# Software Flow

## define constants:

- flashRateHz = 2
- buttonIn = D2
- whiteLEDout = D3
- redLEDout = D4
- greenLEDout = D5
- blueLEDout = D6
- pwmMax = 255
- debounceDelay = 50
- batteryVoltageIn = A7
- lowBatteryVoltage = 7.5
- medBatteryVoltage = 8.5

## declare variables:

- mode = 0
- buttonPushed = True
- pwmOut = 0
- prevButtonState = 0
- lastDebounceTime = 0
- batteryVoltage = 0

## Setup:

attachInterrupt(digitalPinToInterrupt(buttonIn), detectButtonPress, CHANGE)

## loop:

### detectButtonPress()

used as ISR

Sets buttonPushed → !buttonPushed

### setMode(buttonPushed)

counter which increments mode

by 1: 0, 1, 2, 3, 4, 0, 1, 2... but

only if buttonPushed changed

and millis() - lastDebounceTime > debounceDelay

### outputLight(mode)

| mode | outcome |
|---|---|
| 0 | pwmOut = 0, shineLED(pwmOut) |
| 1 | pwmOut = pwmMax, shineLED(pwmOut) |
| 2 | " = int(pwmMax/2), " |
| 3 | " = int(pwmMax/4), " |
| 4 | flashLED() |

### shineLED(pwmOut)

analogWrite(whiteLEDout, pwmOut)

### flashLED()

digitalWrite(whiteLEDout, HIGH)

delay(1000 * 1/flashRateHz)

digitalWrite(whiteLEDout, LOW)

delay(1000 * 1/flashRateHz)

White LED functionality

### checkBattery()

batteryVoltage = analogRead(batteryVoltageIn)

* note:

102.4 is a conversion factor to convert battery's actual voltage to 0-1023 10 bit scale;

((batt. voltage)/5V) · 1024 = batt. voltage · 102.4

### outputBatteryIndicator(batteryVoltage)

| if/elseif | digitalWrite() outputs on tricolor LED |
|---|---|
| batteryVoltage < 100 | red = LOW, blue = LOW, green = LOW |
| batteryVoltage < 102.4 * lowBatteryVoltage | red = HIGH, blue = LOW, green = LOW |
| " < 102.4 * medBatteryVoltage | red = LOW, blue = HIGH, green = LOW |
| else | red = LOW, blue = LOW, green = HIGH |

battery indicator functionality