CamJam EduKit Worksheet Six

Project Morse code SOS using a Buzzer

Description In this project, you will learn how to wire and program a buzzer, and use it to produce Morse code.

You will be using 'user-defined functions'.

Equipment Required

The circuit built in CamJam EduKit Worksheet Five, plus the following:

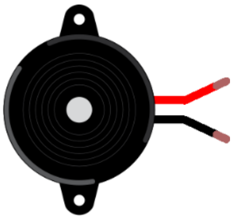
- Buzzer
- 1 x M/F Jumper Wires
- 1 x M/M Jumper Wire

Additional Parts

You will be adding a buzzer to the LED and switch circuit that you made in CamJam EduKit Worksheet Five. Let us look at the additional components.

Do not skip this section, as you need to know how to connect the buzzer.

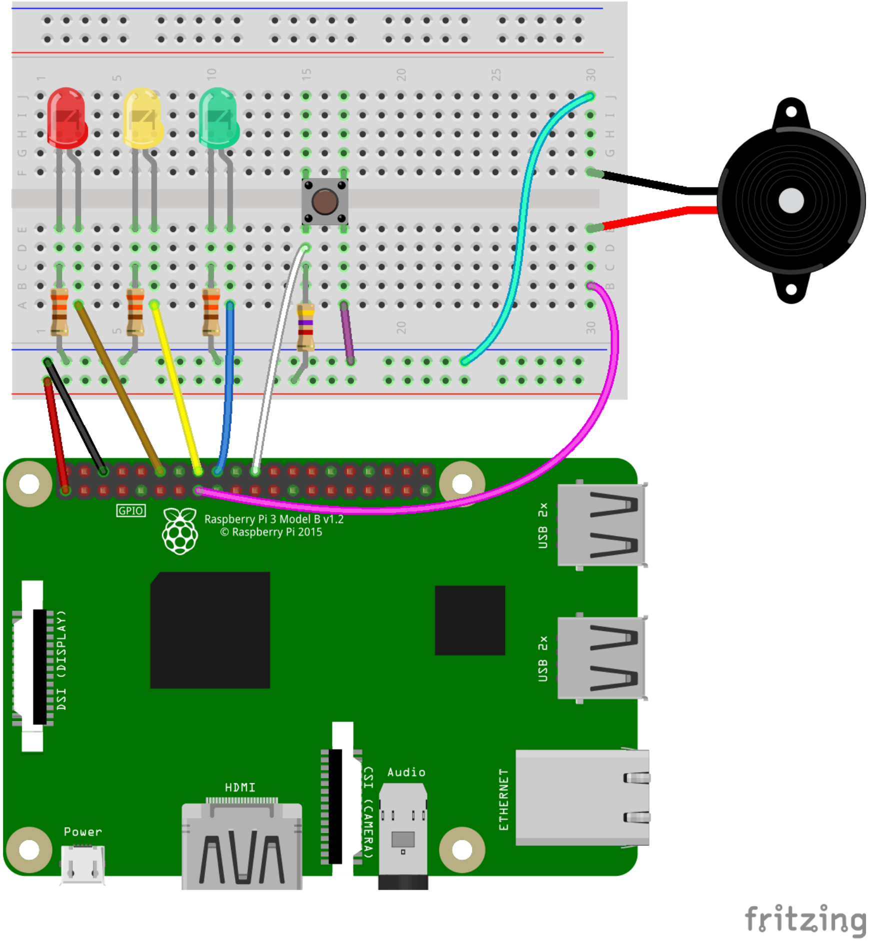
Buzzer



The buzzer supplied in the EduKit is an 'active' buzzer, which means that it only needs an electric current to make a noise. In this case, you are using the Raspberry Pi to supply that current.

The buzzer has positive and negative legs. The longer leg is positive (shown in red in the diagram), the shorter leg is negative (shown in black on the diagram).

Building the Circuit



Before you connect additional components to your circuit, you should turn off your Pi.

Leave the LED and switch circuit from Worksheet 5 in place.

Place the buzzer on the breadboard straddling the middle divide. The longer leg should be connected via a jumper wire to pin 22.

The other leg should be connected to the ground rail.

Pin 22 will be an output pin, and when it is set on, the buzzer will sound.

Concepts

You are going to be using 'user-defined functions' in the code below. These are pieces of code that you may want to run more than once, but by using functions, you only have to write them once. You then 'call' that function from within your code each time you want to run it.

To define a function, you first need to tell Python that you are writing a function. For this, use the 'def' command followed by the name of the function (you choose the name), followed by brackets and a colon ():

```
def hello():
```

Everything indented after this line will be included in the function, for example:

```
    Print(“Hello World!”)
```

To use the function, you must 'call' it by simply using the name of your new function:

```
hello()
```

Now, every time Python sees 'hello()' in your code, it will print 'Hello World!'

Code

Use IDLE3 to create a new file. Type in the following code below exactly as seen, including the important indents.

```
# CamJam EduKit 1 - Basics
# Worksheet 6 - Morse Code

# Import Libraries
import os  # Gives Python access to Linux commands
import time  # Proves time related commands
import RPi.GPIO as GPIO  # Gives access to the GPIO Pins

# Set the GPIO pin naming mode
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

PINBuzzer = 22  # Sets the buzzer pin 22

# Sets PINBuzzer as an output pin and initialise it to 'off'
GPIO.setup(PINBuzzer, GPIO.OUT)
GPIO.output(PINBuzzer, GPIO.LOW)

def dot():  # A single Morse dot
    GPIO.output(PINBuzzer, GPIO.HIGH)
    time.sleep(0.1)
    GPIO.output(PINBuzzer, GPIO.LOW)
    time.sleep(0.1)

def dash():  # A single Morse dash
    GPIO.output(PINBuzzer, GPIO.HIGH)
    time.sleep(0.3)
    GPIO.output(PINBuzzer, GPIO.LOW)
    time.sleep(0.1)

def letterspace():  # The space between letters
    time.sleep(0.2)

def wordspace():  # The space between words
    time.sleep(0.6)

def morse_s():  # The Morse for S, ...
    dot()
    dot()
    dot()

def morse_o():  # The Morse for O, ---
    dash()
    dash()
    dash()

os.system('clear')  # Clears the terminal window

print("Morse Code")

# Prompt the user for input
loop_count = input("How many times would you like SOS to loop? ")
loop_count = int(loop_count)  # Convert text input into an integer

while loop_count > 0:  # Loop around the chosen number of times
    loop_count = loop_count - 1
    morse_s()
    letterspace()
    morse_o()
    letterspace()
    morse_s()
    wordspace()

GPIO.cleanup()
```

Save the file in the EduKit directory as `6-morsecode.py`.

Running the Code

Run this code by pressing F5 or using the menus. You will be prompted for the number of times you want to repeat ‘SOS’.

Challenge

Using the above code as your template, write another program that will allow you to sound any Morse code you choose. Use the following rules:

- The length of a dot is one unit.
- The length of a dash is three units.
- The space between the parts of each letter is one unit.
- The space between letters is three units.
- The space between words is seven units.
- The letter and number codes are:

A	•–	K	–•–	U	••–	0	–––––
B	–•••	L	•–••	V	•••–	1	•––––
C	–•–•	M	––	W	•––	2	••–––
D	–••	N	–•	X	–••–	3	•••––
E	•	O	–––	Y	–•––	4	••••–
F	••–•	P	•––•	Z	––••	5	•••••
G	––•	Q	––•–			6	–••••
H	••••	R	•–•			7	––•••

I	• •	S	• • •	8	– – – • •
J	• – – –	T	–	9	– – – – •

Advanced Challenge

Using what you have learned so far, especially from Worksheet 5, make your own Morse machine by making the buzzer sound when you press the button.