

MultiWingSpan

[Home](#) [Programming](#) [Web Design](#) [Computer Science](#) [Twisting Puzzles](#) [Arduino](#) [BBC micro:bit](#)

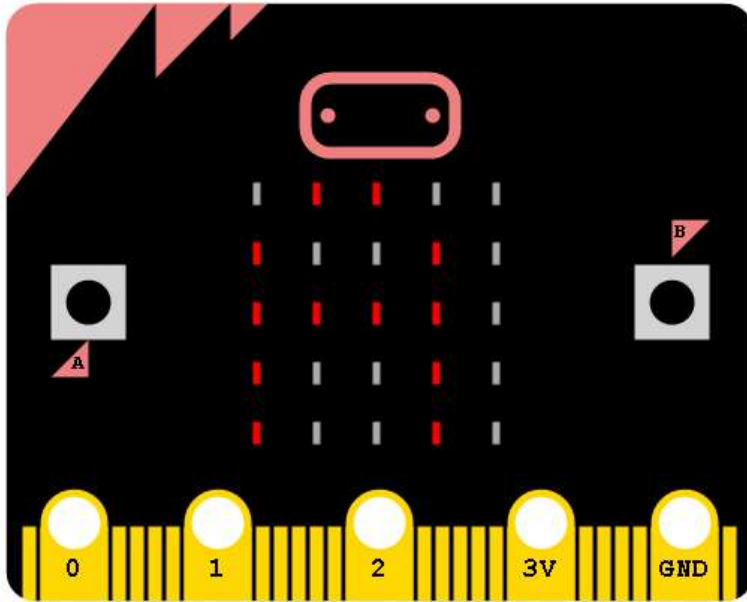
BBC micro:bit Text Entry - Accelerometer

Introduction

This example makes use of the push buttons and accelerometer to allow the user to enter text that can be used in a micro:bit program. Here, we will just scroll the entered text across the screen.

The Idea

The micro:bit will start with a blank display until you press the A button. Then it will show the letter A,



Tilt the micro:bit away from you to move through the alphabet. Press the A button each time you want to select the letter for your word.

When you enter your last letter, hold down the B button. Your text should now scroll repeatedly across the micro:bit display.

The Program

```
from microbit import *

# charset is a list of ASCII codes in our character set
charset = [i for i in range(65,91)] # upper case letters
charset += [k for k in range(48,58)] # digits
charset += [l for l in range(32,48)] # punctuation and symbols

# chars is a list of characters in our character set
chars = [chr(i) for i in charset]

def GetLetter():
    current = 0
    display.show(chars[current])
    while button_a.was_pressed() == False:
        if accelerometer.get_y() > 300:
            current -= 1
            sleep(500)
        elif accelerometer.get_y() < -300:
            current += 1
            sleep(500)
        current = max(0, min(current, len(chars)-1))
        display.show(chars[current])
    return chars[current]

def GetString():
    usertext = ""
    while button_b.is_pressed() == False:
        usertext += GetLetter()
    return usertext

currentWord = ""

while True:
    while button_a.was_pressed() == False:
        display.scroll(currentWord)
        currentWord = GetString()
```

BBC Microbit

- + [Block Editor - The Basics](#)
- + [Block Editor - Components](#)
- + [Kodu - micro:bit Worlds](#)
- + [JavaScript Blocks](#)
- + [JavaScript Blocks - Exercises](#)
- + [Blocks - Bit:Bot](#)
- + [Blocks - Bit:Commander](#)
- + [MicroPython - Starting Off](#)
- [MicroPython - Examples](#)
 - ✱ [Dice Rolling](#)
 - ✱ [Shut The Matrix](#)
 - ✱ [Encoding Morse Code](#)
 - ✱ [Encoding Ciphers](#)
 - ✱ [Drawing A Maze](#)
 - ✱ [Scrolling Race Track](#)
 - ✱ [Vertical Scroller](#)
 - ✱ [Concentration Game](#)
 - ✱ [Text Entry - Accelerometer](#)
 - ✱ [Charlie's Python Game](#)
 - ✱ [Lights Out Game](#)
 - ✱ [Sam's French Number Game](#)
 - ✱ [Bryn's Concentrated Clocks](#)
 - ✱ [Lattice Paths](#)
 - ✱ [Knight Moves](#)
- + [MicroPython - Components](#)
- + [MicroPython - Breakout Boards](#)
- + [MicroPython - Exercises](#)
- + [MicroPython - Pi Accessories](#)
- + [MicroPython - Bit:Bot](#)
- + [MicroPython - Bit:Commander](#)
- + [MicroPython - Projects](#)
- + [MicroPython - Visual Basic](#)
- + [Other - Odds & Ends](#)



The variable declarations and assignments at the top of the program are a quick way to build up the list of characters you want to allow the user to enter. Sticking to upper case makes it quicker to enter text but you can add the lower case letters if you wish.

The GetLetter() function manages the process of entering a single letter. It loops until the user presses the A button, having chosen their letter by tilting the device.

The GetString() function keeps fetching letters from the user until they are holding down the B button when they choose a letter.

The rest of the program will scroll the word across the screen until the user presses the A button again.

Challenges

1. Start by making a few tweaks to the code to have it working as you'd like. Consider the delays and the amount of tilt you want to require. You could make the speed at which letters change depend on the amount of tilt. You could also have the letter blinking until it is entered.
2. This is not very practical for something like a name badge. An accidental knock of the micro:bit and the name is deleted. You could overcome this by looking for a more complicated set of inputs and gestures that are unlikely to occur in sequence by accident.
3. Entering a single letter or digit can be used as a menu system when you have more options than you can handle with buttons and gestures. You could, for example, make a menu that allowed you to choose a number from 1 to 6. These could correspond to the strings on a music instrument. You could sound the appropriate note with a buzzer or MIDI out connection for tuning purposes.
4. Still on a music theme, you could make a table of letters and BMP tempo numbers. Get a buzzer or MIDI drum to tick tock with the appropriate intervals for time keeping purposes.