

## MICROBIT PROGRAMMING (BLOCK-BASED)

(Time required 30-45 minutes session)

### Introduction

In this activity, we will again bridge the gap between the digital world and the analog world. We'll be using a piezo buzzer that makes a small "click" when you apply voltage to it (try it!). By itself that isn't terribly exciting, but if you turn the voltage on and off hundreds of times a second, the piezo buzzer will produce a tone. And if you string a bunch of tones together, you've got music! This circuit and set of code blocks will create a simple sound machine.

Parts Needed (included in the kit):

- 1x micro:bit
- 1x Micro B USB Cable
- 1x micro:bit Breakout (with Headers)
- 1x Breadboard
- 14x Jumper Wires
- 1x Piezo Buzzer
- 2x Momentary Push Buttons
- 2x 10k $\Omega$  Resistors

### Buzzer Introduction

The buzzer is a small component with a piece of metal in it that moves when you apply a voltage across it. This motion causes a small sound, or "click."

If you turn the voltage on and off fast enough, you get different beeps, squeals, chirps and buzzes. You will use PWM to control the speed of turning the piezo on and off --- and, in turn, the audio frequency coming out of the buzzer. Adjusting the PWM enables you to get legitimate notes out of the buzzer. If you flip the buzzer over and look at the bottom, you will see that one pin has a (+) next to it. That pin gets connected to a signal from the P0 pin. The other pin should be connected to ground.



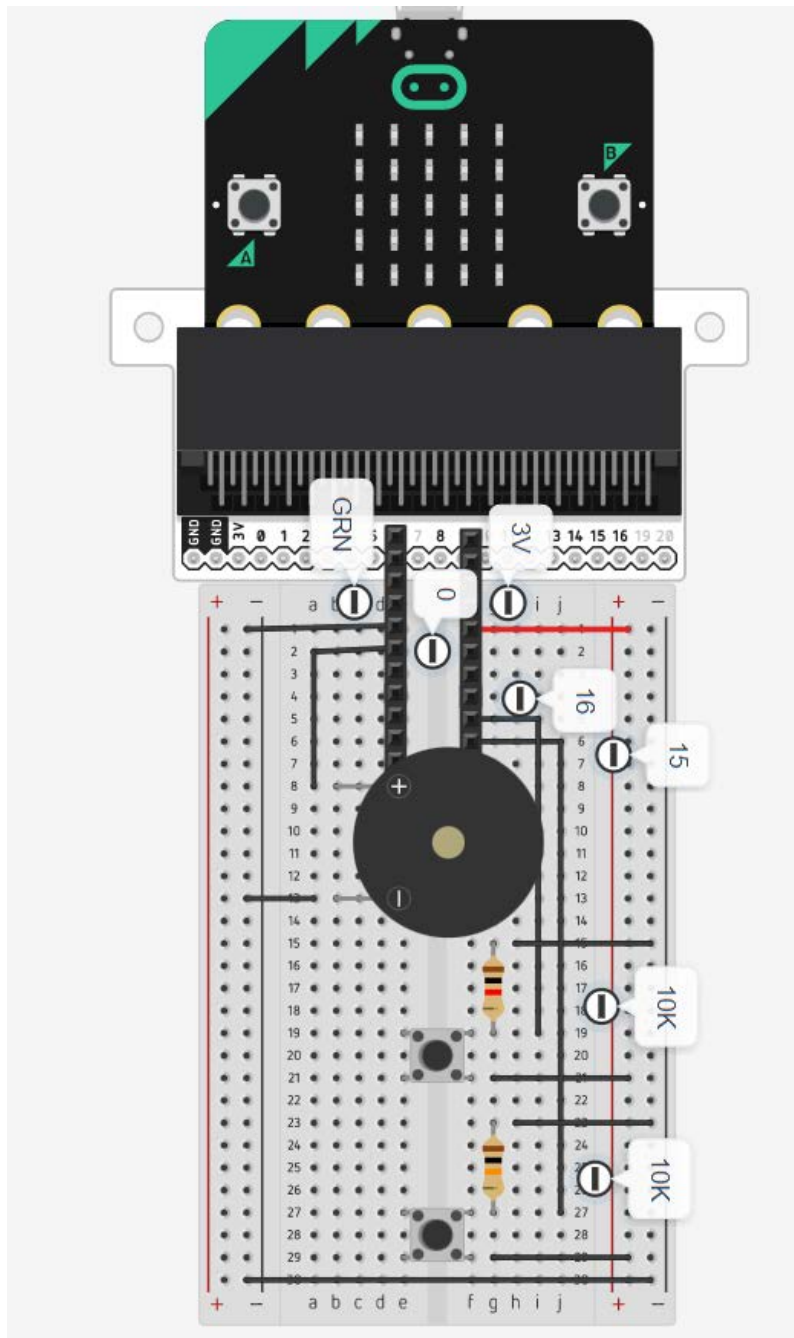
### Start Melody Repeating

The Start Melody Repeating block takes all of the frustration out of getting music out of a microcontroller. It is as simple as selecting one of a number of songs that are preprogrammed into MakeCode and how many times you want it to repeat and you are done! Note that when a melody is playing no other code can run, this is called "blocking" code and has to be accounted for you in your program.

## Play Tone for

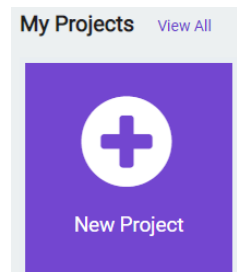
The play tone for block is pretty standard if you are used to making sound with other microcontrollers. For example, tone() function in Arduino is pretty much the same as this block. The play tone for block accepts a note that you would like the buzzer to produce and the length of time in beats per second that you would like it to play. So if you are a musician, you are golden to write horrible robot music for your friends!

### Buzzer Wiring Diagram

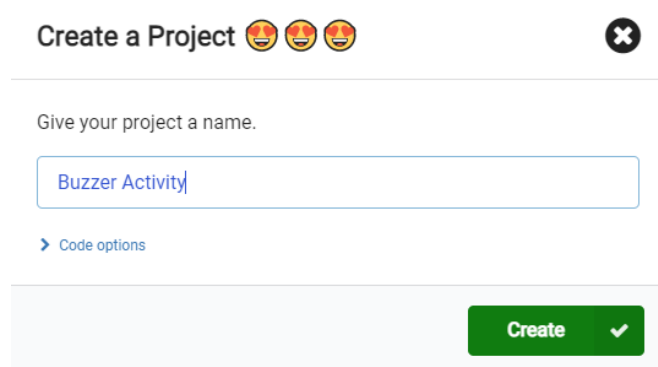


## Building the “Buzzer Activity”

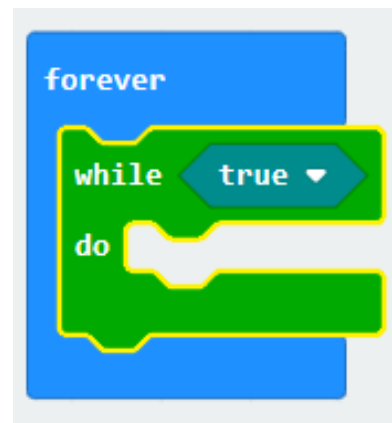
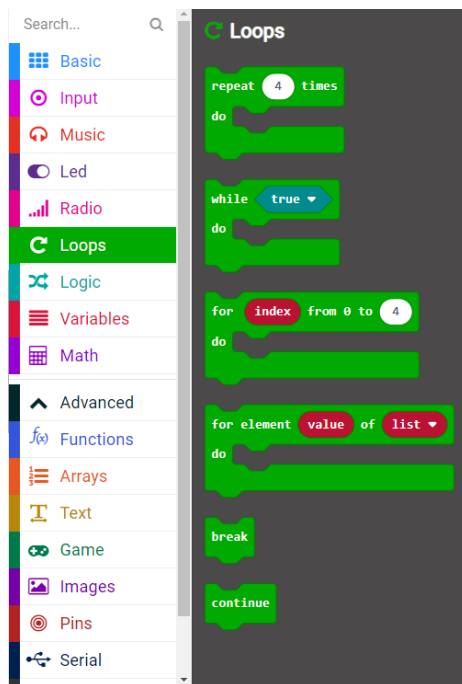
**Step 1:** Go to <https://makecode.microbit.org/#> and create a New Project



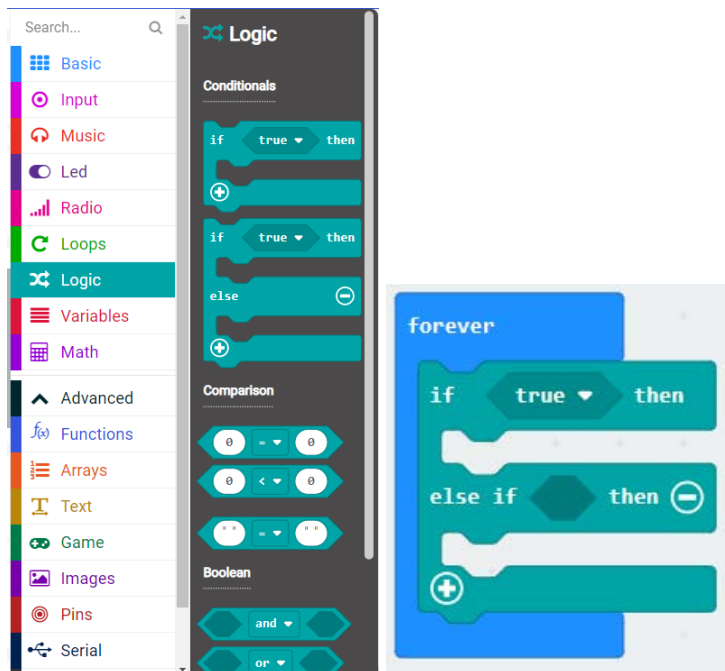
**Step 2:** Click on New Project and give it a project name **Buzzer Activity** and click Create



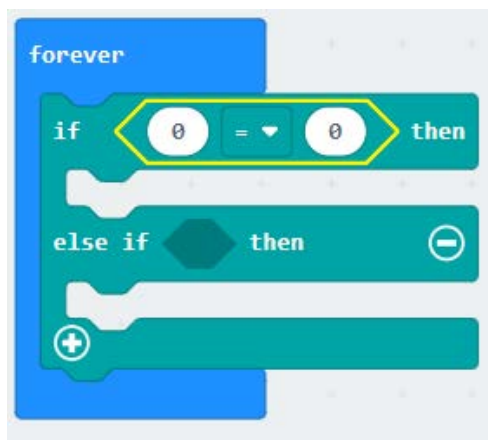
**Step 3:** Once the MakeCode is launched, click on the **Loops** category and then select and drag **while <true> do** block into the forever block.



**Step 4:** Click on the **Logic | Conditionals** category and select and drag the **if <true> then else** block into the **while <true> do** block.



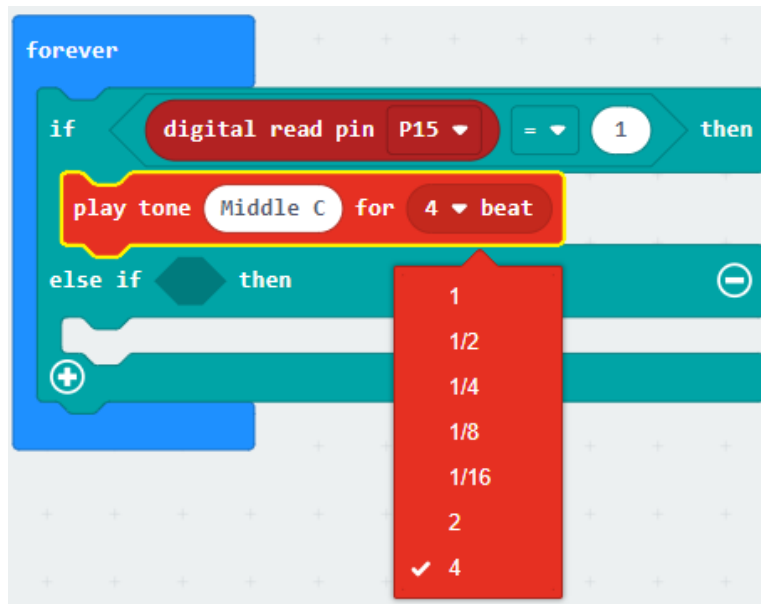
**Step 5:** Click on the **Logic | Comparison** category then select and drag the **<0> = <0>** block into the **if <true>** block



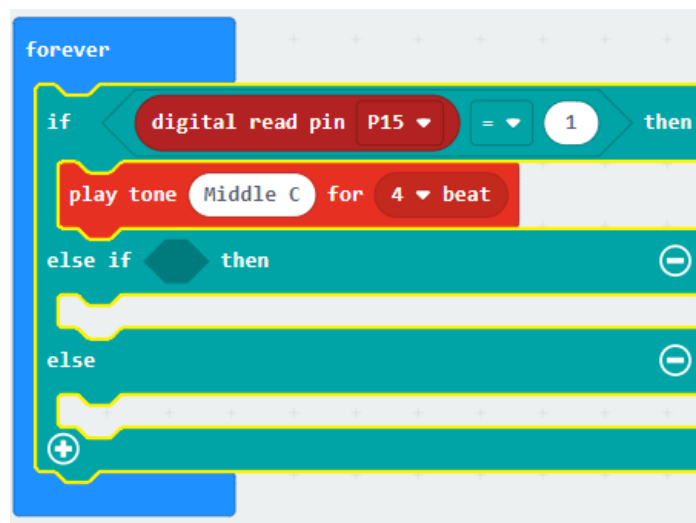
**Step 6:** Click on the **Advance** category then **Pins** then select and drag the **digital read pin P0** block into the **left comparison** block. Change the pin value to **pin 15 (P15)**. Finally, replace the 0 by 1.



**Step 7:** Click on the **Music | Tone** category then select and drag the **play tone Middle C for 1 beat** block into the **if <true> then** block. Change the number of repetition to 4 beats.



**Step 8:** Click on the plus (+) to add of the **if <true> then else** block to add an additional condition for button 2



**Step 9:** Click on the **Logic | Comparison** category then select and drag the **<0> = <0>** block into the **else if <true> then** block.

**Step 10:** Click on the **Advance** category then **Pins** then select and drag the **digital read pin P0** block into the **left comparison** block. Change the pin value to **pin 16 (P16)**. Finally, replace the 0 by 1.

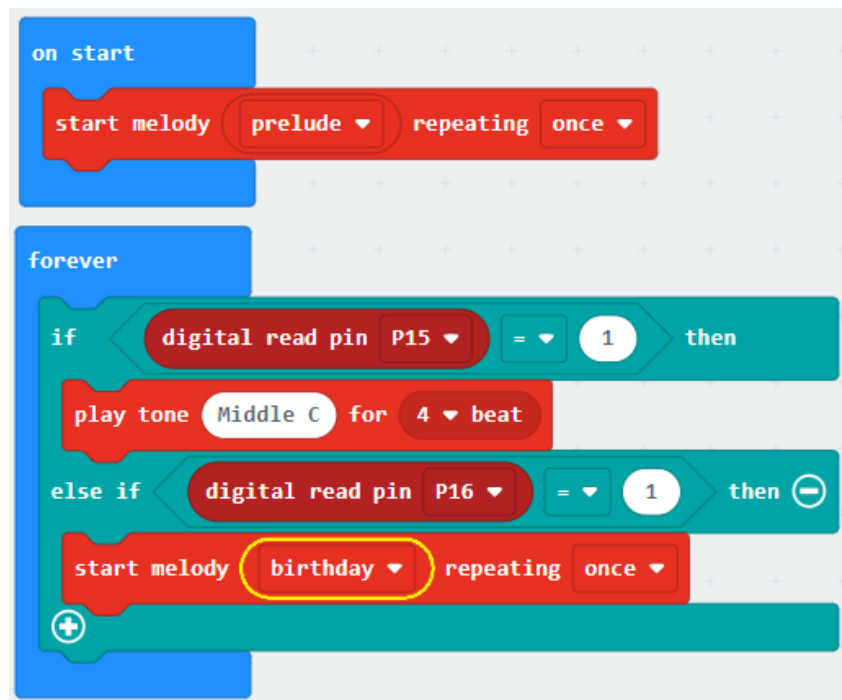
**Step 11:** Click on the **Music | Melody Advance** category then select and drag the **start melody dadadum repeating once** block into the **if <true> then** block.

**Step 12:** Click on the minus (-) of the **else** block to remove the else statement block

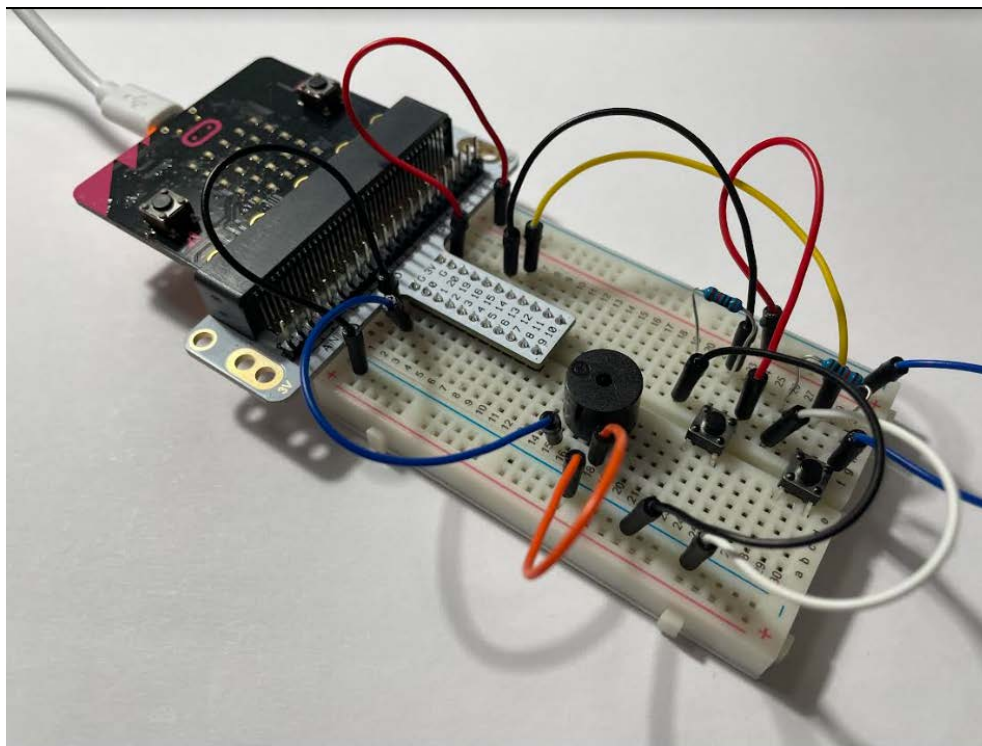
**Step 13:** Change the melody to birthday



**Step 14:** Click on the **Music | Melody Advance** category then select and drag the **start melody dadadum repeating once** block into the **on start** block. Change **dadadum** to **prelude repeating once**



## Final Prototype

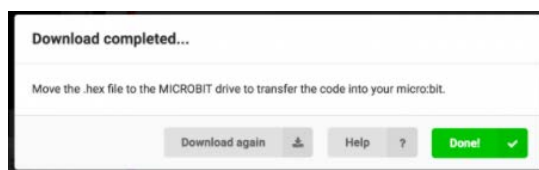


*Plug the USB cable to the micro:bit*

**Step 7:** Click the download button in the lower lefthand corner of the code window. It will download most probably in the Downloads folder



**Step 8:** Simply click and drag your program file from its download location to your micro:bit drive, which shows up as an external device.



**Step 9:** Your micro:bit will flash for a few seconds, and then your program will start automatically. A prelude melody will start playing once. If you want to hear the birthday melody press the button2 that is connected to P16. If you want to hear Middle C tone press the button1 that is connected to P15

**Step 10:** Make changes to your program by replacing the melody of your choice.

**Congratulations! You have successfully completed this activity.**

*Reference: Sparkfun Inventor's Kit for micro:bit Experiment Guide*

<https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-for-microbit-experiment-guide>