

# Experiment 20 – Car remote control

## Instruction

Just like our brains, controllers play a very important role in remotely controlled cars. It is responsible for telling the car what to do and how to do it. Why not try to use a micro:bit as a controller to simulate how the car is controlled, such as lights, horns, steering, etc. This is what we will do in this experiment.

## Target

- Make a remote controller and simulate controlling the car.
- Learn how to use gesture blocks to detect gestures you take when holding a micro: bit (such as shaking a micro: bit)

## Required Parts

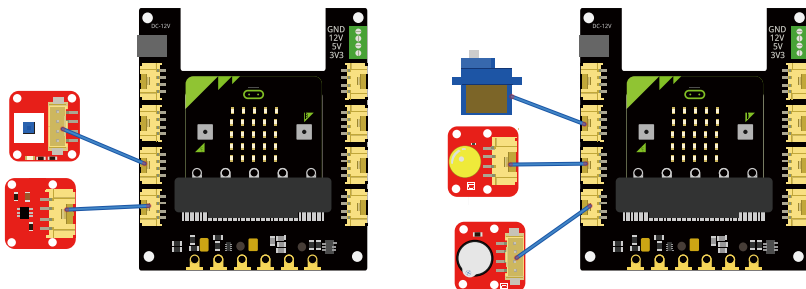
- Micro:bit x2
- Crowtail-Base shield for Micro:bit x2
- Crowtail-Touch Sensor x1
- Crowtail-Switch x1
- Crowtail-LED(Yellow) x1
- Crowtail-Buzzer x1
- Crowtail-9G Servo x1
- Crowtail-Cable x4
- USB cable x1

## Hardware learning and connection

The circled position in the picture is the accelerometer on the micro: bit. We can use this accelerometer to measure micro: bit acceleration and detect micro: bit movement. It can also detect other micro: bit gestures, such as shaking, tilting, and free fall.



Connect Crowtail-Touch Sensor and Crowtail-Switch to P0 and P1 ports of Crowtail-Base shield for Micro:bit(named bit1). Connect Crowtail-Buzzer, Crowtail-LED and Crowtail-9G Servo to P0, P1 and P2 ports of Crowtail-Base shield for Micro:bit(named bit2). **Note: You need to connect 12V power supply for bit2.** The hardware connection are as follows:



## Programming and note

Send\_code(bit1):

```
on start
  radio set group 10

on logo down
  radio send number 1

on tilt left
  radio send number 2

on tilt right
  radio send number 3

forever
  if digital read pin P0 = 1 then
    radio send number 4
  if digital read pin P1 = 1 then
    radio send number 5
  else
    radio send number 6
```

Receiver\_code(bit2):

```
on start
  radio set group 10
  digital write pin P1 to 0

on radio received receivedNumber
  if receivedNumber = 1 then
    servo write pin P2 to 90
    show leds
  else if receivedNumber = 2 then
    servo write pin P2 to 0
    show leds
  else if receivedNumber = 3 then
    servo write pin P2 to 180
    show leds
  else if receivedNumber = 4 then
    start melody ba ding repeating once in background
  else if receivedNumber = 5 then
    digital write pin P1 to 1
  else if receivedNumber = 6 then
    digital write pin P1 to 0
```

- **On Gesture:** On logo down, on tilt left and on tilt right these three blocks are all called On gesture block. This type of block will start an event handler(part of the program that will run when something happens). This handler works when you do a gesture, for example, shaking the micro:bit. In send\_code, when bit1 logo down, we send 1; when bit1 tilts to the left, we send 2; when bit1 tilts to the right, we send 3.
- **Show leds:** This block is to shows a picture on the LED screen. There are 4 \* 4 LEDs, you can control any one of the LEDs above to draw your own picture. Here the picture we need to draw is 3 direction arrows.
- **If...else if...:** We use this statement to run the corresponding code based on different numbers received by radio.

## Blocks Overview

### Send\_code(bit1):

1. Set radio group for micro:bit.
2. Send corresponding code based on different gesture detected.
3. If the touch sensor is touched(1), send number 4.
4. If the switch is pressed(1), send number 5.
5. If the switch pops up(0), send number 6.

### Receiver\_code(bit2):

1. Set radio group of micro:bit.
2. If the number received is 1, the servo shaft rotates to 90 degrees and a forward picture is displayed on the micro:bit matrix LEDs.
3. If the number received is 2, the servo shaft rotates to 0 degrees and a left picture is displayed on the micro:bit matrix LEDs.
4. If the number received is 3, the servo shaft rotates to 180 degrees and a right picture is displayed on the micro:bit matrix LEDs.
5. If the number received is 4, play a melody in background.
6. If the number received is 5, turn LED on.
7. If the number received is 6, turn LED off.

- **Download the send\_code program to bit1 and receiver\_code program to bit2 and see what happens.**

### Result

When the bit1 logo tilts down, the matrix LED on bit2 displays the forward picture and the servo rotates to 90 degrees; when bit1 tilts to the left, the matrix LED on bit2 displays the left picture and the servo rotates to 0 degrees; when bit1 tilts to the right, the matrix LED on bit2 displays the right picture and the servo rotates to 180 degrees; when you press the touch sensor connected to bit1, the buzzer connected to bit2 will play the melody. When you press the switch connected to bit1, the LED connected to bit2 will light up. When you pop the switch, the LED will go out.



*Try to use what you have learned in this project to make a real remote control car!*

