

NeoPixel Library - Documentation

Library Description:

The Neopixel library contains an assortment of functions useful for controlling the neopixel, an LED that can change color via serial communication, on the PIC24FJ64GA002. Connect the neopixel to pin RB13. The Neopixel library uses Timer1 module on the microcontroller. Ensure this module is not being used elsewhere. Ensure that the Neopixel_asmLib library (.h and .s files) are also included as well, the Neopixel library will internally include the Neopixel_asmLib library.

Hardware Description:

Microcontroller: [PIC24FJ64GA002 | Microchip Technology](#)

Neopixel LED: [Neopixel LED](#)

Full Documentation:

void initNeopixel()

- No arguments, no output
- Description: Initializes pin RB13 to digital and to output.

void initTimer1()

- No arguments, no output
- Description: Enables timer 1 to a 0.2 second period with a 1:64 prescale

void writeColor(int r, int g, int b)

- Arguments are the amount of red, green, and blue desired from 1 to 255. No output.
- Description: Displays the exact color desired on LED by using the exact amount of red, green, and blue.

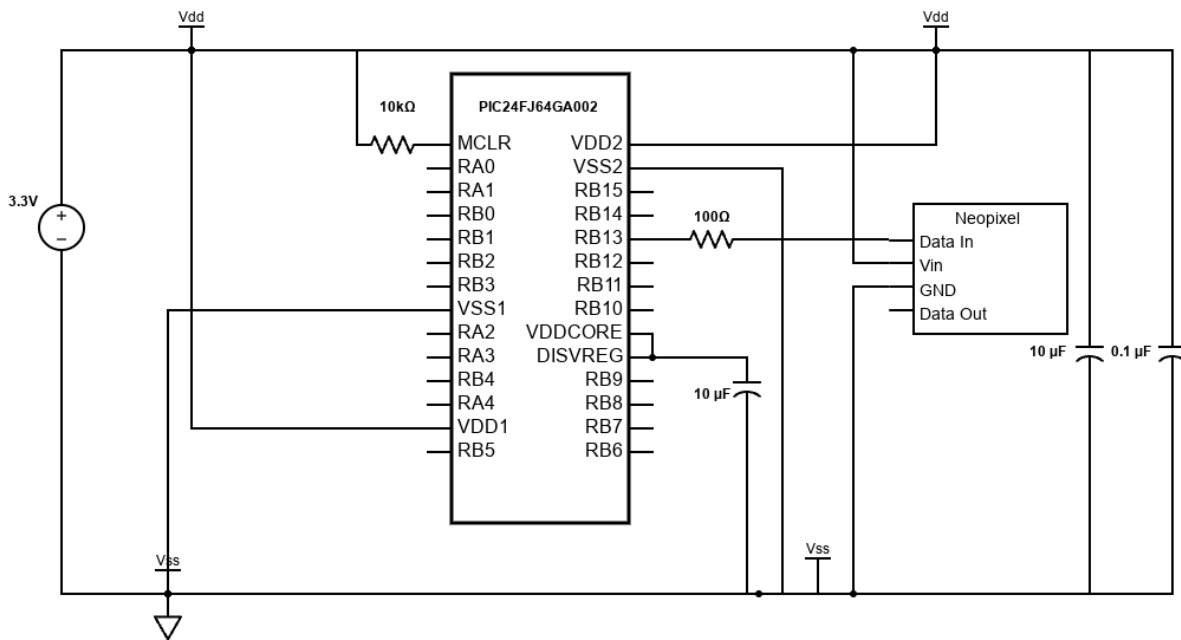
void blinkGreen()

- No arguments, no output
- Description: Blinks the LED green 4 times in 1.6 seconds

void blinkRed()

- No arguments, no output
- Description: Blinks the LED red 4 times in 1.6 seconds

Basic Usage Example:



First, wire up the Neopixel as shown in the schematic above. Leave the data out pin unconnected.

In the main code, include the “xc.h” library. Additionally, ensure to include the necessary flash configuration words into the main file (see Section 24.1 of the PIC24 Family Reference Manual for details). Also include “Neopixel.h”.

Main Code Example:

```
// CW1: FLASH CONFIGURATION WORD 1 (see PIC24 Family Reference Manual 24.1)
#pragma config ICS = PGx1      // Comm Channel Select (Emulator EMUC1/EMUD1 pins are shared with PGC1/PGD1)
#pragma config FWDTEN = OFF    // Watchdog Timer Enable (Watchdog Timer is disabled)
#pragma config GWRP = OFF      // General Code Segment Write Protect (Writes to program memory are allowed)
#pragma config GCP = OFF       // General Code Segment Code Protect (Code protection is disabled)
#pragma config JTAGEN = OFF    // JTAG Port Enable (JTAG port is disabled)
```

```
// CW2: FLASH CONFIGURATION WORD 2 (see PIC24 Family Reference Manual 24.1)
#pragma config I2C1SEL = PRI    // I2C1 Pin Location Select (Use default SCL1/SDA1 pins)
#pragma config IOL1WAY = OFF    // IOLOCK Protection (IOLOCK may be changed via unlocking seq)
#pragma config OSCIOFNC = ON    // Primary Oscillator I/O Function (CLKO/RC15 functions as I/O pin)
#pragma config FCKSM = CSECME    // Clock Switching and Monitor (Clock switching is enabled,
// Fail-Safe Clock Monitor is enabled)
#pragma config FNOSC = FRCPLL    // Oscillator Select (Fast RC Oscillator with PLL module (FRCPLL))
```

```
#include "xc.h"
```

```
#include "Neopixel.h"
```

```
int main(void) {
    initNeopixel();
```

```

    blinkRed(); // the LED should blink RED
    loop();
}

void loop() {
    while(1);
}

```

Advanced Usage Example:

The Neopixel LED is can be used to signify the state of other sensors. In the example below, the LED is blinked red when movement is detected by an accelerometer. It is then blinked green when there is no movement.

```

// CW1: FLASH CONFIGURATION WORD 1 (see PIC24 Family Reference Manual 24.1)
#pragma config ICS = PGx1      // Comm Channel Select (Emulator EMUC1/EMUD1 pins are shared with PGC1/PGD1)
#pragma config FWDTEN = OFF    // Watchdog Timer Enable (Watchdog Timer is disabled)
#pragma config GWRP = OFF      // General Code Segment Write Protect (Writes to program memory are allowed)
#pragma config GCP = OFF       // General Code Segment Code Protect (Code protection is disabled)
#pragma config JTAGEN = OFF    // JTAG Port Enable (JTAG port is disabled)

```

```

// CW2: FLASH CONFIGURATION WORD 2 (see PIC24 Family Reference Manual 24.1)
#pragma config I2C1SEL = PRI    // I2C1 Pin Location Select (Use default SCL1/SDA1 pins)
#pragma config IOL1WAY = OFF    // IOLOCK Protection (IOLOCK may be changed via unlocking seq)
#pragma config OSCIOFNC = ON    // Primary Oscillator I/O Function (CLKO/RC15 functions as I/O pin)
#pragma config FCKSM = CSECME   // Clock Switching and Monitor (Clock switching is enabled,
                                // Fail-Safe Clock Monitor is enabled)
#pragma config FNOSC = FRCPLL   // Oscillator Select (Fast RC Oscillator with PLL module (FRCPLL))

```

```

#include "xc.h"
#include "Accelerometer.h"
#include "Neopixel.h"

```

```

void setup();
void loop();

```

```

int main(void) {
    setup();
    loop();
}

```

```

void setup() {
    initAccelerometer();
    initNeopixel();
}

```

```
void loop() {  
  while(1) {  
    if(movementDetected()) { //device is in motion  
      blinkRed();  
    }  
    while(movementDetected());  
    blinkGreen(); // Indicate motion is stopped  
  }  
}
```

Another use case involves writing a certain color pattern to the Neopixel, for example, `writeColor(255,0,0)` will make the Neopixel completely red.