Driver Project 2: Basic GPIO Control using C programing and Hardware (Group Project) – 20 points

Assignment Due as per D2L

Overview: The aim of this assignment is to become familiar with the hardware kit, ensure proper kit wiring, MPLAB X IDE, and writing basic C code to control the I/O.

Assignment: Using the PIC 24F, you will design a simple IO controller to “test” the hardware kit provided. Design a state machine to turn on, turn off and blink an LED connected to port RB8 based on the push buttons (PBs) connected to the input ports RA2, RA4 and RB4 as shown in the schematic in the lecture slide ‘HW and IO Control.pdf’. PB1, PB2 and PB3 represent push buttons connected to ports RA2, RA4 and RB4 respectively. The state machine should operate as follows:

User input(s)	**Output(s)**
While PB1 is pressed	LED blinks at approximately 1 sec interval (1 sec on and 1 sec off)
While PB2 is pressed	LED blinks at approximately 2 sec interval (2sec on and 2 sec off)
While PB3 is pressed	LED blinks at approximately 3 sec interval (3sec on and 3 sec off)
While 2 or more PBs are pressed together	LED stays on without blinking
No PBs pressed	LED stays off

Additional info:

- Implement the above controller using the hardware kit provided
- **Implement your code using the sample Code template provided on D2L**. You will design your code using basic ANSI C commands. You will not use Timer functions or peripherals for this assignment – Timers and built in delay functions will be covered later in class.
- Your code organization can be as per your preference.
- Note: To use any of the digital IO (RAx or RBx) that is multiplexed with the analog pins (AN1-AN12), the corresponding PCFGx bit in the AD1PCFG SFR must be set to “1” as shown in sample code.
- ***Hint on generating delay cycles:*** The default clock speed of the microcontroller is 8MHz. The program counter increments from one instruction to the next at every other clock cycle (i.e. 4 MHz). The number of clock cycles needed per instruction varies between 2 and 20 for different types of instructions with the free C compiler used.

Deliverables:

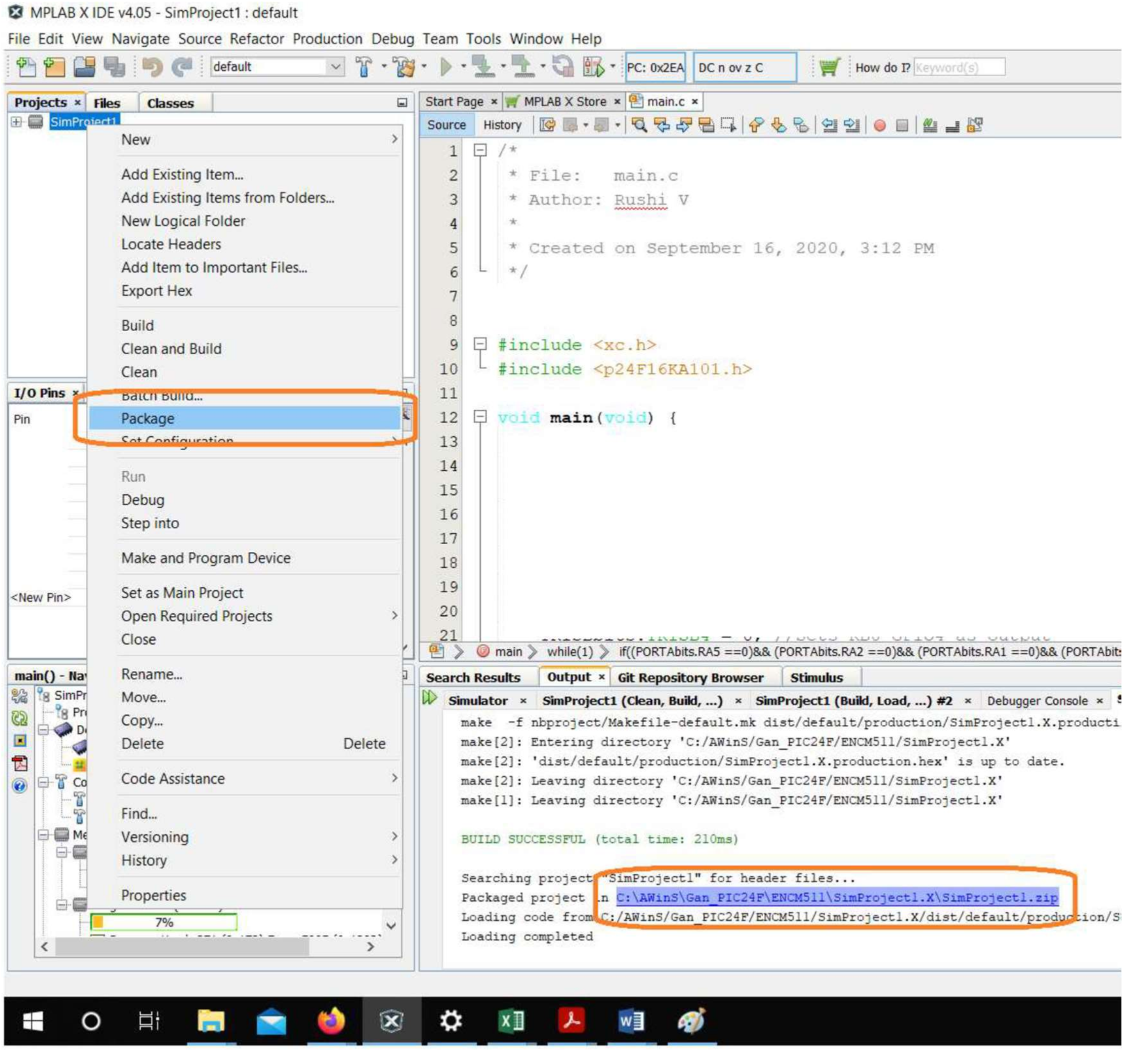
This is a group project. Each group should upload the following 2 items onto their respective group D2LDropbox folder created:

1. Link to a single video recording **<u>no longer than 5 mins</u>** long showing the following:
    a. UCID card of one group member placed in front of the computer with MPLAB running
    b. Demo of the code and hardware operation showing the following:
        i. Each of the PBs pressed individually
        ii. 2 PBs and 3 PBs pressed simultaneously
        iv. No PBs pressed
        v. Repeat with each of the PBs pressed individually
    c. Demo should contain a spoken explanation of what is being performed by one of the group members. For example: “I ‘Group member name’ am pressing PB1 connected to RA2 and you

can see LED blinking for xx seconds. Next I am pressing PB2 connected to ….. My IO state machine is in function names ____ placed in source file ___.c ”

NOTE: Video recordings of the HW and computer screen should be made using a smartphone or camera. Screenshots and screen recordings of individual segments using Yuja or Zoom or any other software will lose points. Each group should upload 1 link to the video recording.

2. Zipped up file of the project. MPLAB projects can be zipped up by right clicking on the project and selecting package (See screenshot below). The zipped project is saved in the same project folder created by user on user's computer.


MPLAB X IDE v4.05 - SimProject1 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
default
PC: 0x2EA
DC n ov z C
How do I? Keyword(s)
Projects × Files Classes
SimProject1
New
Add Existing Item...
Add Existing Items from Folders...
New Logical Folder
Locate Headers
Add Item to Important Files...
Export Hex
Build
Clean and Build
Clean
Batch Build...
Package
Set Configuration
Run
Debug
Step into
Make and Program Device
Set as Main Project
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Code Assistance
Find...
Versioning
History
Properties
I/O Pins ×
Pin
<New Pin>
main() - Nav
7%
Start Page × MPLAB X Store × main.c ×
Source History
main
while(1)
if((PORTAbits.RA5 ==0)&& (PORTAbits.RA2 ==0)&& (PORTAbits.RA1 ==0)&& (PORTAbit
Search Results Output × Git Repository Browser Stimulus
Simulator × SimProject1 (Clean, Build, ...) × SimProject1 (Build, Load, ...) #2 × Debugger Console ×

```
/*
 * File:   main.c
 * Author: Rushi V
 *
 * Created on September 16, 2020, 3:12 PM
 */


#include <xc.h>
#include <p24F16KA101.h>

void main(void) {
```

```
make  -f nbproject/Makefile-default.mk dist/default/production/SimProject1.X.producti
make[2]: Entering directory 'C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X'
make[2]: 'dist/default/production/SimProject1.X.production.hex' is up to date.
make[2]: Leaving directory 'C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X'
make[1]: Leaving directory 'C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X'

BUILD SUCCESSFUL (total time: 210ms)

Searching project "SimProject1" for header files...
Packaged project in C:\AWinS\Gan_PIC24F\ENCM511\SimProject1.X\SimProject1.zip
Loading code from C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X/dist/default/production/S
Loading completed
```

Grading rubric

-Correct Peripheral setup, Program logic and working of all IO states = 16 points (4 points per state)
-Code (including properly uploaded code with comments) = 2 points
-Proper video demo (includes one UCID card displayed, meeting of demo time limit, brief explanation of Hardware and software operation) = 2 points

Driver Project Generic Rubric – Fall 2023

	Fails to meet expectations	**Minimally meets expectations**	**Adequately Meets Expectations**	**Exceeds Expectations**	**Score awarded**
Peripheral Configuration and Use	None of the peripherals and states working correctly **0 to 2 points**	Some of the peripherals and states working correctly **4 points**	Most of the peripherals and states working correctly **6 points**	All peripherals and states correctly working **8 points**	
Program Logic	Does not provide evidence of appropriate program flow **0 to 2 points**	Provides evidence of attempting to use C control statements but has some errors or does not cover all scenarios **4 points**	Provides evidence of attempting to use C control statements to cover most but not all scenarios or has some errors **6 points**	Provides evidence of appropriate C control statements and implements all scenarios correctly **8 points**	
Code Quality	No evidence of code commenting or reasonable variable names **0 points**	Some evidence of code comments, but infrequent or incomplete **1 point**	Evidence that most elements of the code are commented **1.5 points**	Evidence that all important elements of the code are commented meaningfully **2 points**	
Demonstration and Video Upload	Demo does not appear to work or does not align with project specifications or Video not provided **0 points**	Demo does not cover all scenarios or no explanation provided **1 point**	Demo shows the design working, covering most states but not all states or proper explanation for all states not provided or Video runs over time limit. **1.5 points**	Demo shows the design working, covering all states. Video meets time limit and contains proper explanation of hardware and software operation **2 points**	
				Total (20):	

Overall

Fails to meet expectations 0 - 8

Minimally meets expectations 9 - 12

Adequately Meets Expectations 13 - 17

Exceeds Expectations 18 - 20