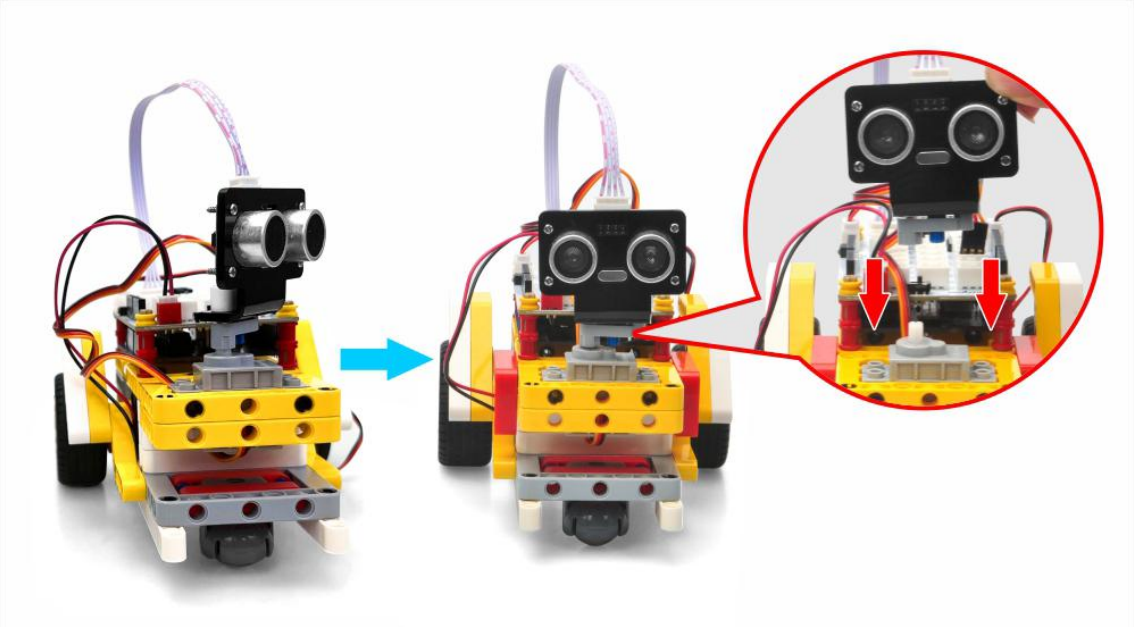


HOW TO PLAY

Adjust the Ultrasonic Sensor direction:

After the code is loaded into robot car, unplug the USB cable from the UNO board

and turn on the power switch on the Magic I/O board. Then the ultrasonic sensor will rotate to face the straight forward direction for 5 seconds. If the Ultrasonic sensor is not facing the front position, please turn off the power. Please use a screw driver to adjust the ultrasonic sensor direction and make it facing straight ahead , then turn on the power again and during the early 5 seconds, the sensor should be facing front direction.



After the 5 seconds , the robot car will start to drive and change direction to avoid obstacles automatically.

Lesson 5: Auto transporter

OBJECTIVE

In this project, we will make a smart robot finger and use it to transfer a pingpong ball to another place. We will use obstacle sensor(ultrasonic distance detector) to detect the coming ball. When a ball is detected, the finger will pickup the ball and the car will drop it to another place, then the car will go back to original place to wait for another ball.

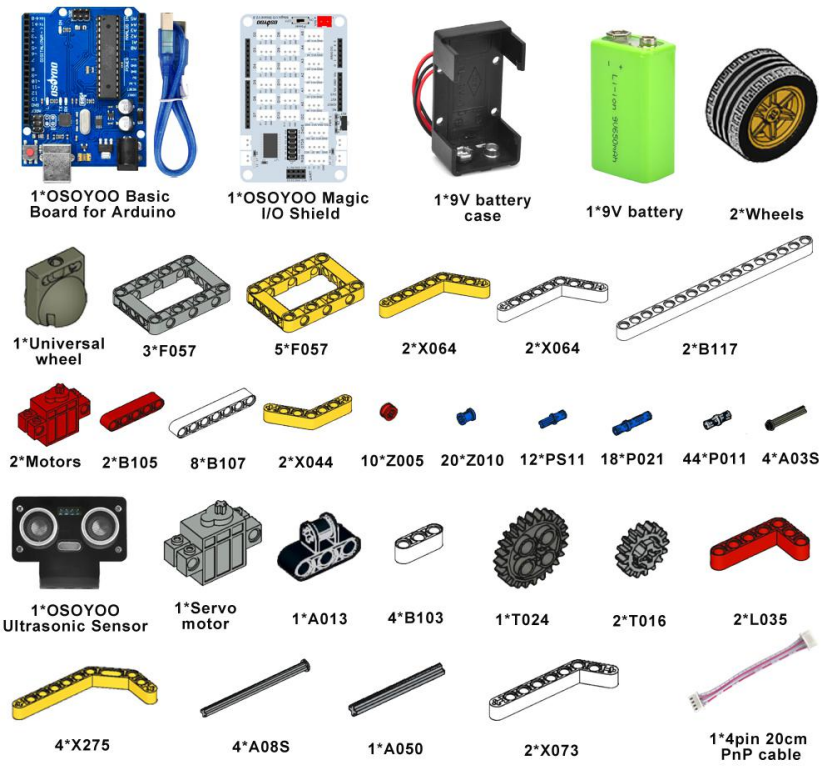
PARTS & DEVICES

Please prepare the following parts to complete this project

NOTE:

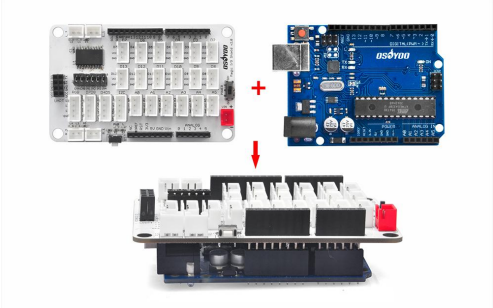
1.the color of the building block is subject to the actual product, which does not affect the use.

2. ALL OSOYOO PRODUCTS FOR ARDUINO ARE THIRD PARTY BOARD WHICH IS FULLY COMPATIBLE WITH ARDUINO



HOW TO MAKE

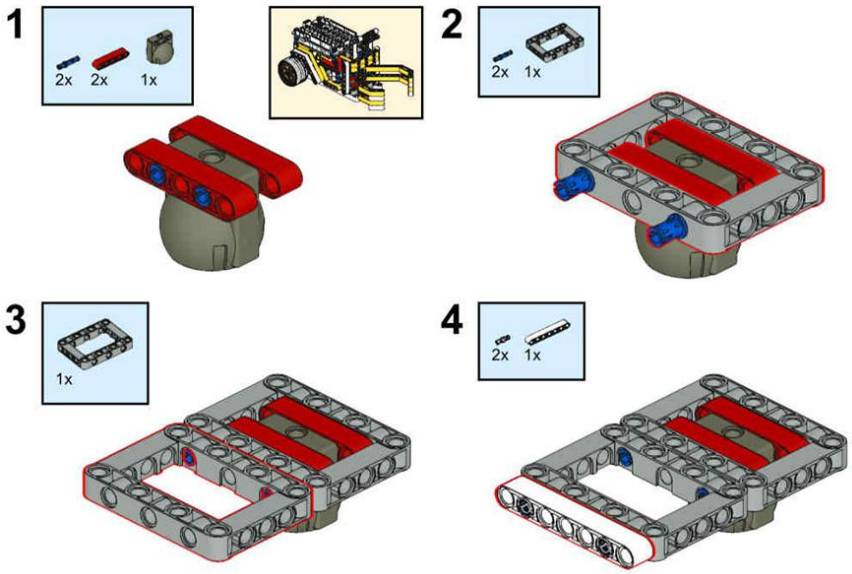
Before you build the robot with blocks, please install OSOYOO basic board for Arduino under OSOYOO Magic I/O shield as following (Attention please : the pins of I/O shield is aligned with the port of basic the board firstly, then press the shield tightly on the board).

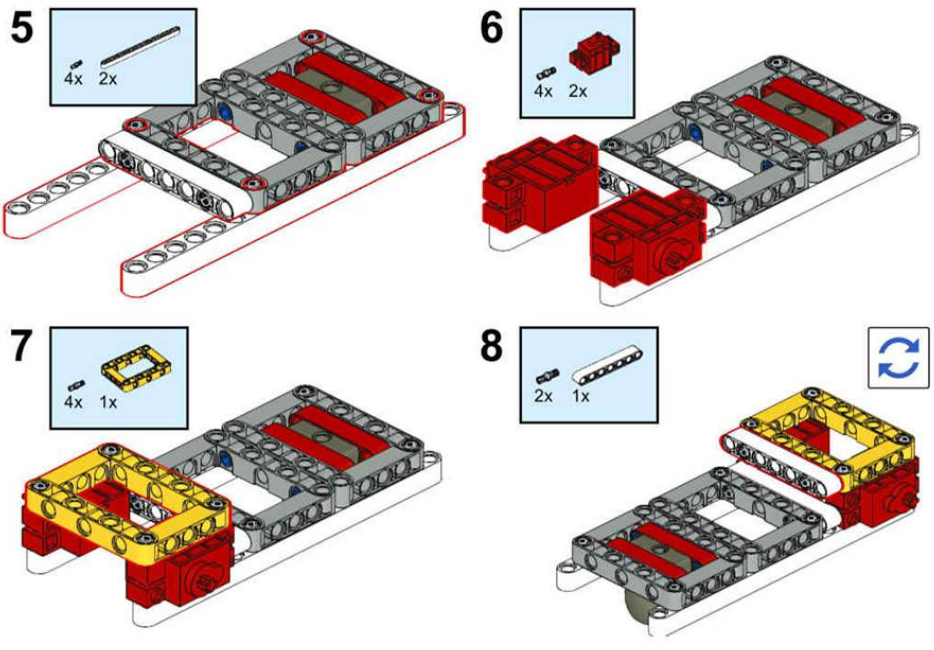


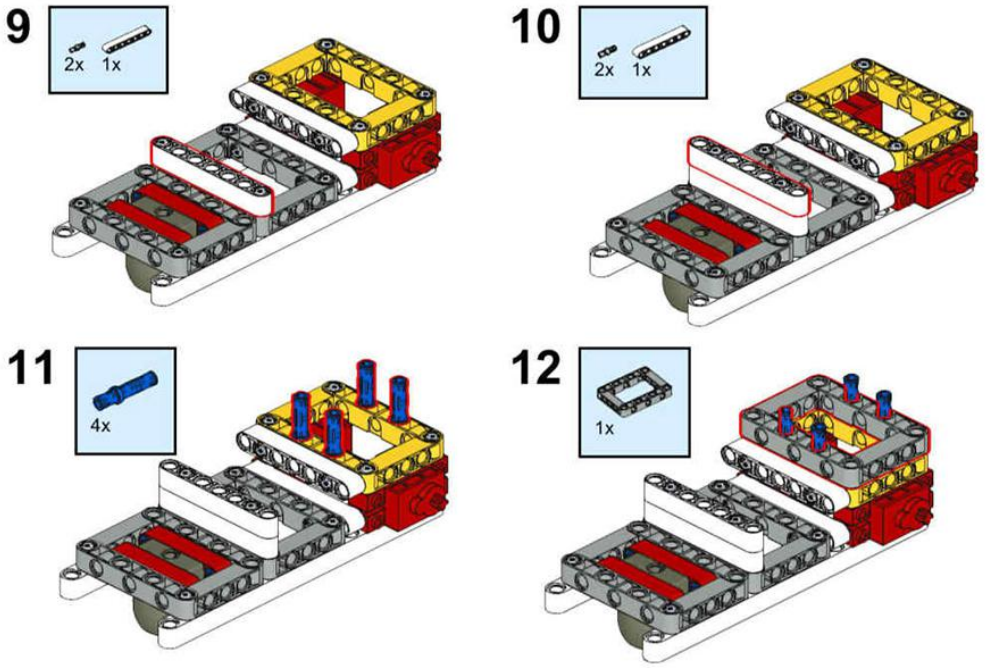
Please follow the building steps to build this robot car, If you want to get clear PDF building steps, please download

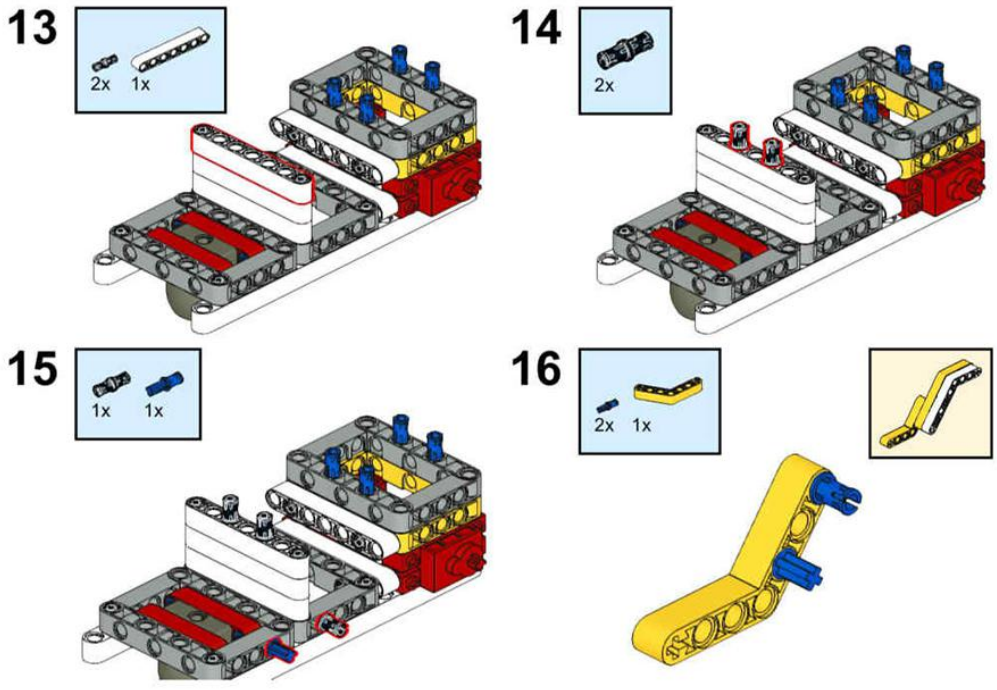
from https://osoyoo.com/picture/Building_Robot_Car/lesson5/LESSON5.pdf

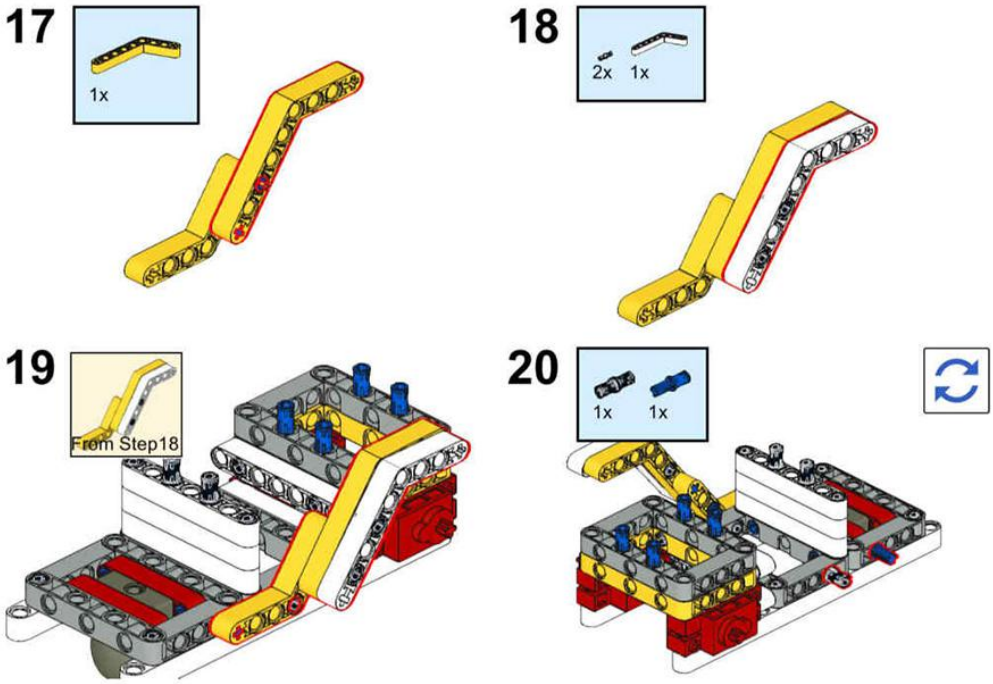
Note: If you have built the robot car for lesson1, please skip to the step35 in this PDF.

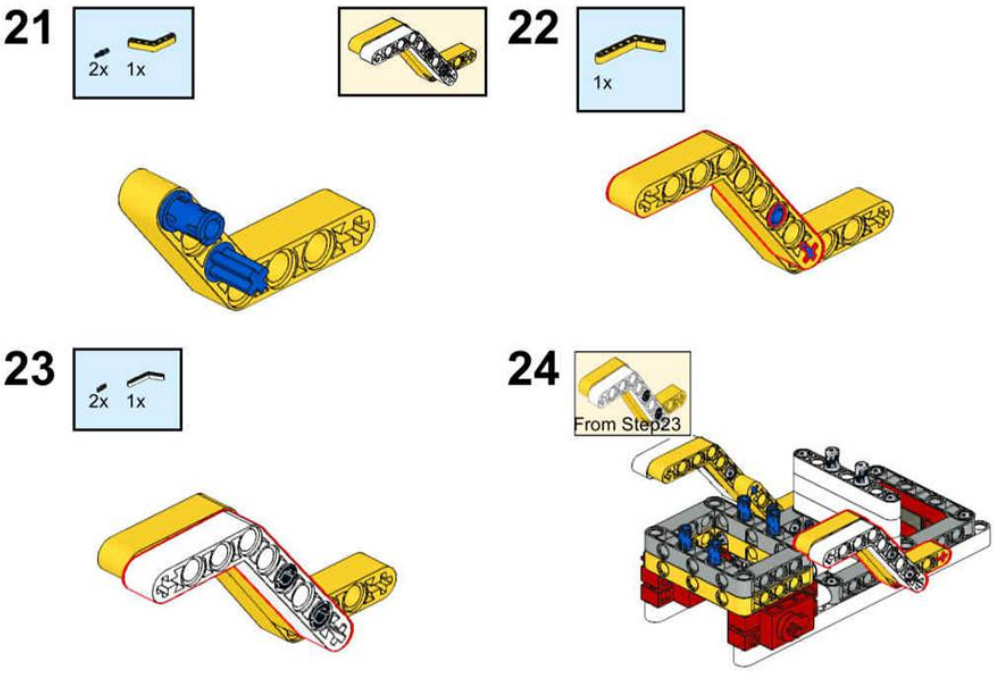


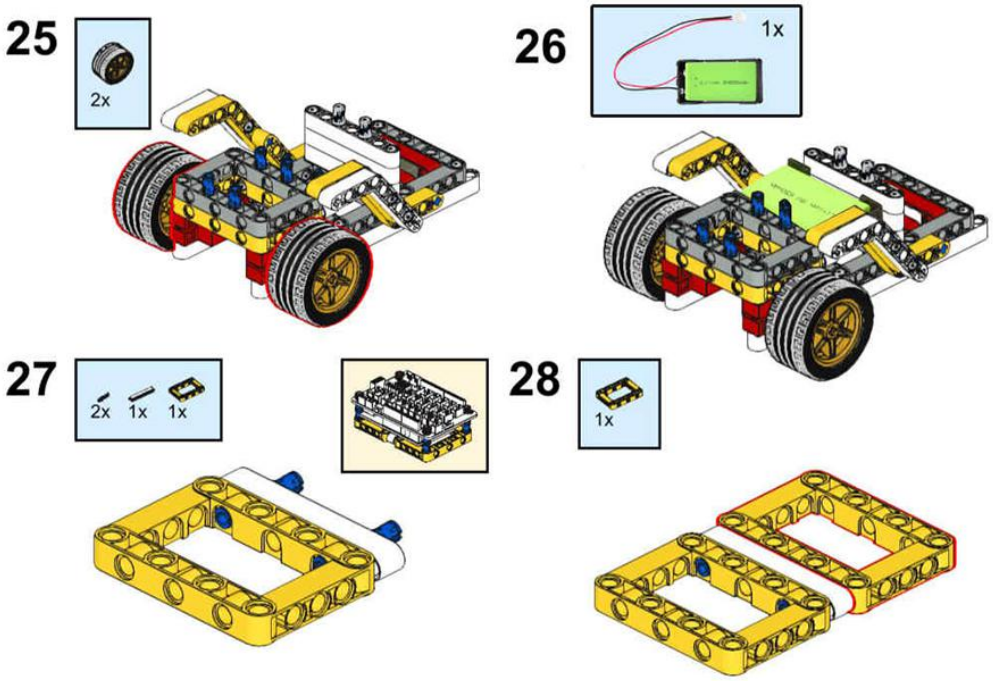


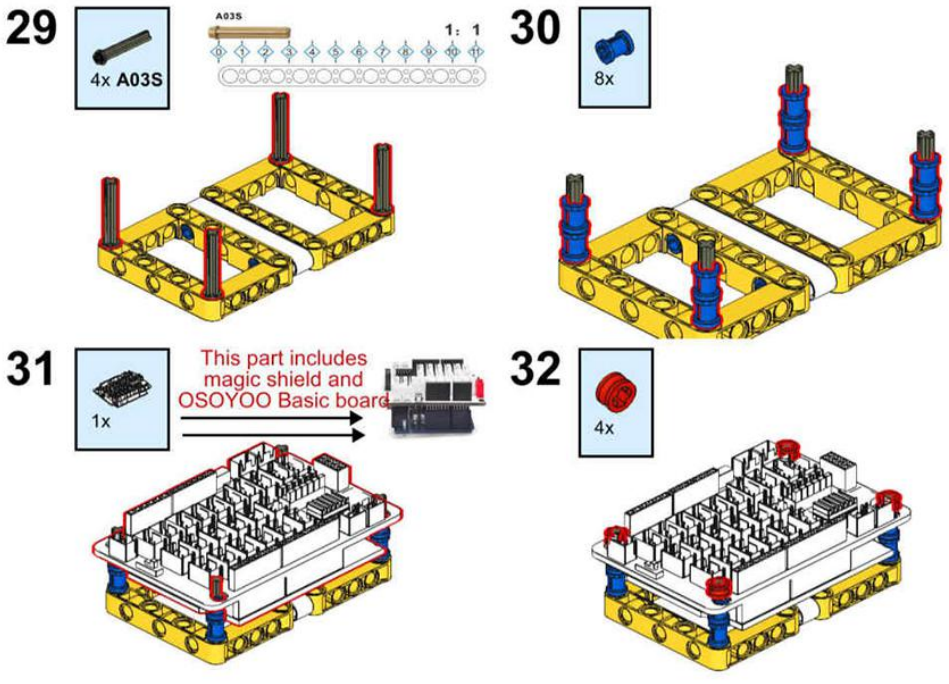

9
2x 1x
10
2x 1x
11
4x
12
1x

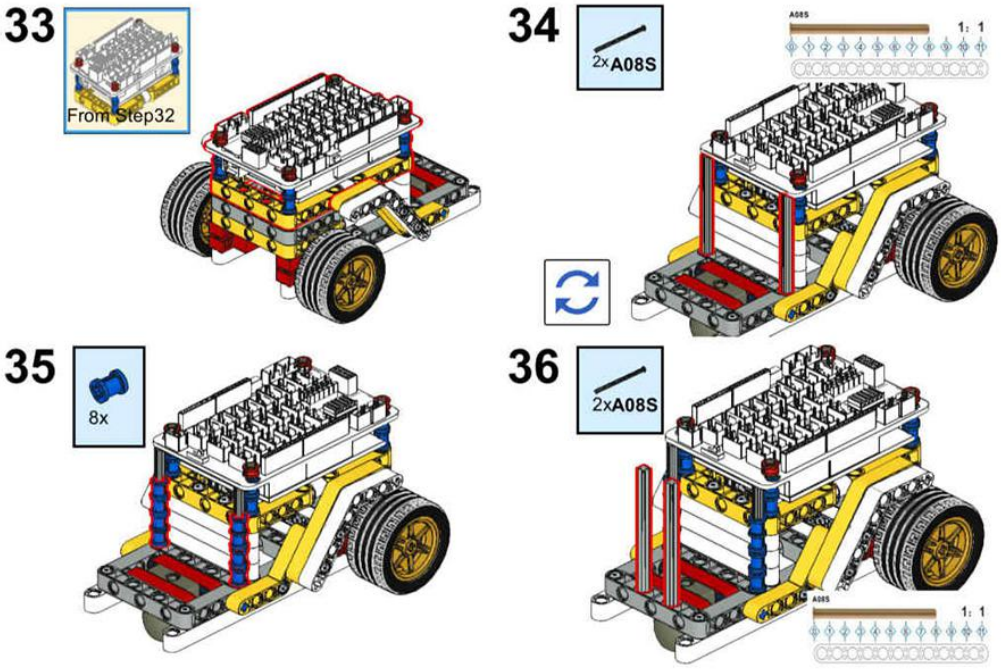

13
2x 1x
14
2x
15
1x 1x
16
2x 1x

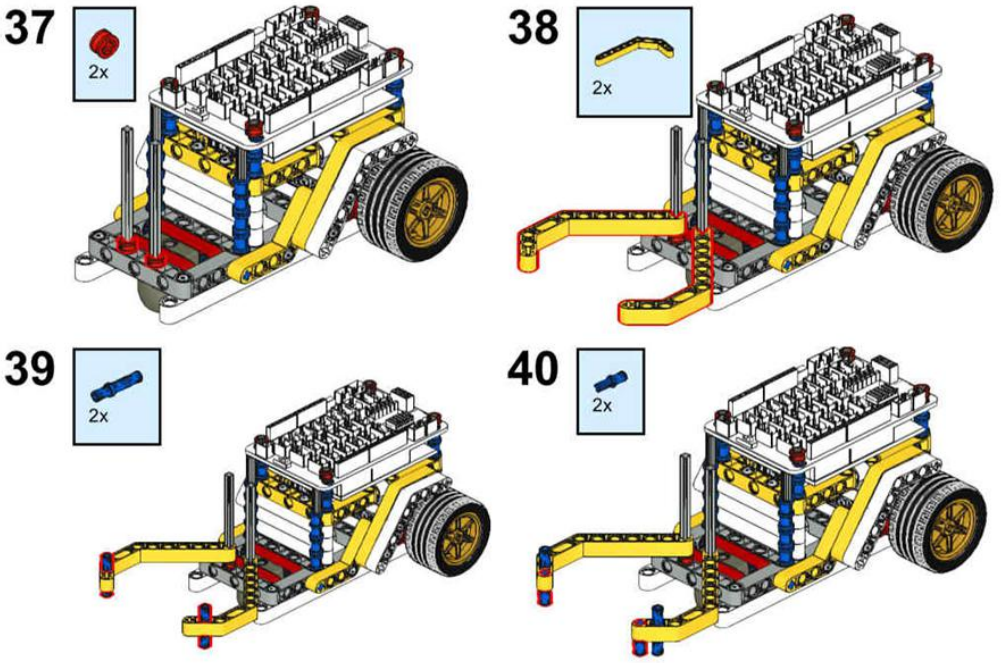

17
1x
18
2x 1x
19
From Step18
20
1x 1x

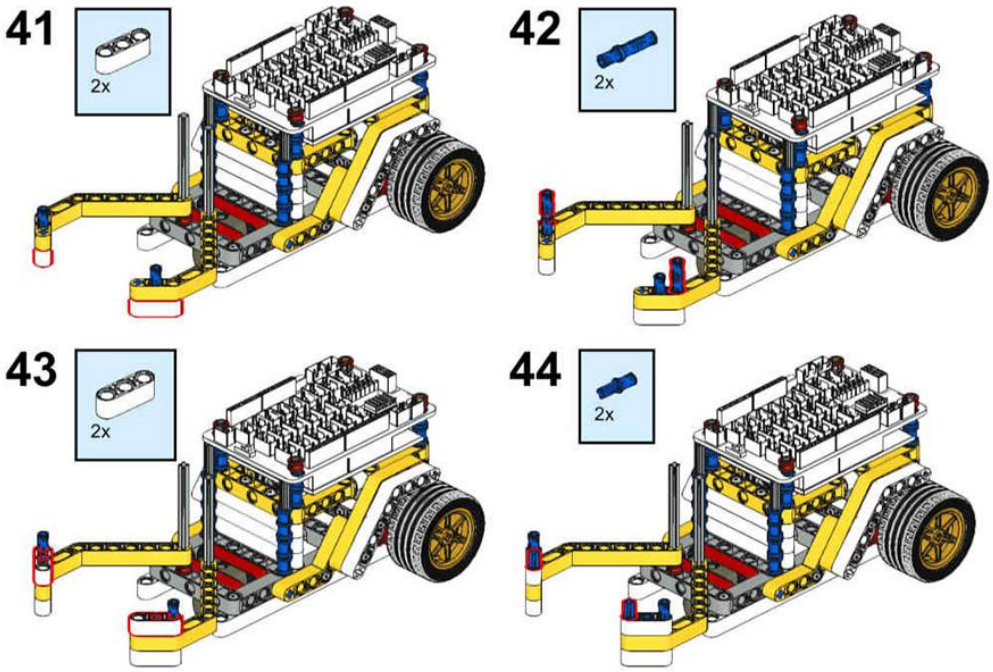

21
2x 1x
22
1x
23
2x 1x
24
From Step23

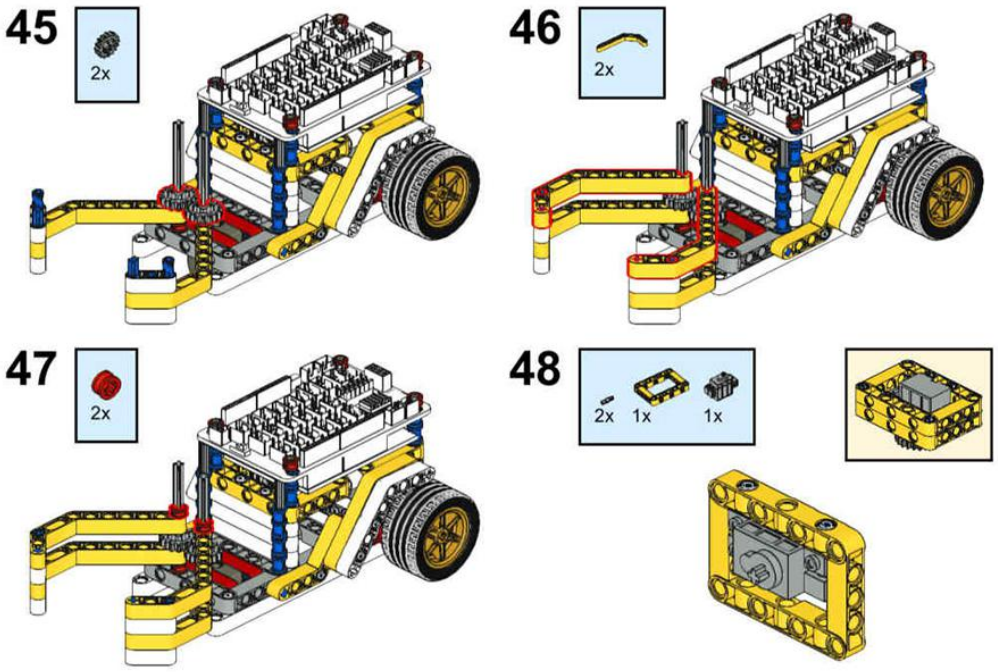

25
2x
26
1x
27
2x 1x 1x
28
1x

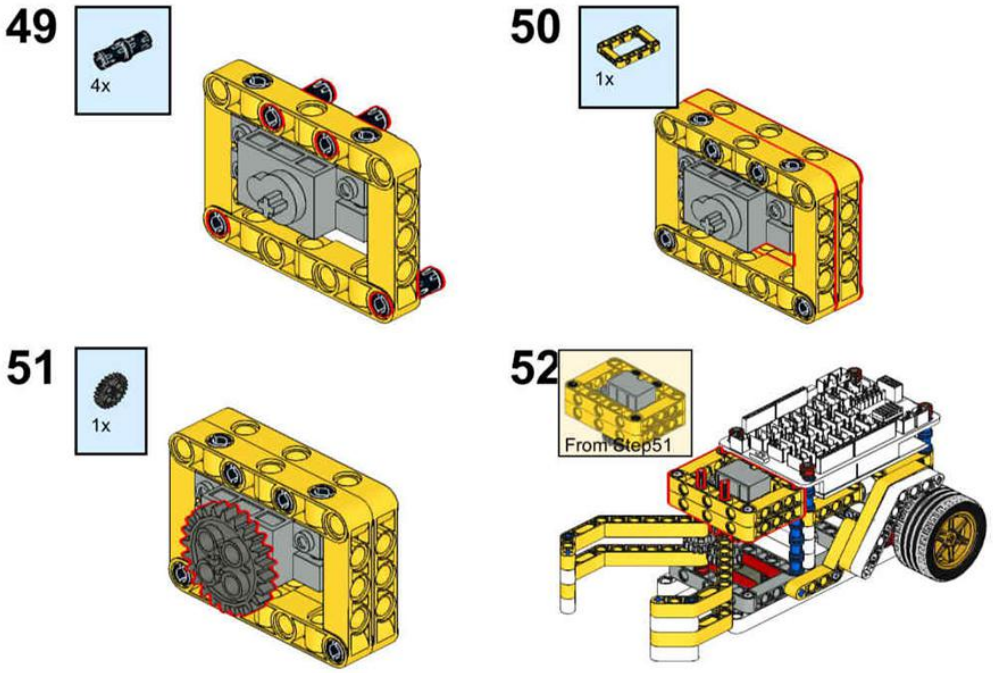

29
4x A03S
A03S
1 : 1
30
8x
31
1x
This part includes magic shield and OSOYOO Basic board
32
4x

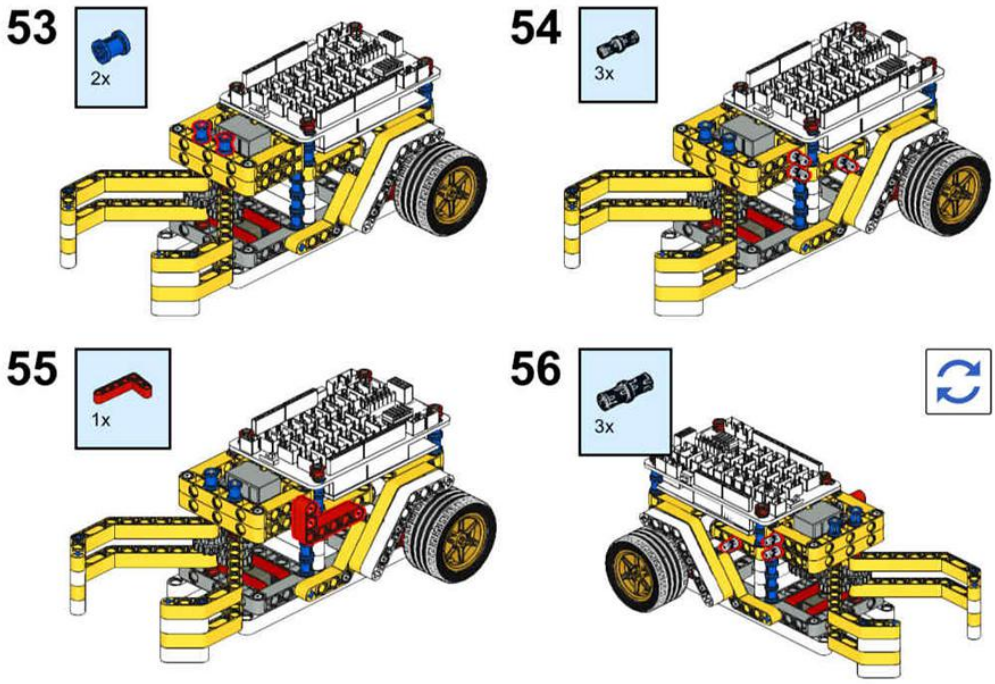

33
From Step32
34
2x A08S
A08S
1: 1
35
8x
36
2xA08S
A08S
1: 1

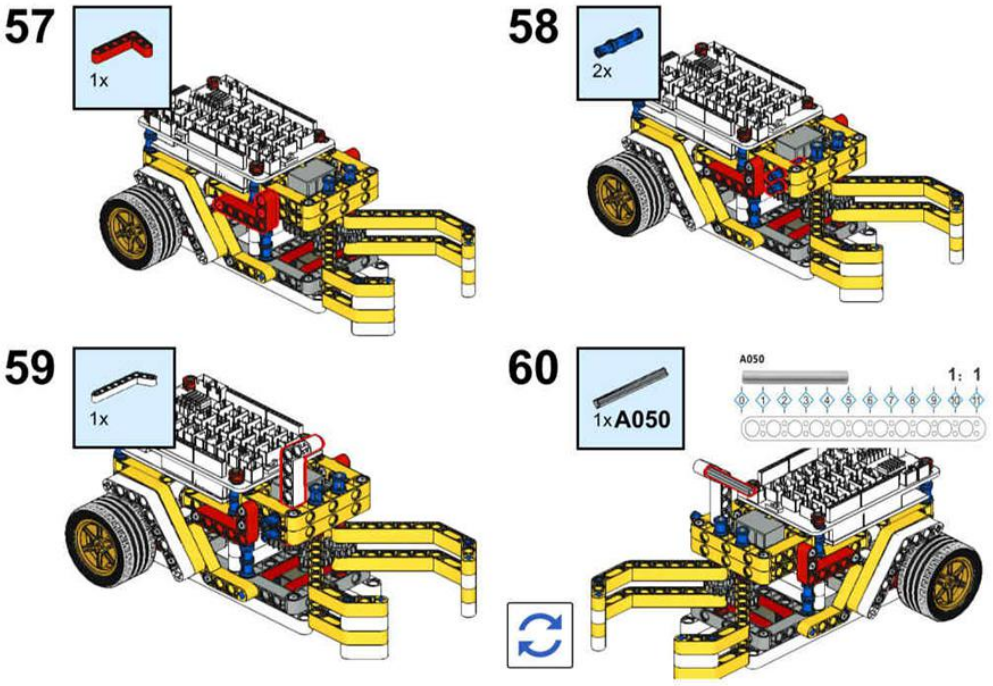

37
2x
38
2x
39
2x
40
2x

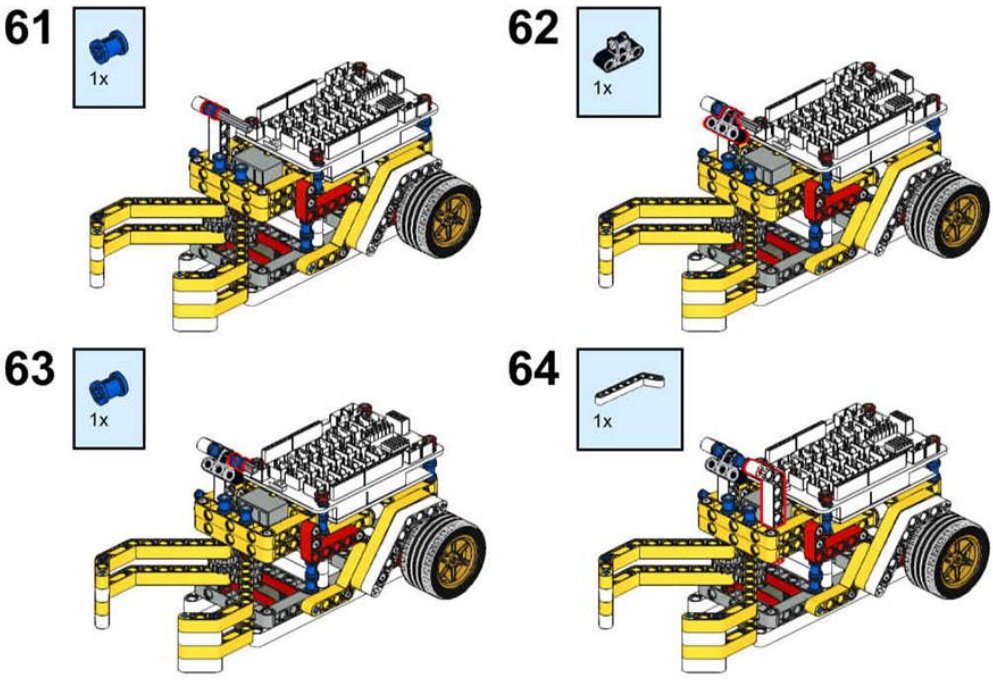

41
2x
42
2x
43
2x
44
2x


45
2x
46
2x
47
2x
48
2x
1x
1x

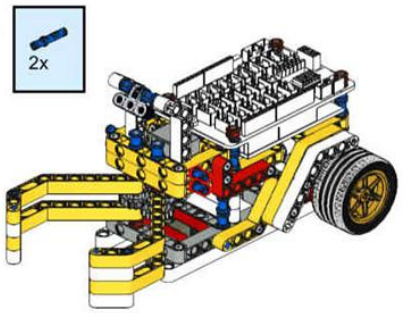

49
4x
50
1x
51
1x
52
From Step51


53
2x
54
3x
55
1x
56
3x

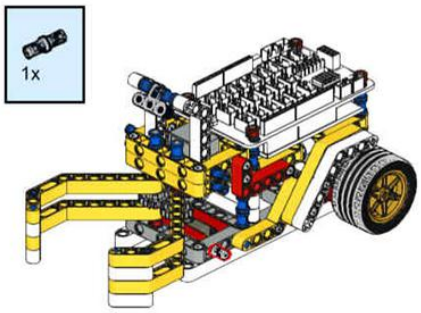

57
1x
58
2x
59
1x
60
1x A050
A050
1 : 1


61
1x
62
1x
63
1x
64
1x

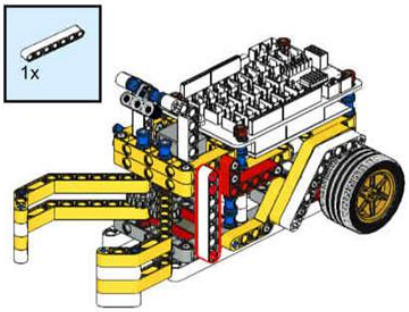
65



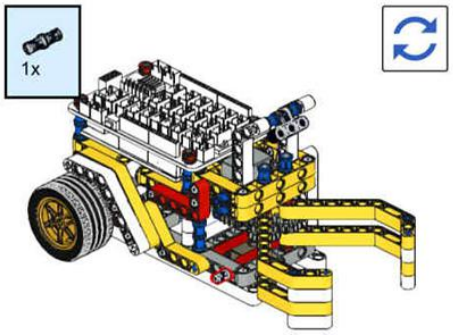
66



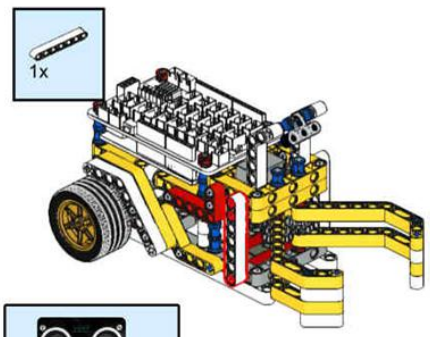
67



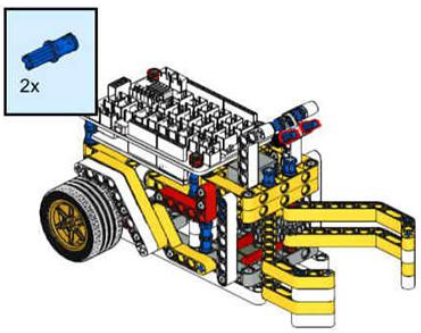
68



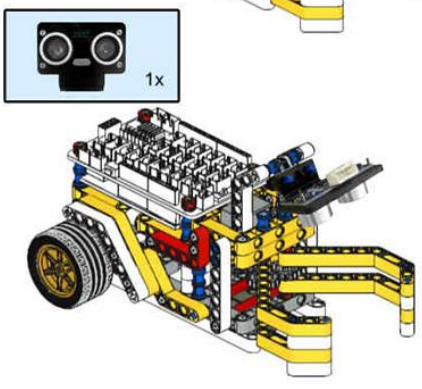
69



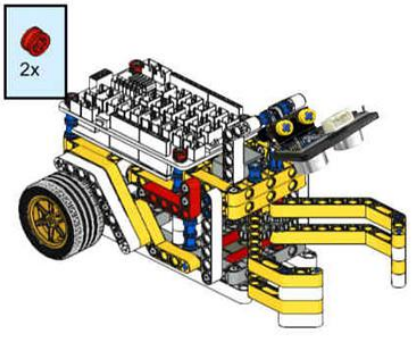
70

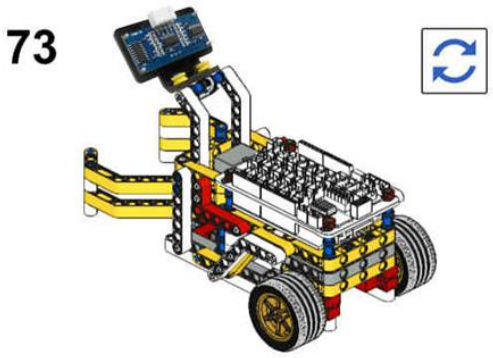


71



72

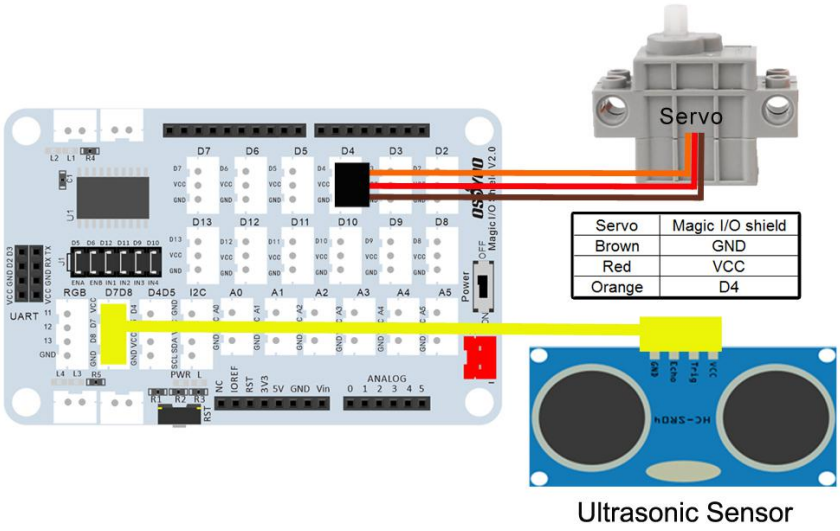




CIRCUIT CONNECTION

Please connect motors and 9V battery case as lesson1.

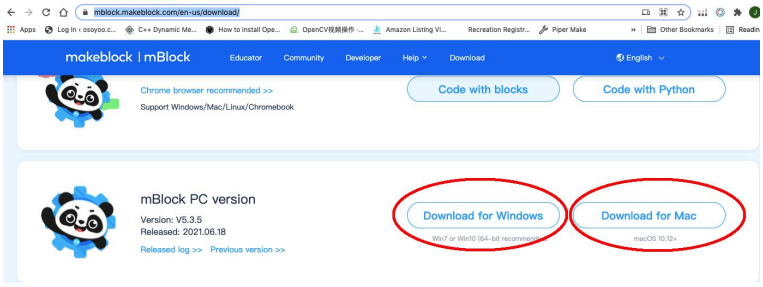
Then Connect SG90 servo motor to D4 port of OSOYOO Magic I/O shield, Ultrasonic module sensor to the D7D8 port of the Magic I/O shield with a 4-pin PNP cable as below:



HOW TO CODE

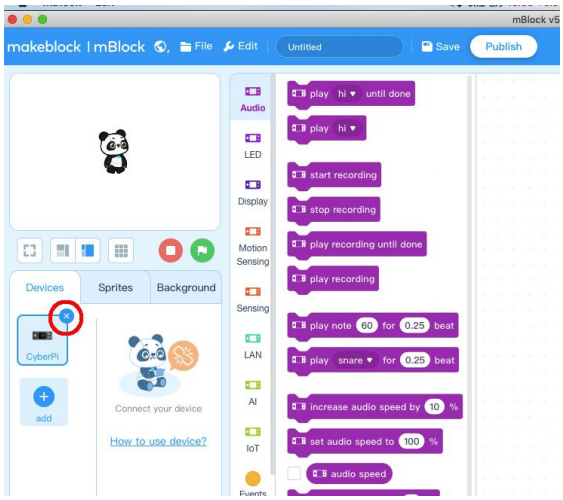
Note: In this kit, we use mblock as programming tool, if you want to learn more about mblock, please visit preparation before class 1: https://osoyoo.com/2021/12/01/preparation-before-class-1

Step 1) Download **mBlock PC version** from https://mblock.makeblock.com/en-us/download/, select the download file as per your computer OS type (Please don't use the browser version):

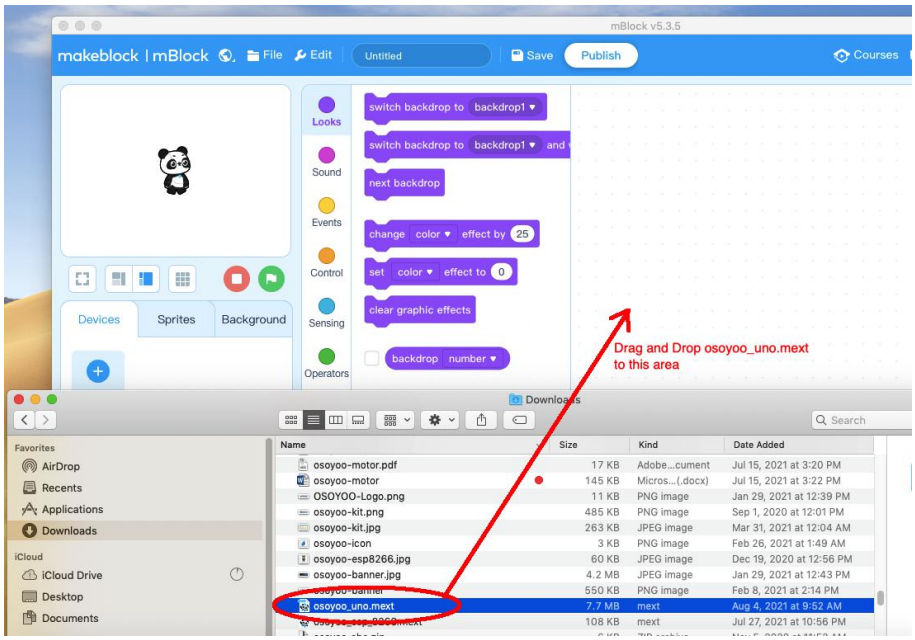


Step 2) Download OSOYOO_UNO.mext device file from https://osoyoo.com/driver/mblock/osoyoo_uno.mext

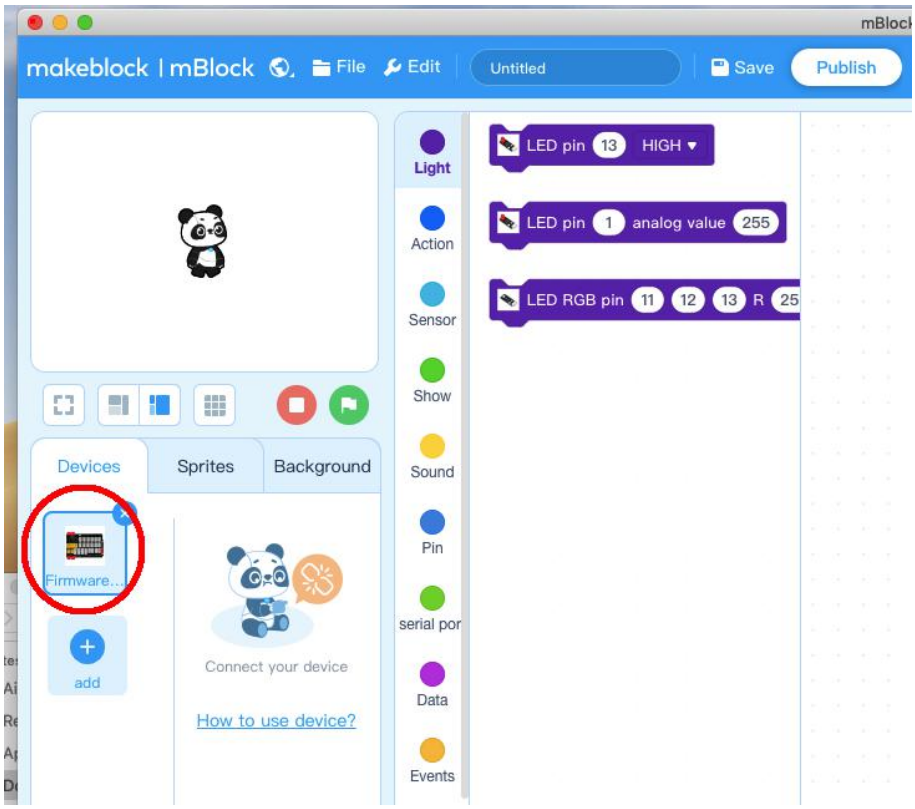
Step 3) Run the mBlock PC software by double click the lovely Panda icon. you will see mBlock UI as following picture. Please delete the default device **CyberPi** by click the cross in the red circle.



Step 4) Drag and Drop osoyoo_uno_mext file(downloaded in Step 2) to mBlock software as following:



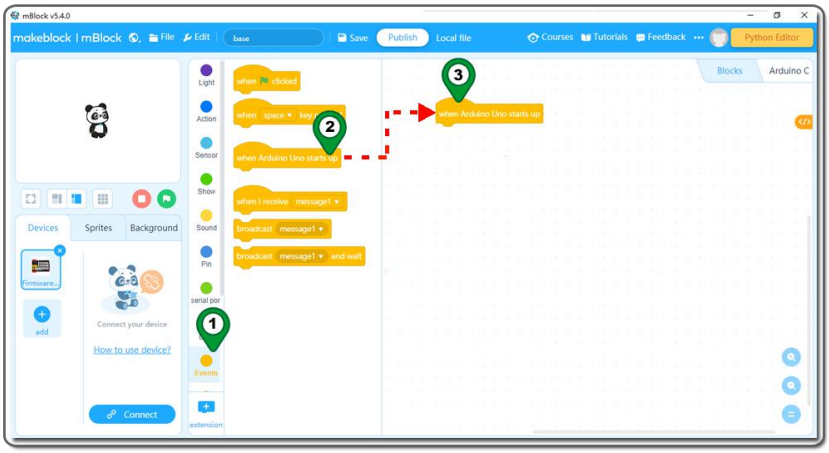
Now you will see a new device firmware in mBlock, see following picture:



Now mBlock software and OSOYOO_UNO device firmware have been successfully installed in our PC!

Now we will show you how to use blocks to turn above idea into reality.

Step 5: Click **Events**, add **when Arduino Uno starts up** block to the top:

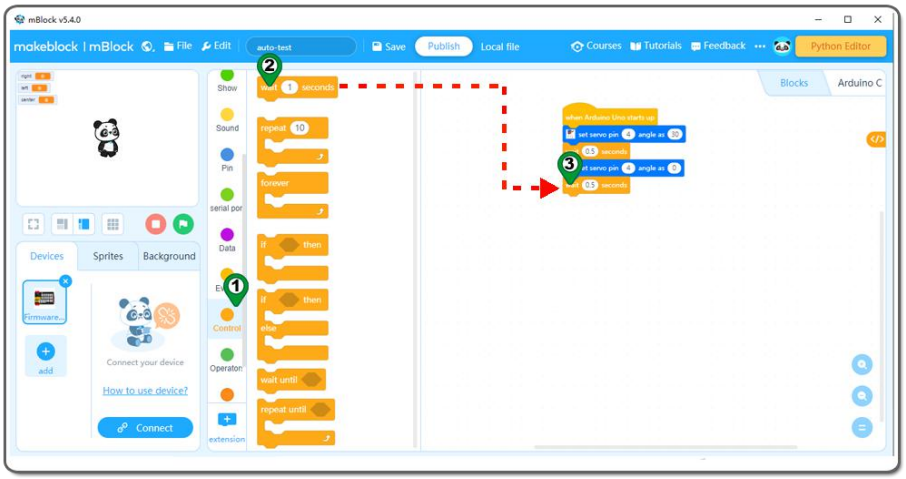


Step 6: Click **Action**, then Drag and drop **set servo pin angle as** block to programming area ,change the pin 9 to 4 ,90 to 30 as following;

Step 7: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed 0.5 seconds as following:

Step 8: Click **Action** again, then Drag and drop **set servo pin angle as** block to programming area ,30 to 0 as following;

Step 9: Repeat the Step7 as following:



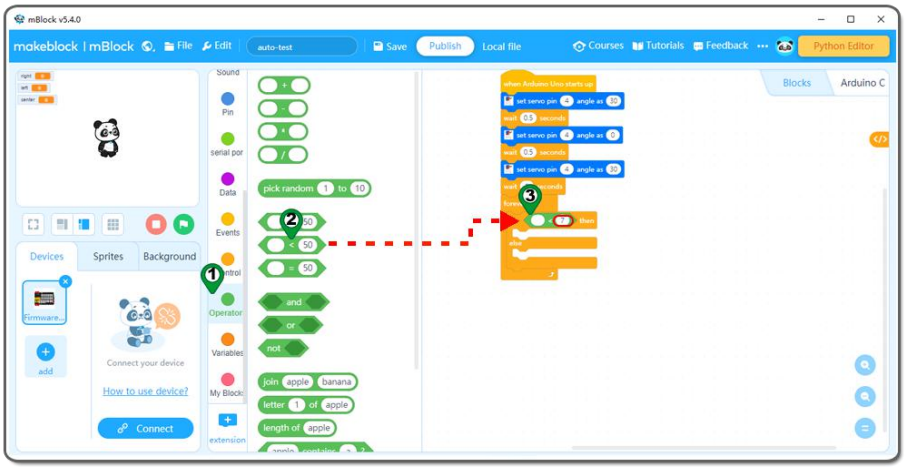
Step 10: Click **Action**, then Drag and drop **set servo pin angle as** block to programming area ,**0** to **30** as following;

Step 11: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed **1** seconds as following:

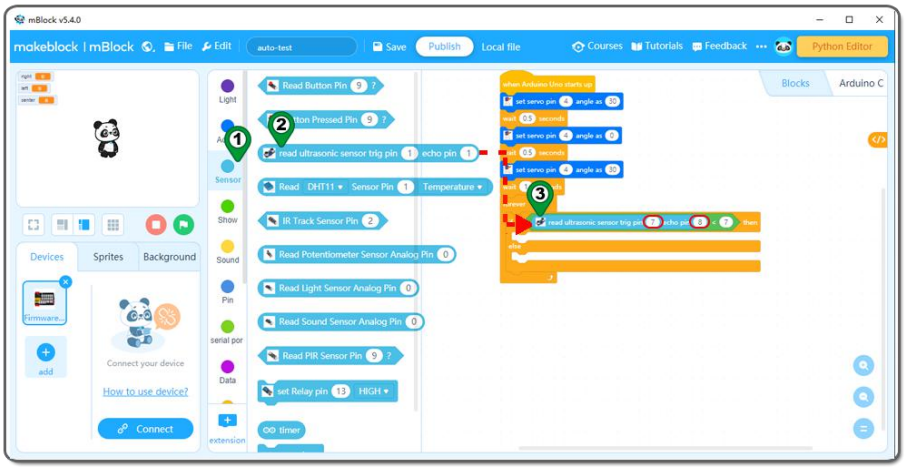
Step 12: Click **Control**, then Drag and drop **forever** block to programming area as following:

Step 13: Click **Control** again,then Drag and drop **if then else** block to programming area as following:

Step 14: Click **Operation**, then Drag and drop **0<0** block to programming area ,change the 0 to 7 as following:



Step 15: Click **Sensor**, then Drag and drop **read ultrasonic sensor trig pin echo pin** block to programming area ,writed pin 7 and pin 8 as following;

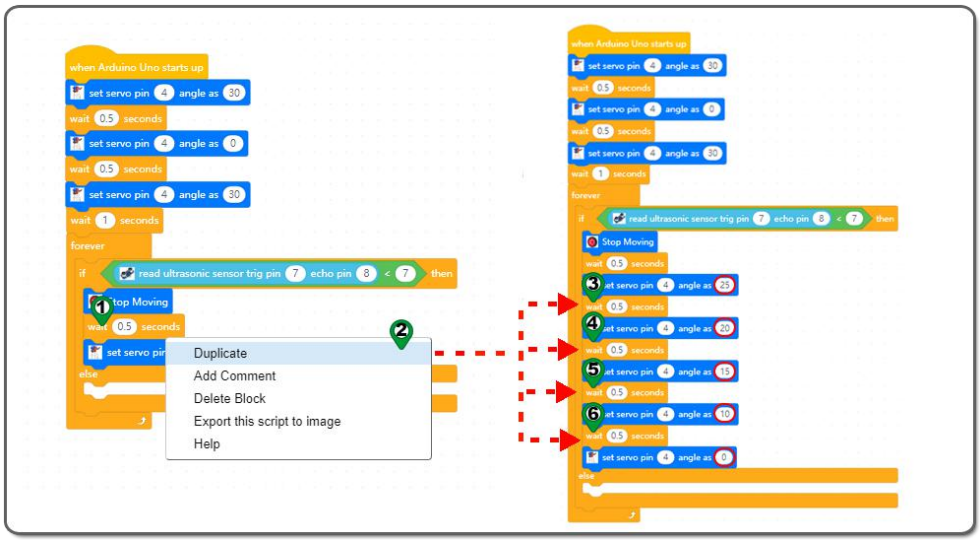


Step 16: Click **Action** , then Drag and drop **Stop Mowing** block to programming area as following;

Step 17: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed **0.5** seconds as following:

Step 18: Click **Action**, then Drag and drop **set servo pin angle as** block to programming area ,**0** to **25** as following;

Step 19: Right click **Wait seconds** block ,Click duplicate from the digital write pin to block over . You will get a duplicated block ,and place it behind **Set servo pin angle as** block .Change the angle value as following figure.

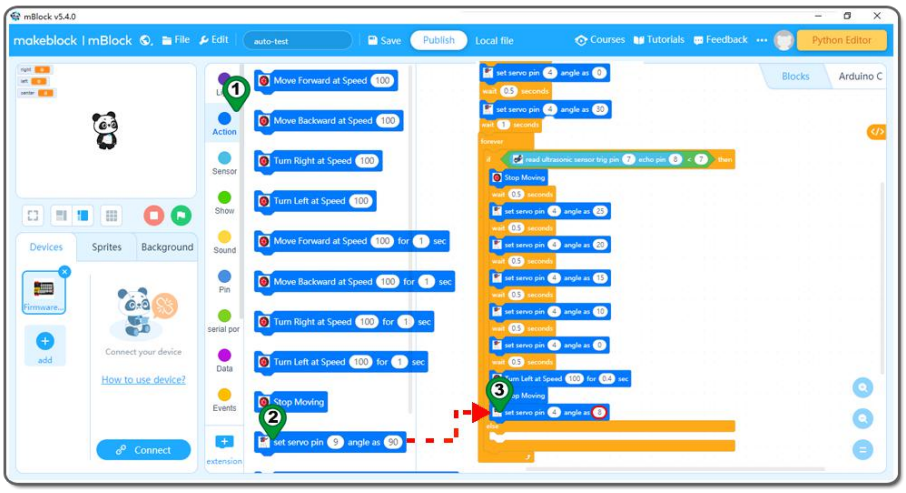


Step 20: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed **0.5** seconds as following:

Step 21: Click **Action**,then Drag and drop **Turn Left Speed for** block to programming area ,change the 1 sec to **0.4 sec** as following;

Step 22: Click **Action** , then Drag and drop **Stop Mowing** block to programming area as following;

Step 23: Click **Action** again, then Drag and drop **set servo pin angle as** block to programming area ,**90** to **0** as following;

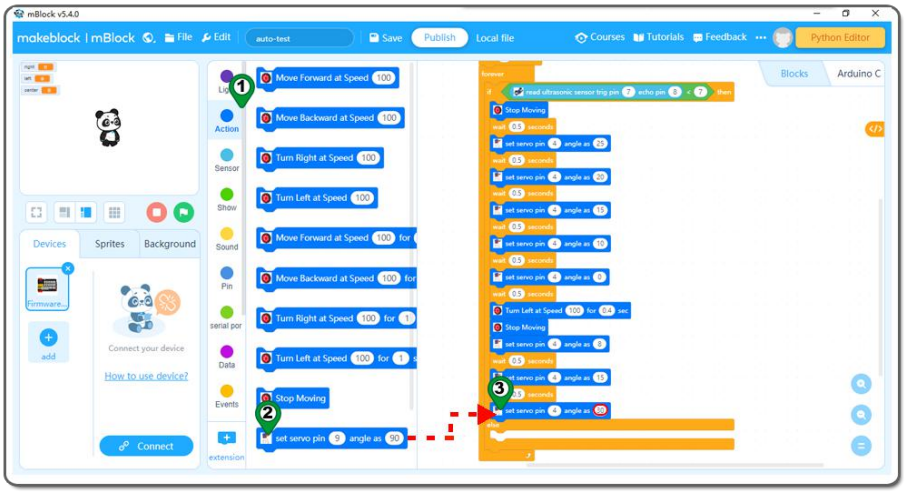


Step 24: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed **0.5** seconds as following:

Step 25: Repeat the **Step 23**,change the **90** to 15 as following:

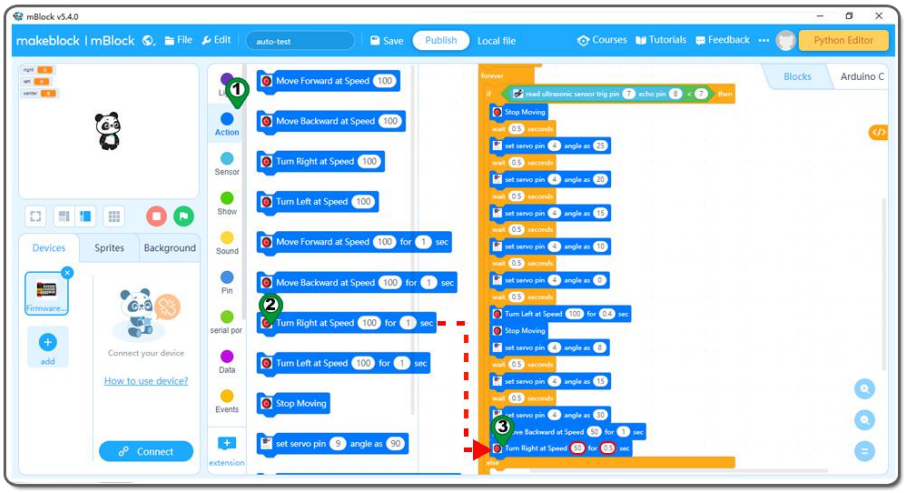
Step 26: Click **Control**, then Drag and drop **wait seconds** block to programming area ,writed **0.5** seconds as following:

Step 27: Click **Action** again, then Drag and drop **set servo pin angle as** block to programming area ,**90** to **30** as following;

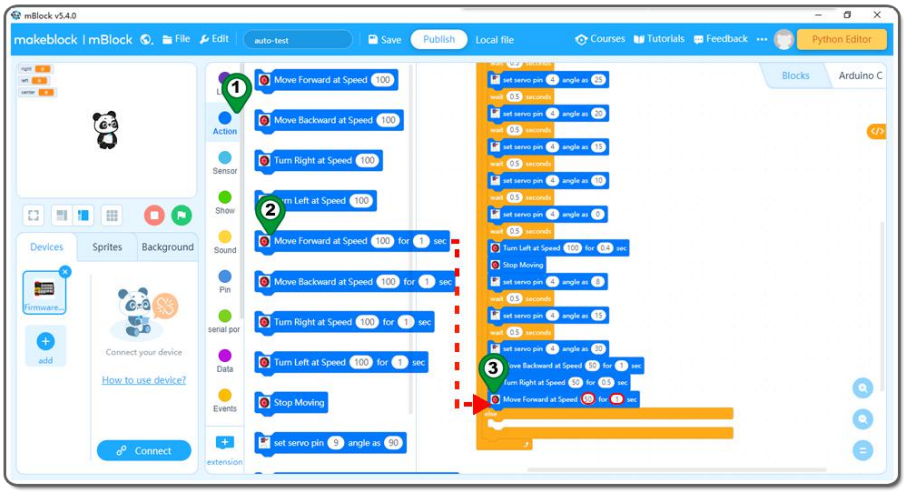


Step 28: Click **Action** , then Drag and drop **Move Backward at speed for sec** block to programming area ,change the speed **100** to **50** as following;

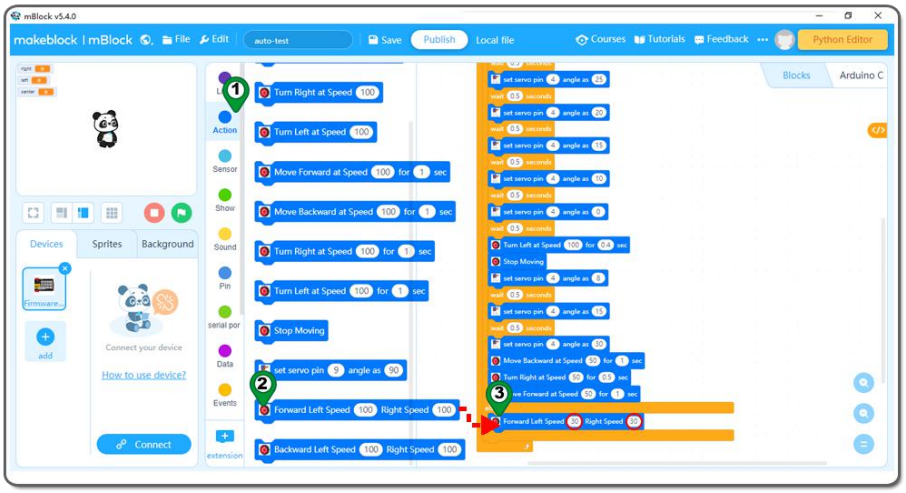
Step 29: Also in **Action** ,then Drag and drop **Turn Right at speed for sec** block to programming area ,change the speed **100** to **50,**the **1 sec** to **0.5 sec** as following;



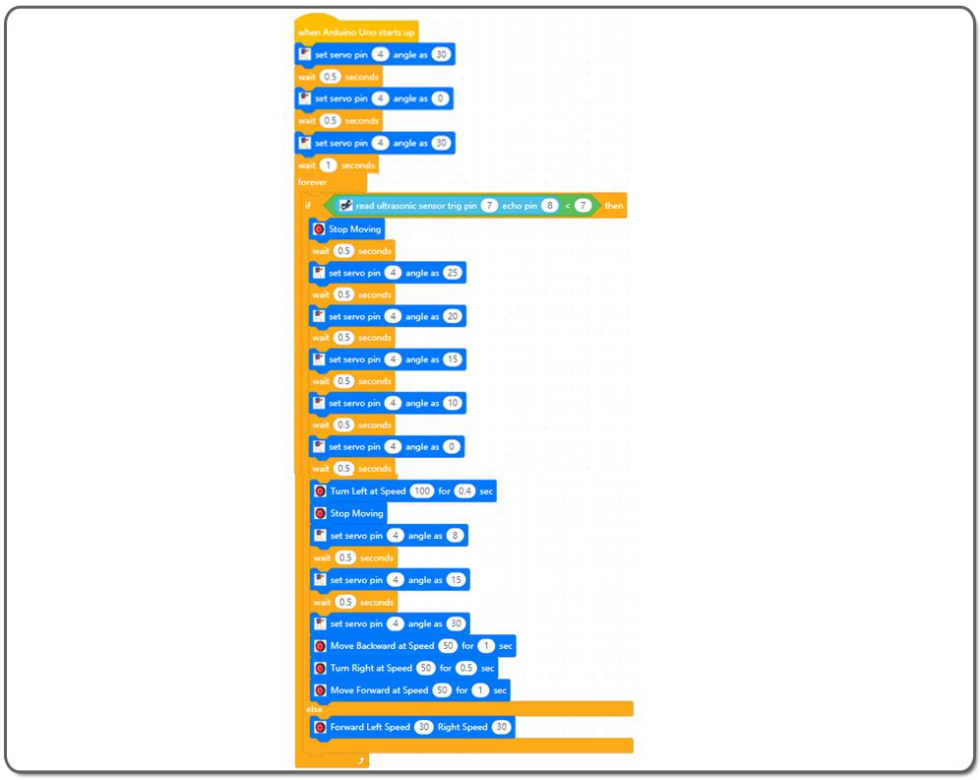
Step 30: Drag and drop **Move Forward at speed for sec** block to programming area ,change the speed **100** to **50** as following;



Step 31: Drag and drop **Forward Left speed Right speed** block to programming area ,change the speed **100** to **30** as following;



Now we have completed the block programming. The final blocks look like following:



Now all the programming blocks have been completed! From above picture, the logic is pretty straight forward:

When Arduino is started, computer will enter a dead loop which will check the button status, when button is not pressed, the servo will stop at original position (in my servo 0 degree) and elevator is in the ground , when button is pressed, the servo arm will

rotate from 0 degree position to 180 degree position and the elevator is lifted to the air.

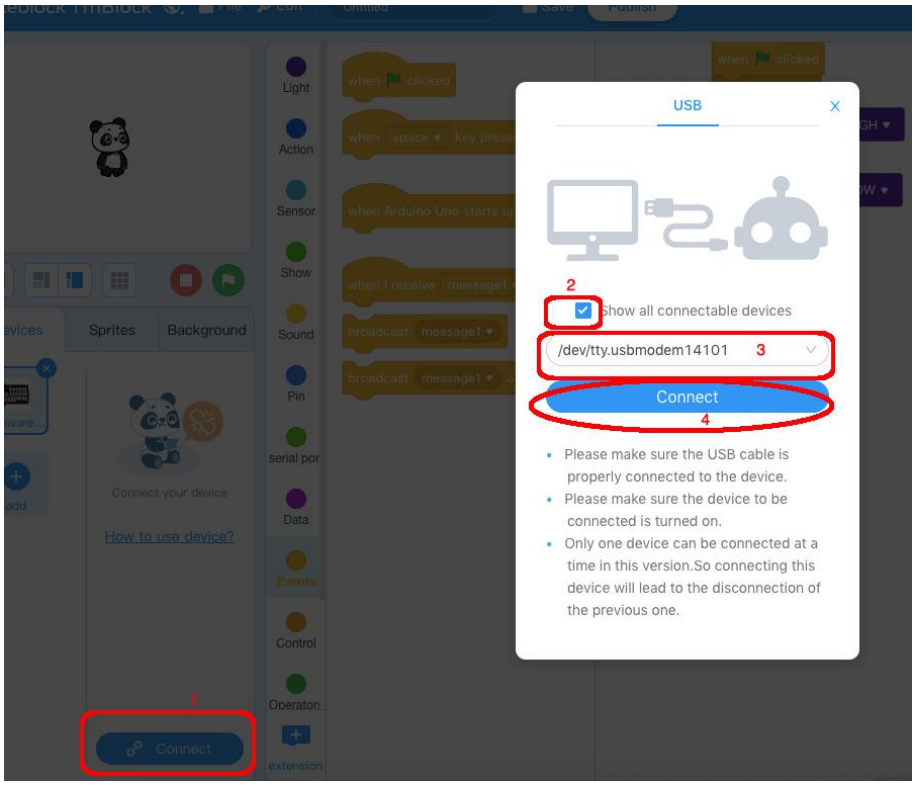
Step 32 Upload the program to OSOYOO basic board

1) Please connect your OSOYOO basic board to your PC with USB cable firstly. Then click the Connect button in the bottom of the mBlock software, you will see a USB window pop up,

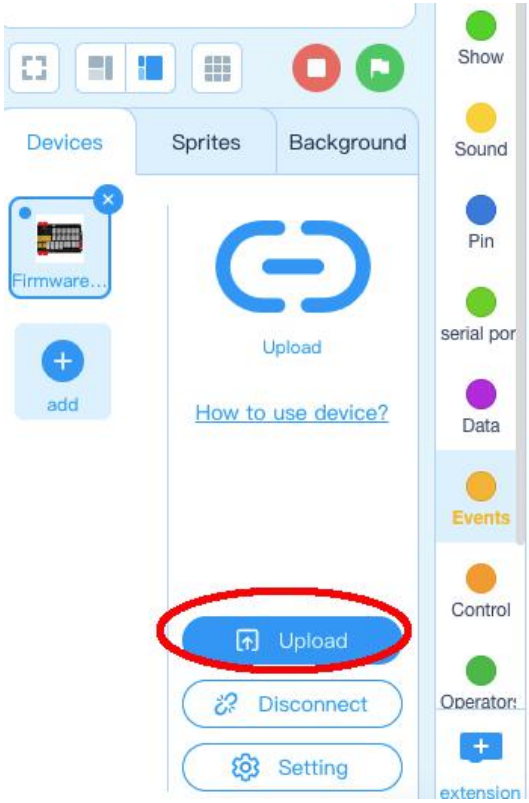
2) select **Show all connectable device** check box , then a device drop-down menu will show up,

3) select your port from device drop-down menu

4) click **Connect** button to connect your PC to OSOYOO basic board.



5)After you PC is connected to OSOYOO basic board, please click Upload button in the bottom of your software, then the code will be uploaded to OSOYOO basic board:



PROGRAM EXPLANATION

This lesson does not use any new program blocks. If you don't understand the block meaning, please review previous lessons.

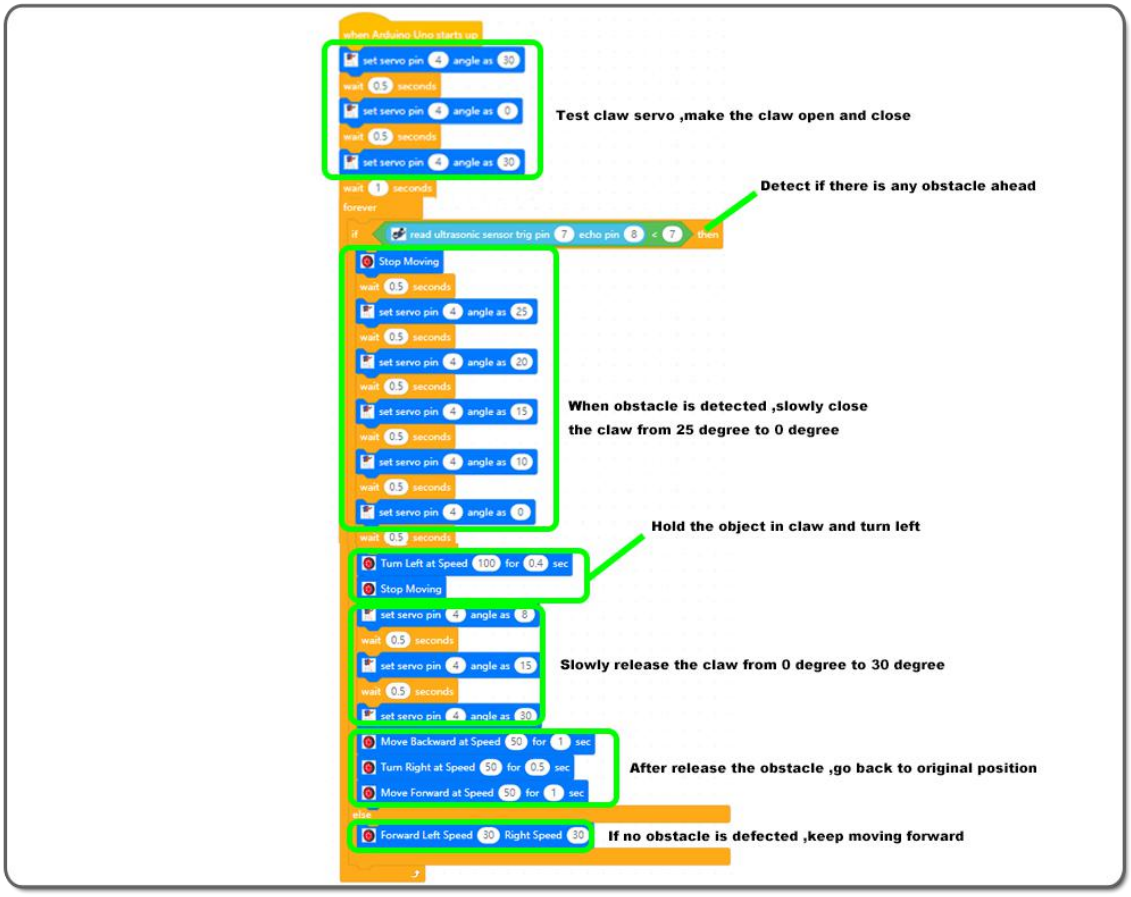
The programming logic explanation is in following graph:

First, program will make servo to rotate from 30 degree to 0 degree and then stop at 30 degree. This will make the finger from opening status to closing status. then opening status. If the finger doesn't work as this, you need to adjust the finger firstly. Then ultrasonic sensor enter into a forever loop which check whether there is obstacle or not on the car's way

If ultrasonic sensor detect an obstacle within 7cm, the car will stop moving for 0.5 second. The servo will rotate slowly from opened status to closed status to make the finger catch the obstacle, then the robot car turn left at speed 100 for 0.4 second, then stop. The servo will rotate slowly from closed status to opened status to make the finger put down the obstacle, then the robot car move backward at speed 50 for 1 second, then turn right at speed 50 for 0.5 second, then move forward at speed 50 for 1 second.

If ultrasonic sensor doesn't detect an obstacle, the car will move forward at speed 30.

Above program blocks are running as a forever loop unless you turn off the power.



HOW TO PLAY

Adjust the finger position:

By manual:

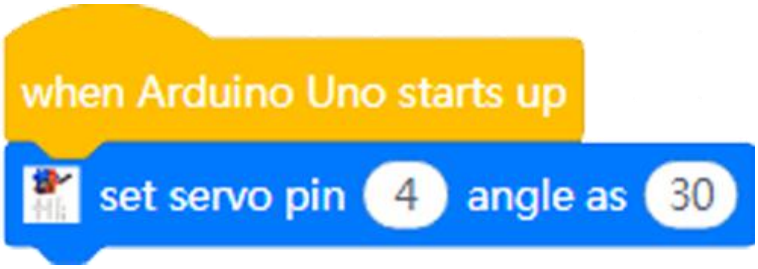
After turning on the power, the finger will open, then close and then open again and stay in open position for about 1 second. If the finger does not stop at open position, please turn off the power, then take out the servo gear and put it to a position where both fingers are in open position.

After you have adjusted finger to right position, please turn on the power again. Then put a ball into the open finger. Once the ultrasonic sensor detected the ball, it will catch the ball and move it to left hand location. After transferring the ball, then the car will turn back to its original position.

By Code:

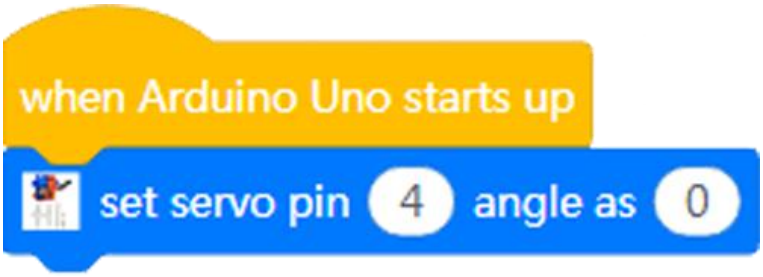
Please upload the following code to your Arduino. Turning on the power and check whether the finger is in opened position or not.

If not, please adjust “30” to other degree and repeat the step till you get the correct value.



Please upload the following code to your Arduino. Turning on the power and check whether the finger is in closed position or not.

If not, please adjust “0” to other degree and repeat the step till you get the correct value.



Please record degrees for your robot finger, and replace the degrees of the sample code with your correct value. so that the code will work perfect.

Working Result:

When you turn on the power and put the car on the ground, the car will go forward, when there is an obstacle(please select the suitable obstacle according the finger size) on its way, the finger will catch it, and put it away) then move on forward.

Lesson 6: WIFI APP control Robot Car and Front Claw

OBJECTIVE

This lesson will use the programming tool —— IDE for Arduino to complete the robot car control by WIFI APP. In this lesson, the car is controlled by phone APP to move forward, backward, turn right and left and open finger servo to catch object and unload object.

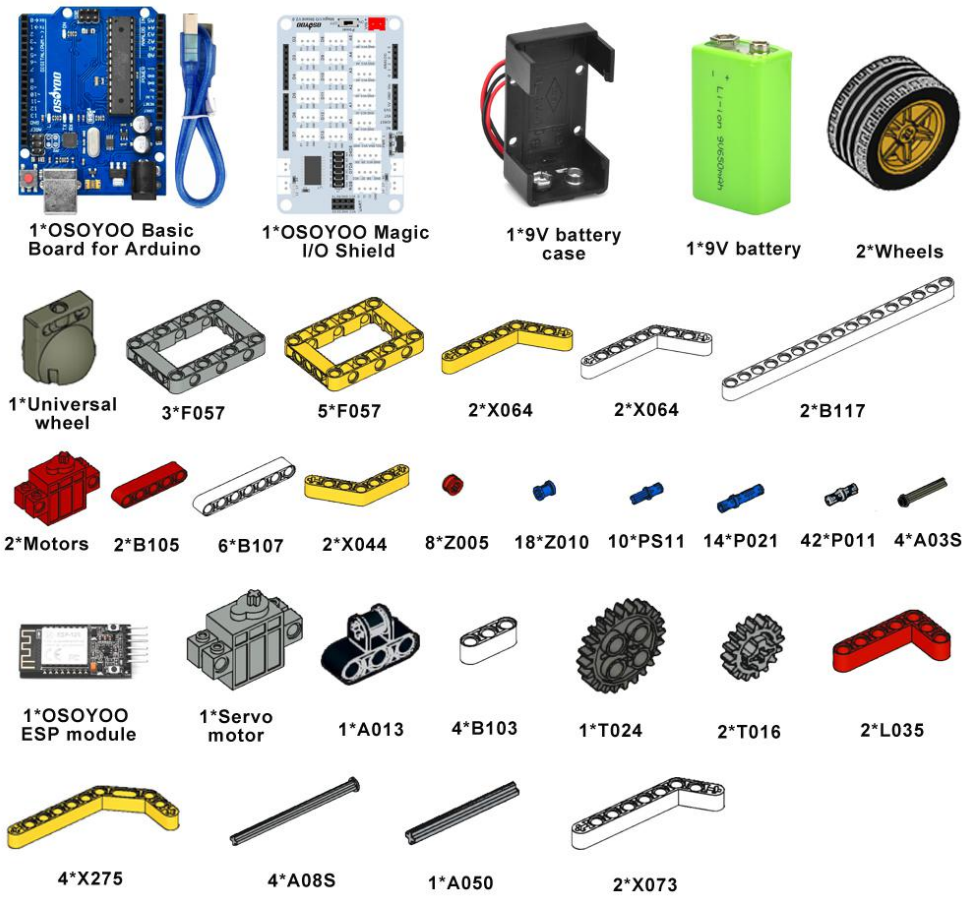
PARTS & DEVICES

Please prepare the following parts to complete this project

NOTE:

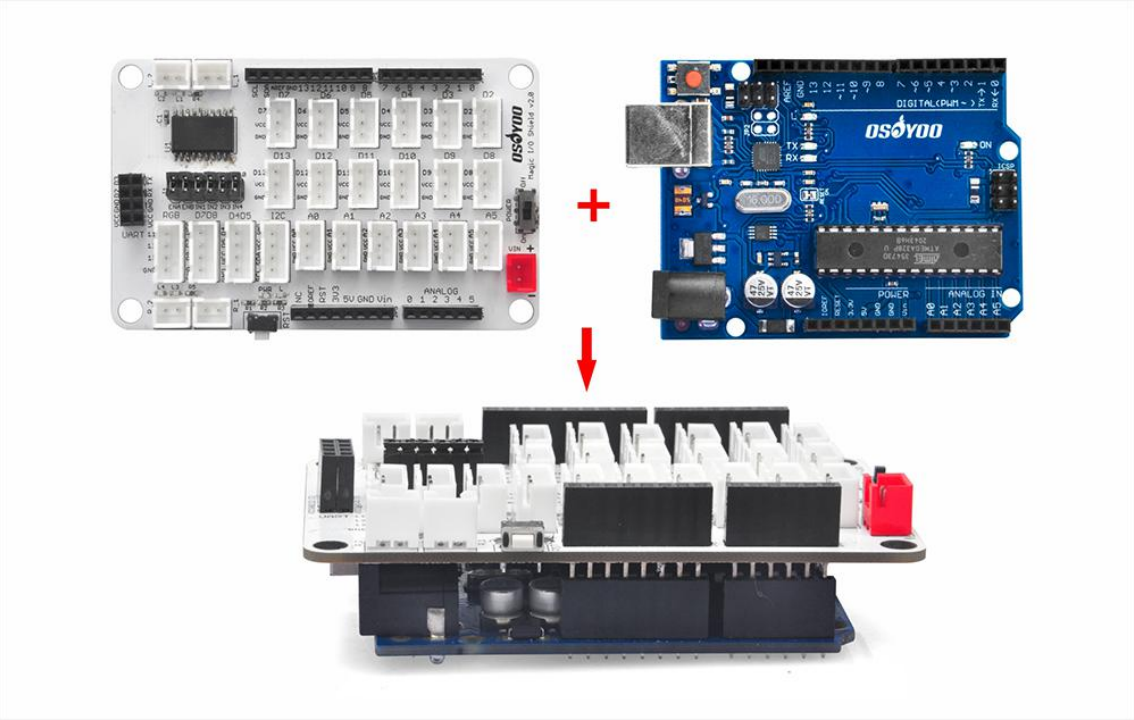
1.the color of the building block is subject to the actual product, which does not affect the use.

2. ALL OSOYOO PRODUCTS FOR ARDUINO ARE THIRD PARTY BOARD WHICH IS FULLY COMPATITABLE WITH ARDUINO



HOW TO MAKE

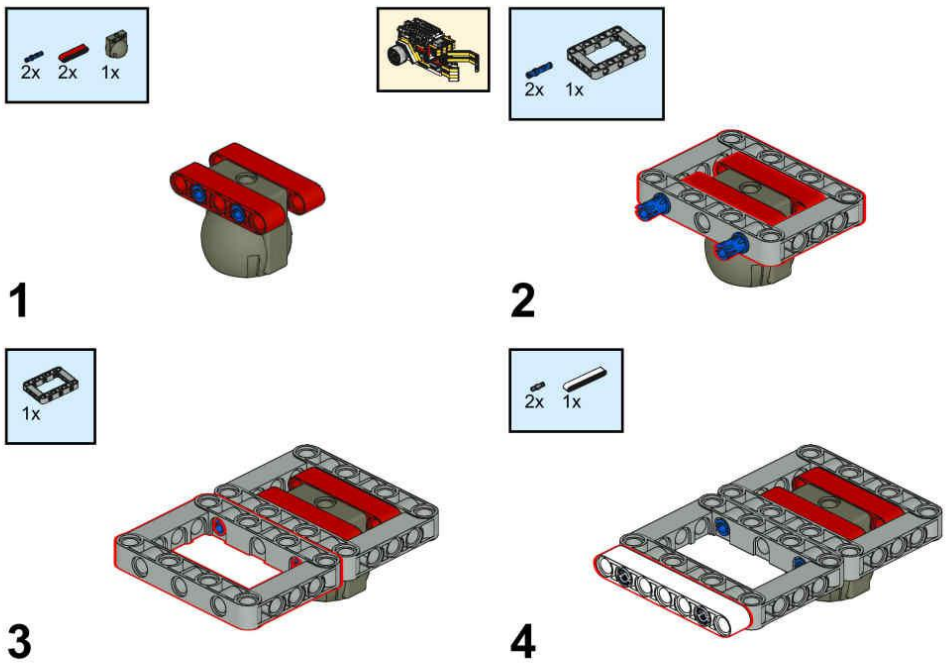
Before you build the robot with blocks, please install OSOYOO basic board for Arduino under OSOYOO Magic I/O shield as following (Attention please : the pins of I/O shield is aligned with the port of basic the board firstly, then press the shield tightly on the board).

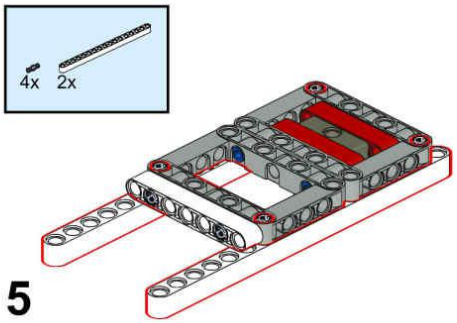


Please follow the building steps to build this robot car, If you want to get clear PDF building steps, please download

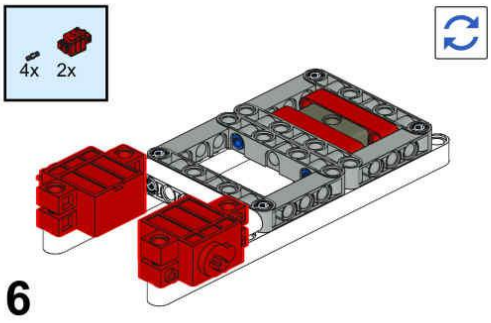
from https://osoyoo.com/picture/Building_Robot_Car/lesson6/lesson6.pdf

Note: If you have built [the robot car for lesson1](), please skip to the step35 in this PDF.

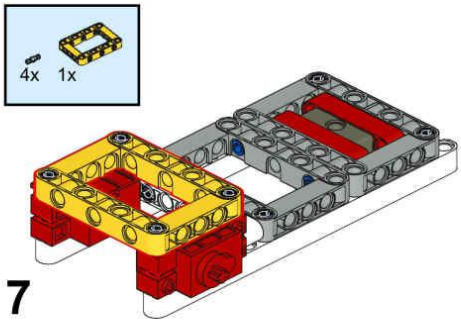




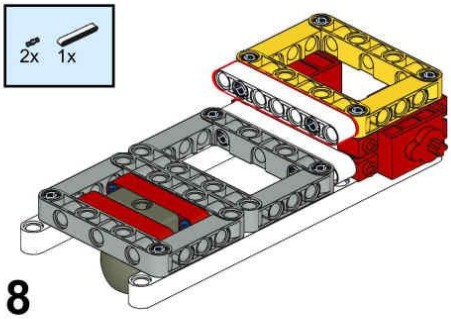
5



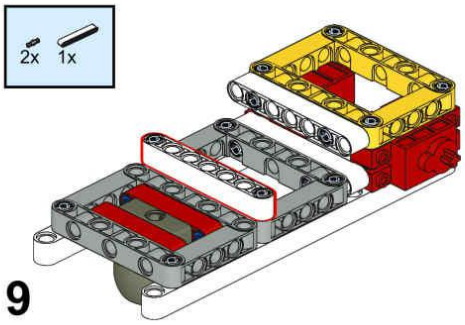
6



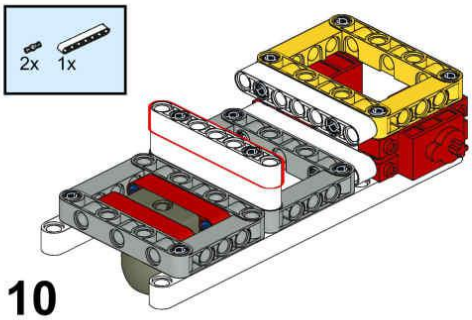
7



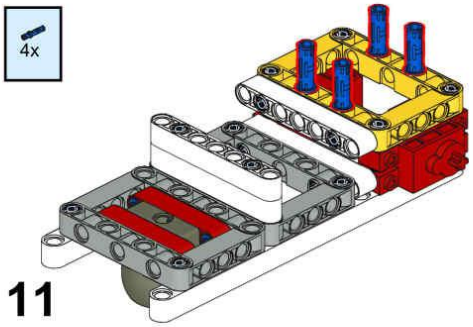
8



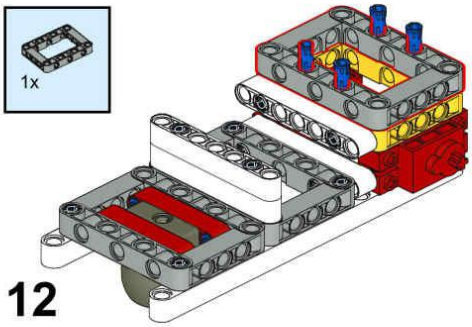
9



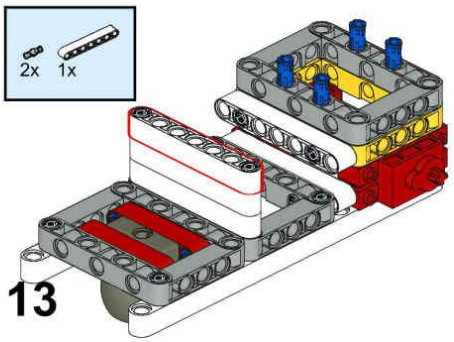
10

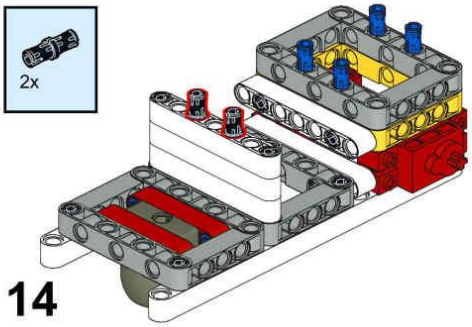


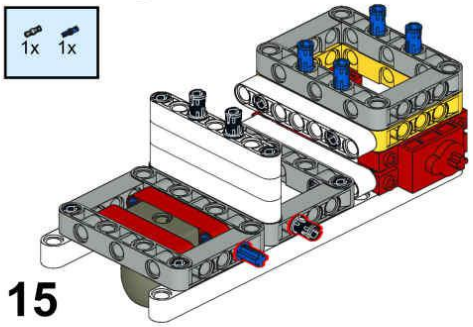
11

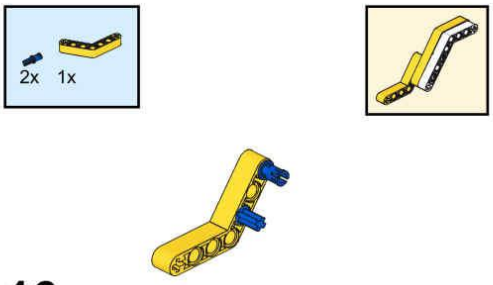


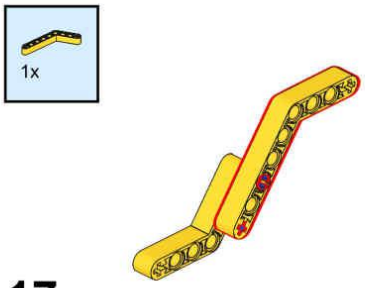
12

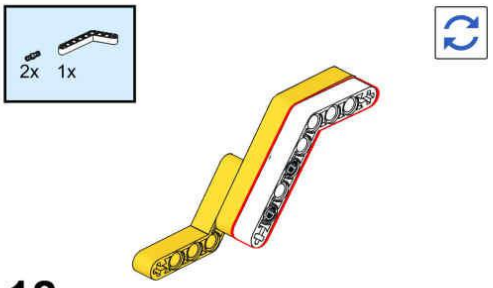

2x 1x
13

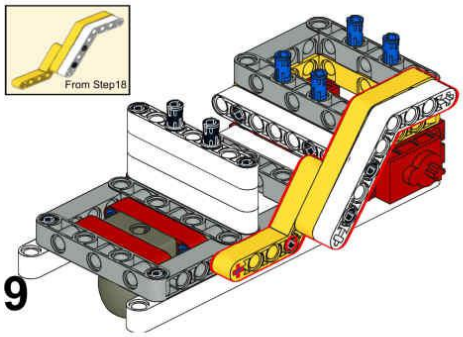

2x
14

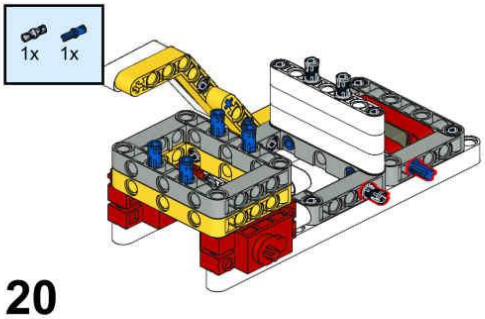

1x 1x
15

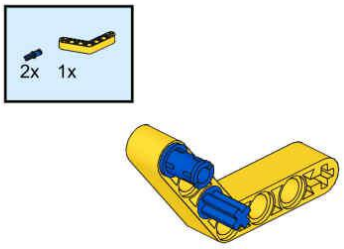

2x 1x
16

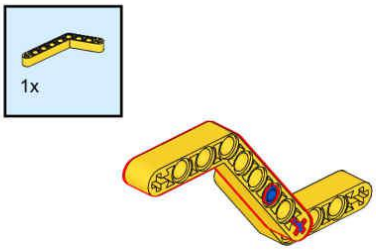

1x
17


2x 1x
18


From Step18
19

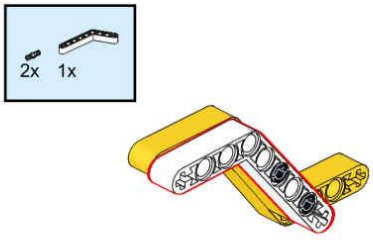

1x 1x
20

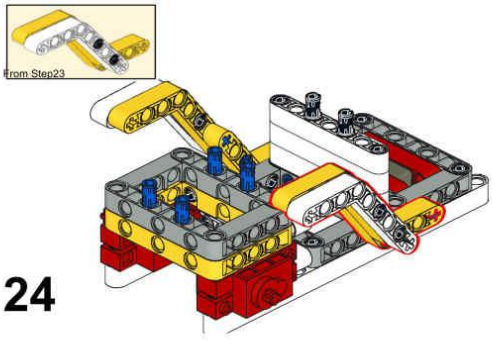




21

22

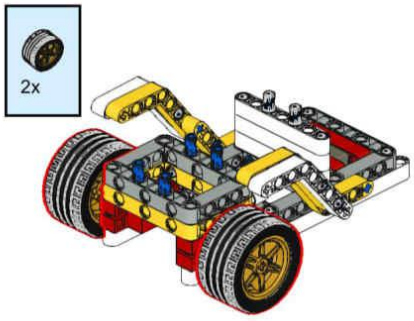




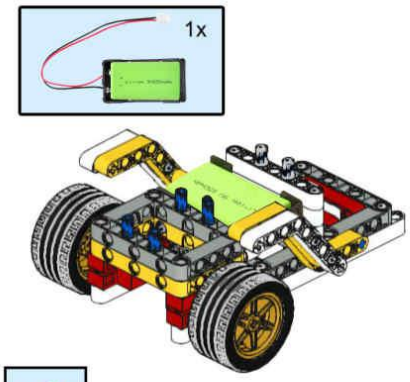
23

24

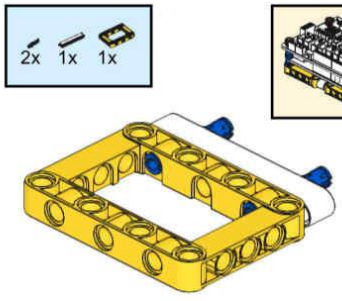
25

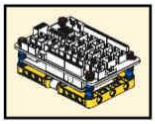


26

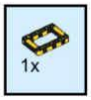


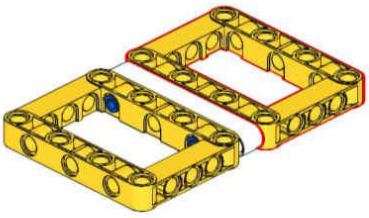
27

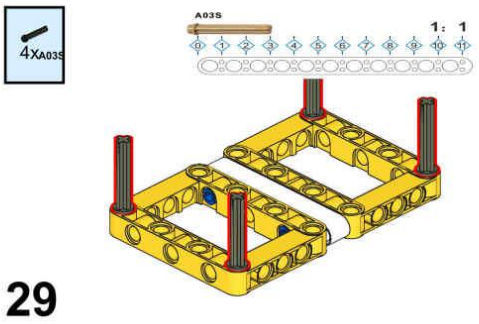




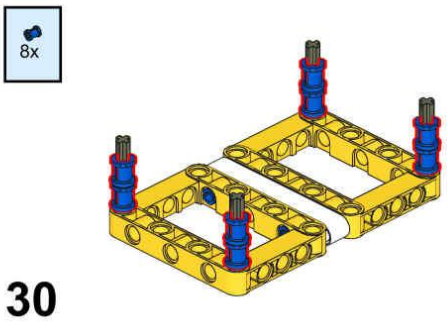
28



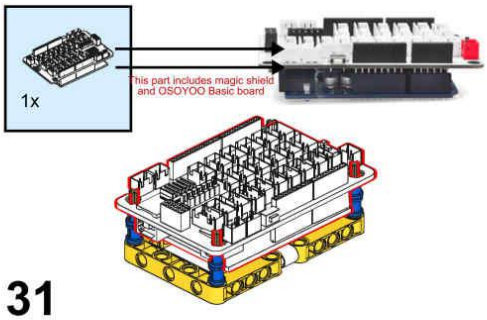




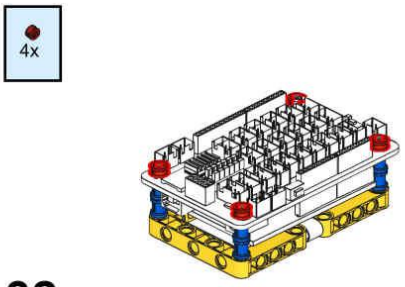
29



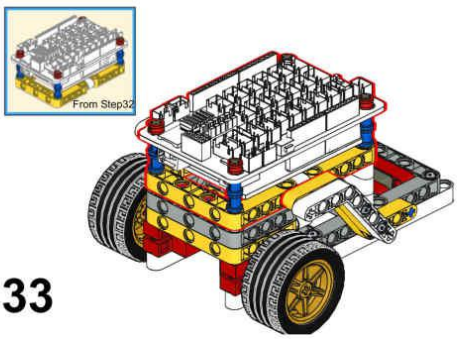
30



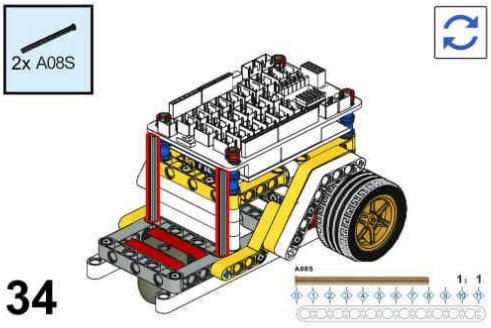
31



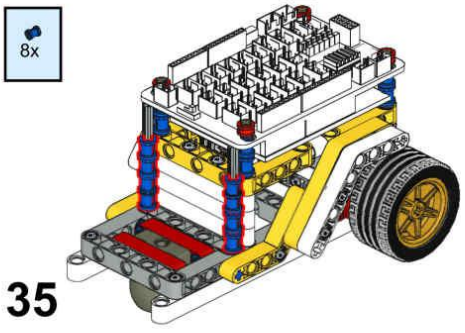
32



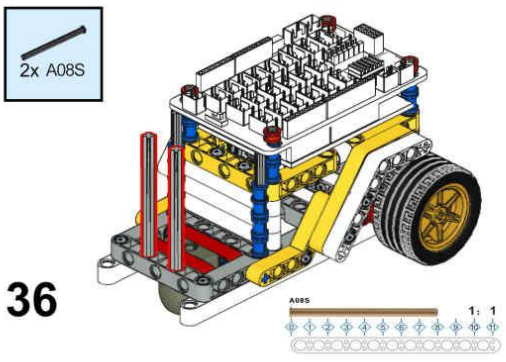
33



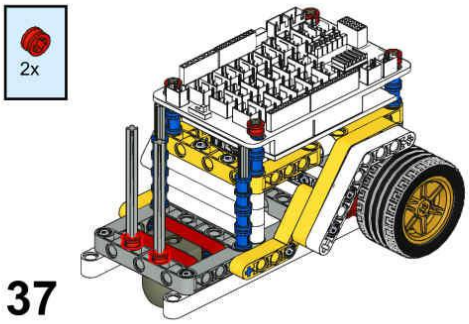
34

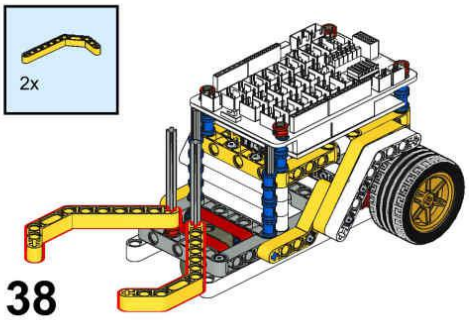


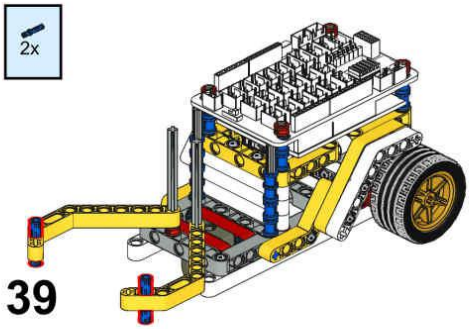
35

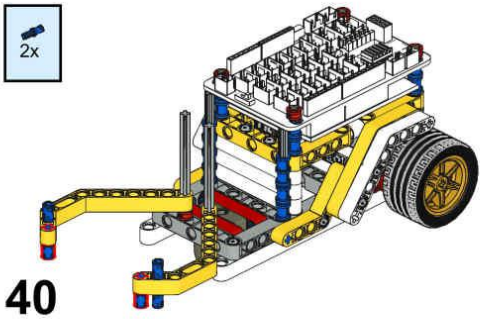


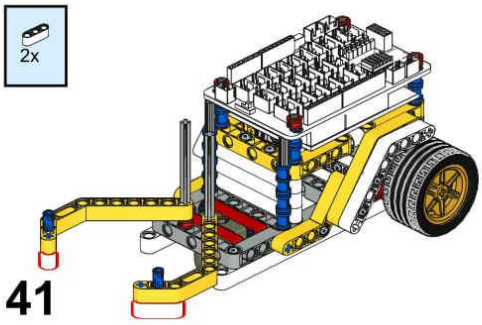
36

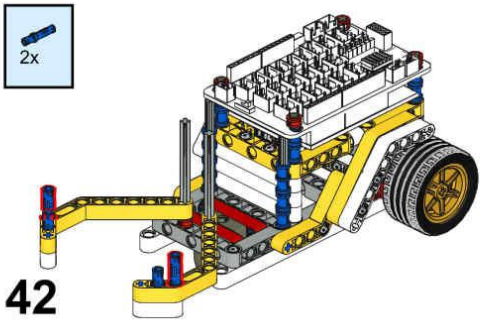

2x
37

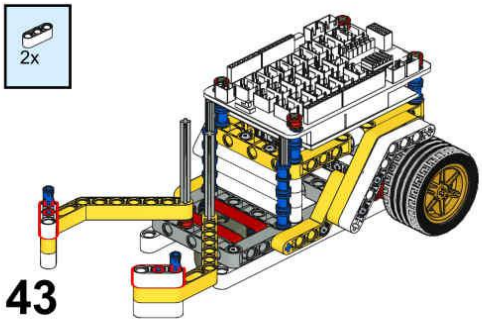

2x
38

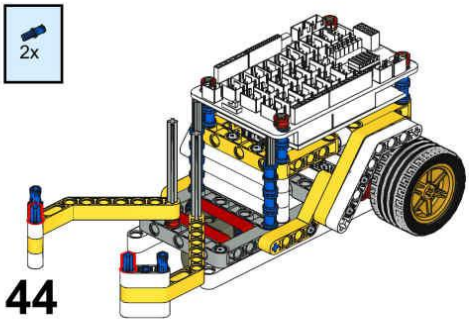

2x
39

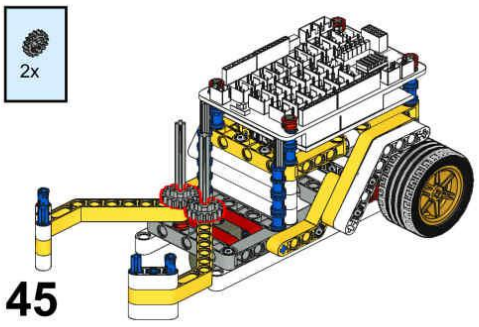

2x
40


2x
41

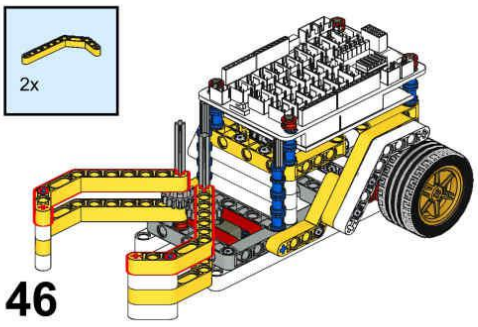

2x
42


2x
43

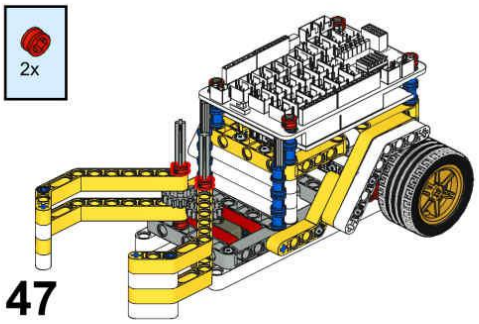

2x
44



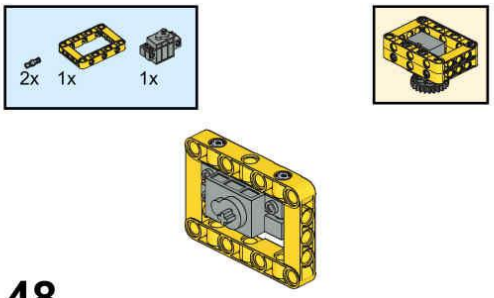
45



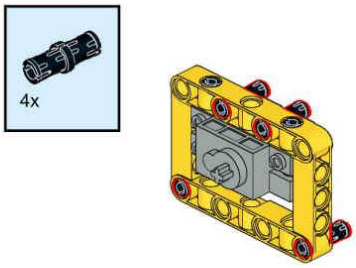
46



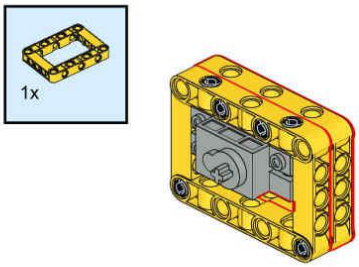
47



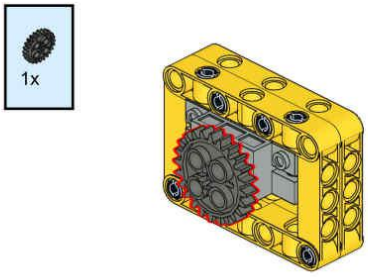
48



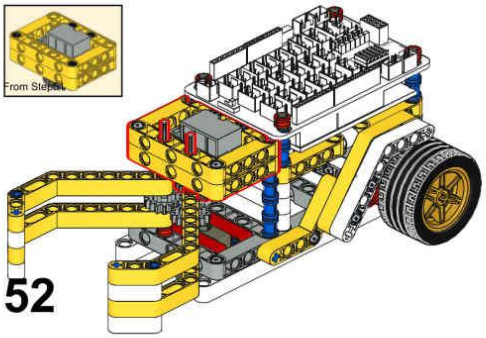
49



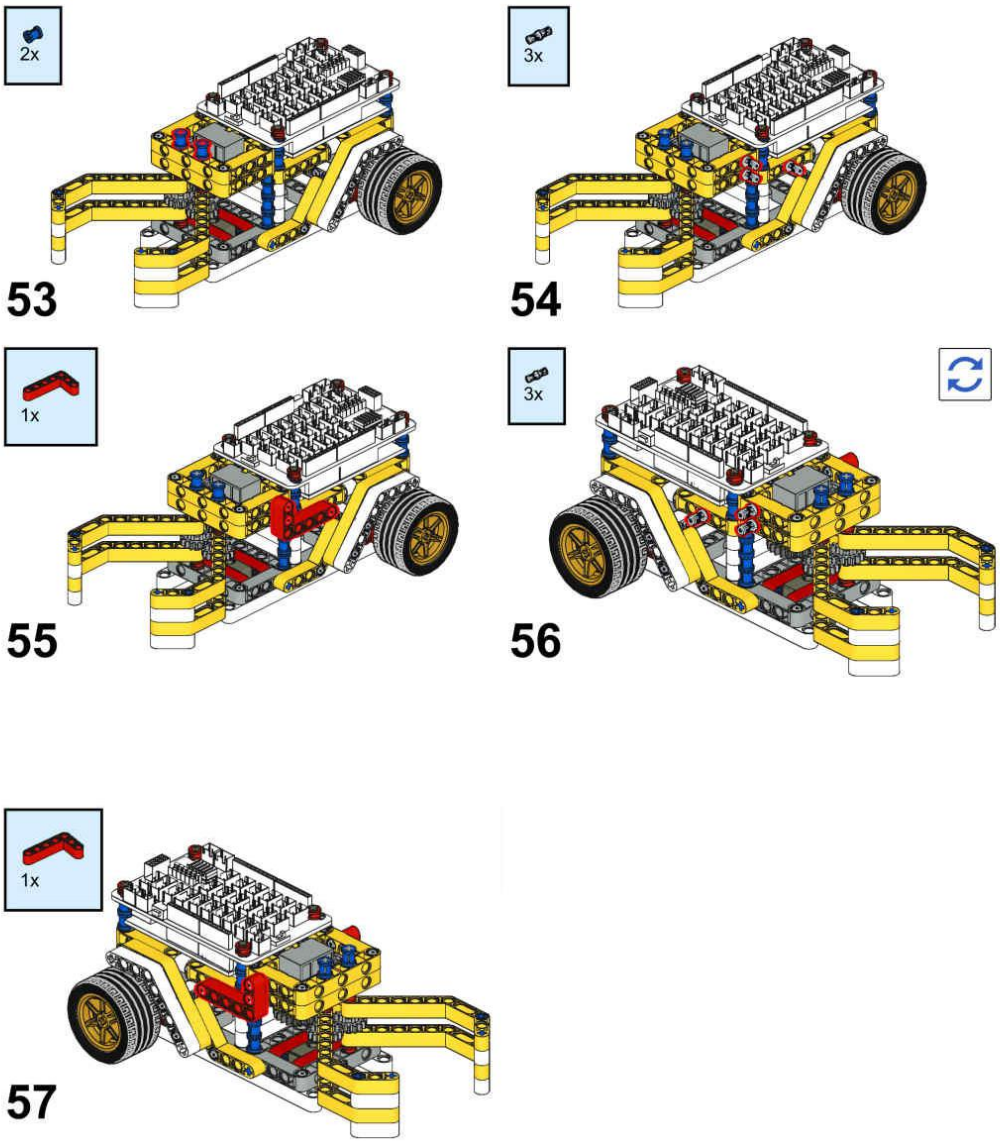
50



51



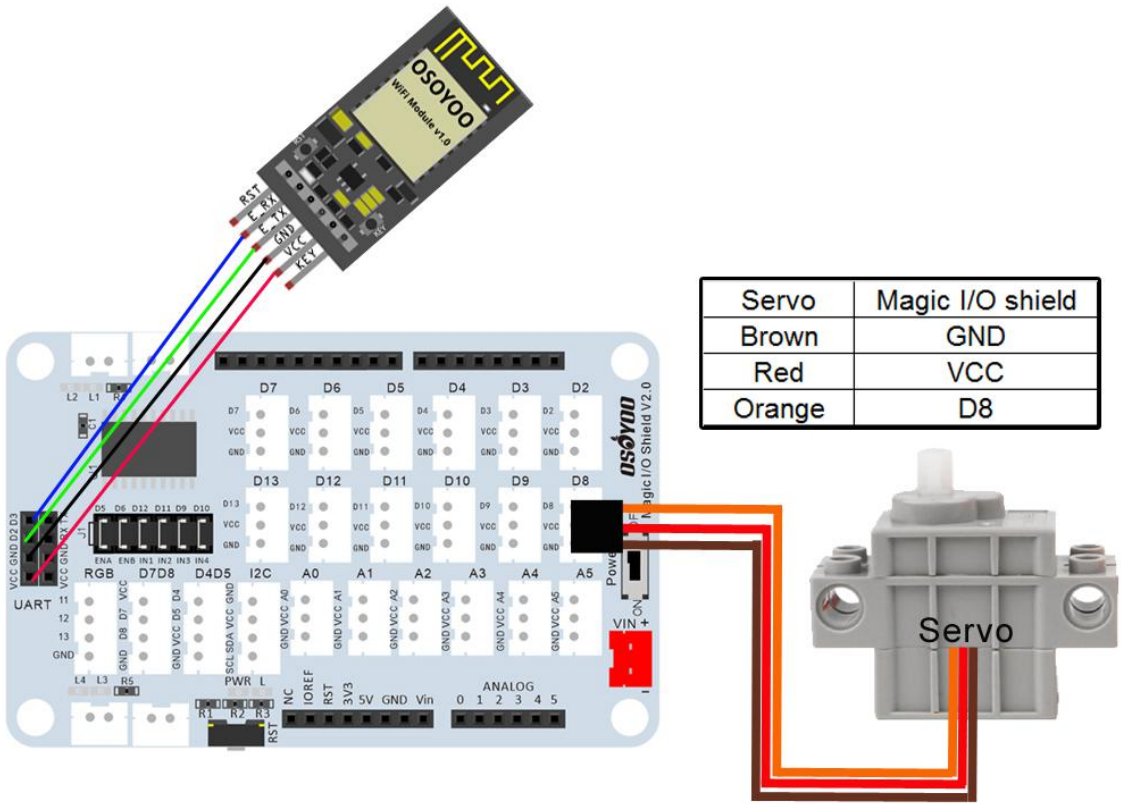
52



CIRCUIT CONNECTION

Please connect motors and 9V battery case as lesson1.

Then connect the **middle four pins** of OSOYOO WIFI Module connect to VCC/GDN/D2/D3 slot of Magic I/O shield, connect servo motor to D8 port as following (Attention please: there are six jumper caps on ENA/ENB/IN1/IN2/IN3/IN4)



Servo	Magic I/O shield
Brown	GND
Red	VCC
Orange	D8

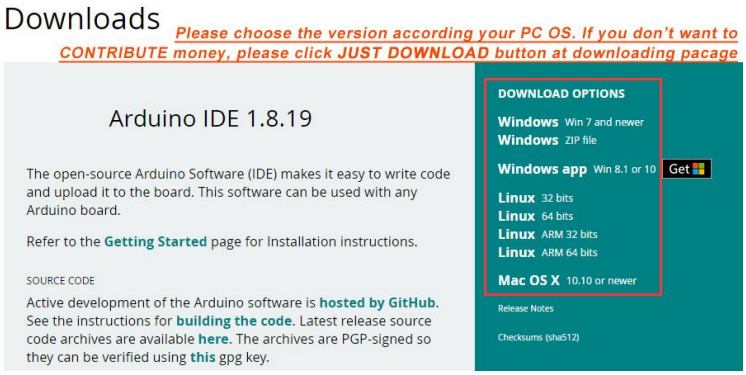
HOW TO CODE

Note: This lesson use Arduino IDE as programming tool.

Step 1: Install Arduino IDE. Download Arduino IDE

from https://www.arduino.cc/en/Main/Software?setlang=en , then install the software.

If you have install Arduino IDE, please skip this step.

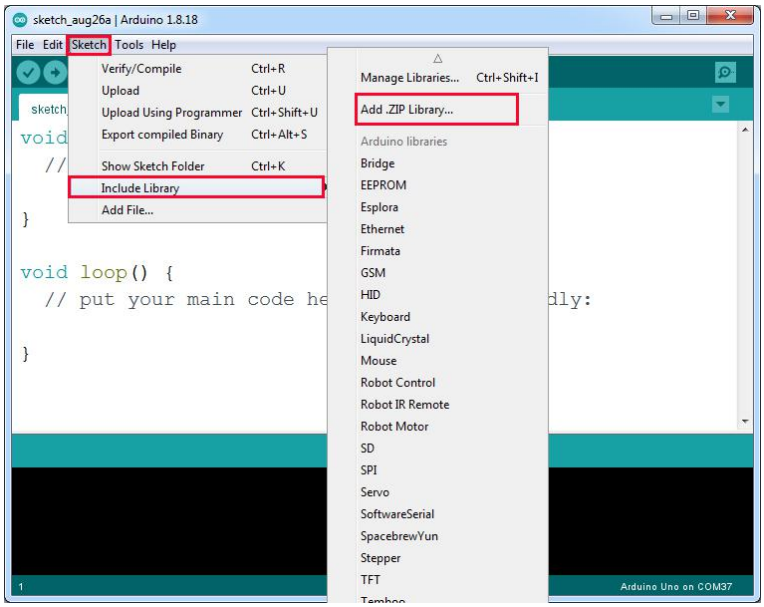


Step 2: Install **WifiEsp-master library** into IDE (If you have already installed

WifiEsp-master library, please skip this step)

Download **WifiEsp-master**

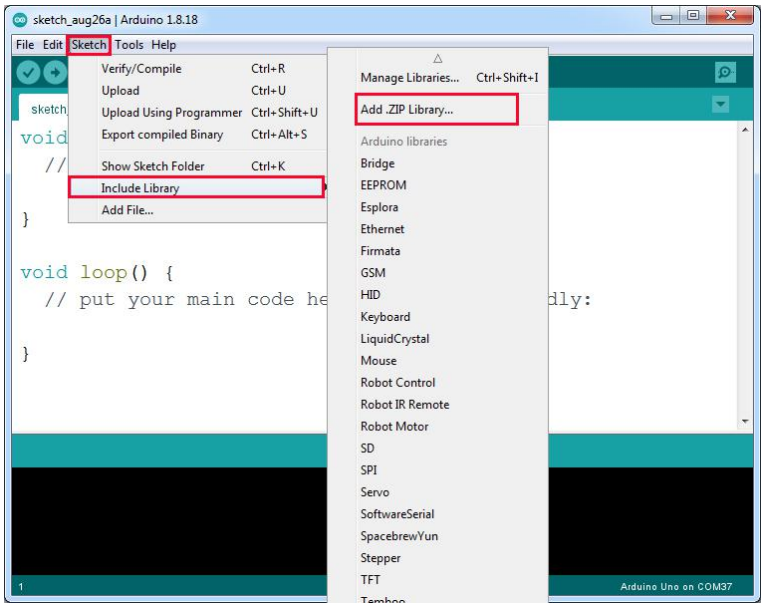
library from https://osoyoo.com/driver/WiFiEsp-master.zip, then import the library into IDE(Open IDE-> click Sketch->Include Library->Add .Zip Library)



Step 3: Install **OsoyooIOT Library** into IDE (If you have already installed OsoyooIOT Library, please skip this step)

Download **OsoyooIOT**

Library from https://osoyoo.com/driver/wifi-iot/OsoyooIOT.zip, then import the library into IDE(Open IDE-> click Sketch->Include Library->Add .Zip Library)



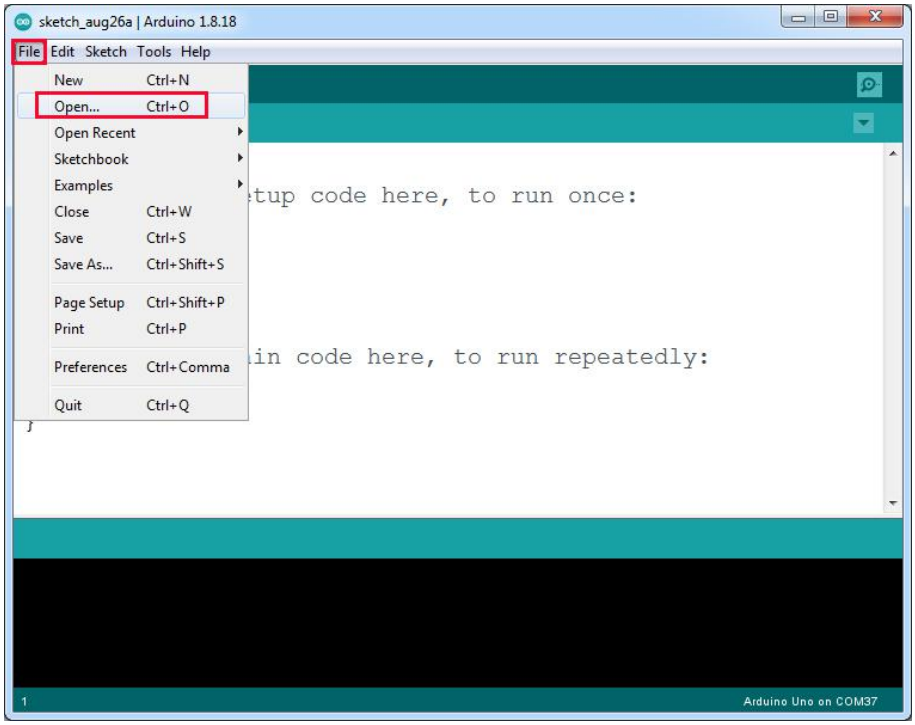
Step 4: Download sample code

from https://osoyoo.com/picture/Building_Robot_Car/lesson6/robot-arm-car.zip,

unzip the download zip file **robot-arm-car.zip**, you will see a folder

called **robot-arm-car** .

Step 5: Connect OSOYOO basic board for Arduino to PC with USB cable, Open Arduino IDE -> click file -> click Open -> choose code "**robot-arm-car.ino**" in **robot-arm-car** folder, load the code into arduino. **(Notice: Please turn off power when your Robot is connected to Personal Computer or Laptop via USB cable)**

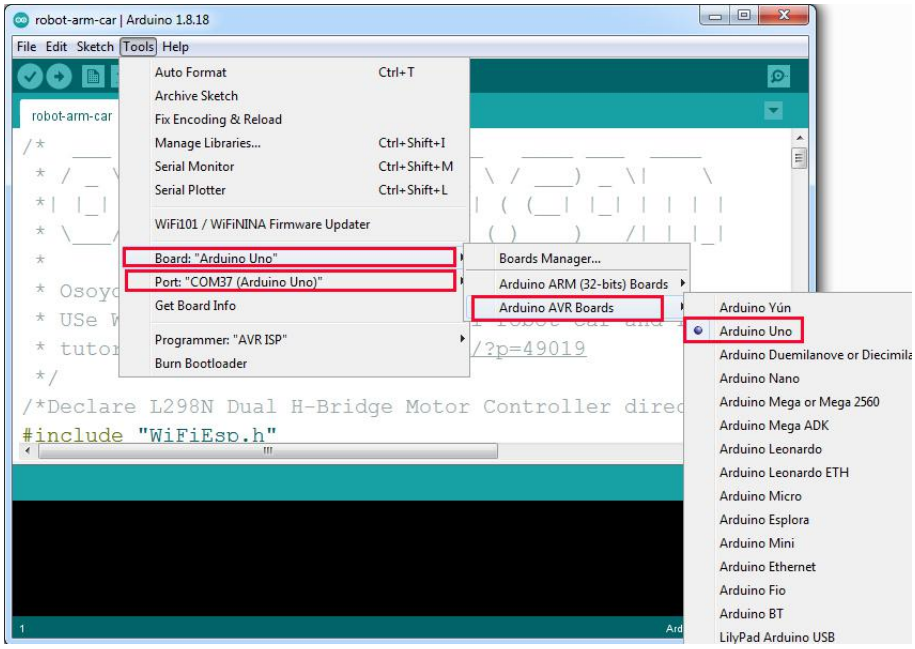


Step 6: You need change the code Line 35 and Line 36 as your router wifi ssid name and password :

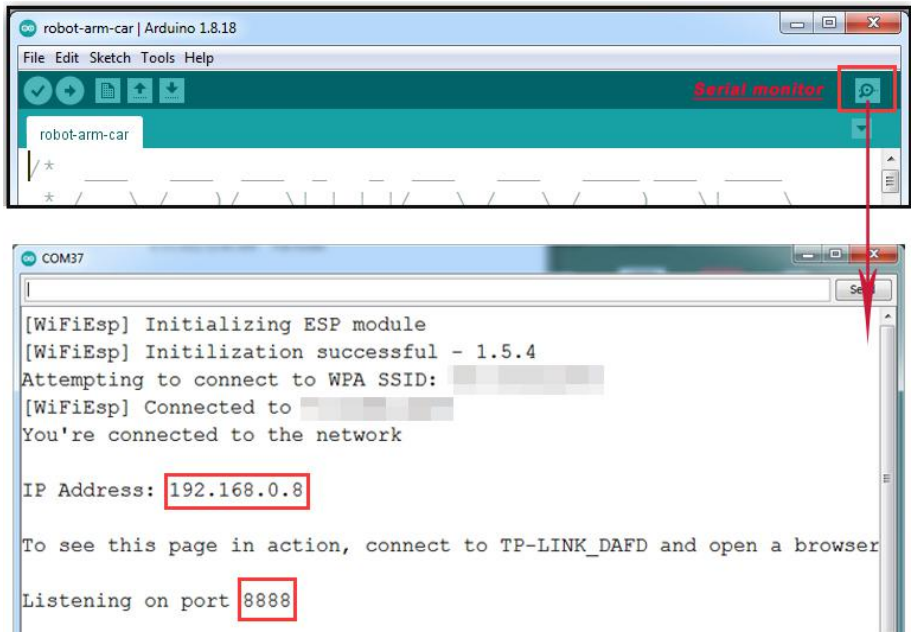
char ssid[] = "***"; // replace *** with your router wifi SSID (name)

char pass[] = "***"; // replace *** with your router wifi SSID (password)

Step 7: Choose corresponding board and port for your project, upload the sketch to the board.



Step 8: click the Serial monitor window in upper right corner of IDE, you will see following result (Note: your PC and the robot are connected with the same LAN network):



TIPS:

If you find the speed for the car is slow or fast, please open the code and change the value of line31 to line33 as the following:

#define FAST_SPEED 50

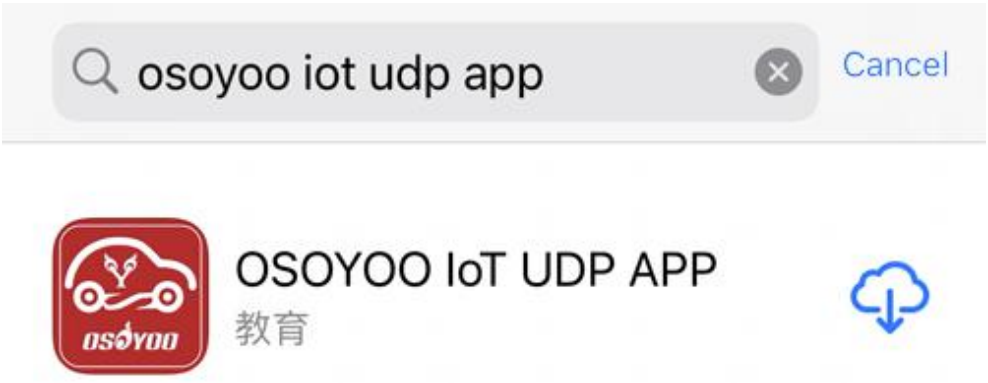
#define MID_SPEED 50

#define SPEED 50

HOW TO PLAY

APP Installation and Setting:

Step 1) Download **OSOYOO Iot udp APP** by searching **OSOYOO Iot udp APP** from Apple Store(iPhone/iPad) or Google Play Store(Android device).



Step 2): Install the **OSOYOO Iot udp APP** in your Phone and make sure your Cell phone is connected with the same LAN Network of your robot car for Arduino.

Step 3): Set IP Address in APP

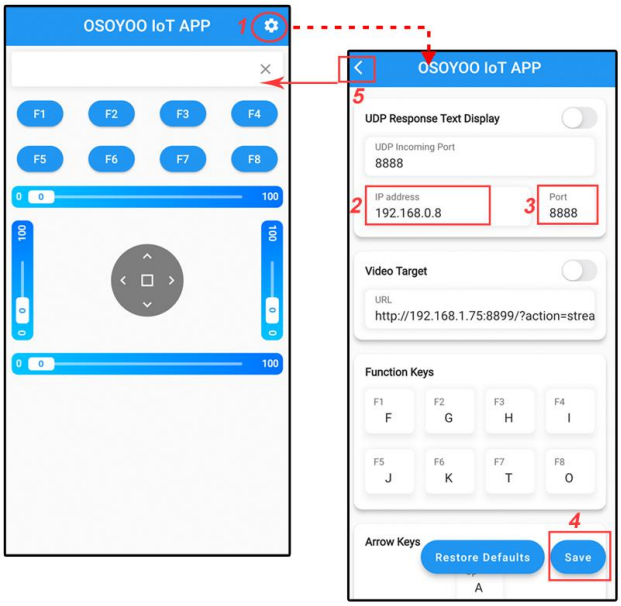
1)Open APP, click Setting button in upper right corner

2)Use the IP address you get in serial monitor window (For our robot is 192.168.0.8) to replace default IP 192.168.1.255

3)keep default port number 8888 without changing

4)Click **Save** button to save the changes you just made

5)Click Back Arrow to go back APP front UI



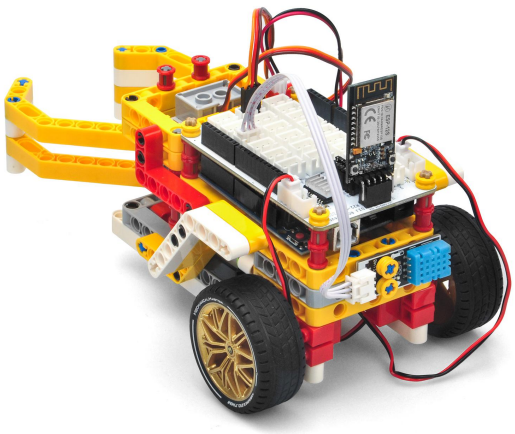
You can use the Arrow keys to control the car movement. APP Button Action

▲	Move Forward
▼	Move Backward
◄	Left Turn
►	Right Turn
[]	Stop
F1	Open Finger
F2	Close Finger

Lesson 7: WIFI APP control Robot Car

OBJECTIVE

In this project we will connect Robot Car to WIFI and Use an APP to control the car through Internet. This is a typical Internet of Things(IoT) Application. Phone APP controls the car movement, and temperature & humidity sensor module sends temperature and humidity to your phone.



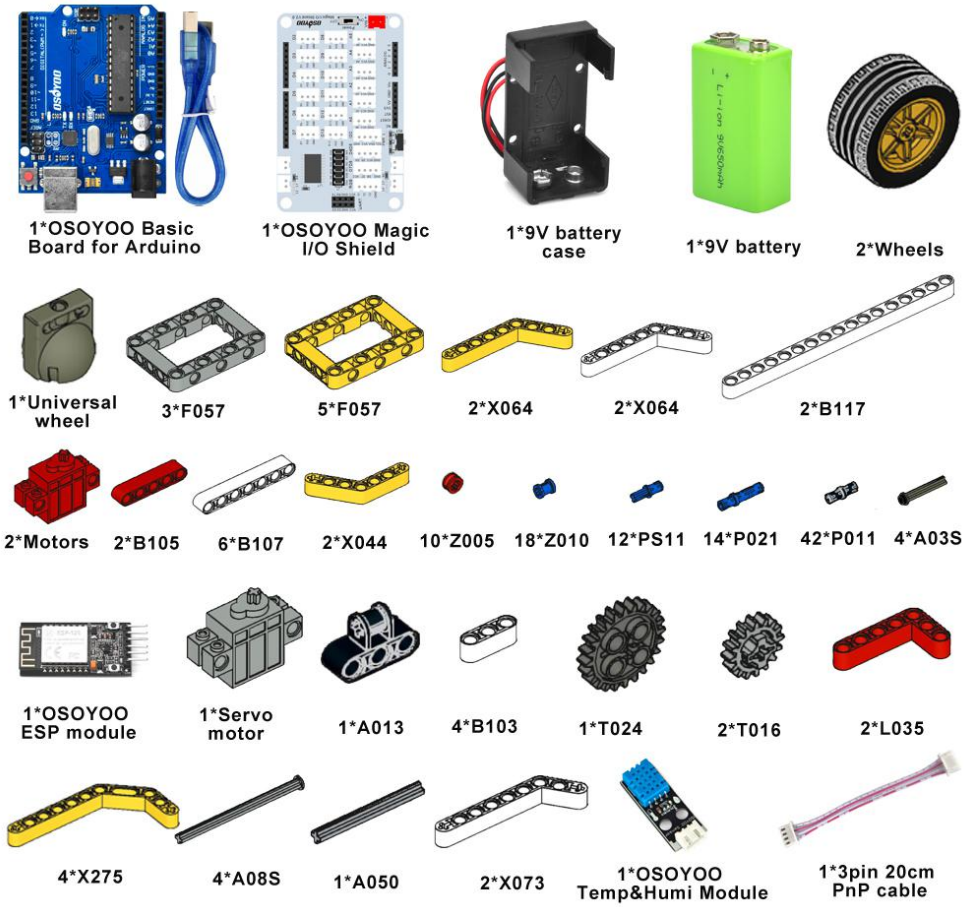
PARTS & DEVICES

Please prepare the following parts to complete this project

NOTE:

1.the color of the building block is subject to the actual product, which does not affect the use.

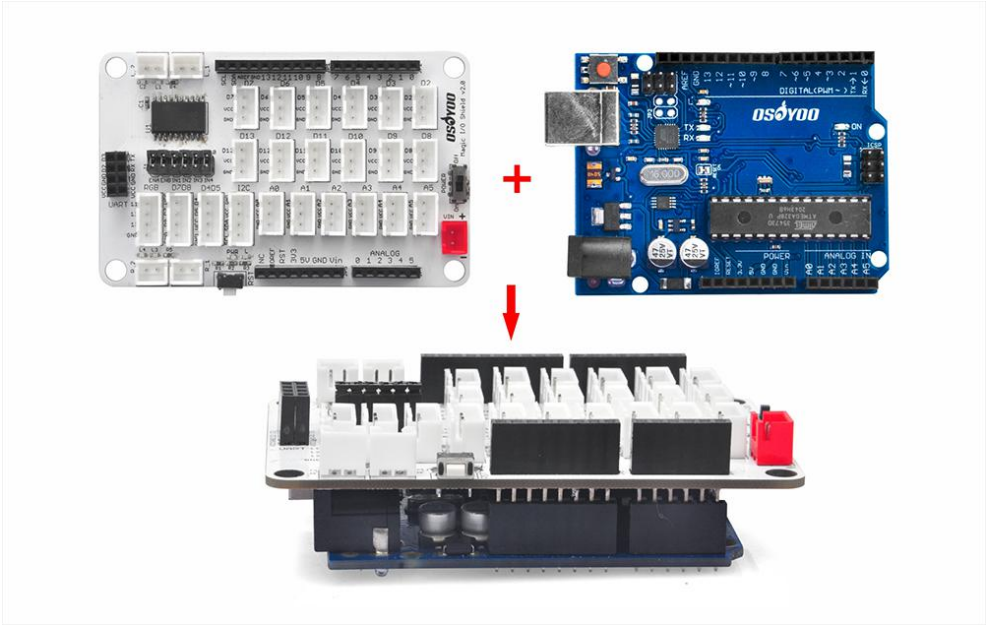
2. ALL OSOYOO PRODUCTS FOR ARDUINO ARE THIRD PARTY BOARD WHICH IS FULLY COMPATITABLE WITH ARDUINO



HOW TO MAKE

Before you build the robot with blocks, please install OSOYOO basic board for Arduino under OSOYOO Magic I/O shield as following (Attention please : the pins of

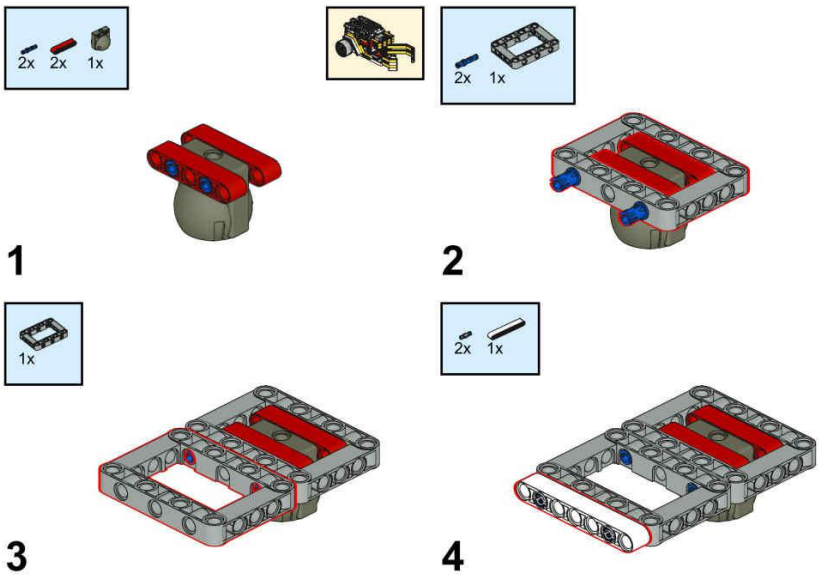
I/O shield is aligned with the port of basic the board firstly, then press the shield tightly on the board).

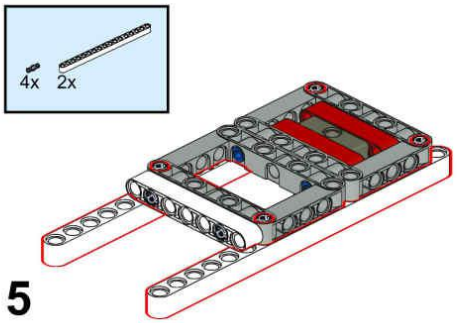


Please follow the building steps to build this robot car, If you want to get clear PDF building steps, please download

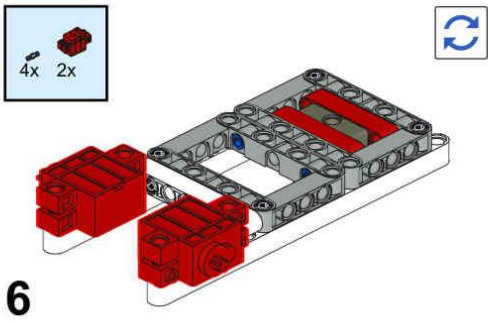
from https://osoyoo.com/picture/Building_Robot_Car/lesson7/lesson7.pdf

Note: If you have built the robot car for lesson6, please skip to the step58 in this PDF.

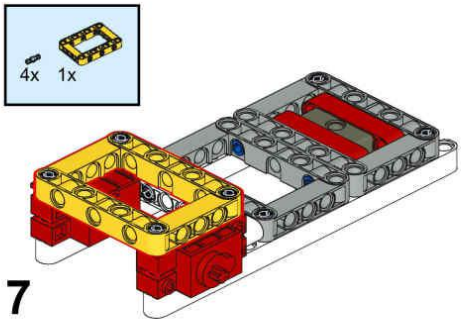




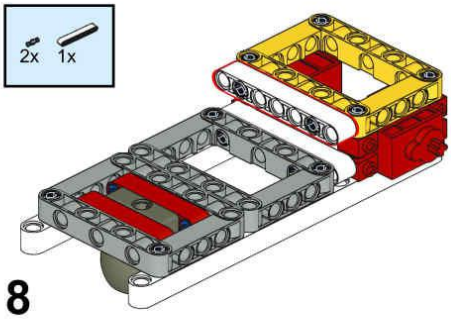
5



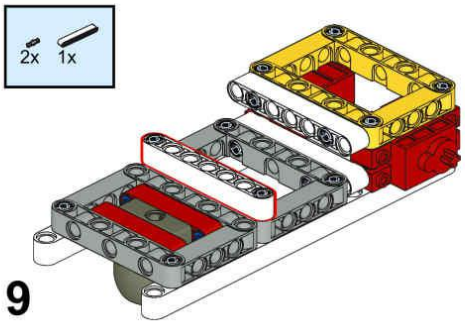
6



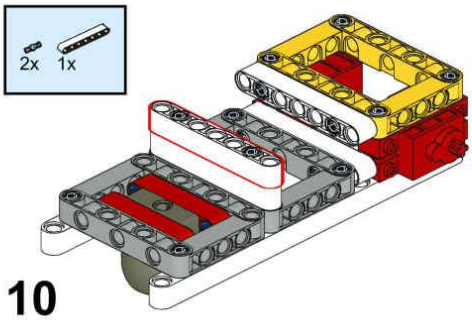
7



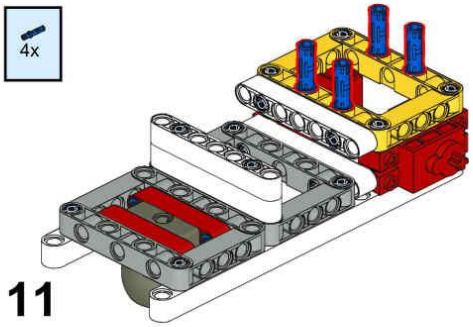
8



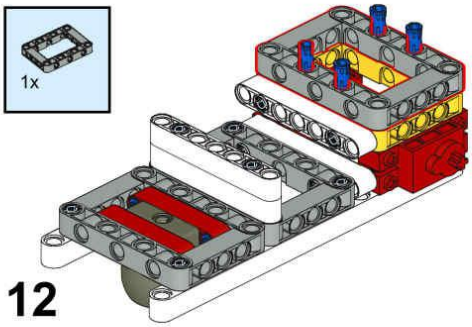
9



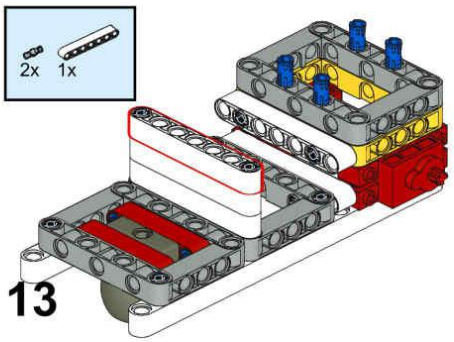
10

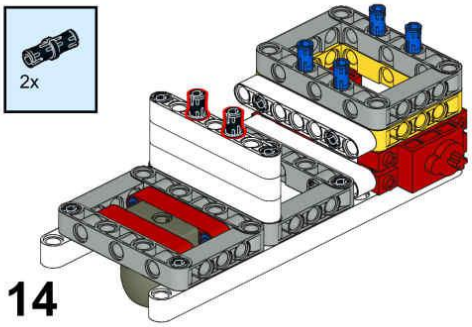


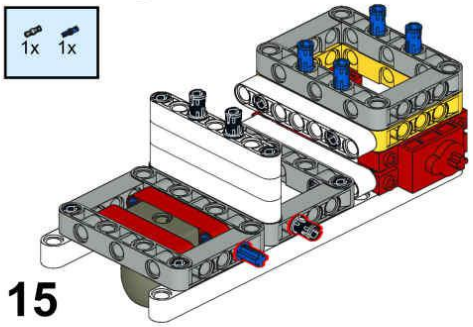
11

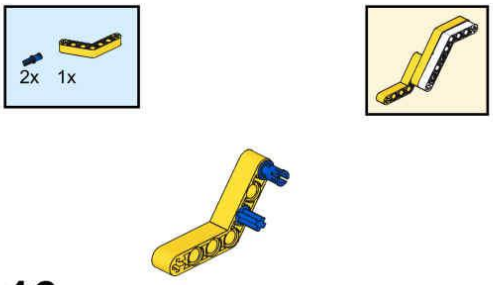


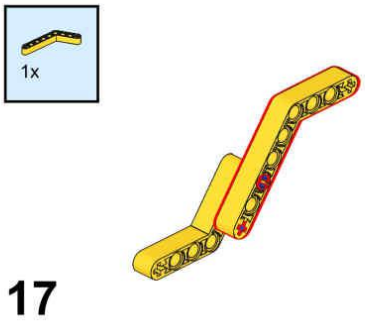
12

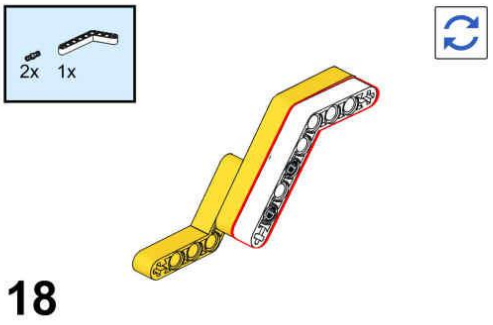

2x 1x
13

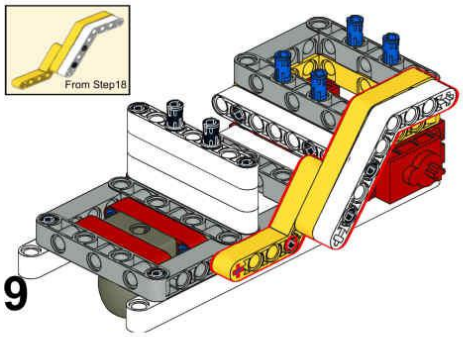

2x
14

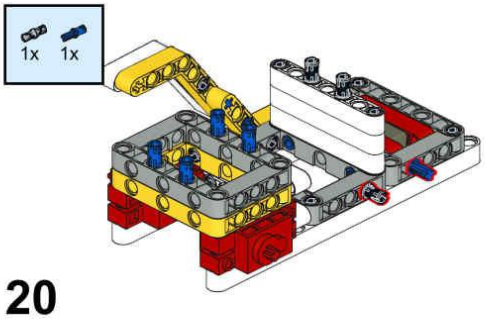

1x 1x
15

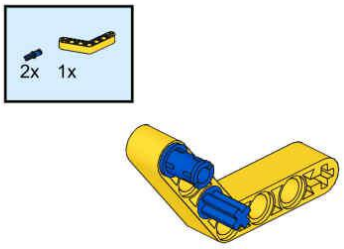

2x 1x
16

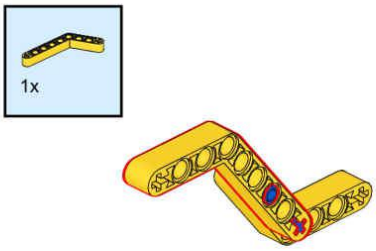

1x
17


2x 1x
18


From Step18
19

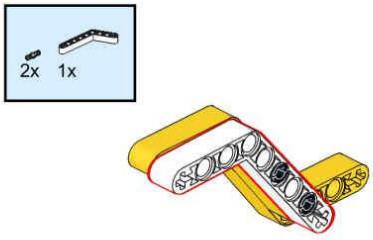

1x 1x
20

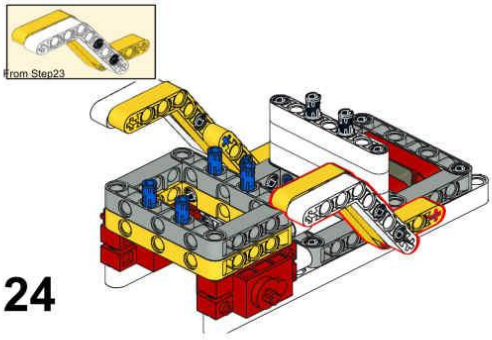




21

22

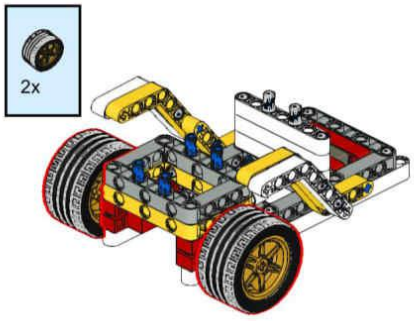




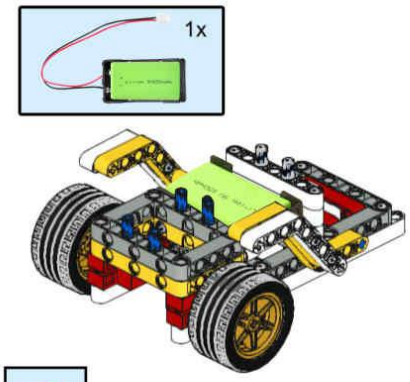
23

24

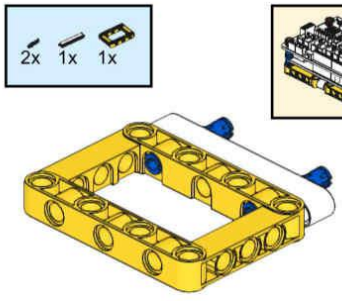
25

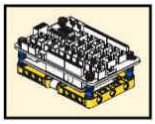


26

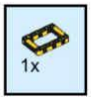


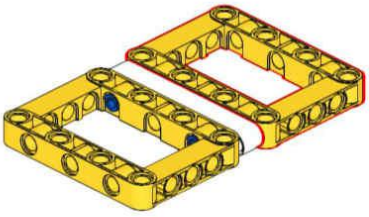
27

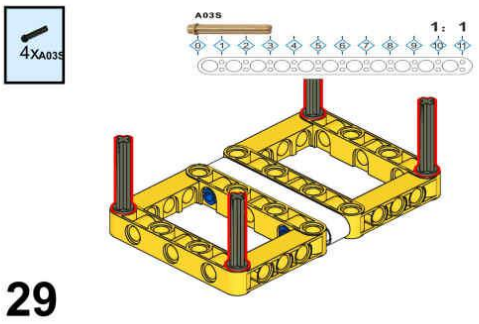




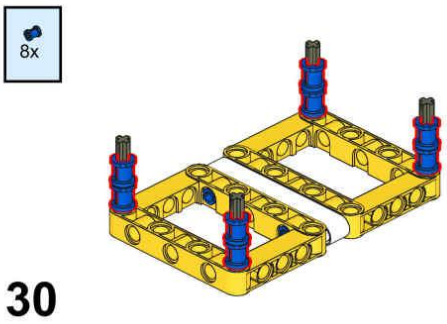
28



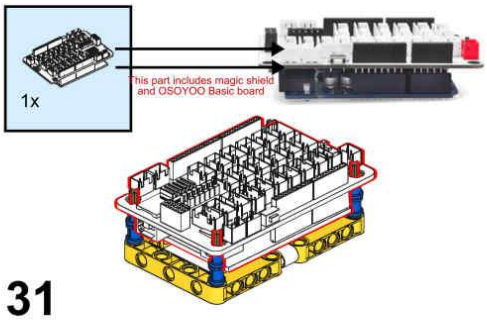




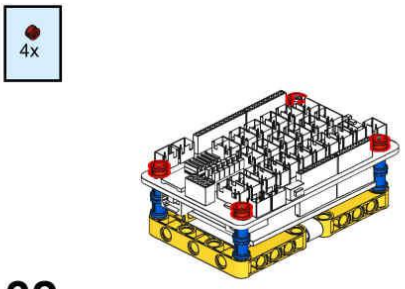
29



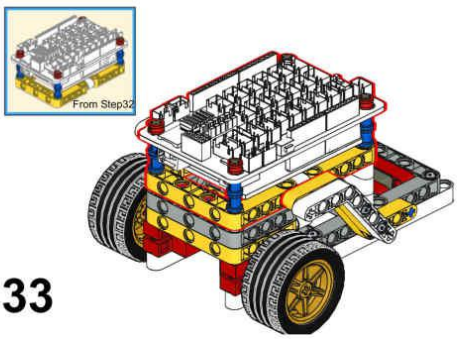
30



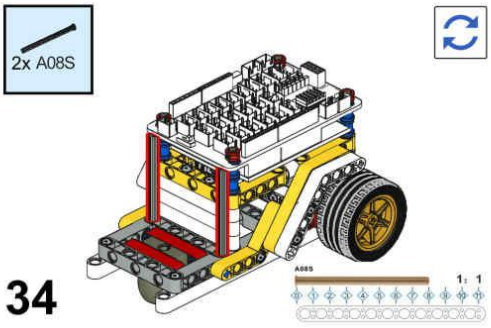
31



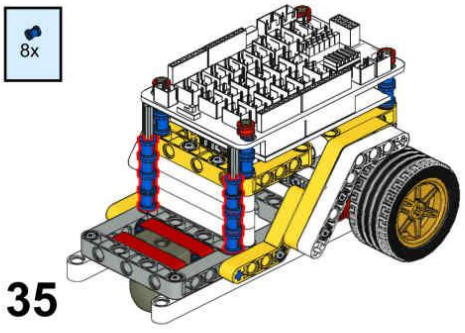
32



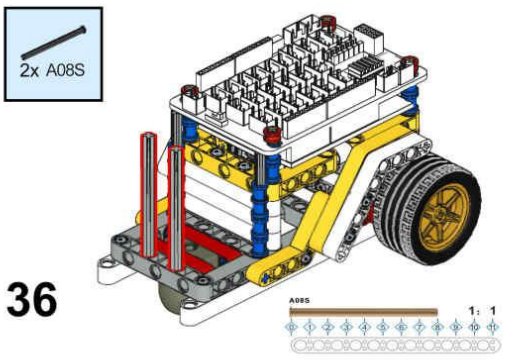
33



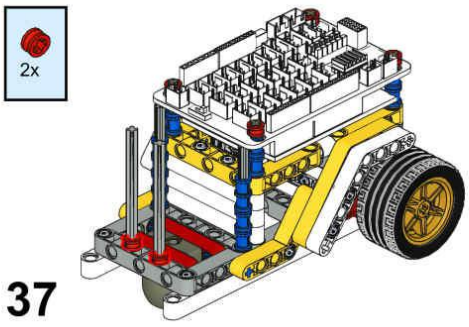
34

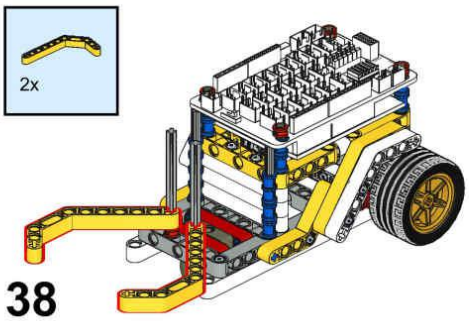


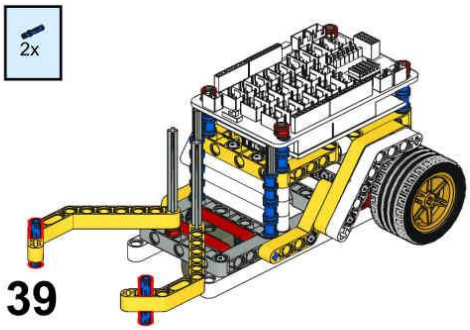
35

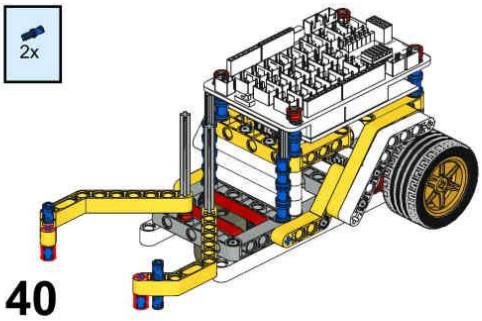


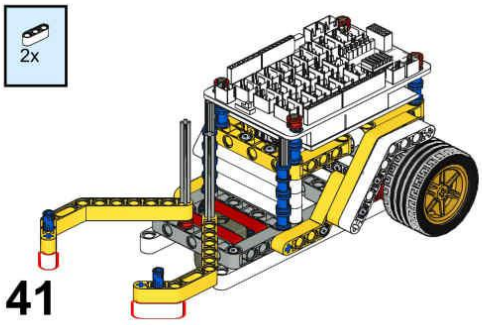
36

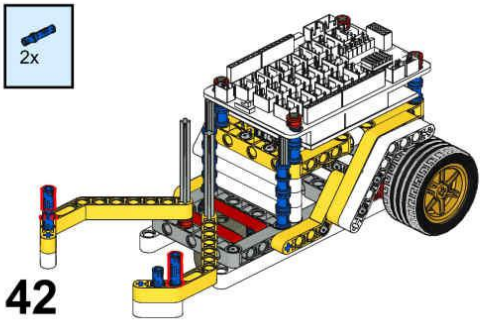

2x
37

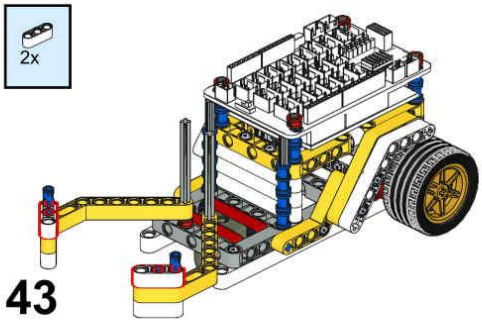

2x
38

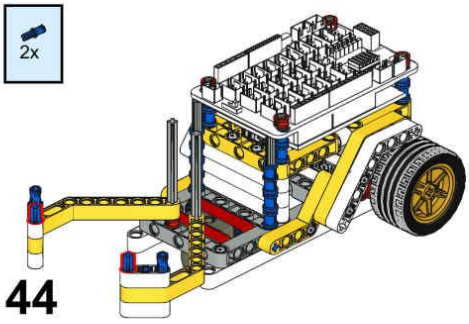

2x
39

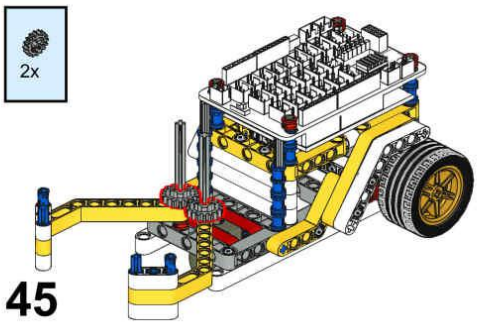

2x
40


2x
41

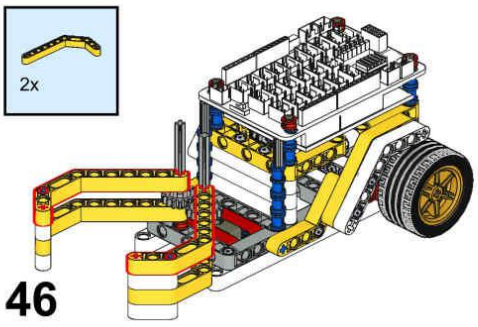

2x
42


2x
43

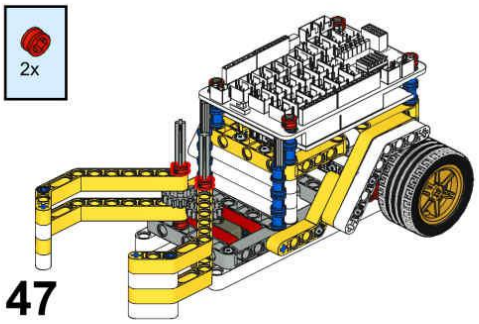

2x
44



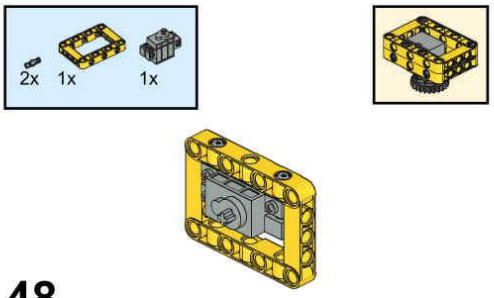
45



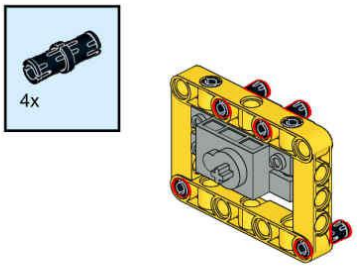
46



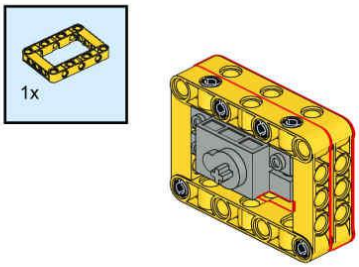
47



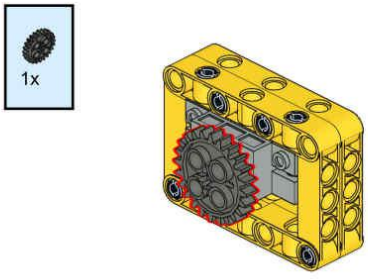
48



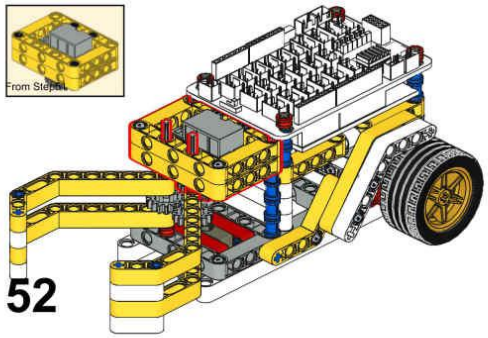
49



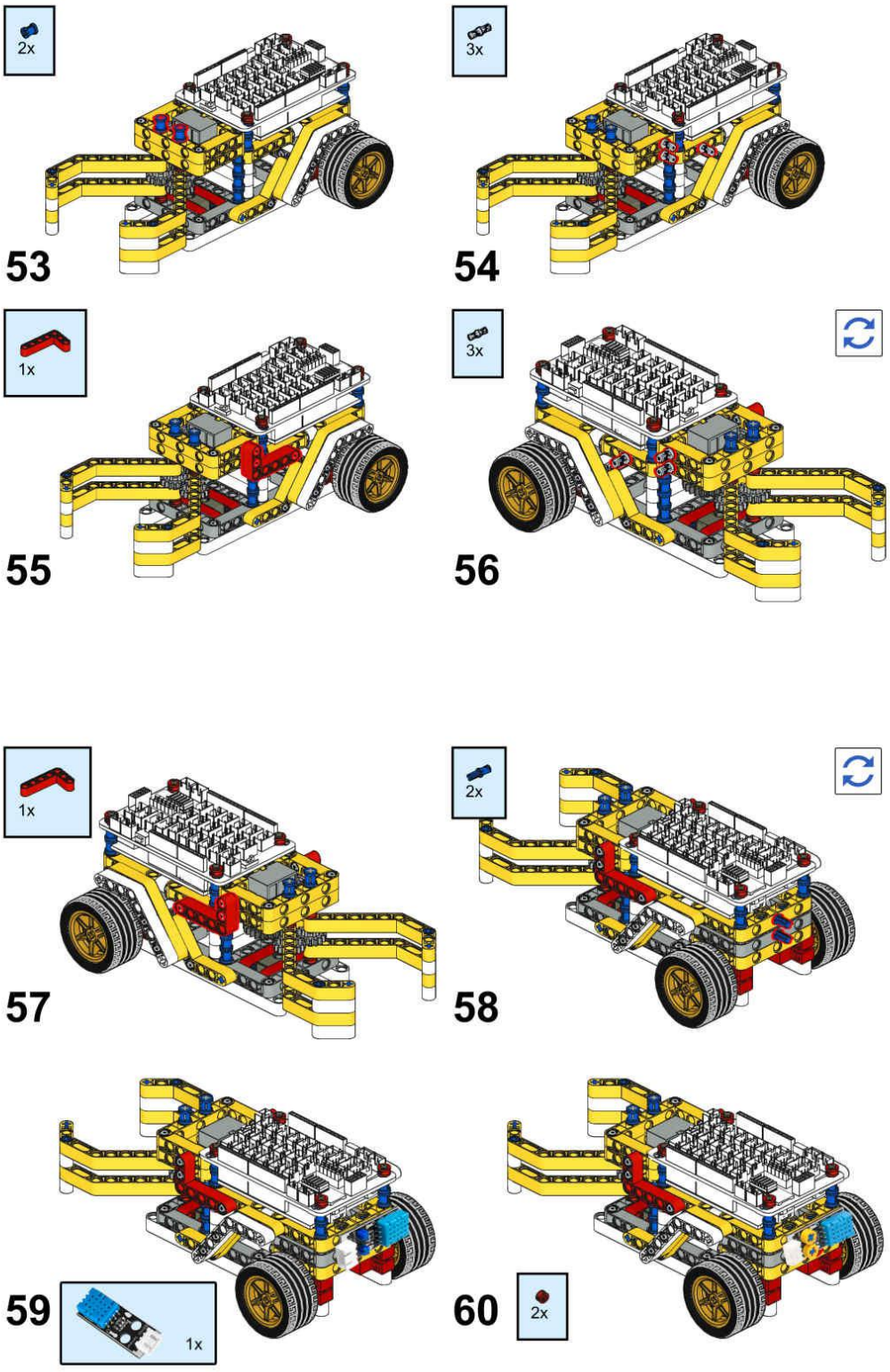
50



51



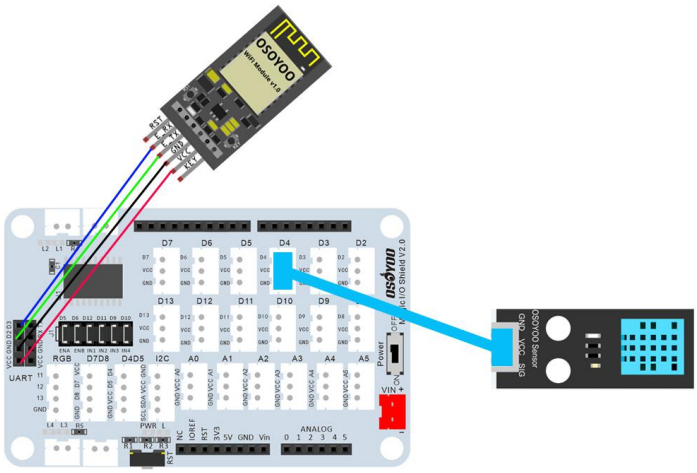
52



CIRCUIT CONNECTION

Please connect motors and 9V battery case as lesson1.

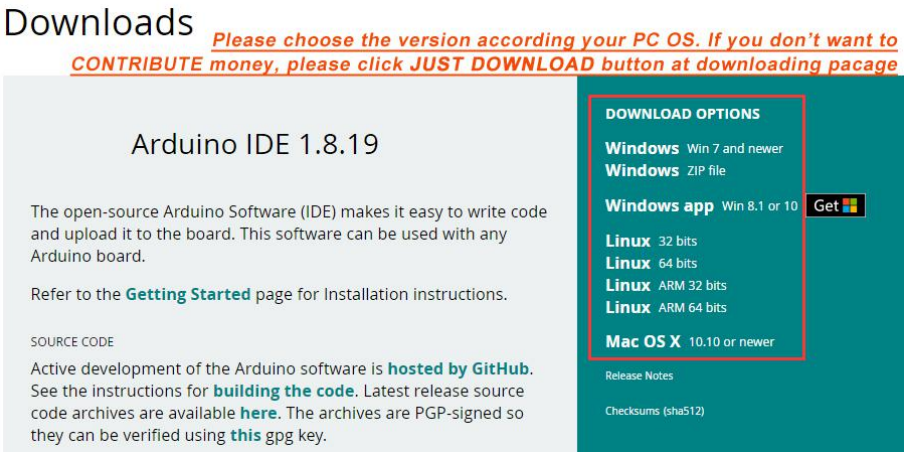
Then connect the **middle four pins** of OSOYOO WIFI Module connect to VCC/GDN/D2/D3 slot of Magic I/O shield, connect servo motor to D4 port as following (Attention please: there are six jumper caps on ENA/ENB/IN1/IN2/IN3/IN4)



HOW TO CODE

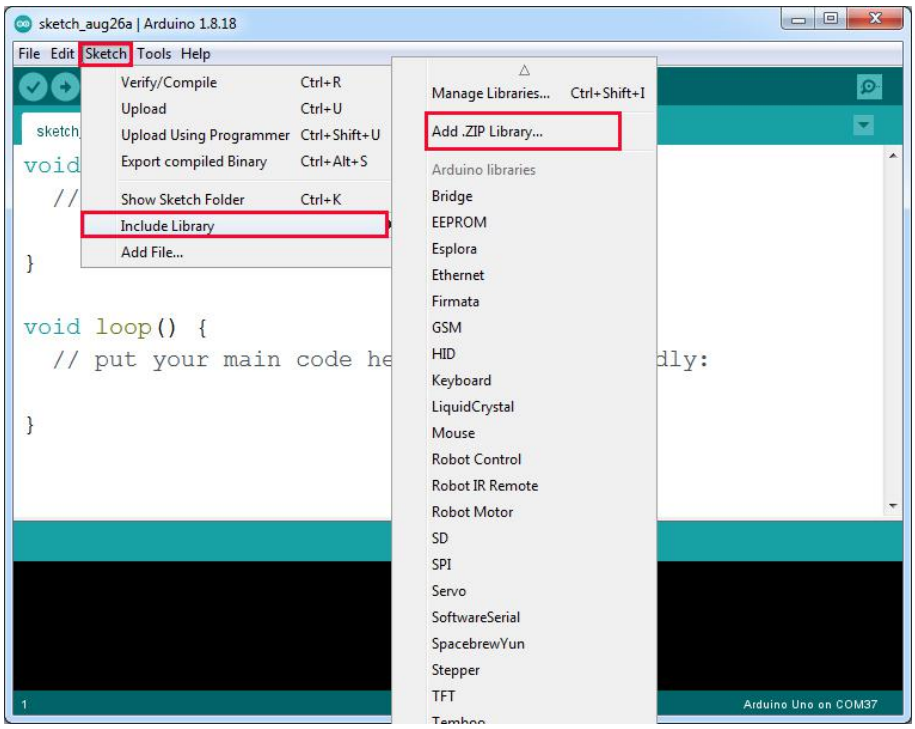
Note:This lesson use Arduino IDE as programming tool.

Step 1: Install Arduino IDE. Download Arduino IDE from https://www.arduino.cc/en/Main/Software?setlang=en , then install the software. If you have install Arduino IDE, please skip this step.



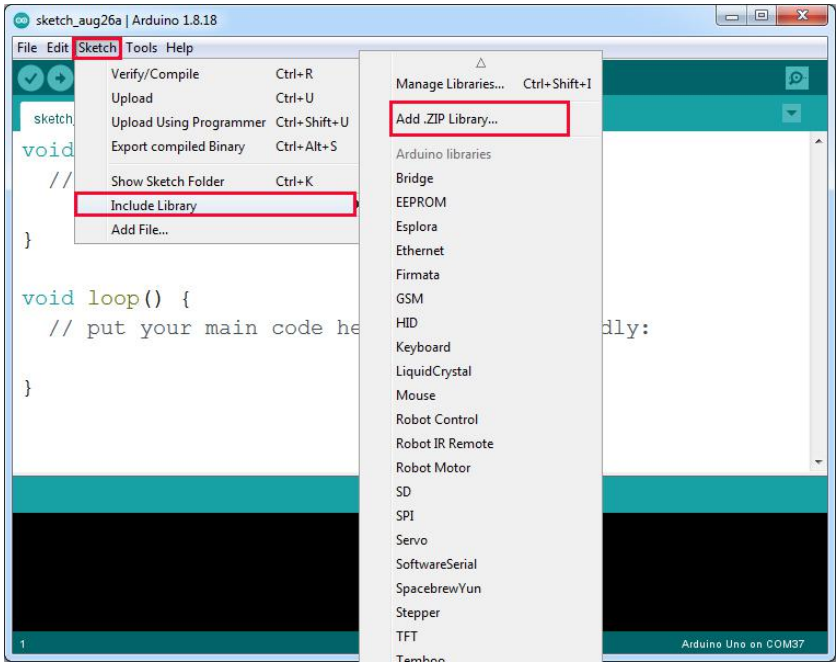
Step 2: Install **WifiEsp-master library** into IDE (If you have already installed WifiEsp-master library, please skip this step)

Download **WifiEsp-master library** from https://osoyoo.com/driver/WiFiEsp-master.zip, then import the library into IDE (Open IDE-> click Sketch->Include Library->Add .Zip Library)



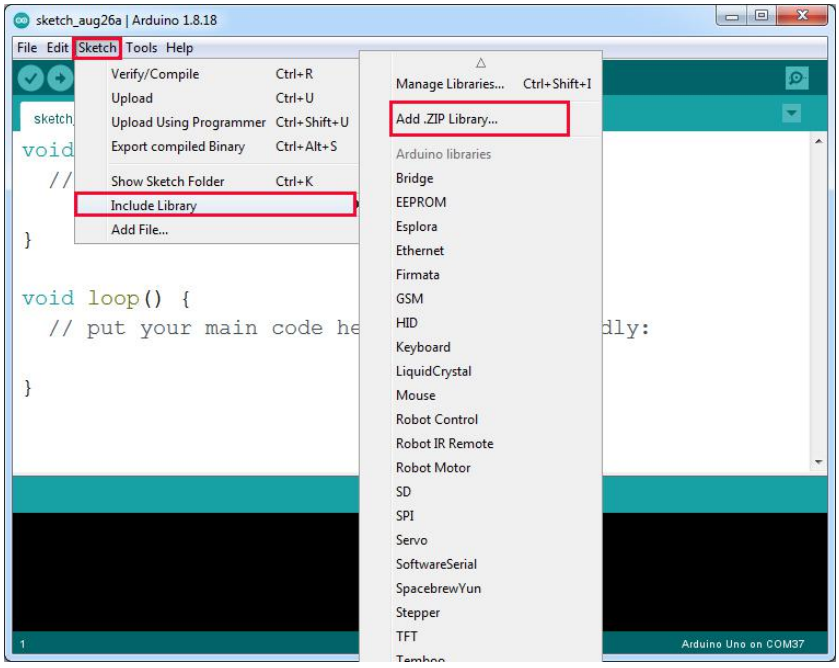
Step 3: Install **OsoyooIOT Library** into IDE (If you have already installed OsoyooIOT Library, please skip this step)

Download **OsoyooIOT Library** from https://osoyoo.com/driver/wifi-iot/OsoyooIOT.zip, then import the library into IDE (Open IDE-> click Sketch->Include Library->Add .Zip Library)



Step 4: Install **DHT Sensor Library** into IDE (If you have already installed DHT Sensor Library, please skip this step)

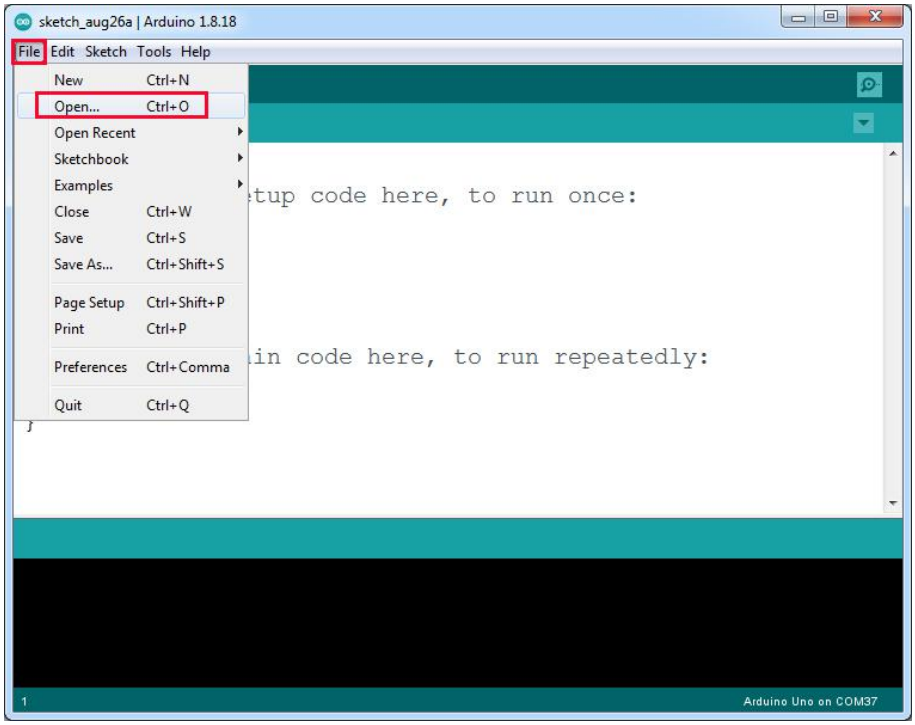
Download **DHT Sensor Library** from https://osoyoo.com/wp-content/uploads/samplecode/DHT.zip then import the library into IDE (Open IDE-> click Sketch->Include Library->Add .Zip Library)



Step 5: Download sample code

from https://osoyoo.com/picture/Building_Robot_Car/lesson7/wifi-block-car.zip, and unzip the downloaded zip file **wifi-block-car.zip**, you will see a folder called **wifi-block-car**.

Step 6: Connect OSOYOO basic board for Arduino to PC with USB cable, Open Arduino IDE -> click file -> click Open -> choose code "**wifi-block-car.ino**" in **wifi-block-car** folder, load the code into Arduino. (Notice: Please turn off power when your Robot is connected to Personal Computer or Laptop via USB cable)

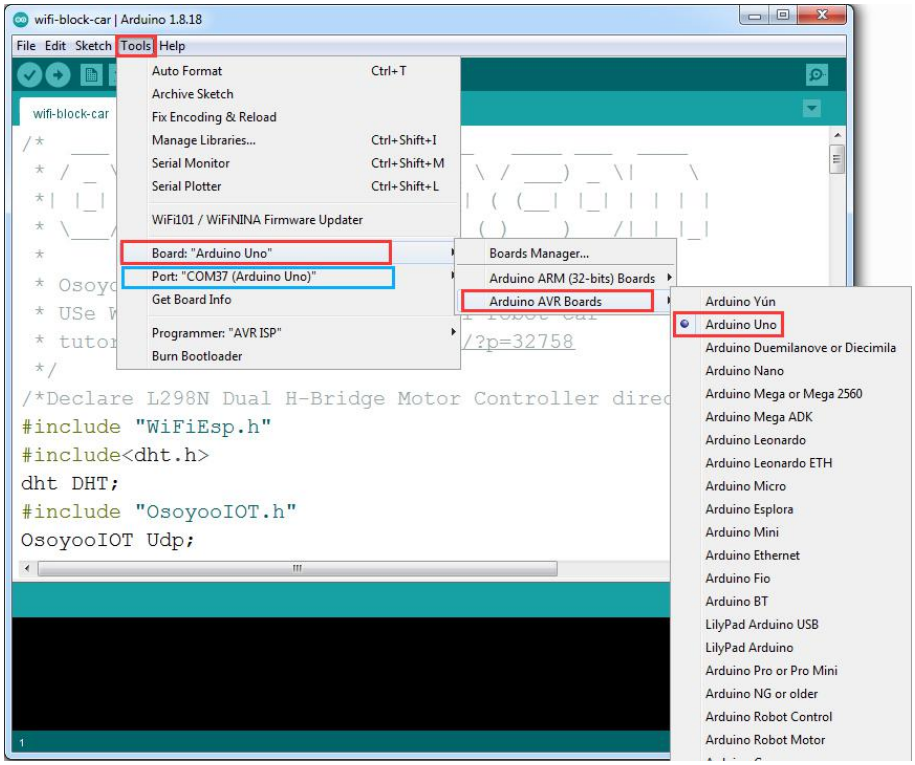


Step 7: You need change the code Line 38 and Line 39 as your router wifi ssid name and password :

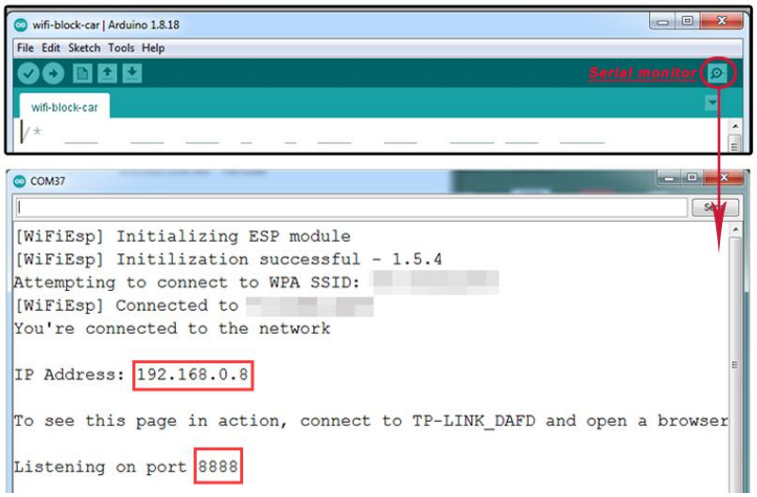
char ssid[] = "***"; // replace *** with your router wifi SSID (name)

char pass[] = "***"; // replace *** with your router wifi SSID (password)

Step 8: Choose corresponding board and port for your project, upload the sketch to the board.



Step 9: click the Serial monitor window in upper right corner of IDE, you will see following result (Note: your PC and the robot are connected with the same LAN network):



TIPS:

If you find the speed for the car is slow or fast, please open the code and change the value of line33 to line35 as the following:

#define FAST_SPEED 180

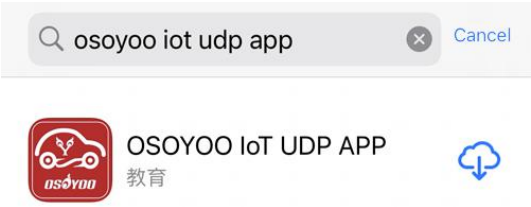
#define MID_SPEED 130

#define SPEED 120

HOW TO PLAY

APP Installation and Setting:

Step 1) Download **OSOYOO Iot udp APP** by searching **OSOYOO Iot udp APP** from Apple Store(iPhone/iPad) or Google Play Store(Android device).



Step 2): Install the **OSOYOO Iot udp APP** in your Phone and make sure your Cell phone is connected with **the same LAN Network** of your robot car for Arduino.

Step 3): Set IP Address in APP

1)Open APP, click Setting button in upper right corner

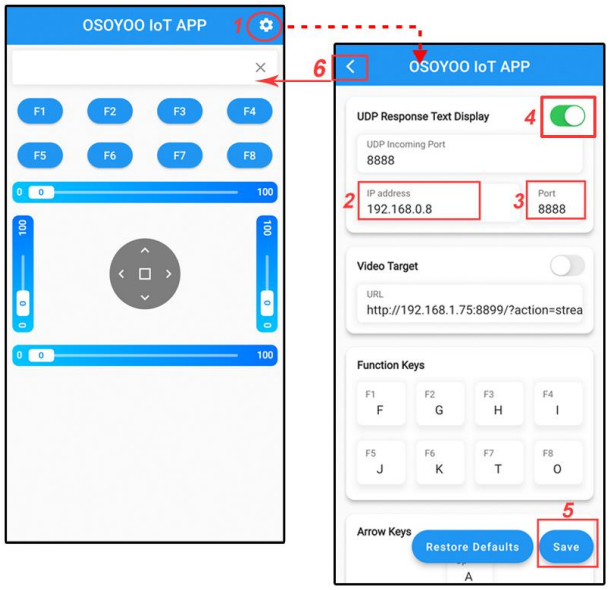
2)Use the IP address you get in serial monitor window (For our robot is 192.168.0.8) to replace default IP 192.168.1.255

3)keep default port number 8888 without changing

4)turn on the switch of UDP Response Text Display

5)Click **Save** button to save the changes you just made

6)Click Back Arrow to go back APP front UI



Control the robot car:

You can use the Arrow keys to control the car movement and see the temperature and humidity detected by the DHT11 sensor in your car when you click any button.

▲	Move Forward
▼	Move Backward
◀	Left Turn
▶	Right Turn
[]	Stop