|  |  |
| --- | --- |
| Course Title: **Microprocessor and Microcontroller** | Course code**: CS43** |
| Lab Session: **6** | Student Name:  USN :  Date : |
| Title: Interface LED, RGB LED & Buzzer |
| Faculty Signature: | Marks(out of 10): |

**Objective:** Interface **NuMicro MCU Learning Board to program flicker LED and beep the buzzer.**

**Answer the following questions:**

|  |
| --- |
| **1) Write a C program to Light a LED connected to port C12.**  //  // Smpl\_GPIO\_LED : GPC12 to control on-board LEDs  // low-active output to control Red LEDs  //  #include <stdio.h>  #include "NUC1xx.h"  #include "Driver\DrvGPIO.h"  #include "Driver\DrvUART.h"  #include "Driver\DrvSYS.h"  void Init\_LED() // Initialize GPIO pins  {  DrvGPIO\_Open(E\_GPC, 12, E\_IO\_OUTPUT); // GPC12 pin set to output mode  DrvGPIO\_SetBit(E\_GPC, 12); // output Hi to turn off LED  }  int main (void)  {  UNLOCKREG(); // unlock register for programming  DrvSYS\_Open(48000000);// set to run at 48MHz  // 12MHz crystal input, PLL output 48MHz  LOCKREG(); // lock register from programming  Init\_LED(); // Initialize LEDs (four on-board LEDs)  while (1) // forever loop to keep flashing four LEDs one at a time  {  DrvGPIO\_ClrBit(E\_GPC, 12); // output Low turn on LED  DrvSYS\_Delay(30000); // delay  DrvGPIO\_SetBit(E\_GPC, 12); // output Hi turn off LED  DrvSYS\_Delay(300000); // delay  }  }    **2. Write a C program to Light a LEDs connected to port C12-15 using macros.**  #include <stdio.h>  #include "NUC1xx.h"  #include "Driver\DrvGPIO.h"  #include "Driver\DrvUART.h"  #include "Driver\DrvSYS.h"  #define INIT\_LED0 DrvGPIO\_Open(E\_GPC, 12, E\_IO\_OUTPUT)  #define INIT\_LED1 DrvGPIO\_Open(E\_GPC, 13, E\_IO\_OUTPUT)  #define INIT\_LED2 DrvGPIO\_Open(E\_GPC, 14, E\_IO\_OUTPUT)  #define INIT\_LED3 DrvGPIO\_Open(E\_GPC, 15, E\_IO\_OUTPUT)  #define LED0\_ON DrvGPIO\_ClrBit(E\_GPC, 12)  #define LED0\_OFF DrvGPIO\_SetBit(E\_GPC, 12)  #define LED1\_ON DrvGPIO\_ClrBit(E\_GPC, 13)  #define LED1\_OFF DrvGPIO\_SetBit(E\_GPC, 13)  #define LED2\_ON DrvGPIO\_ClrBit(E\_GPC, 14)  #define LED2\_OFF DrvGPIO\_SetBit(E\_GPC, 14)  #define LED3\_ON DrvGPIO\_ClrBit(E\_GPC, 15)  #define LED3\_OFF DrvGPIO\_SetBit(E\_GPC, 15)  #define DELAY DrvSYS\_Delay(300000)  // Initial GPIOs  void Init\_LED()  {  // initialize GPIO pins to OUTPUT mode  INIT\_LED0;  INIT\_LED1;  INIT\_LED2;  INIT\_LED3;  // set GPIO pins to output Low  LED0\_OFF;  LED1\_OFF;  LED2\_OFF;  LED3\_OFF;  }  int main (void) {  UNLOCKREG(); // unlock register for programming  DrvSYS\_Open(48000000);// set to run at 48MHz  LOCKREG(); // lock register from programming  Init\_LED(); // Initialize LEDs (four on-board LEDs)  while(1) {  LED0\_ON;  DELAY;  LED0\_OFF;  DELAY;  LED1\_ON;  DELAY;  LED1\_OFF;  DELAY;  LED2\_ON;  DELAY;  LED2\_OFF;  DELAY;  LED3\_ON;  DELAY;  LED3\_OFF;  DELAY;  }  } |
| **3. Write a C program to Light a RGBLED connected to port A12-14.**  //  // Smpl\_GPIO\_RGBled : GPA12,13,14 output control RGB LED  // output low to enable LEDs  #include <stdio.h>  #include "NUC1xx.h"  #include "Driver\DrvGPIO.h"  #include "Driver\DrvUART.h"  #include "Driver\DrvSYS.h"  // Initial GPIO pins (GPA 12,13,14) to Output mode  void Init\_LED()  {  // initialize GPIO pins  DrvGPIO\_Open(E\_GPA, 12, E\_IO\_OUTPUT); // GPA12 pin set to output mode  DrvGPIO\_Open(E\_GPA, 13, E\_IO\_OUTPUT); // GPA13 pin set to output mode  DrvGPIO\_Open(E\_GPA, 14, E\_IO\_OUTPUT); // GPA14 pin set to output mode  // set GPIO pins output Hi to disable LEDs  DrvGPIO\_SetBit(E\_GPA, 12); // GPA12 pin output Hi to turn off Blue LED  DrvGPIO\_SetBit(E\_GPA, 13); // GPA13 pin output Hi to turn off Green LED  DrvGPIO\_SetBit(E\_GPA, 14); // GPA14 pin output Hi to turn off Red LED  }  int main (void)  {  UNLOCKREG(); // unlock register for programming  DrvSYS\_Open(48000000); // set System Clock to run at 48MHz (PLL with 12MHz crystal input)  LOCKREG(); // lock register from programming  Init\_LED();    while (1)  {  // GPA12 = Blue, 0 : on, 1 : off  // GPA13 = Green, 0 : on, 1 : off  // GPA14 = Red, 0 : on, 1 : off  // set RGBled to Blue  DrvGPIO\_ClrBit(E\_GPA,12); // GPA12 = Blue, 0 : on, 1 : off  DrvGPIO\_SetBit(E\_GPA,13);  DrvGPIO\_SetBit(E\_GPA,14);  DrvSYS\_Delay(1000000);  // set RGBled to Green  DrvGPIO\_SetBit(E\_GPA,12);  DrvGPIO\_ClrBit(E\_GPA,13); // GPA13 = Green, 0 : on, 1 : off  DrvGPIO\_SetBit(E\_GPA,14);  DrvSYS\_Delay(1000000);  // set RGBled to Red  DrvGPIO\_SetBit(E\_GPA,12);  DrvGPIO\_SetBit(E\_GPA,13);  DrvGPIO\_ClrBit(E\_GPA,14); // GPA14 = Red, 0 : on, 1 : off  DrvSYS\_Delay(1000000);    // set RGBled to off  DrvGPIO\_SetBit(E\_GPA,12); // GPA12 = Blue, 0 : on, 1 : off  DrvGPIO\_SetBit(E\_GPA,13); // GPA13 = Green, 0 : on, 1 : off  DrvGPIO\_SetBit(E\_GPA,14); // GPA14 = Red, 0 : on, 1 : off  DrvSYS\_Delay(1000000);  }  } |
| **4. Write a C program to beep a buzzer connected to port B11.**  //  // Smpl\_GPIO\_Buzzer : GPB11 low-active output control Buzzer  // Note: Nu-LB-NUC140 R1 should be 0 ohm  //  #include <stdio.h>  #include "NUC1xx.h"  #include "Driver\DrvSYS.h"  #include "Driver\DrvGPIO.h"  #include "Driver\DrvADC.h"  int main (void)  {  UNLOCKREG(); // unlock register for programming  DrvSYS\_Open(48000000); // set System Clock to run at 48MHz  LOCKREG(); // lock register from programming  DrvGPIO\_Open(E\_GPB, 11, E\_IO\_OUTPUT); // initial GPIO pin GPB11 for controlling Buzzer  while(1) {  DrvGPIO\_ClrBit(E\_GPB,11); // GPB11 = 0 to turn on Buzzer  DrvSYS\_Delay(100000); // Delay  DrvGPIO\_SetBit(E\_GPB,11); // GPB11 = 1 to turn off Buzzer  DrvSYS\_Delay(100000); // Delay  }  } |