Project 18: Play Music

Overview

In this project, you will learn how to play music with keyestudio passive buzzer module. We are going to complete two experiments.

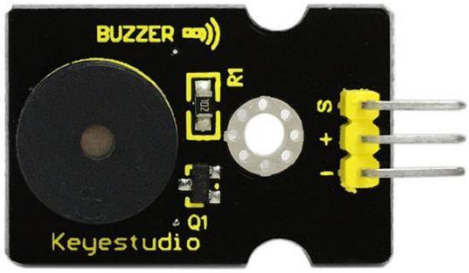
One is to directly control the High and Low level input of micro:bit P0 end, set two square waves to control the buzzer sound. The other is to use the software's own function, input the square waves of different frequencies and different lengths on the P0 end. Finally make the buzzer module play the song "Ode to Joy".

(The input PIO port can only be P0, can not be other interfaces).

Component Required:

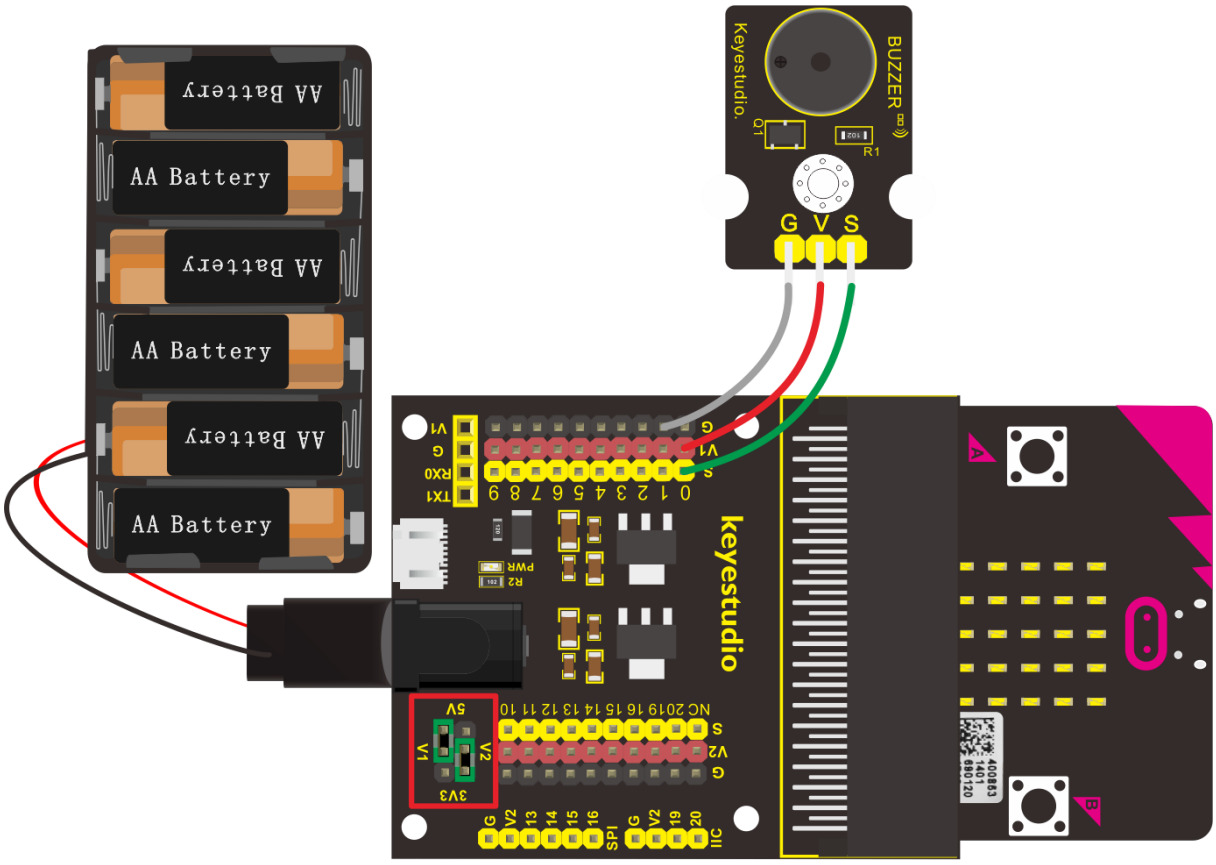
- Micro:bit main board*1
- Keyestudio Micro bit Sensor V2 Shield*1
- USB Cable*1
- keyestudio Passive Buzzer Module*1
- Dupont jumper wire*3
- Premium Battery Holder 6-cell AA*1
- 1.5V AA Battery*6

Passive Buzzer Module:



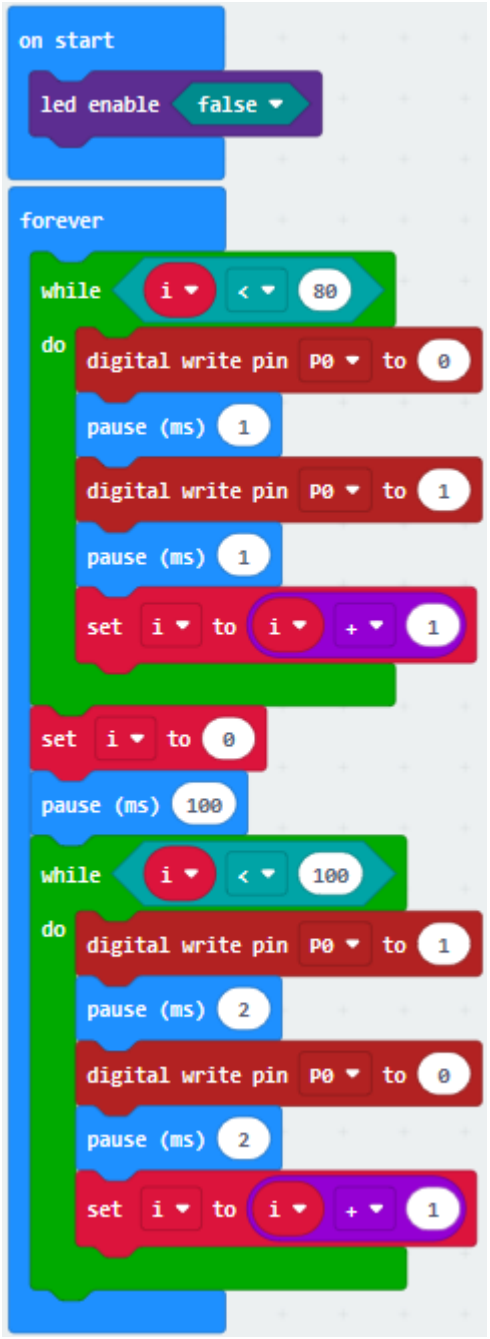
Buzzers can be categorized as active and passive ones. The difference between the two is that an active buzzer has a built-in oscillating source, so it will generate a sound when electrified. The buzzer used on this module is a passive buzzer. A passive buzzer does not have such a source, so DC signal cannot drive it beep. Instead, you need to use square waves whose frequency is between 2K and 5K to drive it. Different frequencies produce different sounds. You can use micro:bit to code the melody of a song, quite fun and simple.

Connection Diagram



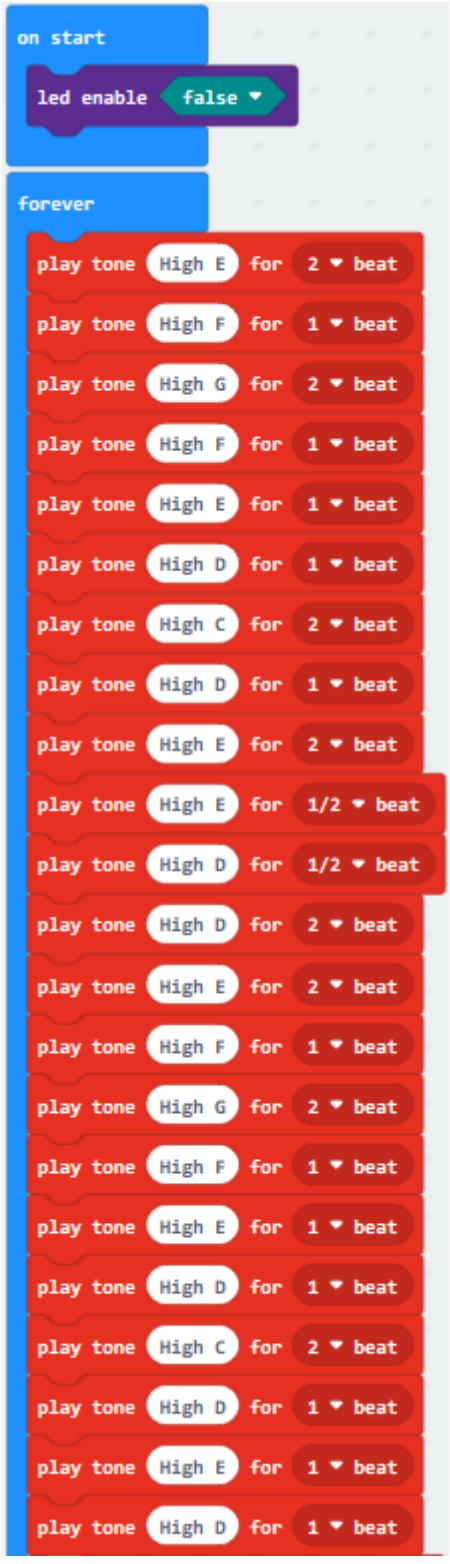
Test Code

Program 1:



"on start" : command block only runs once to start program.
Turn off LED matrix
The program under the block " forever" runs cyclically.
When i<80, run the program under do block
Set P0 to low level (0) to stop passive buzzer from emitting sound.
Delay in 1ms。
Set P0 to high level(1) to make passive buzzer emit sound.
Delay in 1ms
Set variable i to value of i +1
Set variable i to 0
Delay in 100ms
When i<100, run the program under do block
When i<100, run the program under do block
Set P0 to high level(1) to make passive buzzer emit sound.
Delay in 1ms
Delay in 2ms
Set P0 to low level (0) to stop passive buzzer from emitting sound.
Delay in 2ms
Set variable i to value of i +1

Program 2:



"on start" : command block only runs once to start program.

Turn off LED matrix

The program under the block "forever" runs cyclically.

Play tone high E for 2 beats

Play tone high F for 1 beat

Play tone high G for 2 beats

Play tone high F for 1 beat

Play tone high E for 1 beat

Play tone high D for 1 beat

Play tone high C for 2 beats

Play tone high D for 1 beat

Play tone high E for 2 beats

Play tone high E for 1/2 beat

Play tone high D for 1/2 beat

Play tone high D for 2 beats

Play tone high E for 2 beats

Play tone high F for 1 beat

Play tone high G for 2 beat

Play tone high F for 1 beat

Play tone high E for 1 beat

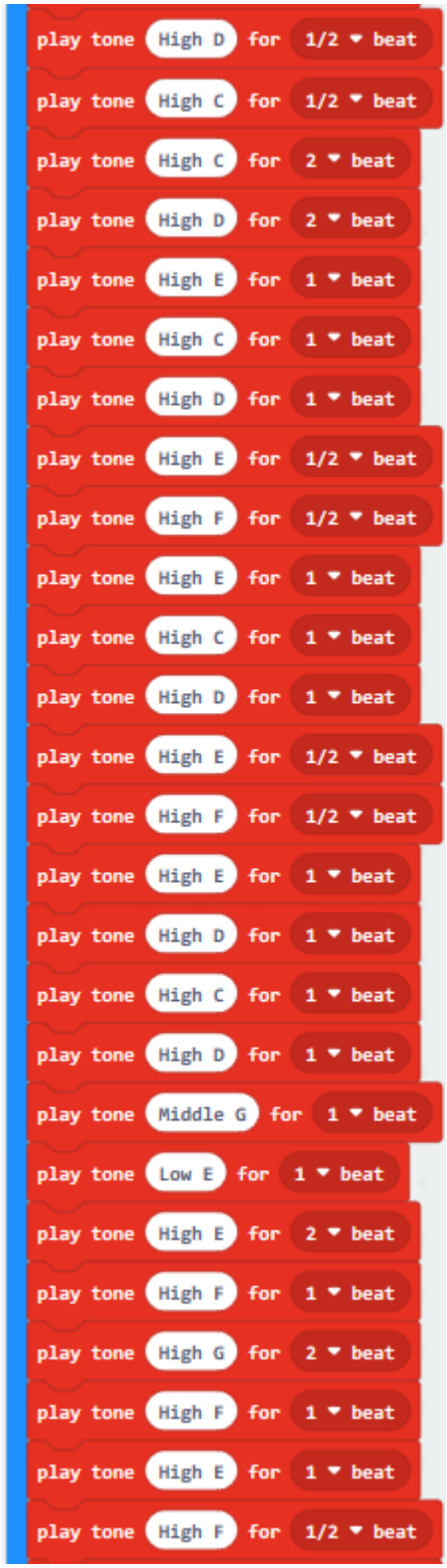
Play tone high D for 1 beat

Play tone high C for 2 beats

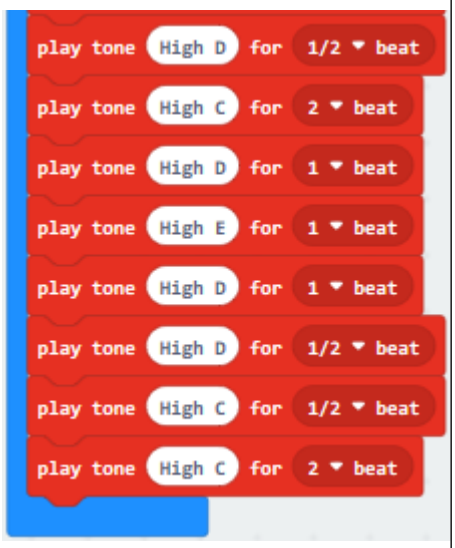
Play tone high D for 1 beat

Play tone high E for 1 beat

Play tone high D for 1 beat



Play tone high D for 1/2 beat
Play tone high C for 1/2 beat
Play tone high C for 2 beats
Play tone high D for 2 beats
Play tone high E for 1 beat
Play tone high C for 1 beat
Play tone high D for 1 beat
Play tone high E for 1/2 beat
Play tone high F for 1/2 beat
Play tone high E for 1 beat
Play tone high C for 1 beat
Play tone high D for 1 beat
Play tone high E for 1/2 beat
Play tone high F for 1/2 beat
Play tone high E for 1 beat
Play tone high D for 1 beat
Play tone high C for 1 beat
Play tone high D for 1 beat
Play tone high G for 1 beat
Play tone high E for 1 beat
Play tone high E for 2 beats
Play tone high F for 1 beat
Play tone high G for 2 beats
Play tone high F for 1 beat
Play tone high E for 1 beat
Play tone high F for 1/2 beat



Play tone high D for 1/2 beat

Play tone high G for 2 beat

Play tone high D for 1 beat

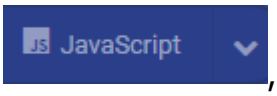
Play tone high E for 1 beat

Play tone high D for 1 beat

Play tone high D for 1/2 beat

Play tone high C for 1/2 beat

Play tone high C for 2 beats

Note: on the MakeCode Block webpage, click the icon 

, you can see the frequency of each tone as follows.


```
led.enable(false)
basic.forever(function () {
    music.playTone(659, music.beat(BeatFraction.Double))
    music.playTone(698, music.beat(BeatFraction.Whole))
    music.playTone(784, music.beat(BeatFraction.Double))
    music.playTone(698, music.beat(BeatFraction.Whole))
    music.playTone(659, music.beat(BeatFraction.Whole))
    music.playTone(587, music.beat(BeatFraction.Whole))
    music.playTone(523, music.beat(BeatFraction.Double))
    music.playTone(587, music.beat(BeatFraction.Whole))
    music.playTone(659, music.beat(BeatFraction.Double))
    music.playTone(659, music.beat(BeatFraction.Half))
    music.playTone(587, music.beat(BeatFraction.Half))
    music.playTone(587, music.beat(BeatFraction.Double))
    music.playTone(659, music.beat(BeatFraction.Double))
    music.playTone(698, music.beat(BeatFraction.Whole))
    music.playTone(784, music.beat(BeatFraction.Double))
    music.playTone(698, music.beat(BeatFraction.Whole))
    music.playTone(659, music.beat(BeatFraction.Whole))
    music.playTone(587, music.beat(BeatFraction.Whole))
    music.playTone(523, music.beat(BeatFraction.Double))
    music.playTone(587, music.beat(BeatFraction.Whole))
    music.playTone(659, music.beat(BeatFraction.Whole))
    music.playTone(587, music.beat(BeatFraction.Whole))
    music.playTone(587, music.beat(BeatFraction.Half))
```

Test Results

Done wiring and powered up, send the code 1 to micro:bit, you should hear two sounds produced from passive buzzer circularly. If send the code 2 to micro:bit, the buzzer will play the song Ode To Joy! Really amazing. Right? You can try to change the tone to play other music.