Sound and Light Gradient

Time required: 45 minutes

Please read all the directions carefully before beginning the assignment.

- Comment your code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Understanding

Demonstrate understanding of:

libraries, functions, loops, variables

Requirements

1. Use variables

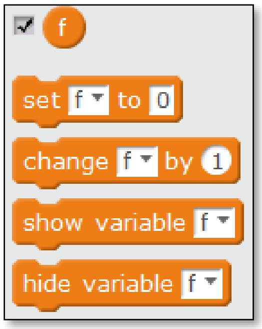
Tutorial Assignment

This assignment experiments with loops, variables, and changing values in the variables.

This assignment introduces variables and Blocks. Modular code is easier to maintain. Blocks are under My Blocks.

Define a Variable

Go to **Variables**. Click Make a Variable to create a variable. Enter the name of the variable and create it. 4 blocks relating to it will appear automatically, as shown in the following picture.



☑ f

set f ▾ to 0

change f ▾ by 1

show variable f ▾

hide variable f ▾

The first two blocks are used to define the variable value and the varied value of the variable. The other two blocks are rarely used.

On the left upper corner of the mBlock stage, there will be

, where the number is the variable value. This is to facilitate users' observation of the variable value changing.

Use of Variables

Variables are values that change when a program runs. A variable has a name and a value.

In mBlock, we can imagine 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. Use repeat 50 to limit the number of changes of f to 50, and the final value of f is 500+20*50=1500Hz.

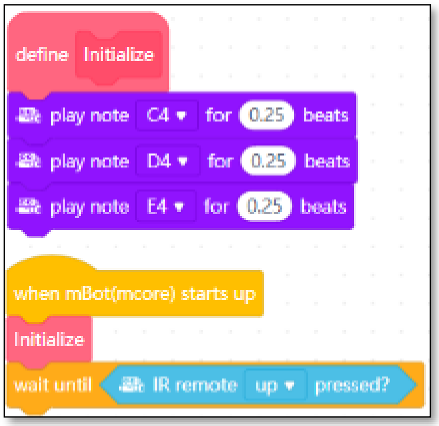
The buzzer sound effects of this example is: It first sounds 500Hz and then the frequency increases by 20Hz each time, i.e. the buzzer sounds respectively 500Hz, 520Hz, and 540Hz … (you can observe the change of the value in f 0 under online debugging status), and each sound lasts for 50ms. It will stop sounding upon the 50th frequency increase.

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

Tutorial Assignment

Create and Use a Block

A Block is a chunk of modular code that can be reused in the program without having to write the code again and again.



```
define Initialize
play note C4 ▾ for 0.25 beats
play note D4 ▾ for 0.25 beats
play note E4 ▾ for 0.25 beats

when mBot(mcore) starts up
Initialize
wait until < IR remote up ▾ pressed? >
```

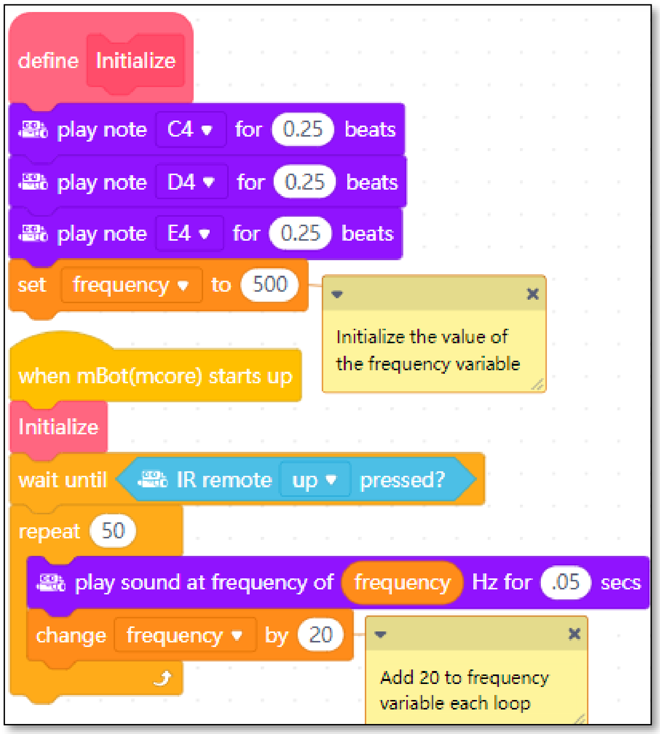
1. Start mBlock and save the program as **Sound Gradient**.
2. Go to **My Blocks**. Click **Make a Block**.
3. Name the block **Initialize**. Click OK.
4. Use the **define** block to create the code for the block. In this example the block is called **Initialize**.
5. Drag the other part of the block as shown in the example.
6. Upload and test the program. You should hear the three notes.
7. Turn the mBot off and then on. You should hear the three notes indicating the mBot is ready to go.

Sound Gradient

The sound starts at a frequency of 500 hertz. Each time through the repeat loop the frequency increases by 20 hertz.

NOTE: Only use letters to start a variable or block name. Don’t use spaces in a variable or block name.

1. Create and test the program as pictured.



```
define Initialize
play note C4 for 0.25 beats
play note D4 for 0.25 beats
play note E4 for 0.25 beats
set frequency to 500

when mBot(mcore) starts up
Initialize
wait until IR remote up pressed?
repeat 50
    play sound at frequency of frequency Hz for .05 secs
    change frequency by 20
```

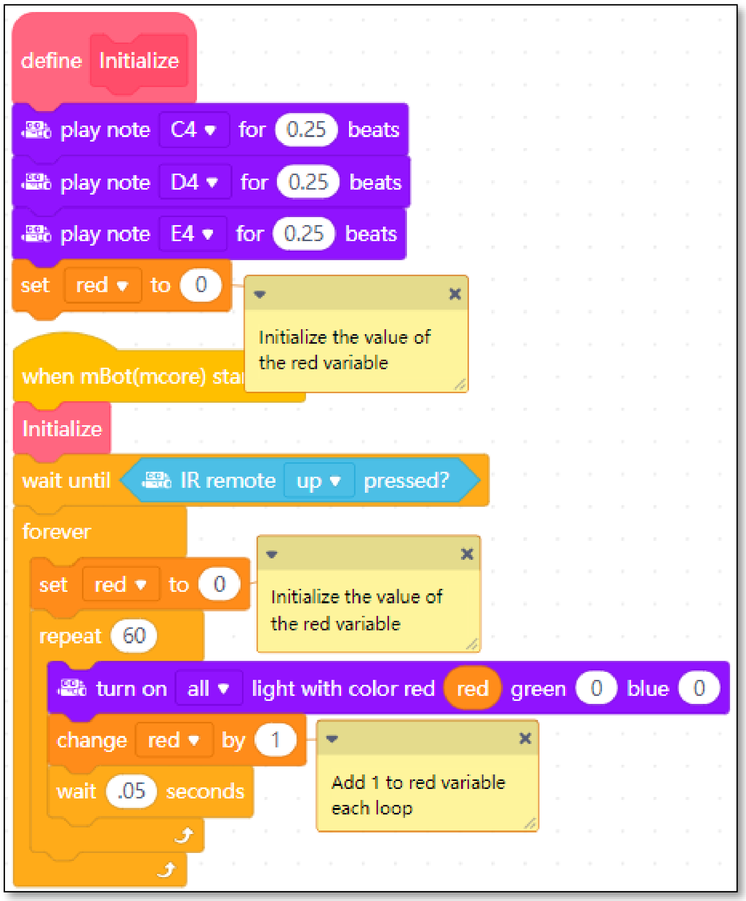
Initialize the value of the frequency variable

Add 20 to frequency variable each loop

Brightness Gradient

The onboard LED lights start dark and the brightness increases gradually. When the brightness reaches a certain value, the onboard LED turns off. Repeat this process.

1. Save this program as **Brightness Gradient**.
2. Complete and test the program as pictured with the requirements listed.



```
define Initialize
play note C4 for 0.25 beats
play note D4 for 0.25 beats
play note E4 for 0.25 beats
set red to 0

when mBot(mcore) sta
Initialize
wait until IR remote up pressed?
forever
    set red to 0
    repeat 60
        turn on all light with color red red green 0 blue 0
        change red by 1
        wait .05 seconds
```

Initialize the value of the red variable

Initialize the value of the red variable

Add 1 to red variable each loop

Requirements

- Complete and successfully run the programs as displayed.

Assignment

Start with your tutorial project and add the following.

- Change either program to loop until a different condition is met.
- Change something else based on a variable.

Assignment Submission

- **All students** → Attach finished programs to the assignment in Blackboard.
- **In class assignment submission** → Demonstrate in person.
- **Online submission** → A link to a YouTube video recording showing the assignment placed in the submission area in BlackBoard.