Faculty of Engineering

Ain Shams University

CSE 211s: Introduction to Embedded Systems

REPORT (4)

Name: Habiba El-Sayed Mowafy

Sec.: 1

Program: CSE

ID: 2100792

Question 2:

Q2. Write Embedded C program that receives commands through UART communication protocol to do the following:

1. When sending "A", all the LEDs are turned off and the Red LED is turned on after 1 minute.
2. When sending "B", all the LEDs are turned off and the Blue LED is turned on after 0.5 minutes.
3. When sending "D", all the LEDs are turned off and the Green LED is turned on after 2 minutes.

Upon starting the program, all the LEDS should be turned off.

Check through the simulated Kit that the behavior of your code is correct.

C code:

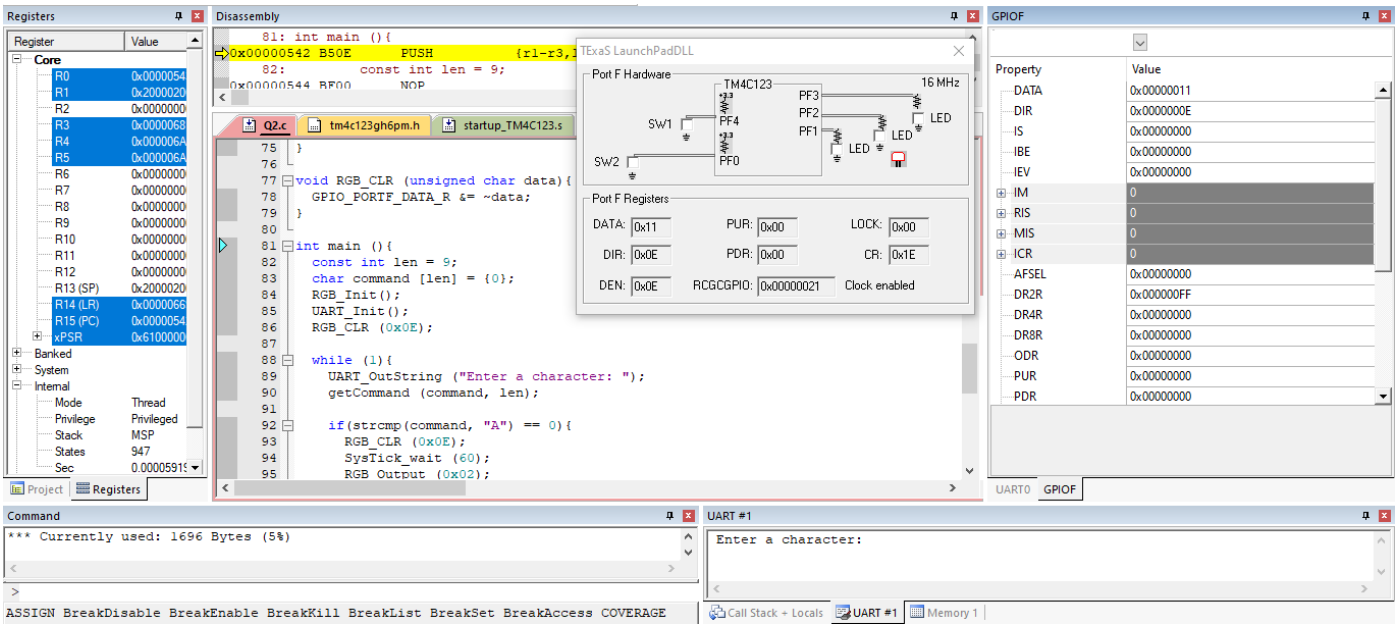
```
#include "tm4c123gh6pm.h"
#include <string.h>
#include <stdio.h>
#define CR 0x0D



void UART_Init (void){
        SYSCTL_RCGCUART_R |= 0x0001;
        SYSCTL_RCGCGPIO_R |= 0x0001;
        UART0_CTL_R &= ~0x0001;
        UART0_IBRD_R = 0x68;
        UART0_FBRD_R = 0xB;
        UART0_LCRH_R |= 0x0070;
        UART0_CTL_R &= 0x0301;
        GPIO_PORTA_AFSEL_R |= 0x03;
        GPIO_PORTA_PCTL_R = (GPIO_PORTA_PCTL_R & 0xFFFFFF00) + 0x00000011;
        GPIO_PORTA_DEN_R |= 0x03;
        GPIO_PORTA_AMSEL_R &= ~0x03;
}

void RGB_Init(void){
        SYSCTL_RCGCGPIO_R |= 0x20; //enable clk for port F
        while ((SYSCTL_PRGPIO_R & 0x20) == 0); //check for the clk delay
        GPIO_PORTF_LOCK_R = GPIO_LOCK_KEY; //break the lock of port F
        GPIO_PORTF_CR_R |= 0x0E; //enable making changes in leds pins (PF1,PF2&PF3)
        GPIO_PORTF_AMSEL_R &= ~0x0E; //disable analog mode
        GPIO_PORTF_AFSEL_R &= ~0x0E; //disable using alternate functions (use the pins
as GPIO)
        GPIO_PORTF_PCTL_R &= ~0x0000FFF0; //clear the bits corresponfing to the 3 pins
        GPIO_PORTF_DEN_R |= 0x0E; //enable using the pins in digital mode
        GPIO_PORTF_DIR_R |= 0x0E; //set the bits corresponding to the pins to 1 to use
them as output
        GPIO_PORTF_DATA_R &= ~0x0E; //initialize the leds to be off
}
```

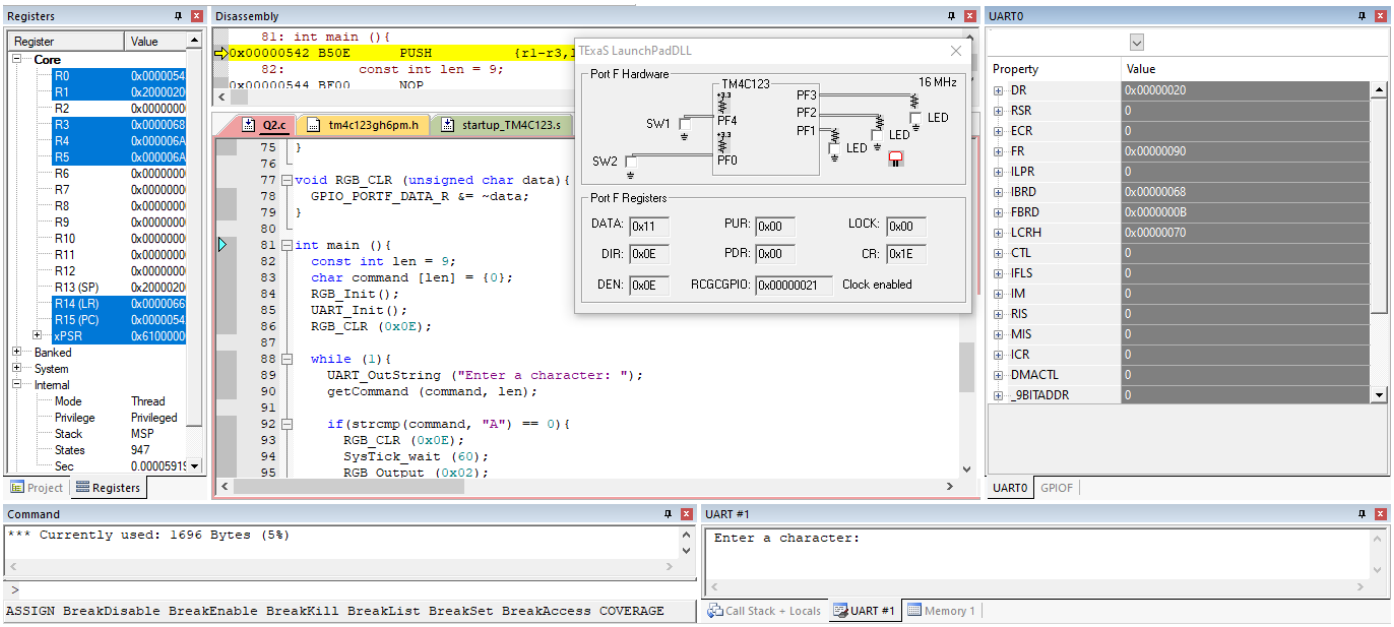
```
void SysTick_wait (){
        NVIC_ST_CTRL_R = 0x00;
        NVIC_ST_RELOAD_R = 16000 -1;
        NVIC_ST_CURRENT_R = 0x00;
        NVIC_ST_CTRL_R = 0x05;
        while ((NVIC_ST_CTRL_R & 0x00010000) == 0); //wait for count flag
}

void delay (int time){
        int i;
        for (i = 0; i <= time; i++){
                SysTick_wait ();
        }
}

char UART_INchar (void){
        while ((UART0_FR_R & 0x0010) != 0);
        return (char) (UART0_DR_R & 0x00FF);
}

void UART_OUTchar (unsigned char data){
        while ((UART0_FR_R & 0x0020) != 0);
        UART0_DR_R = data;
}

void getCommand (char * command, int len){
        char ch;
        int i;
        for (i=0; i<len; i++){
                ch = UART_INchar ();
                if (ch != CR){
                        command [i] = ch;
                        UART_OUTchar (command [i]);
                }
                else if (ch == CR || i == len) break;
        }
}

void UART_OutString (char *pt){
        while (*pt){
                UART_OUTchar (*pt);
                pt++;
        }
}

void RGB_Output (unsigned char data){
        GPIO_PORTF_DATA_R |= data;
}

void RGB_CLR (unsigned char data){
        GPIO_PORTF_DATA_R &= ~data;
}
```

```
int main (){
        const int len = 9;
        char command [len] = {0};
        RGB_Init();
        UART_Init();
        RGB_CLR (0x0E);

        while (1){
                UART_OutString ("Enter a character: ");
                getCommand (command, len);

                if(strcmp(command, "A") == 0){
                        RGB_CLR (0x0E);
                        delay (60000);
                        RGB_Output (0x02);
                        memset(command, 0, len);
                }
                else if(strcmp(command, "B") == 0){
                        RGB_CLR (0x0E);
                        delay (30000);
                        RGB_Output (0x04);
                        memset(command, 0, len);
                }
                else if(strcmp(command, "D") == 0){
                        RGB_CLR (0x0E);
                        delay (120000);
                        RGB_Output (0x08);
                        memset(command, 0, len);
                }

                UART_OutString ("\n");
        }
}
```

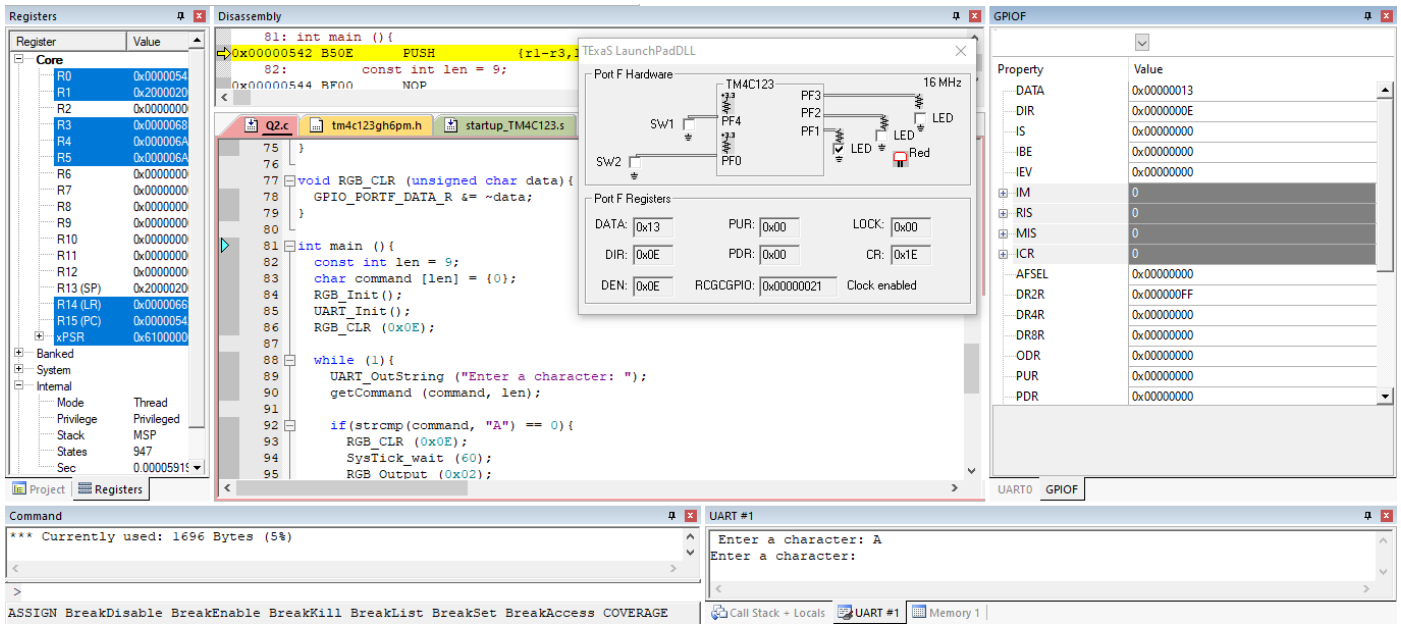
SnapShots:

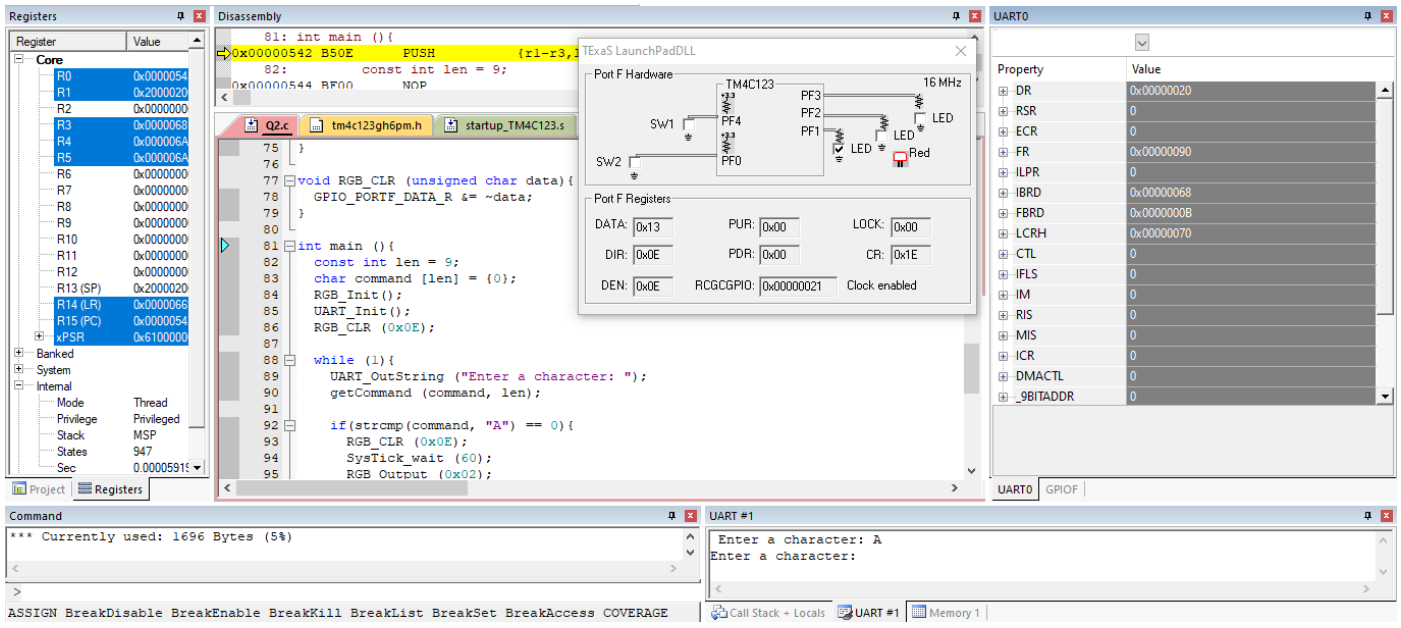
Program Started, all LEDs are OFF



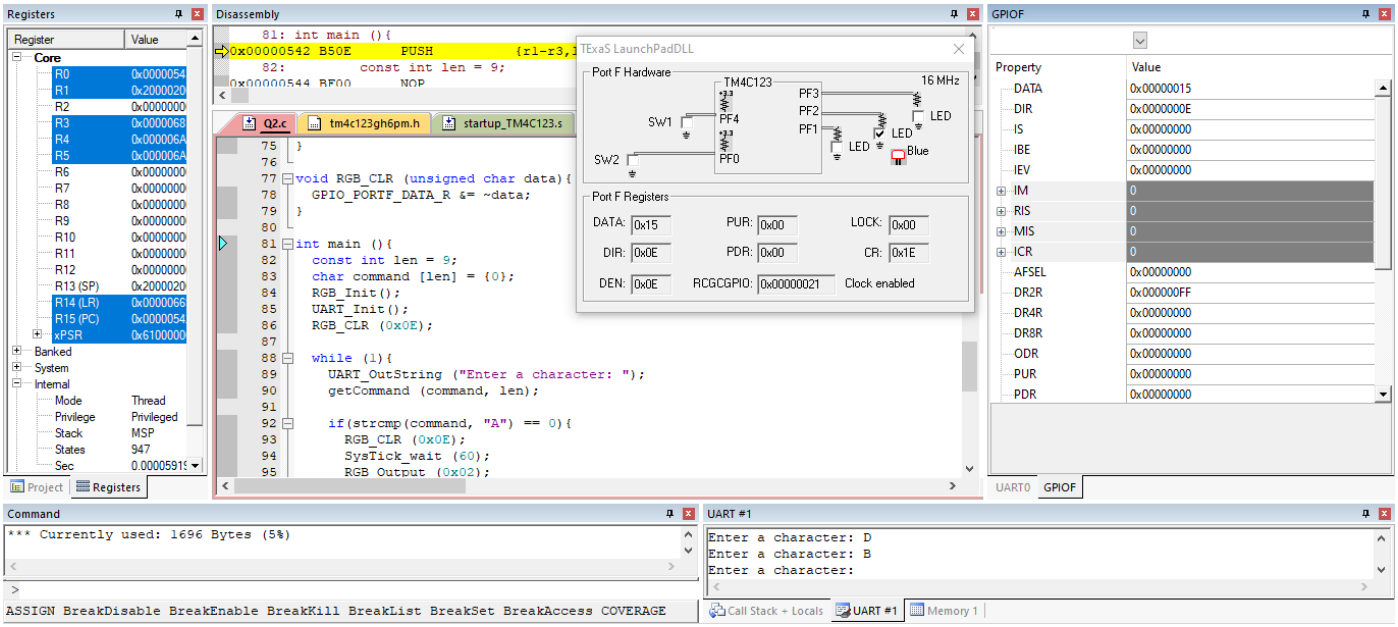


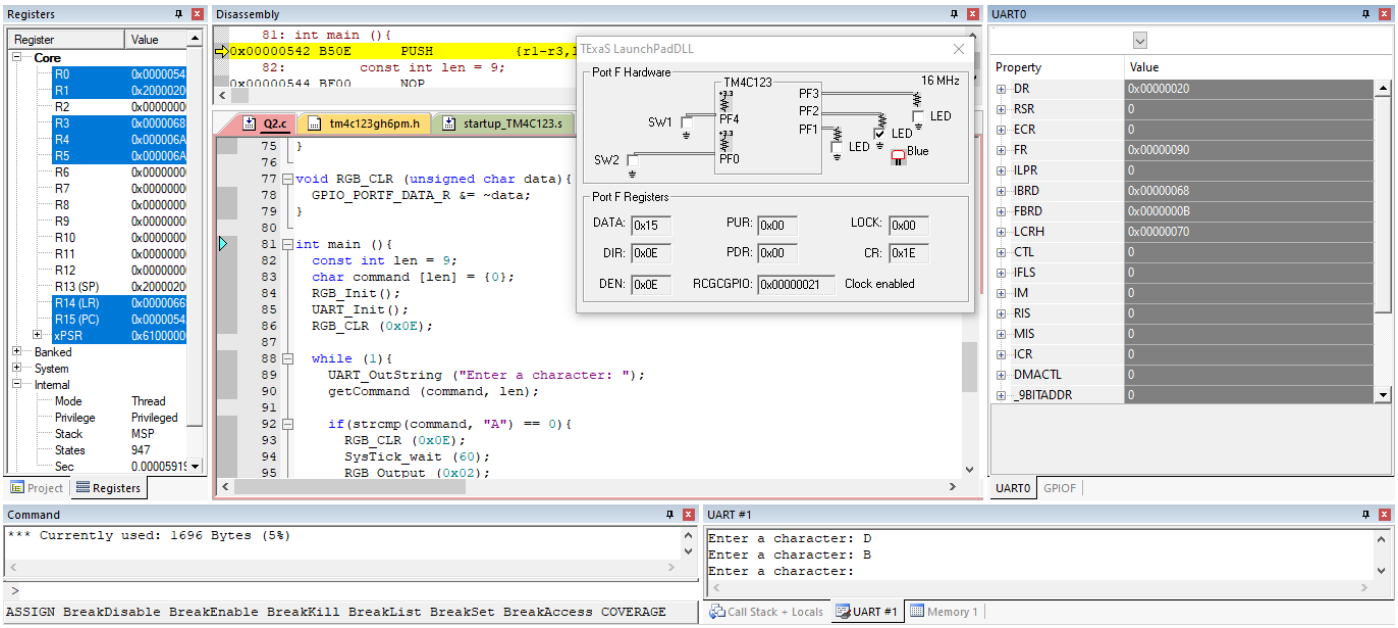
Sending “A”


Registers
Disassembly
GPIOF
TExaS LaunchPadDLL
Port F Hardware
TM4C123
16 MHz
SW1
SW2
LED
Red
Port F Registers
DATA: 0x13
PUR: 0x00
LOCK: 0x00
DIR: 0x0E
PDR: 0x00
CR: 0x1E
DEN: 0x0E
RCGCGPIO: 0x00000021
Clock enabled
Q2.c
tm4c123gh6pm.h
startup_TM4C123.s
Command
*** Currently used: 1696 Bytes (5%)
UART #1
Enter a character: A
Enter a character:
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
Call Stack + Locals
Memory 1

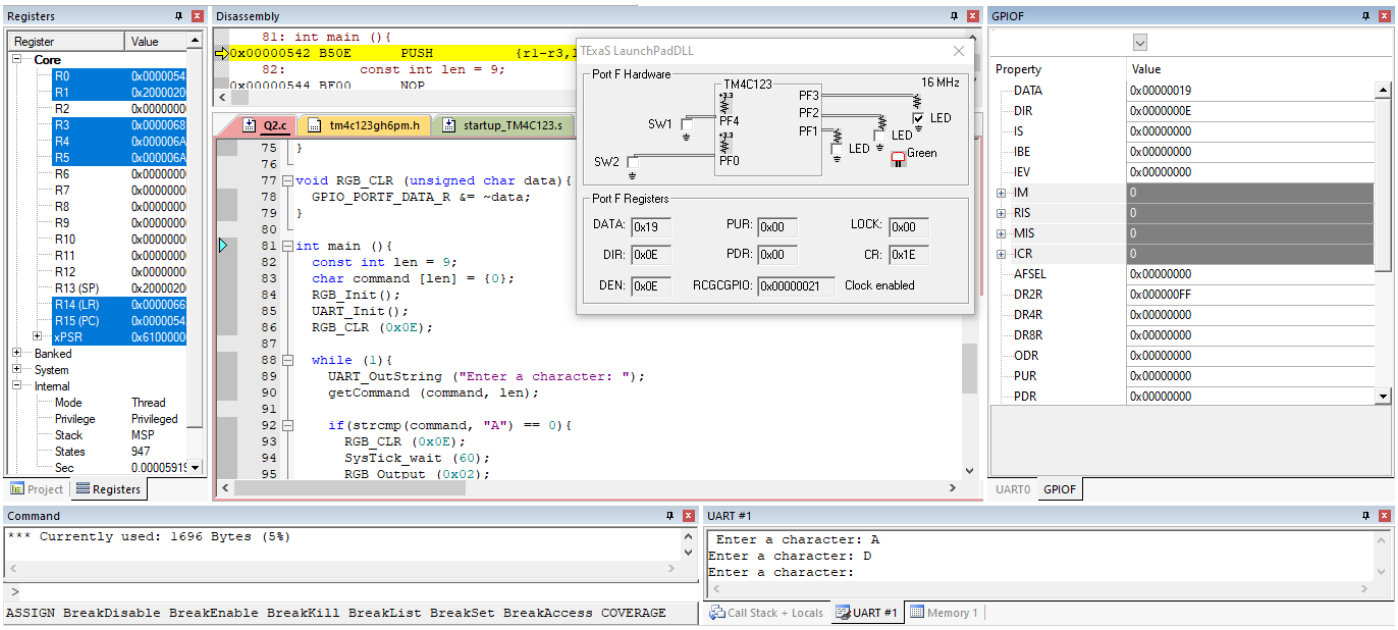

Registers
Disassembly
UART0
TExaS LaunchPadDLL
Port F Hardware
TM4C123
16 MHz
SW1
SW2
LED
Red
Port F Registers
DATA: 0x13
PUR: 0x00
LOCK: 0x00
DIR: 0x0E
PDR: 0x00
CR: 0x1E
DEN: 0x0E
RCGCGPIO: 0x00000021
Clock enabled
Q2.c
tm4c123gh6pm.h
startup_TM4C123.s
Command
*** Currently used: 1696 Bytes (5%)
UART #1
Enter a character: A
Enter a character:
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
Call Stack + Locals
Memory 1

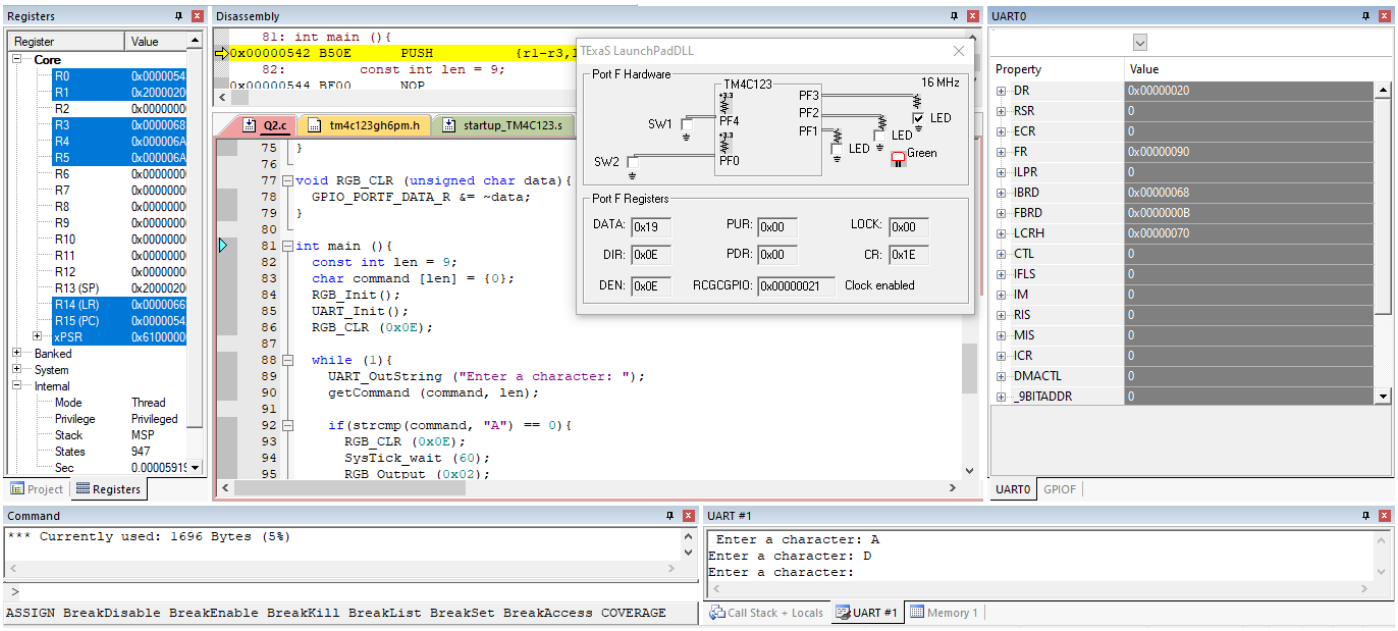
Sending “B”


Registers
Register
Value
Core
R0 0x00000054
R1 0x20000020
R2 0x00000000
R3 0x00000068
R4 0x0000006A
R5 0x0000006A
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x20000020
R14 (LR) 0x00000066
R15 (PC) 0x00000054
xPSR 0x61000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 947
Sec 0.00005919
Project
Registers
Disassembly
81: int main (){
0x00000542 B50E PUSH {r1-r3,lr}
82: const int len = 9;
0x00000544 BF00 NOP
Q2.c
tm4c123gh6pm.h
startup_TM4C123.s
75 }
76
77 void RGB_CLR (unsigned char data){
78 GPIO_PORTF_DATA_R &= ~data;
79 }
80
81 int main (){
82 const int len = 9;
83 char command [len] = {0};
84 RGB_Init();
85 UART_Init();
86 RGB_CLR (0x0E);
87
88 while (1){
89 UART_OutString ("Enter a character: ");
90 getCommand (command, len);
91
92 if(strcmp(command, "A") == 0){
93 RGB_CLR (0x0E);
94 SysTick_wait (60);
95 RGB_Output (0x02);
TExaS LaunchPadDLL
Port F Hardware
TM4C123
16 MHz
SW1
SW2
PF4
PF0
PF3
PF2
PF1
LED
Blue
Port F Registers
DATA: 0x15
PUR: 0x00
LOCK: 0x00
DIR: 0x0E
PDR: 0x00
CR: 0x1E
DEN: 0x0E
RCGCGPIO: 0x00000021
Clock enabled
GPIOF
Property
Value
DATA 0x00000015
DIR 0x0000000E
IS 0x00000000
IBE 0x00000000
IEV 0x00000000
IM 0
RIS 0
MIS 0
ICR 0
AFSEL 0x00000000
DR2R 0x000000FF
DR4R 0x00000000
DR8R 0x00000000
ODR 0x00000000
PUR 0x00000000
PDR 0x00000000
UART0
GPIOF
Command
*** Currently used: 1696 Bytes (5%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
UART #1
Enter a character: D
Enter a character: B
Enter a character:
Call Stack + Locals
UART #1
Memory 1


Registers
Register
Value
Core
R0 0x00000054
R1 0x20000020
R2 0x00000000
R3 0x00000068
R4 0x0000006A
R5 0x0000006A
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x20000020
R14 (LR) 0x00000066
R15 (PC) 0x00000054
xPSR 0x61000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 947
Sec 0.00005919
Project
Registers
Disassembly
81: int main (){
0x00000542 B50E PUSH {r1-r3,lr}
82: const int len = 9;
0x00000544 BF00 NOP
Q2.c
tm4c123gh6pm.h
startup_TM4C123.s
75 }
76
77 void RGB_CLR (unsigned char data){
78 GPIO_PORTF_DATA_R &= ~data;
79 }
80
81 int main (){
82 const int len = 9;
83 char command [len] = {0};
84 RGB_Init();
85 UART_Init();
86 RGB_CLR (0x0E);
87
88 while (1){
89 UART_OutString ("Enter a character: ");
90 getCommand (command, len);
91
92 if(strcmp(command, "A") == 0){
93 RGB_CLR (0x0E);
94 SysTick_wait (60);
95 RGB_Output (0x02);
TExaS LaunchPadDLL
Port F Hardware
TM4C123
16 MHz
SW1
SW2
PF4
PF0
PF3
PF2
PF1
LED
Blue
Port F Registers
DATA: 0x15
PUR: 0x00
LOCK: 0x00
DIR: 0x0E
PDR: 0x00
CR: 0x1E
DEN: 0x0E
RCGCGPIO: 0x00000021
Clock enabled
UART0
Property
Value
DR 0x00000020
RSR 0
ECR 0
FR 0x00000090
ILPR 0
IBRD 0x00000068
FBRD 0x0000000B
LCRH 0x00000070
CTL 0
IFLS 0
IM 0
RIS 0
MIS 0
ICR 0
DMACTL 0
_9BITADDR 0
UART0
GPIOF
Command
*** Currently used: 1696 Bytes (5%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
UART #1
Enter a character: D
Enter a character: B
Enter a character:
Call Stack + Locals
UART #1
Memory 1

Sending “D”





Video showing the LEDs with timer changing:

Click here