



End of Universe Countdown Timer

Objectives:

The goal of this lab was to interface multiple LEDs to the microcontroller and write C code to use the LEDs as a countdown to the end of the universe counter.

- Interfacing LEDs
- C coding

Procedures:

Pre-Lab: Reviewed LED operation and wiring

Design: Created C-Code flow diagram

- Created Code for the counter
- Designed the 4 LED driver circuit
- Built the LED circuits

Test: Downloaded the program onto the microcontroller

- Used button 1 as the “start” input

Verified:

- Count did not start until button 1 pushed
- Counted down
- Stopped at 0000
- Reset put the design back to the beginning

Results:

At first my LEDs did not light up. I determined I had placed them in the board backwards. Once the LEDs were wired properly the design worked. On reset, all 4 LEDs were on. When I pushed the button the LEDs started to count down in a binary pattern until all the LEDs were off.

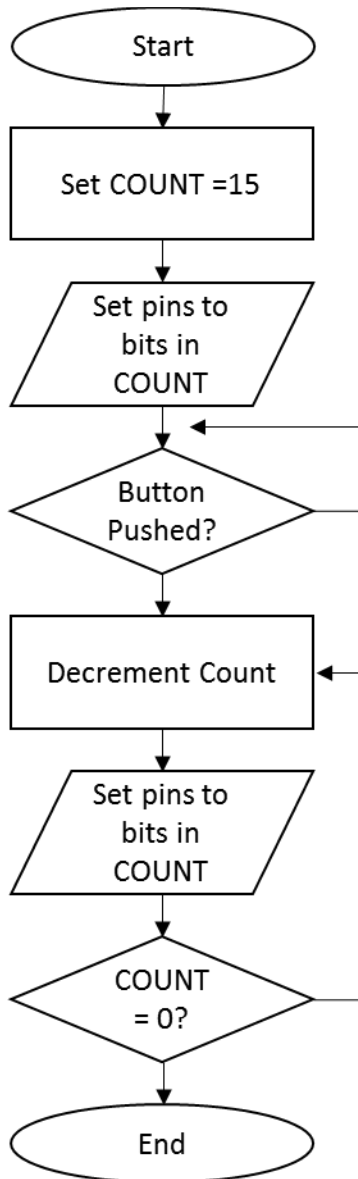
The appendix has the following: C-Code flow diagram, LED circuit design equations, Design schematic, C-Code, Picture of the working design (part way through countdown).

Observations and Conclusions:

I was reminded in this lab that it's important which direction I wire up my LEDs. I also started the design without creating a flow diagram. I quickly realized that I was not sure what to do and once I created the flow diagram the code development went very quickly.

My design met all of the design criteria.

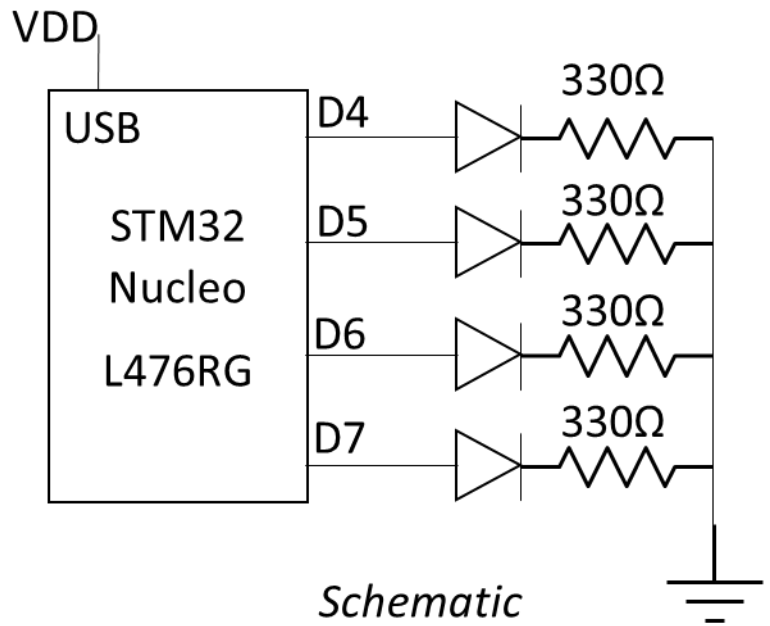
Appendix and References:



Flow Diagram

LED circuit design calculations
 $V+ = 3.3V$
 $V_{diode} = 1.8V$
 $V_{res} = 3.3V - 1.8V = 1.5V$
 $I_{out-max} = 6mA$
 $R = V_{res} / I_{out-max} = 1.5V / 6mA = 250\Omega$

Calculations



Schematic

```

////////////////////////////////////
//
// my_blink_wait project
//
// created (date) by (you)
// rev 0
//
////////////////////////////////////
//
// My version of the blink program
//
// This program prints out a simple
// and causes an external LED to
// specific interval
//
////////////////////////////////////
#include "mbed.h"
#include <stdio.h>          // only needed when printing
#define T_WAIT 2000000    // in us
int main(void){
    setbuf(stdout, NULL); // disable buffering when printing
    // splash
    printf("\n\nmy_flash\n");
    printf("Using Mbed OS version %d.%d.%d\n\n",
           MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // create the LED object tied to D5
    DigitalOut MyLED(D4);
    // my splash
    printf("My first mbed program\n");

    // run an infinite loop
    while(1){
        // flash the LED and print to the terminal
        printf("off\n");
        MyLED.write(0);
        wait_us(T_WAIT);
        printf("on\n");
        MyLED.write(1);
        wait_us(T_WAIT);
    }// end while

    return 0;
} // end main

```

To include your code in the report (un-formatted)

In Keil Studio – Rt Click on the file (main.cpp) → download selected file

- Print and staple to your report

or

- Included in a MS Word File

In MS Word – insert → object ↓ → Text from file
Point to the .cpp file and it will be included

Code