**Date Submitted: 10/26/2019**



**Task 01:**

**Youtube Link:**

<https://youtu.be/jz0GHeqmh00>

**Modified Code:**

**#include** <stdbool.h>

**#include** <stdint.h>

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\inc\hw\_memmap.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\gpio.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\pin\_map.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\ssi.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\sysctl.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\uart.h"

**#include** "uartstdio.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\adc.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\debug.h"

**#define** NUM\_SSI\_DATA 3

**void**

**InitConsole**(**void**)

{

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

**GPIOPinConfigure**(GPIO\_PA0\_U0RX);

**GPIOPinConfigure**(GPIO\_PA1\_U0TX);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_UART0);

UARTClockSourceSet(UART0\_BASE, UART\_CLOCK\_PIOSC);

**GPIOPinTypeUART**(GPIO\_PORTA\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1);

UARTStdioConfig(0, 115200, 16000000);

}

**int**

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

{

**#if** defined(TARGET\_IS\_TM4C129\_RA0) || \

defined(TARGET\_IS\_TM4C129\_RA1) || \

defined(TARGET\_IS\_TM4C129\_RA2)

uint32\_t ui32SysClock;

**#endif**

**#if** defined(TARGET\_IS\_TM4C129\_RA0) || \

defined(TARGET\_IS\_TM4C129\_RA1) || \

defined(TARGET\_IS\_TM4C129\_RA2)

ui32SysClock = SysCtlClockFreqSet((SYSCTL\_XTAL\_25MHZ |

SYSCTL\_OSC\_MAIN |

SYSCTL\_USE\_OSC), 25000000);

**#else**

**SysCtlClockSet**(SYSCTL\_SYSDIV\_1 | SYSCTL\_USE\_OSC | SYSCTL\_OSC\_MAIN |

SYSCTL\_XTAL\_16MHZ);

**#endif**

InitConsole();

UARTprintf("SSI ->\n");

UARTprintf(" Mode: SPI\n");

UARTprintf(" Data: 8-bit\n\n");

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_SSI0);

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

**GPIOPinConfigure**(GPIO\_PA2\_SSI0CLK);

**GPIOPinConfigure**(GPIO\_PA3\_SSI0FSS);

**GPIOPinConfigure**(GPIO\_PA4\_SSI0RX);

**GPIOPinConfigure**(GPIO\_PA5\_SSI0TX);

**GPIOPinTypeSSI**(GPIO\_PORTA\_BASE, GPIO\_PIN\_5 | GPIO\_PIN\_4 | GPIO\_PIN\_3 |

GPIO\_PIN\_2);

**#if** defined(TARGET\_IS\_TM4C129\_RA0) || \

defined(TARGET\_IS\_TM4C129\_RA1) || \

defined(TARGET\_IS\_TM4C129\_RA2)

SSIConfigSetExpClk(SSI0\_BASE, ui32SysClock, SSI\_FRF\_MOTO\_MODE\_0,

SSI\_MODE\_MASTER, 1000000, 8);

**#else**

**SSIConfigSetExpClk**(SSI0\_BASE, **SysCtlClockGet**(), SSI\_FRF\_MOTO\_MODE\_0,

SSI\_MODE\_MASTER, 1000000, 8);

**#endif**

uint32\_t ui32ADC0Value[4];

**volatile** uint32\_t ui32TempAvg;

**volatile** uint32\_t ui32TempValueC;

**volatile** uint32\_t ui32TempValueF;

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

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_ADC0);

**ADCSequenceConfigure**(ADC0\_BASE, 1, ADC\_TRIGGER\_PROCESSOR, 0);

**ADCSequenceStepConfigure**(ADC0\_BASE, 1, 0, ADC\_CTL\_TS);

**ADCSequenceStepConfigure**(ADC0\_BASE, 1, 1, ADC\_CTL\_TS);

**ADCSequenceStepConfigure**(ADC0\_BASE, 1, 2, ADC\_CTL\_TS);

**ADCSequenceStepConfigure**(ADC0\_BASE,1,3,ADC\_CTL\_TS|ADC\_CTL\_IE|ADC\_CTL\_END);

**ADCSequenceEnable**(ADC0\_BASE, 1);

**SSIEnable**(SSI0\_BASE);

**while**(1){

**ADCIntClear**(ADC0\_BASE, 1);

**ADCProcessorTrigger**(ADC0\_BASE, 1);

**while**(!**ADCIntStatus**(ADC0\_BASE, 1, false))

{

}

**ADCSequenceDataGet**(ADC0\_BASE, 1, ui32ADC0Value);

ui32TempAvg = (ui32ADC0Value[0] + ui32ADC0Value[1] + ui32ADC0Value[2] + ui32ADC0Value[3] + 2)/4;

ui32TempValueC = (1475 - ((2475 \* ui32TempAvg)) / 4096)/10;

ui32TempValueF = ((ui32TempValueC \* 9) + 160) / 5;

**while**(**SSIDataGetNonBlocking**(SSI0\_BASE, &pui32DataRx[0]))

{

}

pui32DataTx[0] = ui32TempValueF;

pui32DataTx[1] = ui32TempValueF;

pui32DataTx[2] = ui32TempValueF;

UARTprintf("\nSent:\n ");

**for**(ui32Index = 0; ui32Index < 1; ui32Index++)

{

UARTprintf("'%u' ", pui32DataTx[ui32Index]);

**SSIDataPut**(SSI0\_BASE, pui32DataTx[ui32Index]);

}

**SysCtlDelay**(10000000);

**while**(**SSIBusy**(SSI0\_BASE))

{

}

UARTprintf("\nReceived:\n ");

**for**(ui32Index = 0; ui32Index < 1; ui32Index++)

{

**SSIDataGet**(SSI0\_BASE, &pui32DataRx[ui32Index]);

pui32DataRx[ui32Index] &= 0x00FF;

UARTprintf("'%u' ", pui32DataRx[ui32Index]);

}

}

**SysCtlDelay**(10000000);

**return**(0);

}

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

**Task 02:**

Youtube Link:

<https://youtu.be/r6PYxtap78E>

**Modified Code:**

**#include** <stdbool.h>

**#include** <stdint.h>

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\inc\hw\_memmap.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\inc\hw\_types.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\gpio.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\pin\_map.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\ssi.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\sysctl.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\uart.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\utils\uartstdio.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\adc.h"

**#include** "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\debug.h"

**#define** NUM\_LEDS 8

uint8\_t frame\_buffer[NUM\_LEDS\*3];

**void** **send\_data**(uint8\_t\* data, uint8\_t num\_leds);

**void** **fill\_frame\_buffer**(uint8\_t r, uint8\_t g, uint8\_t b, uint32\_t num\_leds);

**static** **volatile** uint32\_t ssi\_lut[] = {

0b100100100,

0b110100100,

0b100110100,

0b110110100,

0b100100110,

0b110100110,

0b100110110,

0b110110110

};

**int** **main**(**void**) {

FPULazyStackingEnable();

// 80MHz

**SysCtlClockSet**(SYSCTL\_SYSDIV\_2\_5 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ |

SYSCTL\_OSC\_MAIN);

//initialize SPI

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

**SysCtlDelay**(50000);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_SSI0);

**SysCtlDelay**(50000);

**GPIOPinConfigure**(GPIO\_PA5\_SSI0TX);

**GPIOPinConfigure**(GPIO\_PA2\_SSI0CLK);

**GPIOPinConfigure**(GPIO\_PA4\_SSI0RX);

**GPIOPinConfigure**(GPIO\_PA3\_SSI0FSS);

**GPIOPinTypeSSI**(GPIO\_PORTA\_BASE, GPIO\_PIN\_5);

**GPIOPinTypeSSI**(GPIO\_PORTA\_BASE, GPIO\_PIN\_2);

**GPIOPinTypeSSI**(GPIO\_PORTA\_BASE, GPIO\_PIN\_4);

**GPIOPinTypeSSI**(GPIO\_PORTA\_BASE, GPIO\_PIN\_3);

//20 MHz data rate

**SSIConfigSetExpClk**(SSI0\_BASE, 80000000, SSI\_FRF\_MOTO\_MODE\_0, SSI\_MODE\_MASTER, 2400000, 9);

**SSIEnable**(SSI0\_BASE);

**while**(1)

{

// RED

fill\_frame\_buffer(255, 0, 0, NUM\_LEDS);

send\_data(frame\_buffer, NUM\_LEDS);

**SysCtlDelay**(**SysCtlClockGet**()/5);

// GREEN

fill\_frame\_buffer(0, 255, 0, NUM\_LEDS);

send\_data(frame\_buffer, NUM\_LEDS);

**SysCtlDelay**(**SysCtlClockGet**()/5);

// BLUE

fill\_frame\_buffer(0, 0, 255, NUM\_LEDS);

send\_data(frame\_buffer, NUM\_LEDS);

**SysCtlDelay**(**SysCtlClockGet**()/5);

// YELLOW

fill\_frame\_buffer(255, 255, 0, NUM\_LEDS);

send\_data(frame\_buffer, NUM\_LEDS);

**SysCtlDelay**(**SysCtlClockGet**()/5);

// PURPLE

fill\_frame\_buffer(255, 0, 255, NUM\_LEDS);

send\_data(frame\_buffer, NUM\_LEDS);

**SysCtlDelay**(**SysCtlClockGet**()/5);

// LIGHT BLUE

fill\_frame\_buffer(0, 255, 255, NUM\_LEDS);

send\_data(frame\_buffer, NUM\_LEDS);

**SysCtlDelay**(**SysCtlClockGet**()/5);

// WHITE

fill\_frame\_buffer(255, 255, 255, NUM\_LEDS);

send\_data(frame\_buffer, NUM\_LEDS);

**SysCtlDelay**(**SysCtlClockGet**()/5);

}

}

**void** **send\_data**(uint8\_t\* data, uint8\_t num\_leds)

{

uint32\_t i, j, curr\_lut\_index, curr\_rgb;

**for**(i = 0; i < (num\_leds\*3); i = i + 3) {

curr\_rgb = (((uint32\_t)data[i + 2]) << 16) | (((uint32\_t)data[i + 1]) << 8) | data[i];

**for**(j = 0; j < 24; j = j + 3) {

curr\_lut\_index = ((curr\_rgb>>j) & 0b111);

**SSIDataPut**(SSI0\_BASE, ssi\_lut[curr\_lut\_index]);

}

}

**SysCtlDelay**(50000); // delay more then 50us

}

**void** **fill\_frame\_buffer**(uint8\_t r, uint8\_t g, uint8\_t b, uint32\_t num\_leds)

{

uint32\_t i;

uint8\_t\* frame\_buffer\_index = frame\_buffer;

**for**(i = 0; i < num\_leds; i++) {

\*(frame\_buffer\_index++) = g;

\*(frame\_buffer\_index++) = r;

\*(frame\_buffer\_index++) = b;

}

}

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