MultiWingSpan

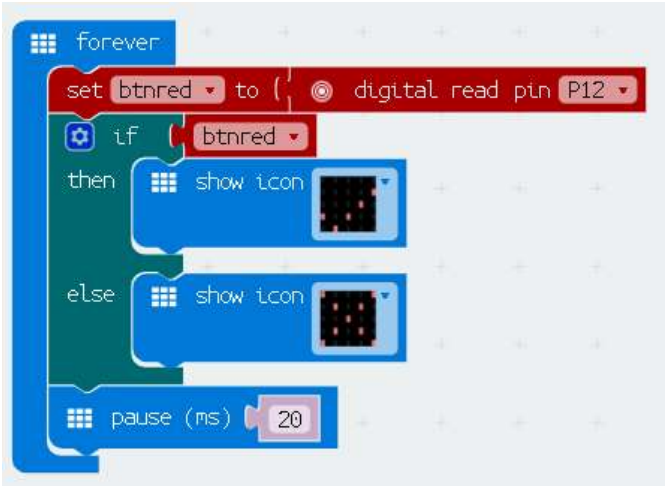
BBC micro:bit
Bit:Commander - The Pushbuttons

Introduction

The 4 pushbuttons on the Bit:Commander are arranged as they would be on a game controller, where they might be used for shooting, action buttons, jumping, selecting and so on. They are also conveniently placed if you want to use them for directional control. When the screen is small that is sometimes better than a joystick.

Programming

You can read the state of a button quite simply.



JavaScript

```
let btnred = 0
basic.forever(() => {
    btnred = pins.digitalReadPin(DigitalPin.P12)
    if (btnred) {
        basic.showIcon(IconNames.Yes)
    } else {
        basic.showIcon(IconNames.No)
    }
    basic.pause(20)
})
```

BBC Microbit

Collapse All Expand All

+ Block Editor - The Basics
+ Block Editor - Components
+ Kodu - micro:bit Worlds
+ JavaScript Blocks
+ JavaScript Blocks - Exercises
+ Blocks - Bit:Bot
- Blocks - Bit:Commander
  - Bit:Commander
  - The Joystick
  - The Neopixels
  - The Potentiometer
  - The Pushbuttons
  - The Buzzer
  - Simon Game
+ MicroPython - Starting Off
+ MicroPython - Examples
+ MicroPython - Components
+ MicroPython - Breakout Boards
+ MicroPython - Exercises
+ MicroPython - Pi Accessories
+ MicroPython - Bit:Bot
+ MicroPython - Bit:Commander
+ MicroPython - Projects
+ MicroPython - Visual Basic
+ Other - Odds & Ends

You can also respond to button presses by using events. The following program displays R,G,B,Y on the matrix when a button is pressed. That is the first letter of the colour of each of the buttons. The screen is cleared when the button is released.

JavaScript

```
pins.onPulsed(DigitalPin.P12, PulseValue.High, () => {
    basic.clearScreen()
})
pins.onPulsed(DigitalPin.P12, PulseValue.Low, () => {
    basic.showString("R")
})
pins.onPulsed(DigitalPin.P15, PulseValue.High, () => {
    basic.clearScreen()
})
pins.onPulsed(DigitalPin.P15, PulseValue.Low, () => {
    basic.showString("B")
})
pins.onPulsed(DigitalPin.P14, PulseValue.High, () => {
    basic.clearScreen()
})
pins.onPulsed(DigitalPin.P14, PulseValue.Low, () => {
    basic.showString("G")
})
pins.onPulsed(DigitalPin.P16, PulseValue.High, () => {
    basic.clearScreen()
})
pins.onPulsed(DigitalPin.P16, PulseValue.Low, () => {
    basic.showString("Y")
})
```