# Chapter 8
## Car Reversing Helper

To help drivers back up and park more safely and easily, there are a lot of car reversing radar alarming systems emerging on the market. Based on ultrasonic measurement principle, the reversing radar system can detect the distance between the car and obstacle, then the drivers can be alerted by beeps or the dashboard display, which could be very helpful for new drivers.
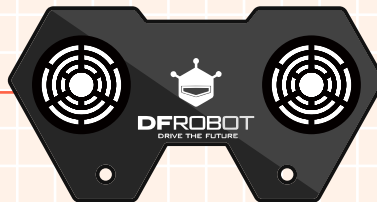
## Goal

1.Learn how to use variables
2.Learn how to embed a condition block inside another one.

## Electronic Component

### Ultrasound Brief

Ultrasonic Sensor

The transmitter sends out ultra-sound, and when hitting the object, the ultrasound reflects as echo and will be sensed by the receiver.

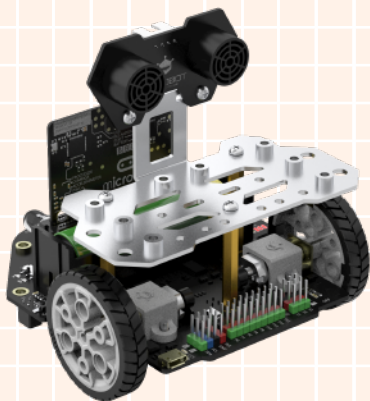## Command Learning

### Block Brief

Variable

distance ▼

Changeable value: number, string, list

Assign a value to variable

set   distance ▼   to   0

Assign a value to the variable.
Default setting:  distance=0

## Hands-on Practice

In this project, our aim is to measure the distance between the end of the robot and obstacles, so we need to inversely install the ultrasonic sensor onto the expansion bracket, as shown below.
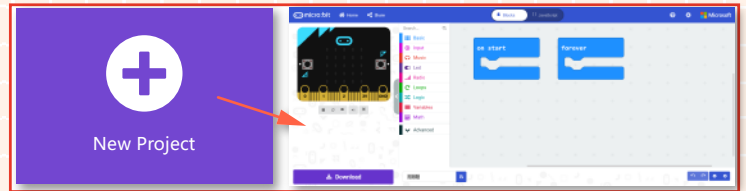
### Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click "new project" to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
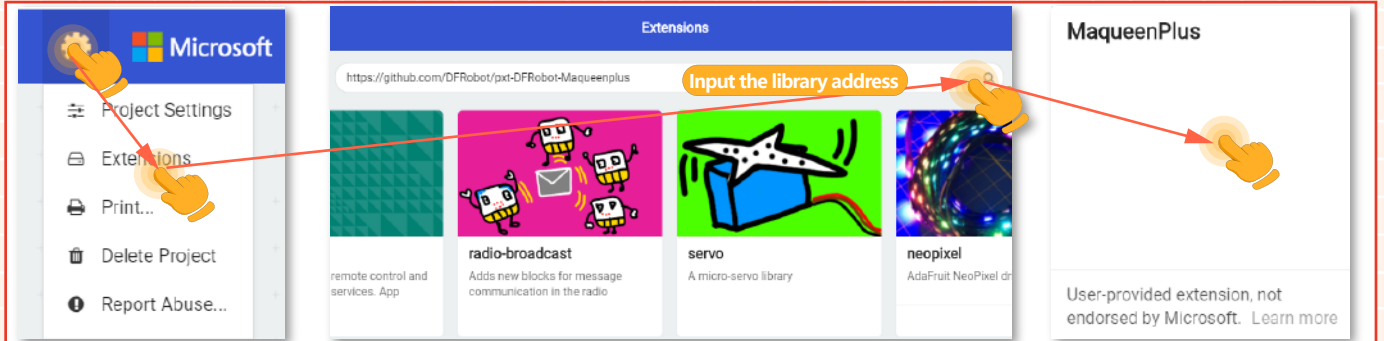
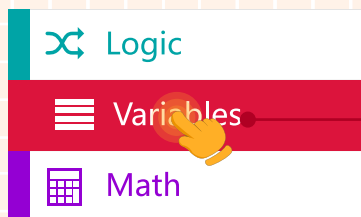https://makecode.microbit.org/

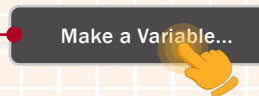1.Enter MakeCode editor

2.Enter programming interface

3.Add the extension library

## Step 2 Programming

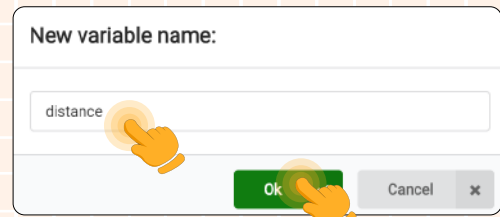1.Create a variable and name it as "distance".



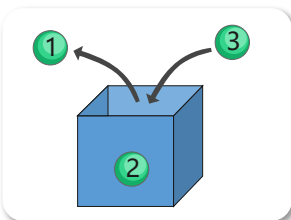1.Click "Variable" in the command block section

2.Click "Make a variable"

New variable name:

distance

3.Name the variable as "distance"
4.Click "OK".

### Knowledge Expansion

**What is a variable?**

We may think of a variable as a container or box where we can store data that we need to use later, and each box can only hold one value (number, text and Boolean data) at a time. For example, use it to store an integer, after we put 1into it, we put 2 into it, then we can only get 2 from this box. The name of the box is the variable's name and the value of the variable is placed inside the box.

2.In this project, we need to monitor the distance value the sensor detected in real-time, so we have to assign the value to the variable "distance".

set distance to read ultrasonic sensor T P1 E P2 cm
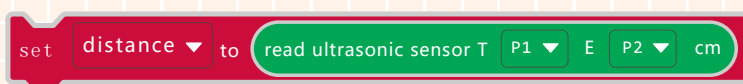
### Knowledge Expansion

distance

Then we can directly use the variable "distance" to call the distance value later.

3.Once the distance between the car and the obstacle is smaller than the preset value (distance <30cm in the example), the car stops. If the distance is long enough, the car goes backward, and the buzzer keeps beeping.

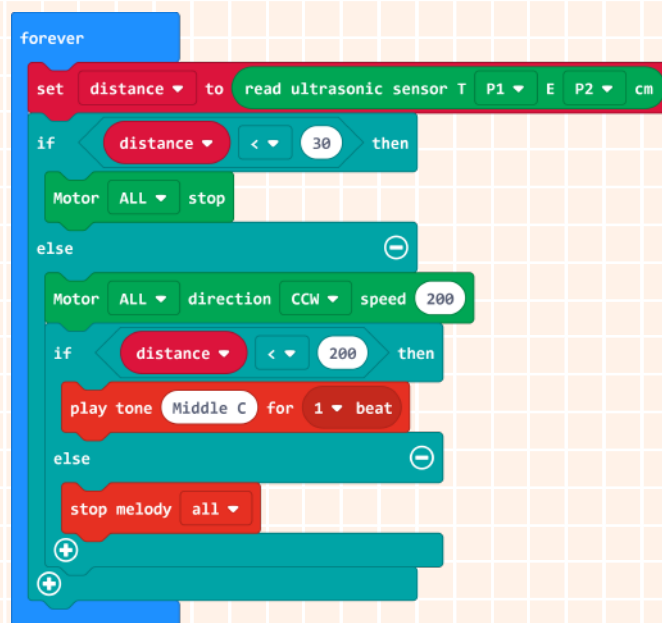Drawing a flowchart is gonna help us understand the program.
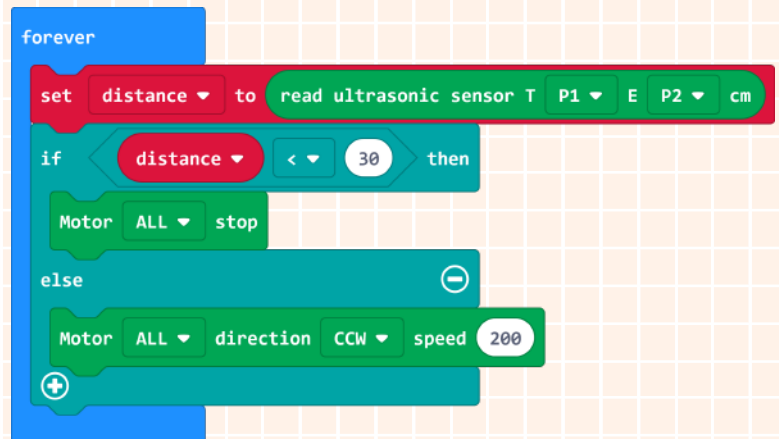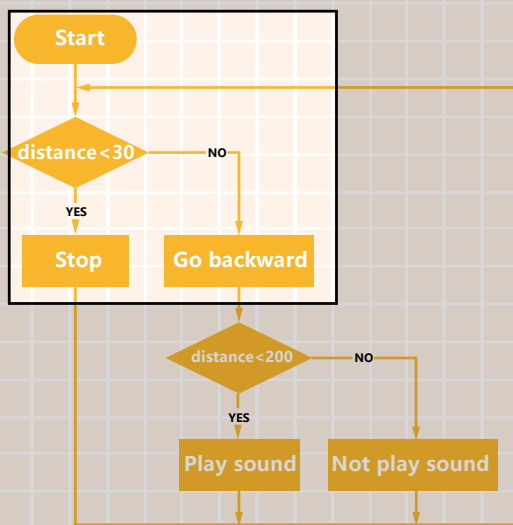


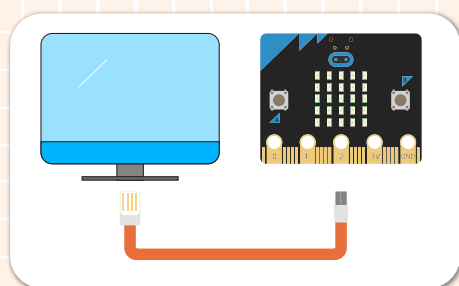4.Complete part of the program: car stops and goes backward.
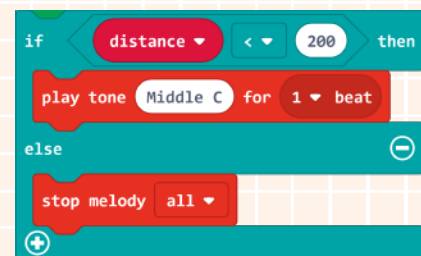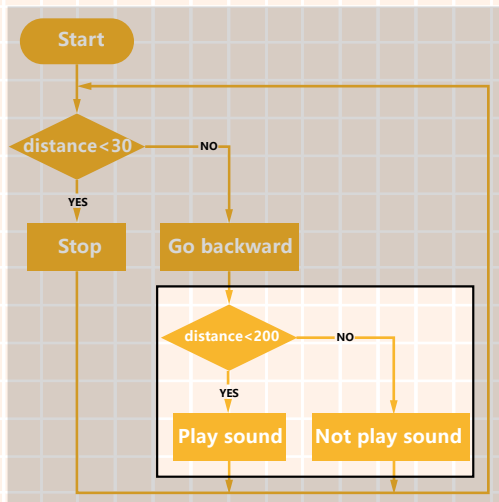


Only when the car is going backward will the buzzer start making sound, so the condition block "distance<200" should be put under the car backward block.
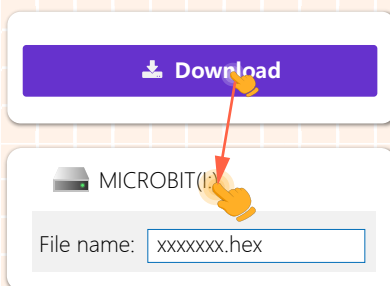
5.The complete program is shown below.

6.Name your project as "Car Reversing Helper" and save it.

## Step 3 Download Program
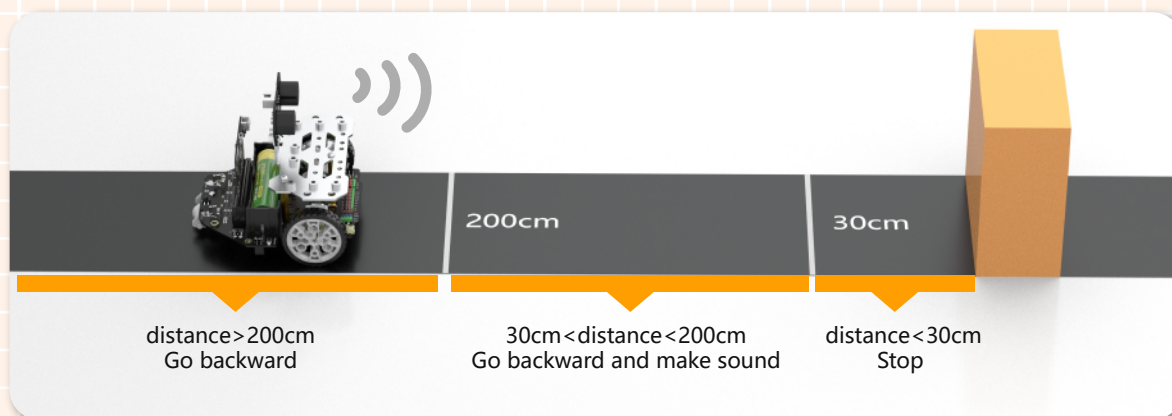


1.Connect to computer

2.Download program

3.Plug in micro:bit

## Step 4 Effect Display

After we completed all the above steps, put an obstacle behind the Maqueen Plus car, turn on the power switch. When the detected distance is smaller than 200, the car starts going backward; when the distance is between 30 to 200, the buzzer keeps beeping; distance <30, the car stops.



| 200cm | 30cm |
|---|---|

distance>200cm
Go backward

30cm<distance<200cm
Go backward and make sound

distance<30cm
Stop

# Think & Explore ❓

We have made a car reversing helper with Maqueen Plus, the car will stop when it is very near an obstacle, but can you let the car bypass it? Make an obstacle avoidance car according to the flowchart below.



Start

distance<30 — NO

YES

turn left/right          go forward
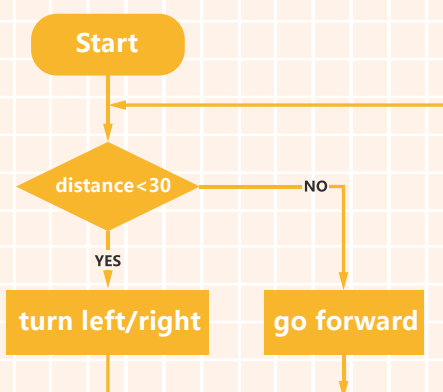
Tip: the ultrasonic sensor should point to the direction the car moves toward.