

Sound gradient

Hardware requirements: mBot/mCore Control Panel

Implementation: Upload the program to Arduino

Example programs



Script description

In our life, we often hear some regular frequency sounds, like the sound when we pour water into a bottle, when we can hear the tone rising. If the tone is changing between some frequencies (such as an ambulance, which has only two frequencies, high and low), we can directly define the frequency value in the tone-playing block. But if the frequency magnitude is small but its range is wide, this method will cause a very long program. So we can use a “variable” block to make a program where these values vary regularly.



Define Variable  to the frequency value of a tone and set the duration to 50ms.



Define the initial value of Variable  is 500



Set the varied value of Variable  and increase it by 20 each time



Set the number of times when Variable  changes to 50

Knowledge points

Point 1 How to define Variable

in **Data&Blocks** the block area. Click **Make a Variable** to create a variable free. Enter the name of the variable and create it, 4 blocks relating to it will appear automatically, as shown in the following picture.



The first two blocks are used to define the variable value and the varied value of the variable. The other two blocks are mostly used for stage creation model.

Also on the left upper corner of the mBlock stage, there will be **f 0**, where the number is the variable value. This is to facilitate users' observation of the variable value changing.

Point 2 Use of Variable Block

Variables, as we know, are values changing when a program runs. A variable has a name and a value. In mBlock, we can image that a variable is a box containing data. A program can store and extract the data within the box. The name of the box is "variable name" and the data in the box is "variable value". When you create a variable, the box is empty. When you set a variable value, you fill the box with data. When you change the value you take the data out and put into new data.

The logic of this example is as follows:

- 1) Define and create a variable **f**.
- 2) Use the block **set f to 500** to set the initial value of **f** to 500.
- 3) Use the block **play tone on note f beat 50** to define Variable **f** to the frequency value of the buzzer.
- 4) Use Block **change f by 20** to increase the value of Variable **f** by 20 each time, i.e. the frequency of the buzzer increases by 20Hz each time.
- 5) Finally use **repeat 50** to limit the number of changes of **f** to 50, and the final

value of **f** is $500+20*50=1500\text{Hz}$.

The buzzer sound effects of this example is: It first sounds 500Hz and then the frequency increases by 20Hz each time, i.e. the buzzer sounds respectively 500Hz,

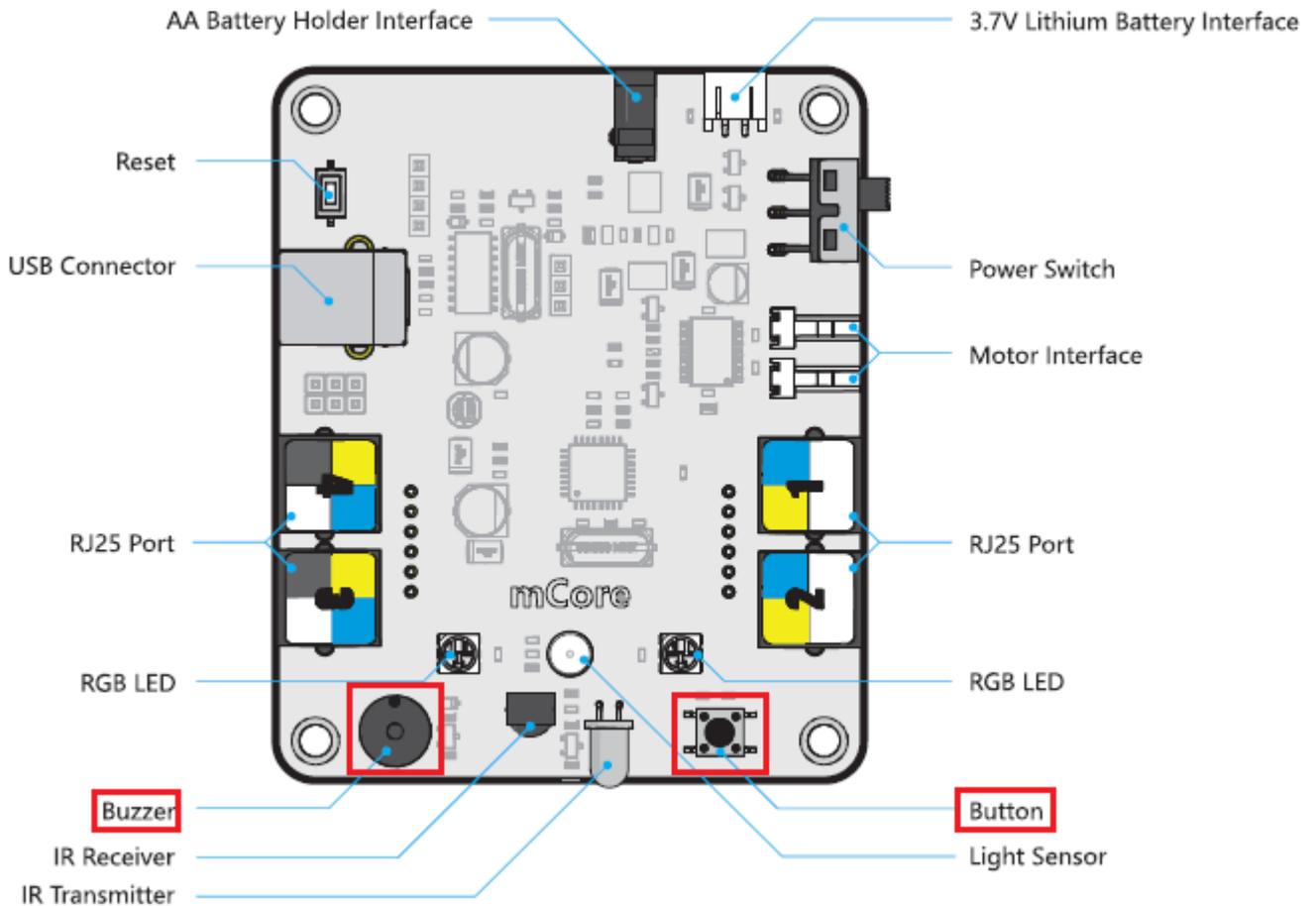
520Hz, and 540Hz ... (you can observe the change of the value in **f 0** under online debugging status), and each sound lasts for 50ms. It will stop sounding upon the 50th frequency increase.

Note: If change the value in **change f by 20** to -20, the value of **f** will be decreased by 20 each time.

Extended tasks

Task 1	Let the buzzer play a frequency-decreasing sound
Task 2	Please think what frequency of the last sound made by the buzzer is in this example?

Attached -mCore main control board buzzer and onboard button diagram



Download: [Sound gradient.sb2](#)