

ESD LAB-3 REPORT

ECEN 5613

Lab #3 Signoff Sheet

Fall 2021

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. 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: Swapnil Ghonge

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: Swapnil Ghonge

Signoff Checklist

Part 1 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: 9600
- ☒ Knows how to use SDCC [IDE or make optional]

P. Tahmami

03/04/2022

TA signature and date

Part 2 Elements

- ☒ 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

P. Tahmami

03/04/2022

TA signature and date

Part 3 Required and Supplemental Elements

- ☒ Required ARM code integration and execution
- ☒ 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

Mansoor M.D.

03/12/22

TA signature and date

Instructor/TA Comments: ☐ ☐ ☐

FOR INSTRUCTOR USE ONLY

Part 1 and 2 Elements

	Not Applicable	Below Expectation	Meets Requirements	Exceeds Requirements	Outstanding
Schematics, SPLD code	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Hardware physical implementation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Part 1 Required Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input 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 type="checkbox"/>	<input type="checkbox"/>
Student understanding and skills	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Overall Demo Quality (Part 2 elements)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

FOR INSTRUCTOR USE ONLY

Part 3 Elements

	Not Applicable	Below Expectation	Meets Requirements	Exceeds Requirements	Outstanding
Part 3 Required Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Supplemental Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Student understanding and skills	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Overall Demo Quality (Part 3 elements)	<input type="checkbox"/>	<input type="checkbox"/>	<input 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

Comments

- [.] Add \overline{WR} connection for NVSRAM
- [.] 2496 case does not work, @ does not go back to buffer 0 entry, ? does not transmit storage characters and does not clear buffer 0.
- [+] Board has all decaps.
- [+] Virtual debug is fully functional.

Comments Part 3

- (+) ARM code functional on MSP432.
- (+) Supplemental completed
 - i) RGB spectrum
 - ii) Command for PWM duty cycle.
- (-) Need to work on the User Interface. Provide an option to list available options.
- (+) Power down & Idle mode verified
- (+) Requir PCA mode tests
- (+) WDT functional.
- (+) High speed toggle mode completed.
- (+) Good UI for 8051 program.

Part-1 & 2

Things Learnt:

Learnt how to use internal and external memory management.

Learnt how to use serial drivers for memory

Learnt how to change the values in memory location both for Internal XRAM and external XRAM.

Learnt how to use SDCC for C compilers

Learnt how to change how to use baud rate on Tera Term.

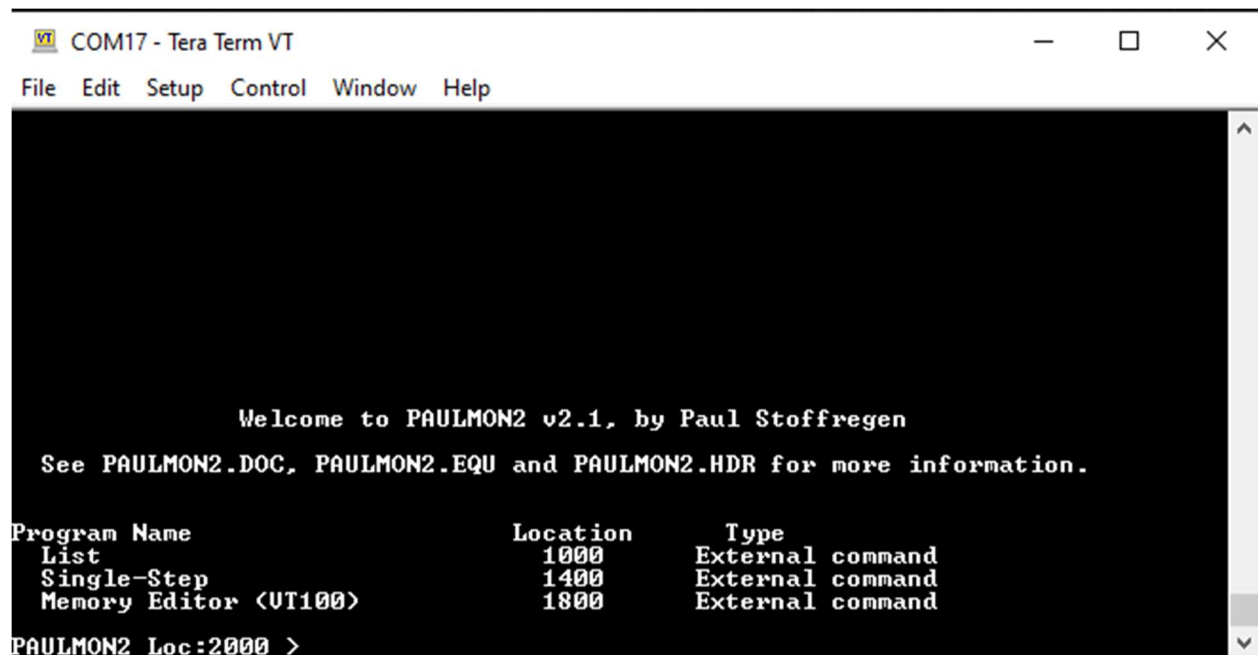
Learnt how to efficiently use FLIP program to program AT89C51RC2.

Learnt how to allocate memory of Heap Size 5000 bytes using malloc function

Learnt how to use UART to which accepts command a function.

Learnt how to free a buffer.

Learnt how to change a buffer using +,-,@,? UART commands



The screenshot shows a Tera Term window titled "COM17 - Tera Term VT". The window contains the following text:

```

Welcome to PAULMON2 v2.1, by Paul Stoffregen
See PAULMON2.DOC, PAULMON2.EQU and PAULMON2.HDR for more information.

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

PAULMON2 Loc:2000 >

```

```
COM17 - Tera Term VT
File Edit Setup Control Window Help

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

PAULMON2 Loc:2000 > Jump to memory location
Jump to memory location <2000>, or ESC to quit: 2000
running program:
Enter a value divisible btw 48 and 4800 and divisible by 16
a= 0
b= 0
c = 6
d = 4
number = 64
valid input
num= 64the number is 64malloc for buffer0 is successful
starting address of buffer 0 is 402
malloc for buffer1 is successful
starting address of buffer 1 is 444
```

Allocating Buffer of size 64 bytes

```
num= 64the number is 64malloc for buffer0 is successful
starting address of buffer 0 is 402
malloc for buffer1 is successful
starting address of buffer 1 is 444
Enter a character
the input character is 27
Enter a character
0the input character is 79
Enter a character
-the input character is 45

Enter a valid buffer number
enter the number of buffer to delete
Buffer 1>1092
Buffer 1 deleted
```

Buffer 1 is deleted by pressing '-'

```
Buffer_2 allocated of size 45
Address of buffer_2 = 0xFFFF
Enter a character
,the input character is 44
Enter a character
+the input character is 43
Allocating a new buffer
enter a new value between 30 and 300
Enter a value divisible btw 30 and 300
q = 0
r = 4
s = 6
new value of buffer is = 46
Memory Allocation Successful for Buffer_3
Buffer_3 allocated of size 46
Address of buffer_3 = 0xFFFF
```

Adding additional buffer of 46 bytes by pressing '+'

```

@the input character is 64
Free Buffer 0
Buffer 1 will now become empty
Freeing Buffer 1
Buffer 1 is empty
Freeing buffer_2
Buffer 2 freed
Freeing buffer_3
Buffer 3 freed

```

Freeing every buffer created

```

count = 64
Enter a character
kthe input character is 107
kbuffer is full

```

Buffer is full condition

```

File Edit Setup Control Window Help
Buffer Size of buffer_0 80
Stored characters in buffer>>>-4302
Free Spaces in buffer4382
Heap status of buffer_1
Start Address of buffer_1 x454
Ending Address of buffer_1x4A4
Buffer Size80
Storage characters in buffer>>>0
Free Spaces in buffer>>>>>>>>80
Number of storage characters = -4302
Total number of characters received = -4269
Total number of buffers that were allocated since the start of the program = -43
68

```

Details for buffer added by pressing '?'

PART-3 Supplemental + Required

1. 8051 Supplemental

Things Learnt:

Learnt to configure Programmable Counter Array in 4 modes

Learnt how to set PWM flags

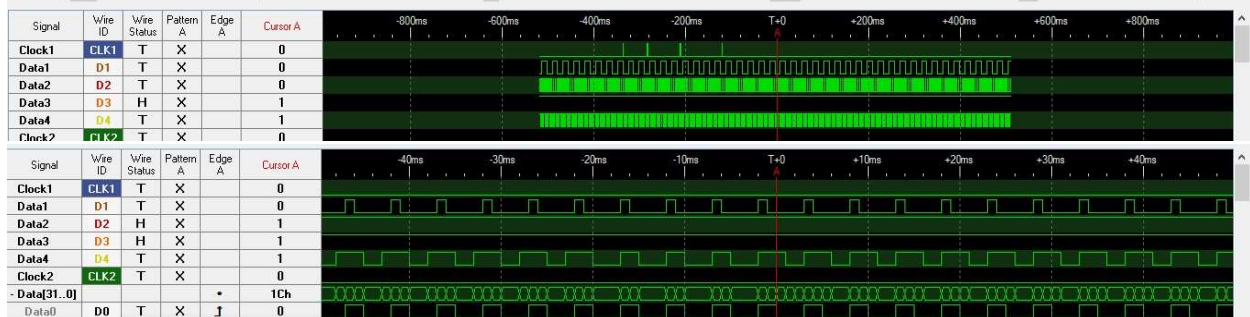
Learnt to stop PWM by setting flags

Learnt to set Maximum and minimum frequency at ALE, PSEN.

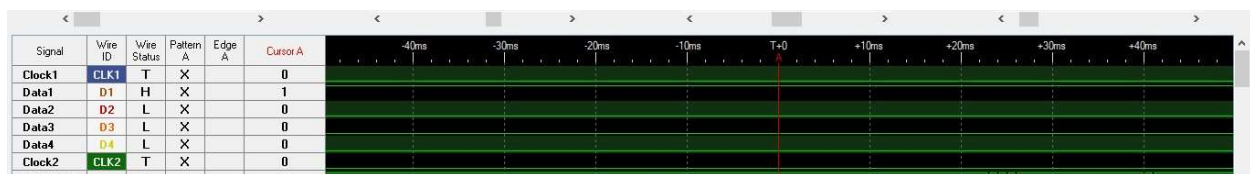
Learnt configure Power Down mode, Idle Mode by setting PCON flags

```
COM17 - Tera Term VT
File Edit Setup Control Window Help
*****USER MENU for LAB 3 Part 3*****
1 for run PWM mode
2 for stop PWM mode
3 forSet FCLK PERIPH at the minimum frequency supported by the CKRL register
4 for Set FCLK PERIPH at the maximum frequency supported by the CKRL register
5 for Enter Idle mode
6 for Enter Power Down mode
7 for Enter for High Speed Toggle
8 for Enter for watchdog timer
```

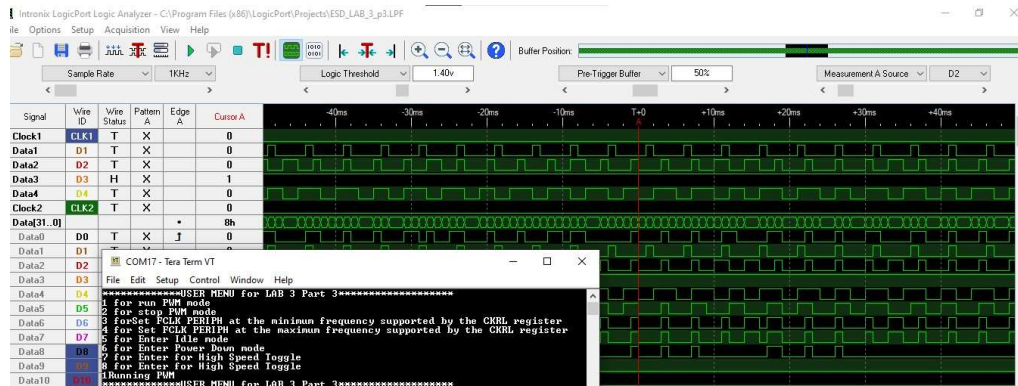
User menu for Modules in PCA modes



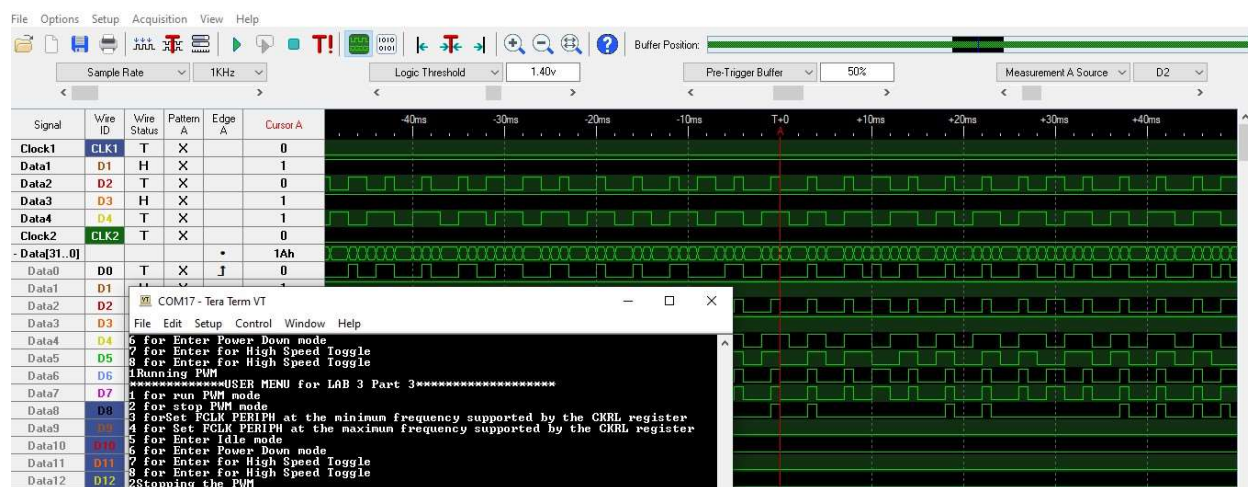
High Speed Toggle



Entering in Power Down Mode



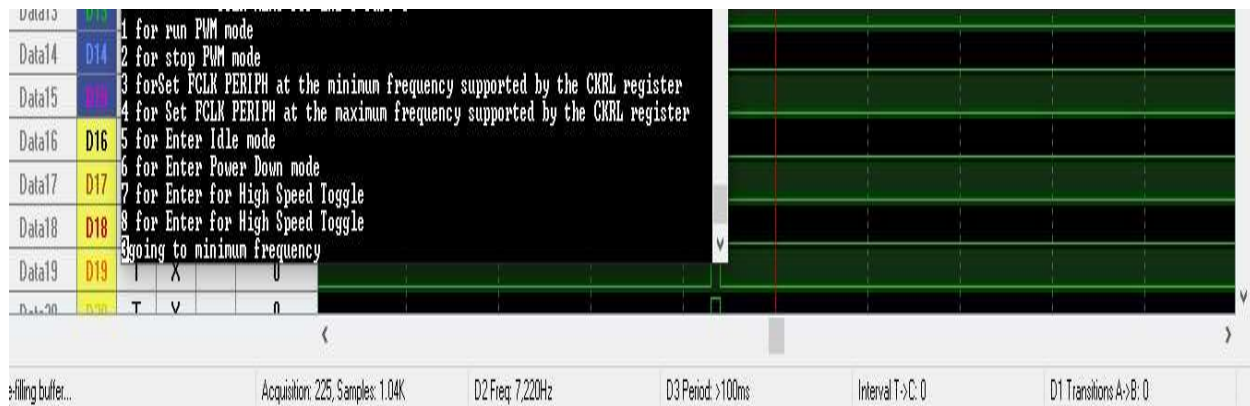
Duty Cycle 25% at port D2



Stopping PWM



Entering Maximum Frequency



Entering Minimum Frequency

2. MSP432 Required + Supplemental

Learnt how to use ADC in temperature Flags

Learnt how to set PWM from examples.

Learnt how to Port mapping

Duty Cycle:

```
COM14 - Tera Term VT
File Edit Setup Control Window Help
Duty cycle at Pin 7.7=500
Duty cycle at Pin 7.7=600
Duty cycle at Pin 7.7=600
Duty cycle at Pin 7.7=700
```

Temperature Values:

```
Temperature in degree (C)=28.419117
Temperature in degree(F)=83.022057
Temperature in degree (C)=28.235294
Temperature in degree(F)=83.286766
Temperature in degree (C)=28.750000
Temperature in degree(F)=32.000000
```


Period changing:

```
Period is =500
```

```
Period is =1000
```

```
Period is =500
```

Spectrum:

