

Lab 3 WriteUp

The write up consists of the verification steps taken during the course of the project.

8051 External Memory Editor, Paul Stoffregen, 1996																		
DATA	ADDR:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII EQUIVILANT
0000:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0010:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0020:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0030:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0040:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0050:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0060:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0070:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0080:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
0090:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
00A0:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
00B0:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
00C0:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
00D0:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
00E0:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU
00F0:	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	UUUUUUUUUUUUUUUUUUUU

To check the memory whether it is being accessed from 0000 to 7FFF, we wrote 55 on the same address, and obtained the above results. After pushing the reset button, we got the result below, as from 0000 to 03FF, we want our memory to be in internal ram, so after pressing reset, all the address will be cleared from 0000 to 03FF.

8051 External Memory Editor, Paul Stoffregen, 1996																		
DATA	ADDR:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII
0100:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0110:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0120:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0130:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0140:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0150:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0160:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0170:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0180:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0190:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
01A0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
01B0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
01C0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
01D0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
01E0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
01F0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

^E=Edit	^G=Goto	^C=Code	^D=Data	^L=Redraw	^Q=Quit
---------	---------	---------	---------	-----------	---------

```
Welcome to PAULMON2 v2.1, by Paul Stoffregen

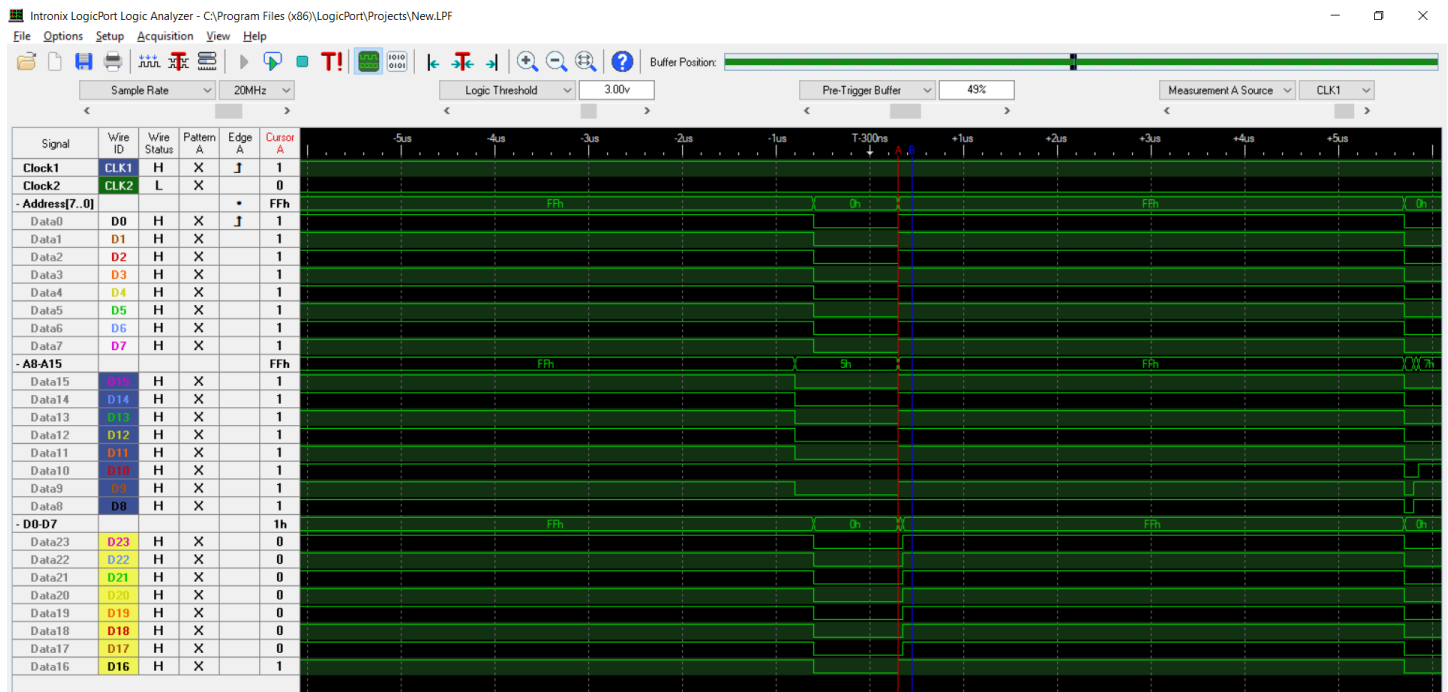
See PAULMON2.DOC, PAULMON2.EQU and PAULMON2.HDR for more info

Program Name      Location      Type
List              1000        External command
Single-Step       1400        External command
Memory Editor <UT100> 1800        External command

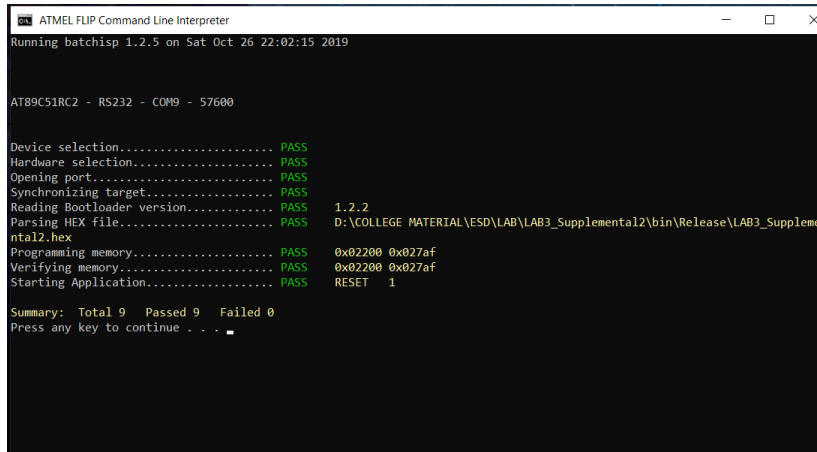
PAULMON2 Loc:2200 > Jump to memory location
Jump to memory location <2200>, or ESC to quit:
```

Since, we didn't want to overwrite the paulmon code, we kept our 8051 heap allocation and execution code at 2200 address, which is shown below. The code was verified by Dominic Doty during the signoff.

For more verification, we used the debug port code and used the logic analyzer to check whether the address which aren't being addressed are written. We wrote 01 to the address FFFF. The image is shown below



For ease of operation, a batch file was created to upload the code to the AT89C51. The below code is written for supplemental 2.



```
AT89C51RC2 - RS232 - COM9 - 57600

Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Synchronizing target..... PASS
Reading Bootloader version..... PASS 1.2.2
Parsing HEX file..... PASS D:\COLLEGE MATERIAL\ESD\LAB\LAB3_Supplemental2\bin\Release\LAB3_Supplemental2.hex
Programming memory..... PASS 0x02200 0x027af
Verifying memory..... PASS 0x02200 0x027af
Starting Application..... PASS RESET 1

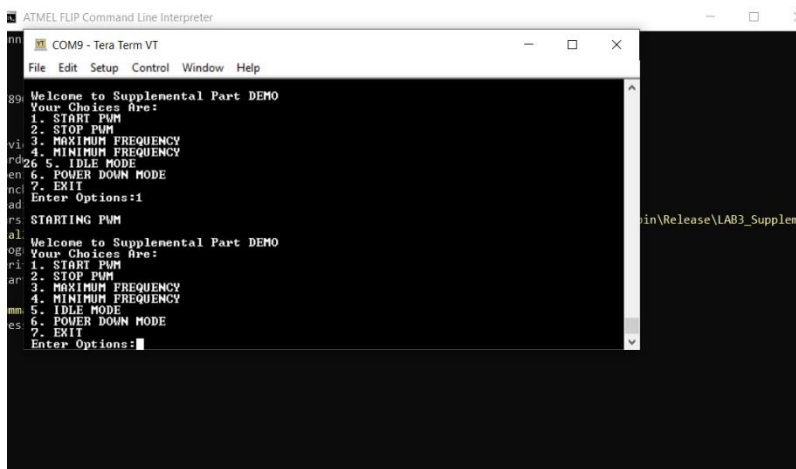
Summary: Total 9 Passed 9 Failed 0
Press any key to continue . . .
```

The whole MSP432 Required Code was written based on the examples given in the resource folder example and was signed off by Tanmay Chaturvedi. Some of the observations include:

- It was observed that if we use the uart to change the pwm duty cycle, then it was much smoother than the changing the pwm duty cycle through switches. This might be due to usage of hardware interrupts in the switches while software interrupts were used in the Keyboard part. (Still to be verified by the TA).
- While doing the 89C51 supplement, there were some bugs experienced but were quickly solved. They were usually because of some silly mistakes like not using the refresh function for Watchdog timer, or not setting the proper value while refreshing.
- One more bug was that while using X2 mode, the display on the UART wasn't proper, i.e the baud rate required was different and wasn't set automatically. It was needed to be changed manually, it still needs some rectified as the supplemental was done after the signoff.
- Also, after setting the minimum frequency, the program stopped on the uart while, the pwm was showing at the minimum frequency.

This work is done till 10/26/2019 and more modification will be informed to the TA as soon as possible.

Screenshots



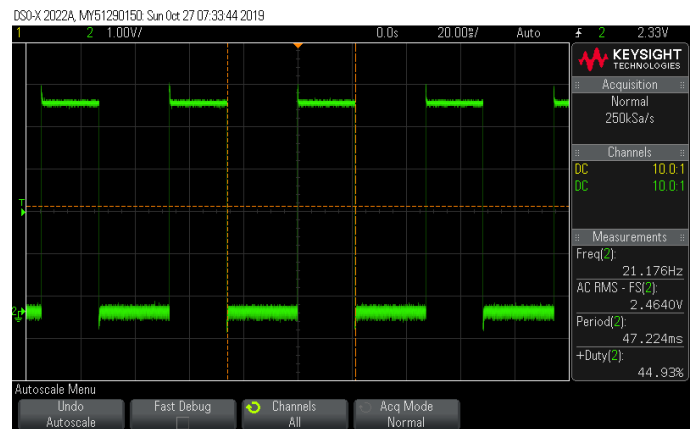
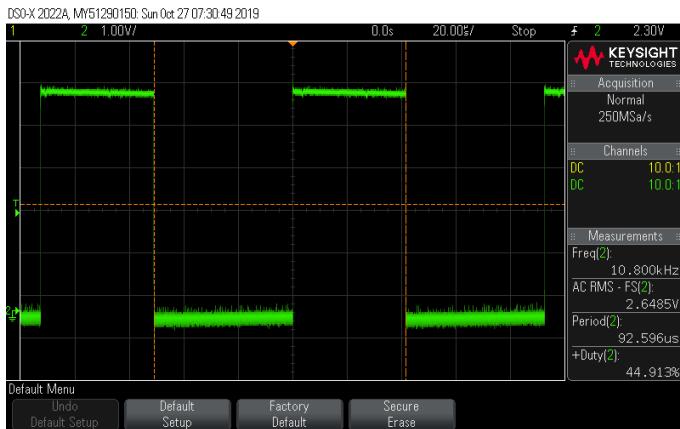
```
AT89C51RC2 - RS232 - COM9 - 57600

Welcome to Supplemental Part DEMO
Your Choices Are:
1. START PWM
2. STOP PWM
3. MAXIMUM FREQUENCY
4. MINIMUM FREQUENCY
5. IDLE MODE
6. POWER DOWN MODE
7. EXIT
Enter Options:1
STARTING PWM

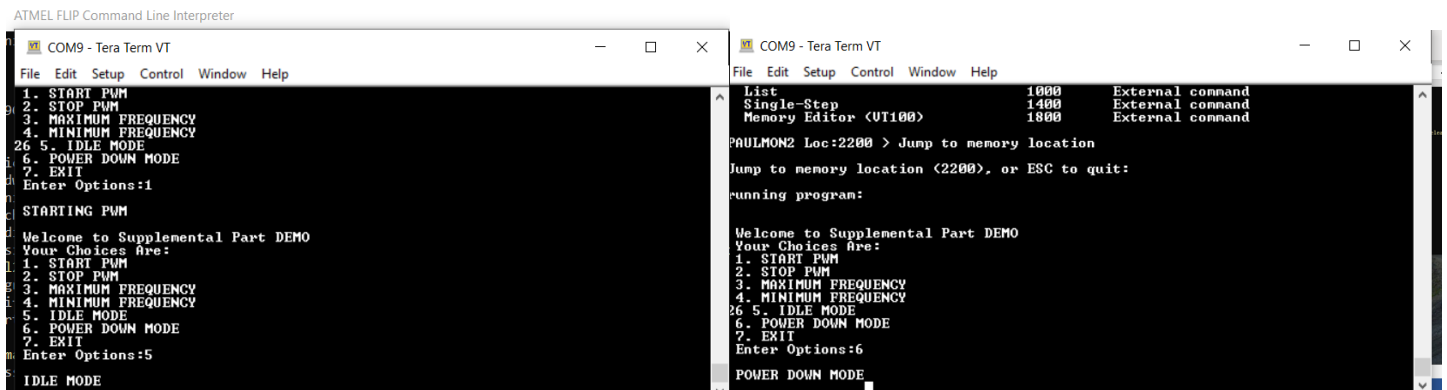
Welcome to Supplemental Part DEMO
Your Choices Are:
1. START PWM
2. STOP PWM
3. MAXIMUM FREQUENCY
4. MINIMUM FREQUENCY
5. IDLE MODE
6. POWER DOWN MODE
7. EXIT
Enter Options:
```

The image is the User interface for the 89C51 supplemental of different choice given to the user.

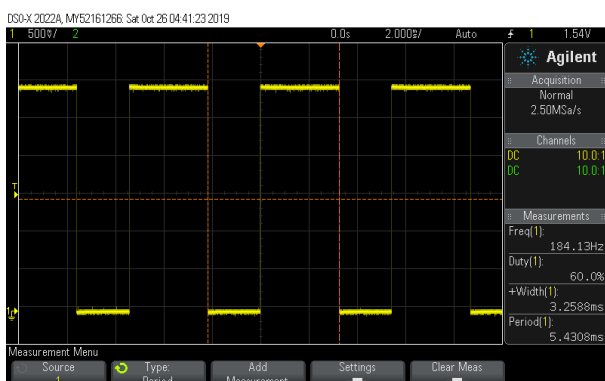
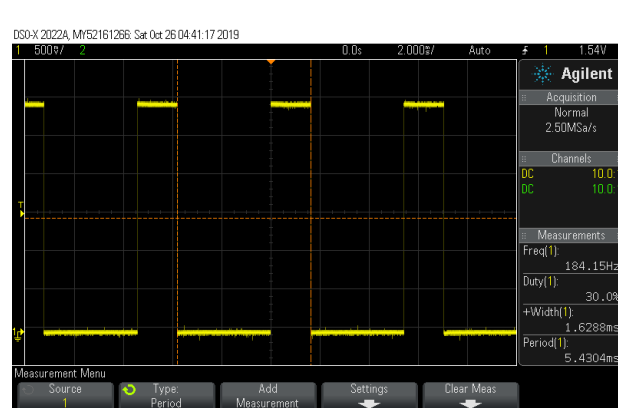
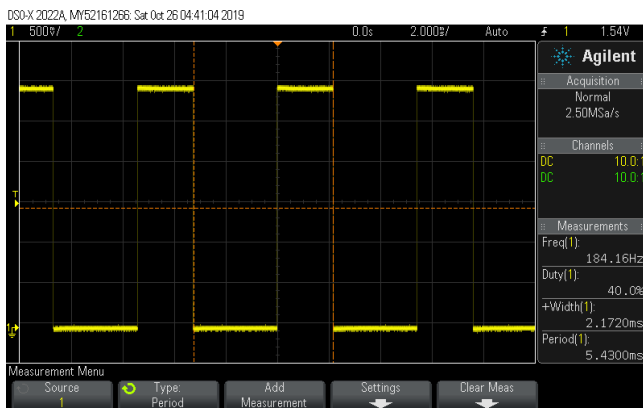
It can clearly viewed that after starting the PWM, the program is still running.



The Above screenshots are for pwm for the maximum and minimum Clock frequency , respectively.(Supplemental)



The above pictures are for the IDLE and Power Down mode, it can be clearly seen that the program has stopped, and requires external interrupt or rest for accessing the board again.



The Three images are for the PWM generated using the MSP432, First at the top left is the default 40%, right to it is the PWM after decreasing it to 30% and the bottom one is the increased duty cycle of the PWM at 60%, it was done by both switches and keyboard.

LAB 3 Part 2 Answers

1. Operating System used for programming: Windows 10.
2. Compiler Used: SDCC 3.9.0
3. IDE Used: Code::Blocks without custom makefile. For ARM, Code Composer Studio.
4. Any other Software: No other software required
5. The global variables were not initializing and faced a lot of problems during compilations and running of the program, in sdcc.

SIGNOFF SHEET

ECEN 5613

Lab #3 Signoff Sheet

Fall 2019

You will need to obtain the signature of your instructor or TA on the following items in order to receive credit for your lab assignment. This assignment is due by **Friday, October 18, 2019 (Part 1 Required Elements)** and **Friday, October 25, 2019 (Part 2 Required and Supplemental Elements)**.

Print your name below, sign the honor code pledge, circle your course number, and then demonstrate your working hardware & firmware in order to obtain the necessary signatures.

Student Name: Nitish Satish Gupta

Honor Code Pledge: "On my honor, as a University of Colorado student, I have neither given nor received unauthorized assistance on this work. I have clearly acknowledged work that is not my own."

Student Signature: [Signature]

Signoff Checklist

Part 1 Required Elements

- ☒ Schematic of acceptable quality (all components shown)
- ☒ Pins and signals labeled, decoupling capacitors, and two 28-pin wire wrap sockets present on board
- ☒ Very good knowledge of a terminal emulator
- ☒ Demonstrates all 32KB of XRAM in memory map are functional, including monitor block fill command
- ☒ Using PAULMON2, demonstrates highest baud rate as: 57600
- ☒ Knows how to use SDCC [IDE or make optional]
- ☒ Knows how to analyze output files (.RST, .MEM, .MAP) for correct addresses
- ☒ C serial program and virtual debug port functional and code commented
- ☒ Hex display of buffer contents PC804 FUNCTIONAL

Part 2 Required and Supplemental Elements

- ☒ ARM code integration and execution → 10/25/2019
- ☒ 8051 PWM control works correctly, X2 mode
- ☒ Correctly enters Idle mode and exits via external interrupt 1
- ☒ Correctly enters Power Down mode
- ☒ All other PCA software menu items function correctly
- ☒ Good understanding of PCA modes
- ☒ Good user interface; program is easy to use

Instructor/TA Comments: ☐ ☐ ☐

TA signature and date: [Signature] 10/18/19

TA signature and date: [Signature] 10/25/19

FOR INSTRUCTOR USE ONLY

Part 1 Elements

	Not Applicable	Below Expectation	Meets Requirements	Exceeds Requirements	Outstanding
Schematics, SPLD code	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Hardware physical implementation	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Part 1 Required Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sign-off done without excessive retries	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Student understanding and skills	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Overall Demo Quality (Part 1 elements)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

FOR INSTRUCTOR USE ONLY

Part 2 Elements

	Not Applicable	Below Expectation	Meets Requirements	Exceeds Requirements	Outstanding
Part 2 Required Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Supplemental Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Student understanding and skills	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Overall Demo Quality (Part 2 elements)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Comments:

- ☒ Optional Challenge: PAULMON2 RUN command
- ☐ Optional Challenge: ISP API calls
- ☐ Optional Challenge: C and Assembly interfacing
- ☐ Optional Challenge: Serial ISR
- ☐ Optional Challenge: SDCC heap memory management analysis

10/23/2019
[Signature]
Partial

- 10/18/19 PT1
- + BATCH ISP SCRIPT FOR PROGRAMMING
 - ONLY ALLOCATES BUFF NO BUF1
 - HEX DUMP IN ASCII NOT HEX
 - NO PROTECTION ON WRITING PAST ARRAY END

10/25/19 Part 2

- (*) NO supplemental for 8051
- (+) PWM +/- for MSP432 [supplemental]
- (+) UART echo works, Temp in °C, °F [supplemental]

10/28/19

- (+) PWM for 8051 works.
- (+) 80 timer for 8051 PCA
- ~~(-) NO UI for WDT~~
- (-) ~~NO~~ UI for PWM, fCLK, IDLE, Powerdown in a separate code.