**Date Submitted: 9/24**

**Task 00: Execute provided code**

**Youtube Link:** [**https://youtu.be/H5WZvWr\_9gw**](https://youtu.be/H5WZvWr_9gw)

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

uint8\_t ui8PinData = 2;

**void** **main**(**void**)

{

// Task 0

**SysCtlClockSet**(SYSCTL\_SYSDIV\_5 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3);

// T = 300 mS where LED stays on for 150 mS

**while**(1)

{

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, ui8PinData);

**SysCtlDelay**(2000000);

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, 0x00);

**SysCtlDelay**(2000000);

**if**(ui8PinData == 8)

ui8PinData = 2; // Reset the LED sequence

**else**

ui8PinData = ui8PinData \* 2; // Increment the LED sequence

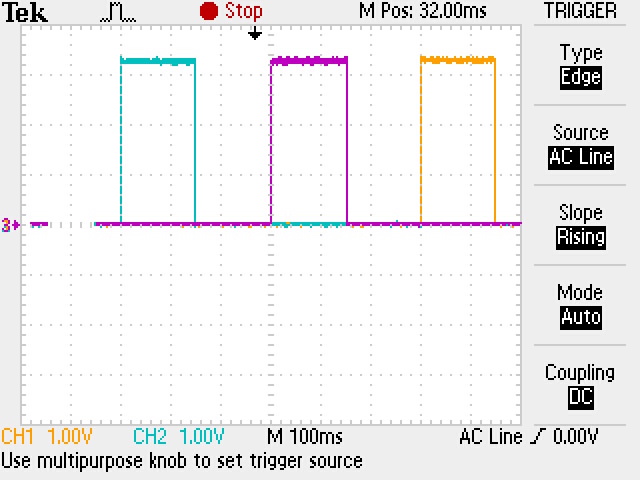
}

}

**------------------------------------------------------------------------------------**

**Task 01:**

**Waveform of task 0:**

****

**The period of the provided code from task 0 is T = 300 mS. The LED turns on for 150 mS.**

**Modified Code:**

Youtube Link: <https://youtu.be/H-G_SkNPPWM>

**To turn on the LED for 500 mS in the period of T = 1 s, the clock is set to 400/2/10 = 20 MHz. The period of a clock cycle is 1/20 MHz = 0.05 μS or 0.00005 mS. Then the value for delay function must be 500 mS/(0.00005 mS\* 3) = 3333333.**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

uint8\_t ui8PinData = 2;

**void** **main**(**void**)

{

// Task 1

**SysCtlClockSet**(SYSCTL\_SYSDIV\_10 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3);

// T = 1 s where LED stays on for 500 mS

**while**(1)

{

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, ui8PinData);

**SysCtlDelay**(3333333);

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, 0x00);

**SysCtlDelay**(3333333);

**if**(ui8PinData == 8)

ui8PinData = 2; // Reset the LED sequence

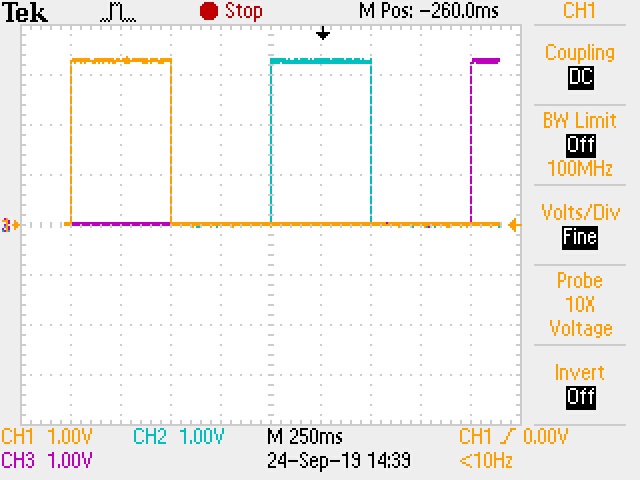
**else**

ui8PinData = ui8PinData \* 2; // Increment the LED sequence

}

}

**Waveform of task 1:**



**Modifying the code from TASK 00 to show a wave with T = 1 s. The LED turns on for 500 mS.**

**------------------------------------------------------------------------------------**

**Task 02:**

**Modified Code:**

1. **Sequence the LED to blink BGR**

Youtube Link: <https://youtu.be/8JrBMiqWbGk>

**To sequence the LED to blink BGR, modifly the ‘if and else’ statement to output the sequence 4, 8, and 2 which corresponds to the color.**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

uint8\_t ui8PinData = 4;

**void** **main**(**void**)

{

// Task 2

**SysCtlClockSet**(SYSCTL\_SYSDIV\_10 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3);

// Color Values

// R = 2

// B = 4

// G = 8

// Part A - Blink the sequence BGR

**while**(1)

{

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, ui8PinData);

**SysCtlDelay**(3333333);

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, 0x00);

**SysCtlDelay**(3333333);

**if**(ui8PinData == 8)

ui8PinData = 2; // Reset the LED to RED

**else**

ui8PinData = ui8PinData \* 2; // Increment the LED sequence

}

}

1. **Sequence the LED to blink R, G, B, RG, RB, GB, RGB,…**

Youtube Link: <https://youtu.be/kfuFvv8Zey4>

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

uint8\_t ui8PinData = 0;

uint8\_t seqCount = 1;

**void** **main**(**void**)

{

// Task 2

**SysCtlClockSet**(SYSCTL\_SYSDIV\_10 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3);

// Part B - Blink R, G, B, RG, RB, GB, RGB, R, G,...

**while**(1)

{

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, ui8PinData);

**SysCtlDelay**(2000000);

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, 0x00);

**SysCtlDelay**(2000000);

// Determines the current sequence

**switch**(seqCount)

{

**case** 1:

**if**(ui8PinData == 0)

ui8PinData = 2; // Set to RED

**else**

{

ui8PinData = 0;

seqCount++;

}

**break**;

**case** 2:

**if**(ui8PinData == 0)

ui8PinData = 8; // Set to GREEN

**else**

{

ui8PinData = 0;

seqCount++;

}

**break**;

**case** 3:

**if**(ui8PinData == 0)

ui8PinData = 4; // Set to BLUE

**else**

{

ui8PinData = 0;

seqCount++;

}

**break**;

**case** 4:

**if**(ui8PinData == 0)

ui8PinData = 2; // Set to RED

**else** **if**(ui8PinData == 2)

ui8PinData = 8; // Set to GREEN

**else**

{

ui8PinData = 0;

seqCount++;

}

**break**;

**case** 5:

**if**(ui8PinData == 0)

ui8PinData = 2; // Set to RED

**else** **if**(ui8PinData == 2)

ui8PinData = 4; // Set to BLUE

**else**

{

ui8PinData = 0;

seqCount++;

}

**break**;

**case** 6:

**if**(ui8PinData == 0)

ui8PinData = 8; // Set to GREEN

**else** **if**(ui8PinData == 8)

ui8PinData = 4; // Set to BLUE

**else**

{

ui8PinData = 0;

seqCount++;

}

**break**;

**case** 7:

**if**(ui8PinData == 0)

ui8PinData = 2; // Set to RED

**else** **if**(ui8PinData == 2)

ui8PinData = 8; // Set to GREEN

**else** **if**(ui8PinData == 8)

ui8PinData = 4; // Set to BLUE

**else**

{

ui8PinData = 0;

seqCount = 1;

}

**break**;

**default**:

ui8PinData = 0; // Clear LED

seqCount = 1; // Reset sequence

}

}

}