

kitronik ultrasonic sensor + servo motor

# Introduction

In this lecture we will get acquainted with Kitronik :move motor extension for mocro:bit. How to use servo motor for destroyed an obstacle that interferes with driving.

# Necessary:

* Micro:bit controller
* Kitronik :MOVE motor kit
* USB cable
* Micro:bit program or internet link in which to do programming

# Process!

1. At **kitronik :move motor kit glue the servo** motor with double-sided adhesive tape, another method can be used (improvise).

A small yellow robot with black wheels and wires

Description automatically generatedA small yellow and black robot with a black antenna

Description automatically generated

1. **Connect** the **servo** to the **connectors provided for it.** If you use the one on the reverse side, then make changes to the code.

A yellow wheel with black wires

Description automatically generated

1. **Add** some **part** **to the servo** motor, you can screw it up or glue it with adhesive tape to the original wings of the servo or make a pantry to the gear, depending on what options are available to you. It can be from any material.
2. **Open the folder** **where** you **saved the code** for the **previous lesson**. **Reduce** the **folder** and mocro:bit programming **platform** so you **can see both** at the same time. With the mouse, **take** the **file** from the folder and **drag** it into the **programming window**, the previous code will open there.

A screenshot of a computer

Description automatically generated

1. Now your code will look like this:

A screenshot of a computer

Description automatically generated

1. To **add** a **servo action,** we create a **function servoBlow.**
2. From **loops** insert **repeat 3 times** in **function**. It will repeat the action three times, you can choose a different number of repetitions
3. From **Pins** take **servo write pin P0 to 0**. **Look**! at the chip at what pin your servo motor is connected to (P15 or P16), **enter** it into the code instead of **P0**. **Enter** the **starting position** of the servo **in degrees**, it will depend on the existing state of your servo and the condition of the connected part. Choose an **angle** so that the **part is raised**.
4. Insert **pause** in (ms) , You can experiment with time and choose the most suitable
5. Again from **Pins** take **servo write pin P0 to 0**. **Enter** the **same** **pin** into the code instead of **P0**. **Enter** the **end position** of the servo **in degrees**, it will depend on the existing state of your servo and the condition of the connected part. Choose an **angle** so that the **part is down**.
6. Insert **pause** in (ms) , You can experiment with time and choose the most suitable
7. **In** your **existing code** after first stop insert **pause (ms) 500** and from **function** insert **call servoBlow**.

A screenshot of a computer program

Description automatically generated

!!! Warning !!! The servo motor has limited power abilities, so choose an area with soft toy obstacles or adjust the angles of position of the motor so that it does not have an excessive load when trying to perform the action. Overload, the motor may burn out

1. **Download** in micro:bit controller

Python code:

**A screenshot of a computer program

Description automatically generated**

# Challange:

Create a code in which to add the sound of beats during the time the servo makes beats.