

Code Sample

```
/**
 * FreeRTOS LED Demo
 *
 * One task flashes an LED at a rate specified by a value set in another task.
 *
 * Date: December 4, 2020
 * Author: Shawn Hymel
 * License: 0BSD
 */

// Needed for atoi()
#include <stdlib.h>

// Use only core 1 for demo purposes
#if CONFIG_FREERTOS_UNICORE
static const BaseType_t app_cpu = 0;
#else
static const BaseType_t app_cpu = 1;
#endif

// Settings
static const uint8_t buf_len = 20;

// Pins
static const int led_pin = LED_BUILTIN;

// Globals
static int led_delay = 500; // ms

//*****

// Tasks

// Task: Blink LED at rate set by global variable
void toggleLED(void *parameter)
{
```

```

while (1)
{
    digitalWrite(led_pin, HIGH);
    vTaskDelay(led_delay / portTICK_PERIOD_MS);
    digitalWrite(led_pin, LOW);
    vTaskDelay(led_delay / portTICK_PERIOD_MS);
}
}

// Task: Read from serial terminal
// Feel free to use Serial.readString() or Serial.parseInt(). I'm going to show
// it with atoi() in case you're doing this in a non-Arduino environment. You'd
// also need to replace Serial with your own UART code for non-Arduino.
void readSerial(void *parameters)
{
    char c;
    char buf[buf_len];
    uint8_t idx = 0;

    // Clear whole buffer
    memset(buf, 0, buf_len);

    // Loop forever
    while (1)
    {
        // Read characters from serial
        if (Serial.available() > 0)
        {
            c = Serial.read();
            // Update delay variable and reset buffer if we get a newline character
            if (c == '\n')
            {
                led_delay = atoi(buf);
                Serial.print("Updated LED delay to: ");
                Serial.println(led_delay);
                memset(buf, 0, buf_len);
                idx = 0;
            }
            else
            {
                // Only append if index is not over message limit

```

```

        if (idx < buf_len - 1)
        {
            buf[idx] = c;
            idx++;
        }
    }
}

}

}

//*****
// Main

void setup()
{
    // Configure pin
    pinMode(led_pin, OUTPUT);

    // Configure serial and wait a second
    Serial.begin(115200);
    vTaskDelay(1000 / portTICK_PERIOD_MS);
    Serial.println("Multi-task LED Demo");
    Serial.println("Enter a number in milliseconds to change the LED delay.");

    // Start blink task
    xTaskCreatePinnedToCore(    // Use xTaskCreate() in vanilla FreeRTOS
        toggleLED,             // Function to be called
        "Toggle LED",          // Name of task
        1024,                  // Stack size (bytes in ESP32, words in FreeRTOS)
        NULL,                   // Parameter to pass
        1,                      // Task priority
        NULL,                   // Task handle
        app_cpu);               // Run on one core for demo purposes (ESP32 only)

    // Start serial read task
    xTaskCreatePinnedToCore(    // Use xTaskCreate() in vanilla FreeRTOS
        readSerial,             // Function to be called
        "Read Serial",          // Name of task
        1024,                  // Stack size (bytes in ESP32, words in FreeRTOS)
        NULL,                   // Parameter to pass
        1,                      // Task priority (must be same to prevent lockup)
        NULL,                   // Task handle

```

```
    app_cpu);          // Run on one core for demo purposes (ESP32 only)

// Delete "setup and loop" task
vTaskDelete(NULL);
}

void loop()
{
    // Execution should never get here
}
```

Revision #1

Created 1 October 2024 04:50:22 by AJ

Updated 1 October 2024 04:50:48 by AJ