Driver Project 5: Analog to Digital Conversion (Group Project)

Due Date: as per D2L

Assignment:

In this project, you will verify the working of the Microcontroller's ADC using the potentiometer connected to IO pin 8 (RA3). Design function uint16_t do_ADC(void) in source files ADC.c and ADC.h to perform the following tasks.

1. Perform Analog to Digital conversion on the analog voltage that the potentiometer induces on pin 8 (RA3) of the Microcontroller
2. Generate a bar graph display on a single line of the PC terminal. The width of the bar should change in proportion to the ADC's digital output value.
3. Displays the ADC's digital output value in Hex at the end of the bar graph on the PC terminal.

   For instance, with the Potentiometer turned all the way in one direction, the PC terminal should display the following on the 1st line:

   ********** **0x3FF**

   For instance, with the Potentiometer turned all the way in the other direction, the PC terminal should display the following on the 1st line:

   *** 0x000**

   For instance, with the Potentiometer turned somewhere in the middle should display the following on the 1st line of the PC terminal:

   ********** **0x135**

You can use any clearly visible ASCII symbol and any proportionate horizontal scale to generate the Bar graph.

4. Returns the value of ADC's digital output to the calling function

Additional info:

- Implement the above controller using the hardware kit and your code, which will be designed using basic ANSI C commands; Microcontroller's ADC and prior driver functions provided/designed.

- Function names: ADC function should be named uint16_t do_ADC(void) and placed in source files ADC.c and ADC.h
- Display instructions: **All displays on the PC terminal window should be on a single line**. **Note that display functions carried out at 32 kHz (300 Baud) can the display speed on the PC terminal. Your code should account for such delays including use of proper clock switching in order to produce Bar graph display clearly and quickly enough on the PC terminal.**

Note:

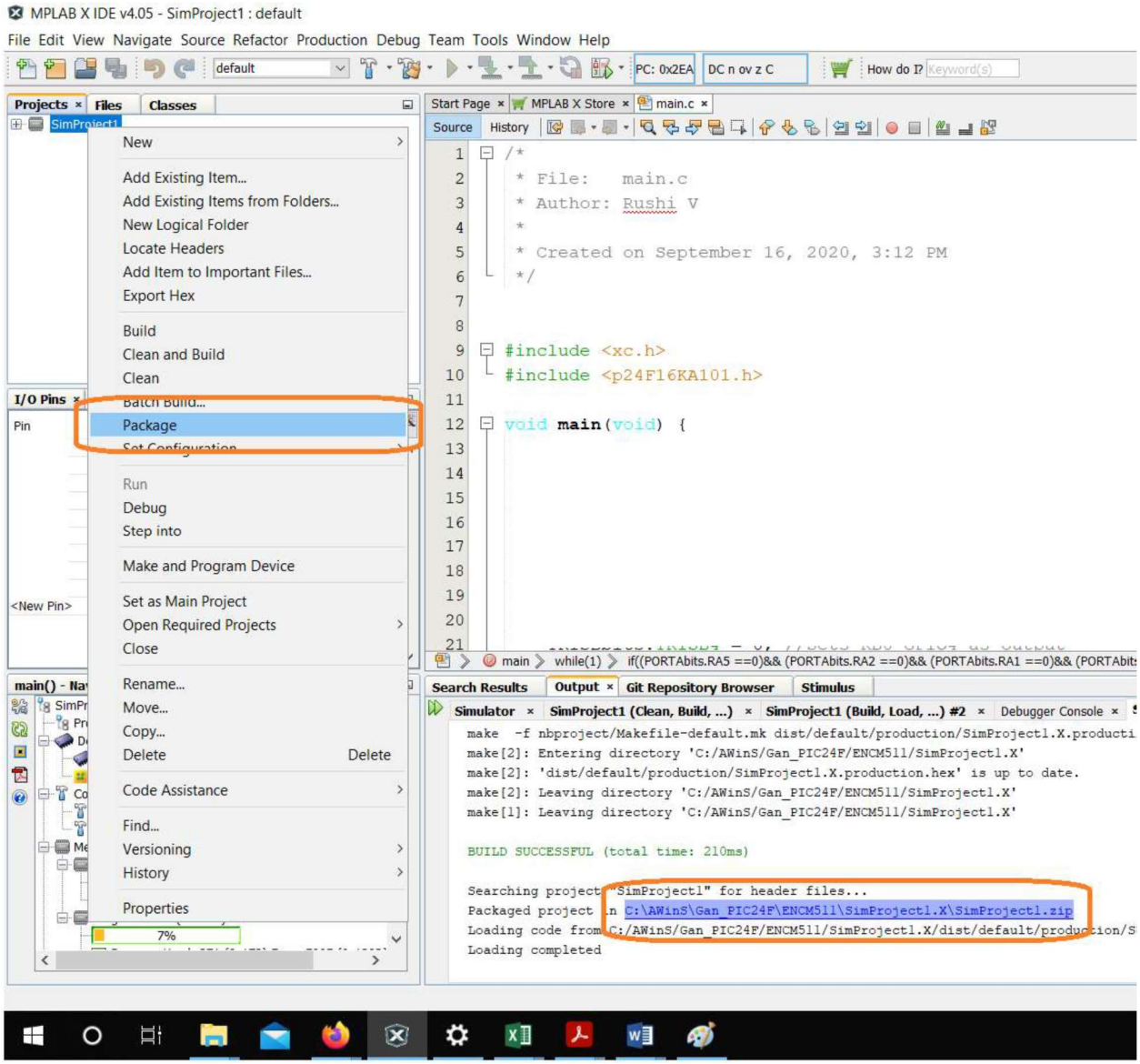
Port RA2 is one of those exceptional ports that is also multiplexed to the input for an external oscillator and an analog input port. To be able to use it as a digital input with a pushbutton, it's multiplexed analog input has to be disabled by including the following line of code in your IOinit() function.

```
AD1PCFG = 0xFFFF; // Turn all analog pins as digital
```

Deliverables:

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

1. **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. Make sure your driver code is commented properly especially any mathematical computations used.**
2. **Link to your video demo uploaded on youtube, Vimeo or similar video hosting website along with the zipped up project. Dropbox or Google or OneDrive links are allowed as well but ensure that videos are in .mp4 or .mov format. Videos uploaded in any other format will lose points. Video demo should be as follows:**
    a. **Single recording no more than 2 mins long**
    b. **Show UCID or government issued ID cards of all 3 group members placed in front of the computer with MPLAB and/or hardware running**
    c. **Demo of the code and hardware operation showing the following:**
        i. **The potentiometer being turned from its minimal to its maximal resistance value**
        ii. **The bar graph width adjusting with respect to potentiometer's turn**
        iii. **ADC's digital output value in Hex displayed at the end of the bar graph**
    d. **Explanation of the code organization in MPLAB including any special power or time saving features (i.e. interrupts, clock switching, sleep/idle) used, and respective contribution of each group member towards code development and hardware/software testing.**


MPLAB X IDE v4.05 - SimProject1 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
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
Run
Debug
Step into
Make and Program Device
Set as Main Project
Open Required Projects
Close
Rename...
Move...
Copy...
Delete
Code Assistance
Find...
Versioning
History
Properties
I/O Pins
Pin
<New Pin>
Start Page
MPLAB X Store
main.c
Source
History
/*
* File: main.c
* Author: Rushi V
*
* Created on September 16, 2020, 3:12 PM
*/
#include <xc.h>
#include <p24F16KA101.h>
void main(void) {
Search Results
Output
Git Repository Browser
Stimulus
Simulator
SimProject1 (Clean, Build, ...)
SimProject1 (Build, Load, ...) #2
Debugger Console
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
7%

Grading rubric (Total 20 points)

1. Correct setup and use of timers, IOs, UART, ADCs, interrupts and clock modules as follows: 16 points
---- Properly displays the bar graph for the lowest values of the Potentiometer and ADC's output
---- Properly displays the bar graph for the highest values of the Potentiometer and ADC's output
---- Properly displays the changing bar graph and ADC output value as the Potentiometer is adjusted
---- Code optimization for better power savings and display speed i.e. clock switching, interrupts, sleep/idle modes for speedy message displays and power savings
2. Code (including properly uploaded code with comments) = 2 points
3. 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):	

<u>Overall</u>

Fails to meet expectations 0 - 8

Minimally meets expectations 9 - 12

Adequately Meets Expectations 13 - 17

Exceeds Expectations 18 - 20