

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: Tharuni Gelli

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: Tharuni

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: 57000
- ☒ Knows how to use SDCC [IDE or make optional]

Reshpaude 10/20/23.
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

Reshpaude 10/20/23.
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

Reshpaude 10/27/23
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 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 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

Part 1 and 2

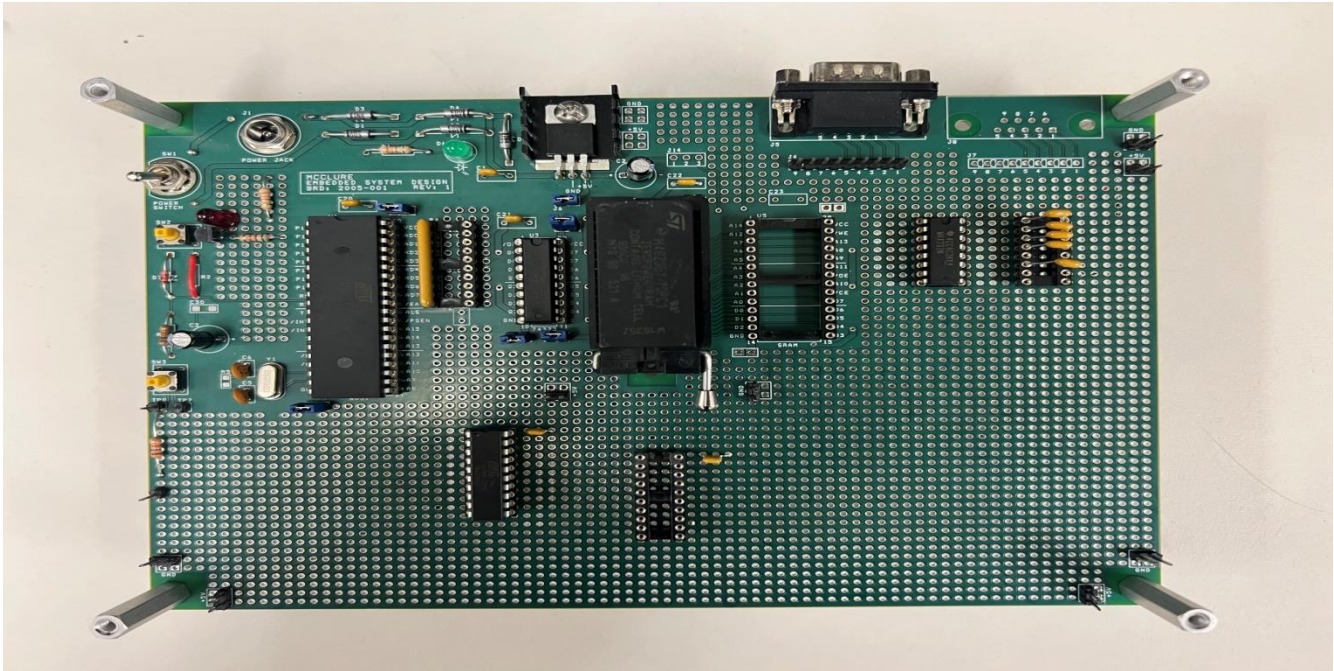
- (+) Good schematics.
- ~~(*) All commands working but are missing/failing~~
- (-) (+), (?), (=) functional correctly.
 - (-) even though buffer is ~~freed~~ free empty still shows buffer size in (?) comment.
- (+) Boundary conditions handled well.
- (-) Can improve UI.

Part 3

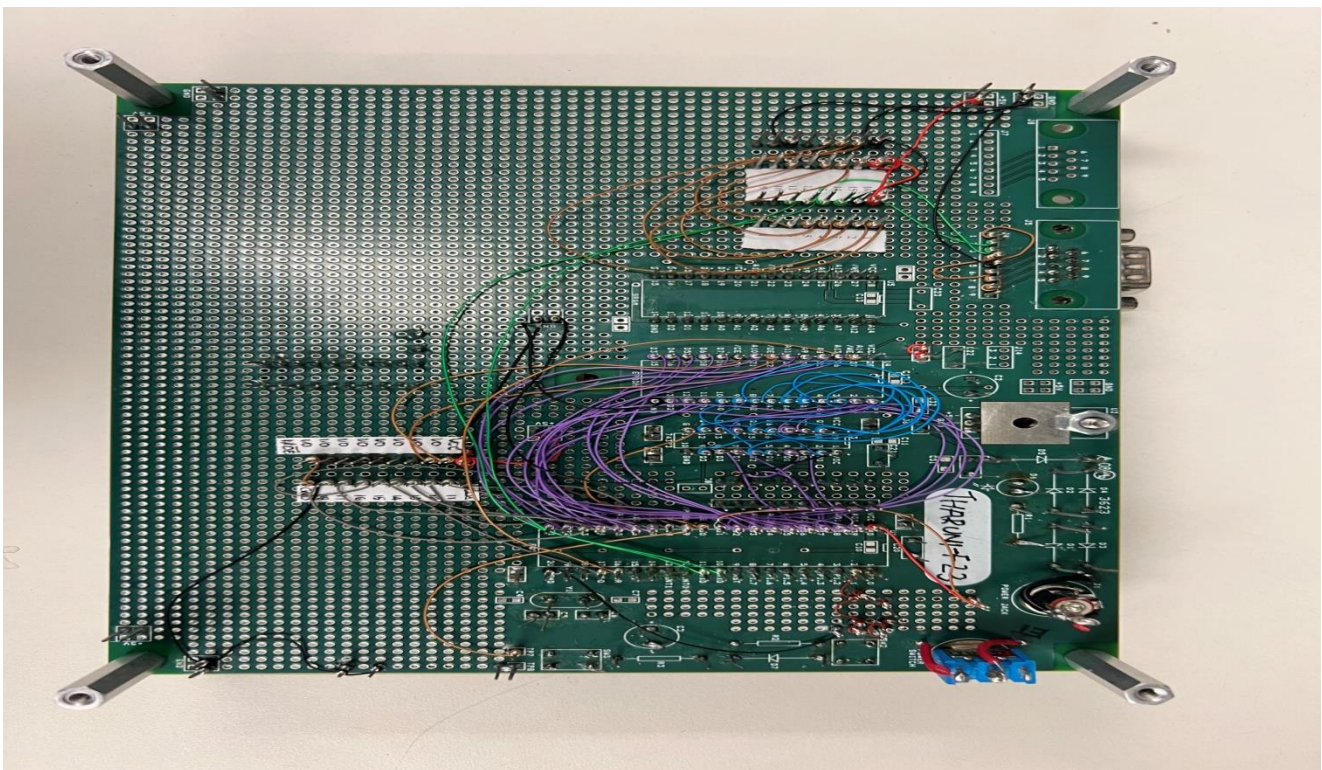
- (+) ARM code functional with PB & UART & PWM mapped to PD12.
- (+) PWM, WDT, High-Speed o/p implemented.
- (+) Idle & Power Down functional.
- (+) Good understanding of PCA model.
- (+) Good understanding of memory management.

ESD LAB-3 WRITE UP – Tharuni Gelli

- 8051 Development board front side



- 8051 Development board back side



PAULMON COMMANDS:

- E Command to see code memory

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
CODE      8051 External Memory Editor, Paul Stoffregen, 1996
ADDR: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII EQUIVILANT
2000: 32 20 06 02 22 B1 75 81 26 12 20 62 E5 82 60 03  "iu & be `
2010: 02 20 03 79 00 E9 44 00 60 1B 7A 00 90 49 2D 78   y iD ` z l-x
2020: 0E 75 A0 17 E4 93 F2 A3 08 B8 00 02 05 A0 D9 F4   u d r# 8 Yt
2030: DA F2 75 A0 FF E4 78 FF F6 D8 FD 78 00 E8 44 00  Zru dx vX>x hD
2040: 60 0A 79 01 75 A0 00 E4 F3 09 D8 FC 78 0D E8 44   `y u ds Xi x hD
2050: 17 60 0C 79 18 90 00 01 E4 F0 A3 D8 FC D9 FA 02   `y C dp#X!Yz
2060: 20 03 75 8F 00 43 8E 0C 90 00 00 22 E5 82 90 15   u C "e
2070: E1 F0 E0 FF 7E 00 90 70 05 EF F0 EE A3 F0 22 AF   ap ` ~ p opn#p"/
2080: 03 E5 82 90 15 E2 F0 EF A3 F0 30 99 FD 90 15 E2   `e bpo#p0 > ` b
2090: E0 FE A3 E0 FF 8E 99 C2 99 8E 82 8F 83 22 10 98   `~#` B " " dd
20A0: 02 00 FB A0 E9 7F 00 8E 82 8F 83 22 90 15 E4 E4   < " " dd
20B0: F0 A3 F0 90 15 E6 F0 90 15 E6 E0 F7 24 E7 FD E4   p#p f p f ` $g>d
20C0: 34 15 FE C0 07 C0 06 C0 05 12 20 9E AB 82 AC 83   4 ~e e e ` +
20D0: D0 05 D0 06 D0 07 8D 82 8E 83 EB F0 EB 70 03 02   P P P ` kpkp
20E0: 21 59 8D 82 8E 83 E0 FC BC 0D 02 80 6C 8D 82 8E   !V `>= `!< l
20F0: 83 E0 FD BD 08 1C 90 00 08 C0 07 12 20 7F 90 00   !V `>= `!< l
^E-Edit ^G-Goto ^C=Code ^D=Data ^L=Redraw ^Q=Quit
```

- Ctrl+D command to see data memory- From 0000 to 7FFF with 55

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
DATA      8051 External Memory Editor, Paul Stoffregen, 1996
ADDR: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII EQUIVILANT
7F00: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F10: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F20: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F30: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F40: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F50: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F60: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F70: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F80: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7F90: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7FA0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7FB0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7FC0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7FD0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7FE0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
7FF0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 0000000000000000
^A=ASCII ^X=Hex ^F=Fill ^G=Goto ^C=Code ^D=Data ^L=Redraw ^Q=Quit
```

- D command to download HEX file directly from terminal emulator, which overwrites the data memory

```

COM5 - Tera Term VT
File Edit Setup Control Window Help
CODE      8051 External Memory Editor, Paul Stoffregen, 1996
ADDR: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII EQUIVILANT

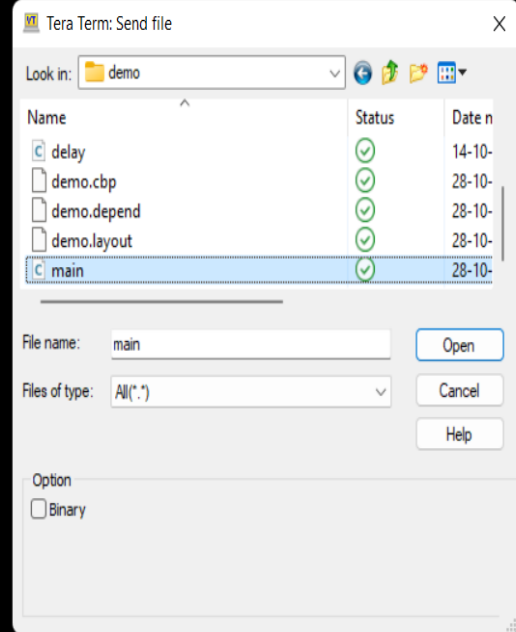
2000: 32 20 06 02 22 B1 75 81 26 12 20 62 E5 82 60 03  "iu & be `
2010: 02 20 03 79 00 E9 44 00 60 1B 7A 00 90 49 2D 78  y iD ` z I-x
2020: 0E 75 A0 17 E4 93 F2 A3 08 B8 00 02 05 A0 D9 F4  u d r# 8 Yt
2030: D0 F2 75 A0 FF E4 78 FF F6 D8 FD 78 00 E8 44 00  Zru dx vX>x hD
2040: 60 0A 79 01 75 A0 00 E4 F3 09 D8 FC 78 0D E8 44  'y u ds Xix hD
2050: 17 60 0C 79 18 90 00 01 E4 F0 A3 D8 FC D9 FA 02  'y dp#X!Yz
2060: 20 03 75 8F 00 43 8E 0C 90 00 00 22 E5 82 90 15  u C "e
2070: E1 F0 E0 FF 7E 00 90 70 05 EF F0 EE A3 F0 22 AF  ap`~ p opn#p"/
2080: 83 E5 82 90 15 E2 F0 EF A3 F0 30 99 FD 90 15 E2  e bpo#p0 } h
2090: E0 FE A3 E0 FF 8E 99 C2 99 8E 82 8F 83 22 10 98  `# B "
20A0: 02 80 FB AE 99 7F 00 8E 82 8F 83 22 90 15 E4 E4  < " dd
20B0: F0 A3 F0 90 15 E6 F0 90 15 E6 E0 FF 24 E7 FD E4  p#p fp f` $g>d
20C0: 34 15 FE C0 07 C0 06 C0 05 12 20 9E AB 82 AC 83  4 ~e e +
20D0: D0 05 D0 06 D0 07 8D 82 8E 83 EB F0 EB 70 03 02  P P P kpkp
20E0: 21 59 8D 82 8E 83 E0 FC BC 0D 02 80 6C 8D 82 8E  tY `i< l
20F0: 83 E0 FD BD 08 1C 90 00 08 C0 07 12 20 7F 90 00  `)= e

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

PAULMON2 Loc:2000 > Download

Begin ascii transfer of Intel hex file, or ESC to abort

```



```

COM5 - Tera Term VT
File Edit Setup Control Window Help
CODE      8051 External Memory Editor, Paul Stoffregen, 1996
ADDR: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII EQUIVILANT

2000: 32 20 06 02 22 B1 75 81 26 12 20 62 E5 82 60 03  "iu & be `
2010: 02 20 03 79 00 E9 44 00 60 1B 7A 00 90 49 2D 78  y iD ` z I-x
2020: 0E 75 A0 17 E4 93 F2 A3 08 B8 00 02 05 A0 D9 F4  u d r# 8 Yt
2030: D0 F2 75 A0 FF E4 78 FF F6 D8 FD 78 00 E8 44 00  Zru dx vX>x hD
2040: 60 0A 79 01 75 A0 00 E4 F3 09 D8 FC 78 0D E8 44  'y u ds Xix hD
2050: 17 60 0C 79 18 90 00 01 E4 F0 A3 D8 FC D9 FA 02  'y dp#X!Yz
2060: 20 03 75 8F 00 43 8E 0C 90 00 00 22 E5 82 90 15  u C "e
2070: E1 F0 E0 FF 7E 00 90 70 05 EF F0 EE A3 F0 22 AF  ap`~ p opn#p"/
2080: 83 E5 82 90 15 E2 F0 EF A3 F0 30 99 FD 90 15 E2  e bpo#p0 } h
2090: E0 FE A3 E0 FF 8E 99 C2 99 8E 82 8F 83 22 10 98  `# B "
20A0: 02 80 FB AE 99 7F 00 8E 82 8F 83 22 90 15 E4 E4  < " dd
20B0: F0 A3 F0 90 15 E6 F0 90 15 E6 E0 FF 24 E7 FD E4  p#p fp f` $g>d
20C0: 34 15 FE C0 07 C0 06 C0 05 12 20 9E AB 82 AC 83  4 ~e e +
20D0: D0 05 D0 06 D0 07 8D 82 8E 83 EB F0 EB 70 03 02  P P P kpkp
20E0: 21 59 8D 82 8E 83 E0 FC BC 0D 02 80 6C 8D 82 8E  tY `i< l
20F0: 83 E0 FD BD 08 1C 90 00 08 C0 07 12 20 7F 90 00  `)= e

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

PAULMON2 Loc:2000 > Download

Begin ascii transfer of Intel hex file, or ESC to abort

Download aborted

Summary:
0 lines received
0 bytes received
0 bytes written
No errors detected

PAULMON2 Loc:2000 > Download

Begin ascii transfer of Intel hex file, or ESC to abort

Download completed

Summary:
142 lines received
2161 bytes received
2161 bytes written
Errors:
1 unexpected begin of line
262 unexpected hex digits
2734 unexpected non hex digits

PAULMON2 Loc:2000 >

```


PART1-CHALLENGE – PAULMON as an on-chip debugger –

- For the code I submitted in debugger.asm and .hex file

```
VT COM10 - Tera Term VT
File Edit Setup Control Window Help

PAULMON2 Loc:2000 > Single-Step

Jump to memory location (2000), or (ESC) to exit: 2000
Now running in single step mode: (RET)= step, ?= Help

ACC B C DPTR R0 R1 R2 R3 R4 R5 R6 R7 SP Addr Instruction
00 00 0 2000 00:00:00:00:00:00:00:00 0A 14DD: JMP EA+DPTR
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2000: LJMP 2016
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2016: CPL P1.1
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2018: SJMP 2016
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2016: CPL P1.1
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2018: SJMP 2016
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2016: CPL P1.1
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2018: SJMP 2016
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2016: CPL P1.1
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2018: SJMP 2016
00 00 0 2000 00:00:00:00:00:00:00:00 0A 2016: CPL P1.1
```

PART2: Buffers allocation using heap of 5000 bytes

- Menu display with buffer 0 and buffer 1 creation

```
COM5 - Tera Term V1
File Edit Setup Control Window Help

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 (VT100) 1800        External command

PAULMON2 Loc:2000 > Jump to memory location
Jump to memory location (2000), or ESC to quit: 2000
running program:

***** HELLO :) *****

Enter a number between 32 & 4800 for buffer size
1600
The input number is:1600
Start Address of buffer0 = 0x3
Start Address of buffer1 = 0x645
Buffer_0 and Buffer_1 successfully created

OPTIONS TO CHOOSE
You can enter characters to be stored in buffer 0
Enter + to create a new buffer of size between 20 and 400 bytes
Enter - to delete a buffer.
Enter ? to generate a heap report.
Enter = to display current contents of buffer 0.
Enter @ to free all buffers and start program again.

*****THANK YOU*****
```

- '+' Character for creation of n buffers with the size specified between 20 to 400

```
COM5 - Tera Term VT
File Edit Setup Control Window Help

+
Creates a new buffer
Enter the new buffer size between 20 and 400
100
Successful allocated memory for Buffer_2
Buffer_2 has allocated a size of 100
Start Address of buffer_2 = 0x0
+
Creates a new buffer
Enter the new buffer size between 20 and 400
200
Successful allocated memory for Buffer_3
Buffer_3 has allocated a size of 200
Start Address of buffer_3 = 0x0
```

- Storing Buffer characters in buffer 0:

```
Creates a new buffer
Enter the new buffer size between 20 and 400
200
Successful allocated memory for Buffer_3
Buffer_3 has allocated a size of 200
Start Address of buffer_3 = 0x0
A
B
C
D
J
H
K
L
I
O
T
N
B
G
H
H
K
K
N
B
U
G
S
B
U
D
G
B
H
W
B
D
J
H
B
W
U
J
S
H
X
C
```

- ‘-’ Character usage for deleting buffers with specified buffer number and heap report after that ‘-’ character

```

COM5 - Tera Term VT
File Edit Setup Control Window Help
Buffer_4 has allocated a size of 150
Start Address of buffer_4 = 0x0
-
Enter a valid buffer number
2
Deleting buffer 2
Buffer 2 is Free
?
***** REPORT OF HEAP *****
Buffer 0
Its Start Address = 0x3
Its Ending Address = 0x643
Its Size = 1600
Current Storage characters in buffer 0 = 0
Current Free Spaces in buffer = 1600
-----
Buffer 1
Its Start Address = 0x645
Its Ending Address = 0xC85
Its Size = 1600
Storage characters in buffer -0, because storage characters are 0 buffer 0
Current Free Spaces in buffer = 1600
-----
Buffer 4
Start Address = 0xCED
Ending Address = 0xD83
Buffer Size = 150
Storage characters in buffer = 0
Free Spaces in buffer = 150
-----
Number of storage characters = 0
Total number of characters received = 108
Total number of buffers that were allocated since the start of the program = 5
Total storage characters stored since last '?' = 0

```

- ‘?’ Character usage for giving heap report and printing buffer 0 32 ASCII characters in line and empty buffer 0

```

COM5 - Tera Term VT
File Edit Setup Control Window Help
X
N
B
U
H
E
J
D
?
***** REPORT OF HEAP *****
Buffer 0
Its Start Address = 0x3
Its Ending Address = 0x643
Its Size = 1600
Current Storage characters in buffer 0 = 54
Current Free Spaces in buffer = 1546
-----
Buffer 1
Its Start Address = 0x645
Its Ending Address = 0xC85
Its Size = 1600
Storage characters in buffer -0, because storage characters are 0 buffer 0
Current Free Spaces in buffer = 1600
-----
Buffer 2
Start Address = 0xC87
Ending Address = 0xCEB
Buffer Size = 100
Storage characters in buffer = 0
Free Spaces in buffer = 100
-----
Number of storage characters = 54
Total number of characters received = 105
Total number of buffers that were allocated since the start of the program = 4
Total storage characters stored since last '?' = 54
-----
DXUHEWDEBCHJQYGHBDXEGHXHSBDGWYU
SJNBGYHEBDUHJXNBUEJD

```


- '@' Character for deleting all the buffers created till now!

```

-----
@
Deleting Buffer 0
Buffer 0 is free
Deleting Buffer 1
Buffer 1 is free
Deleting buffer_2
Buffer 2 is free
***** HELLO :) *****
Enter a number between 32 & 4800 for buffer size

```

- '=' Character for hex dump with 16bytes in each line with address of each storage character and its hex ASCII value.

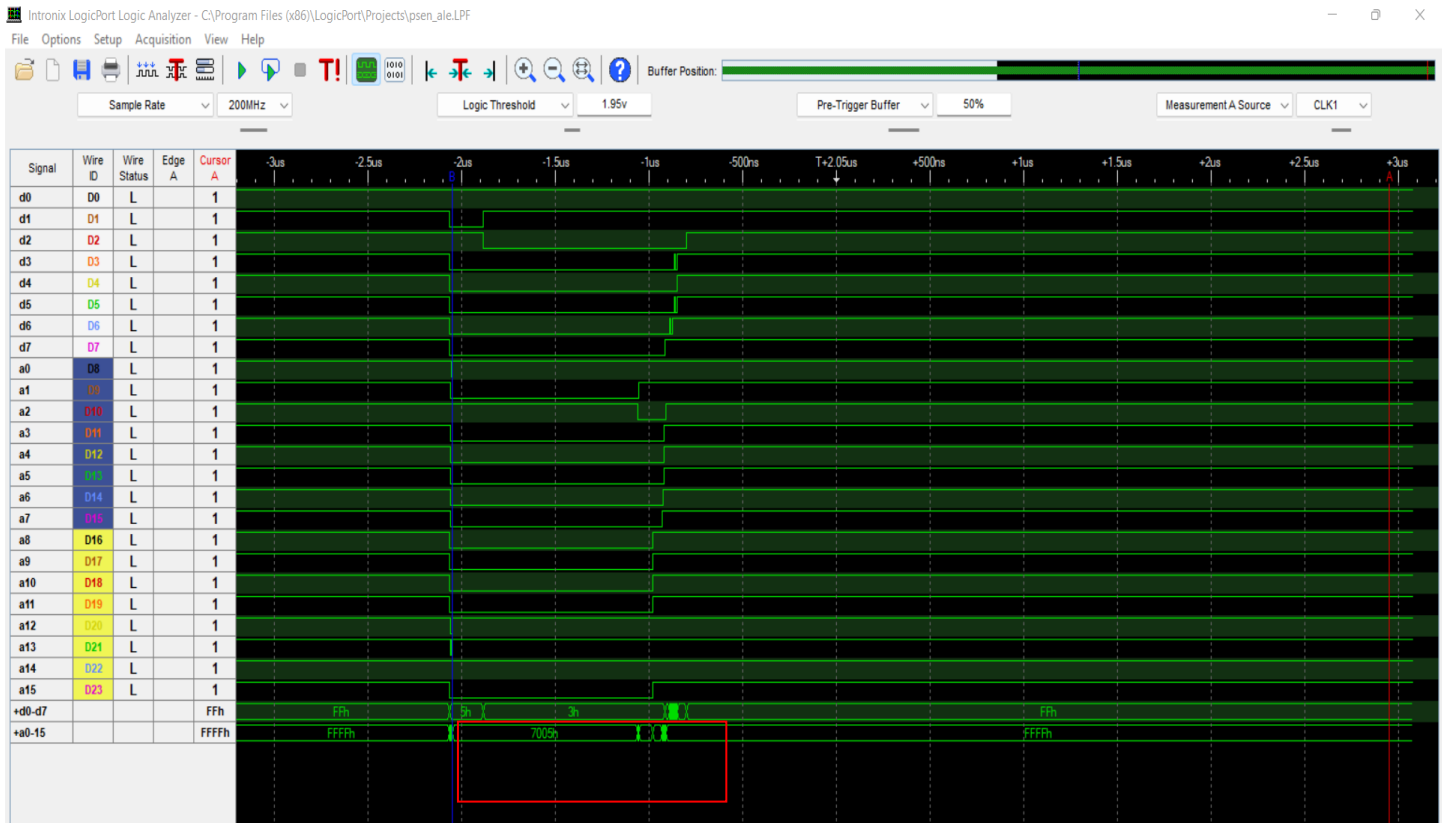
```

COM5 - Tera Term VT
File Edit Setup Control Window Help
+
Creates a new buffer
Enter the new buffer size between 20 and 400
200
Successful allocated memory for Buffer_3
Buffer_3 has allocated a size of 200
Start Address of buffer_3 = 0x0
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
[
\
]
^
_
`
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
{
|
}
~
=
The Stored Characters in Buffer 0 are
0x3 >> 41 47 48 4E 4D 44 55 43 48 55 45 48 49 52 43 4E
0x13 >> 45 55 52 42 56 4E 45 52 55 44 40 49 40 45 57 53
0x23 >> 55 40 4E 45 43 44 40 44 45 57 49 53 40 45 44

```

Virtual Debug Port:

- The address location 7005h is used for passing unique characters onto external data memory for each special characters. I am passing values 1,2,3,4 & 5 integers for each special characters in buffer. I am checking for '?' which passes a 3 integer value in 7005h.



PART3-STM32F411E:

- 'P' character prints the present duty cycle, which is 60%.

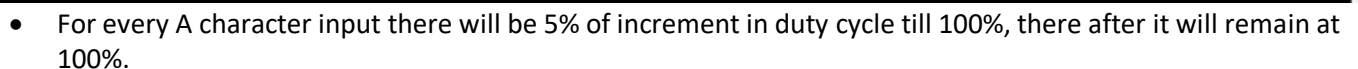
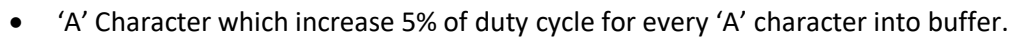
A screenshot of a Tera Term VT window titled "COM10 - Tera Term VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The main area is black with white text. It shows a repeating pattern of a red prompt character "P" followed by the text "Current Duty Cycle: 60%". This sequence appears six times, with a small white cursor visible at the end of the last line.

COM10 - Tera Term VT

File Edit Setup Control Window Help

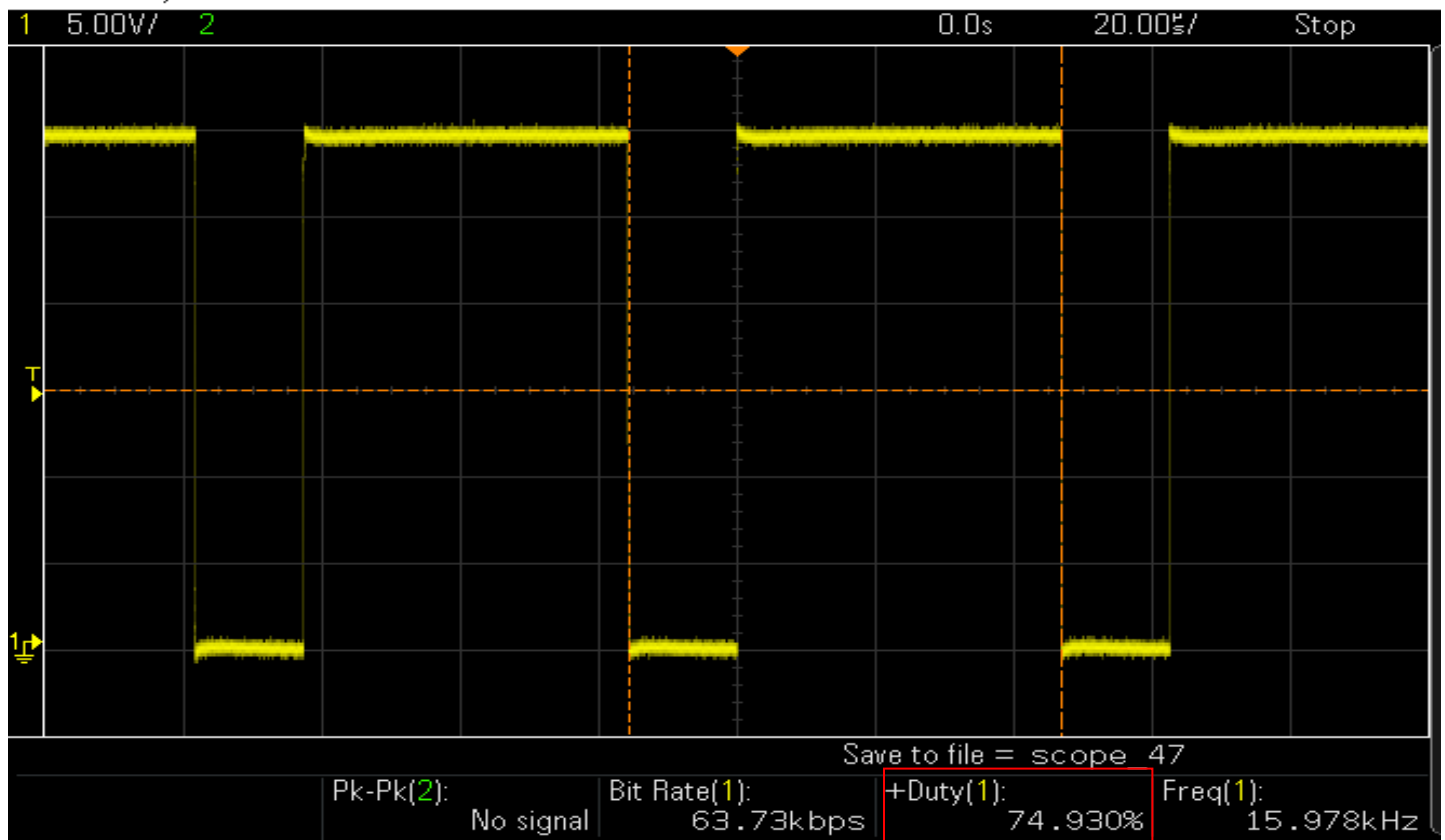
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%

- DSO-X 1102G, CN57266514: Fri Oct 27 08:05:51 2023



- Oscilloscope output at 75% duty cycle.

DSO-X 1102G, CN57246526: Mon Oct 30 06:44:17 2023

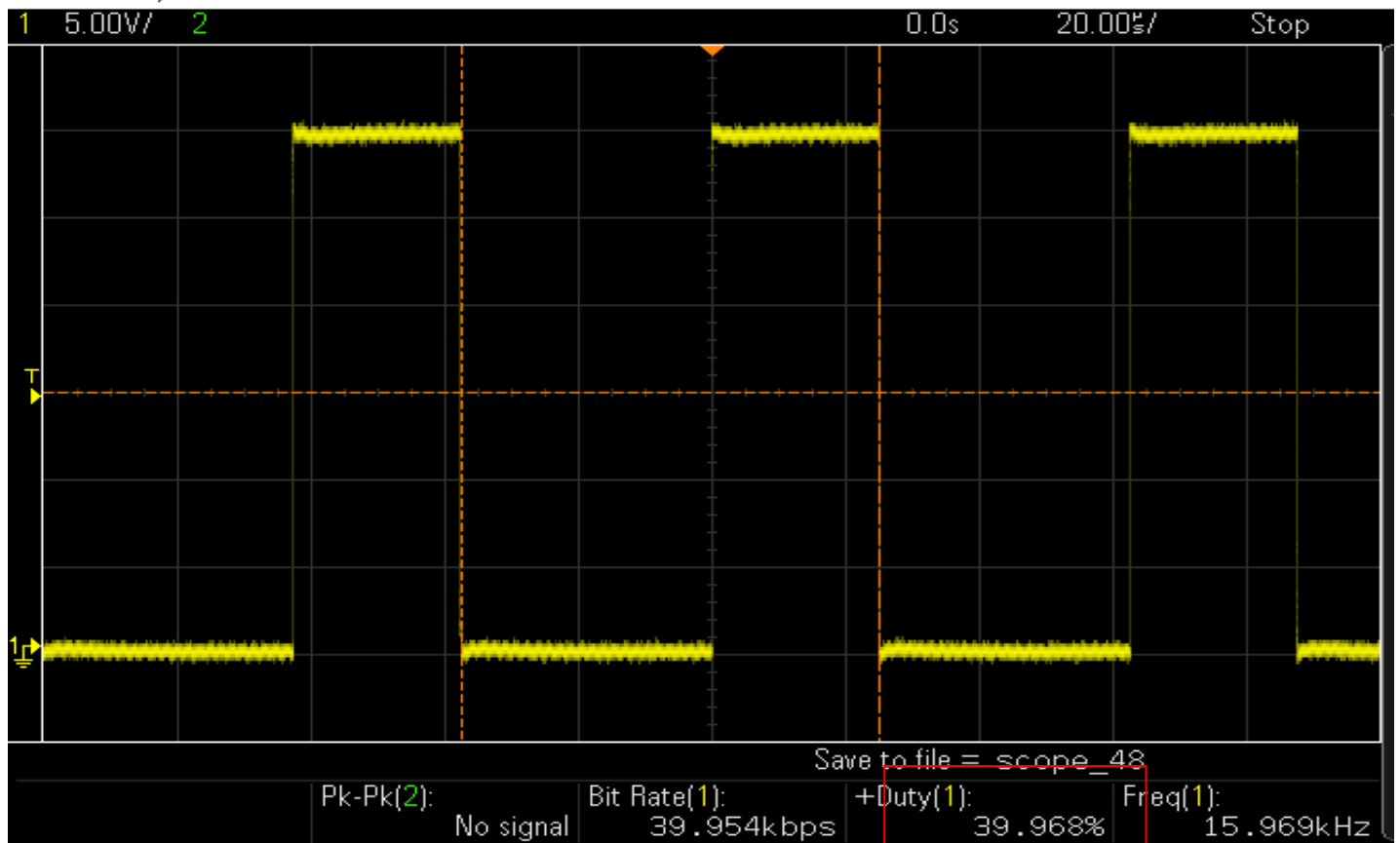


- 'B' Character which decrease 5% of duty cycle for every 'B' character entry

```
COM10 - Tera Term VT
File Edit Setup Control Window Help
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
P
Current Duty Cycle: 60%
AAAP
Current Duty Cycle: 75%
AAAAAP
Current Duty Cycle: 100%
AAAAAP
Current Duty Cycle: 100%
AAAAAP
Current Duty Cycle: 100%
BBBBP
Current Duty Cycle: 85%
BBBBBP
Current Duty Cycle: 60%
BBBBBP
Current Duty Cycle: 40%
```

- Oscilloscope output @ 40% duty cycle for 'B' character entries

DSO-X 1102G, CN57246526: Mon Oct 30 06:45:01 2023



- For every 'B' character there will be a 5% of decrement in duty cycle till 0% and it stays there only for further 'B' characters as well.

```
COM10 - Tera Term VT
File Edit Setup Control Window Help
Current Duty Cycle: 60%
AAAA
Current Duty Cycle: 75%
AAAAA
Current Duty Cycle: 100%
AAAAA
Current Duty Cycle: 100%
AAAAA
Current Duty Cycle: 100%
BBBB
Current Duty Cycle: 85%
BBBBBB
Current Duty Cycle: 60%
BBBBB
Current Duty Cycle: 40%
BBBBB
Current Duty Cycle: 15%
BBBBBB
Current Duty Cycle: 0%
BBBB
Current Duty Cycle: 0%
BBBB
Current Duty Cycle: 0%
```

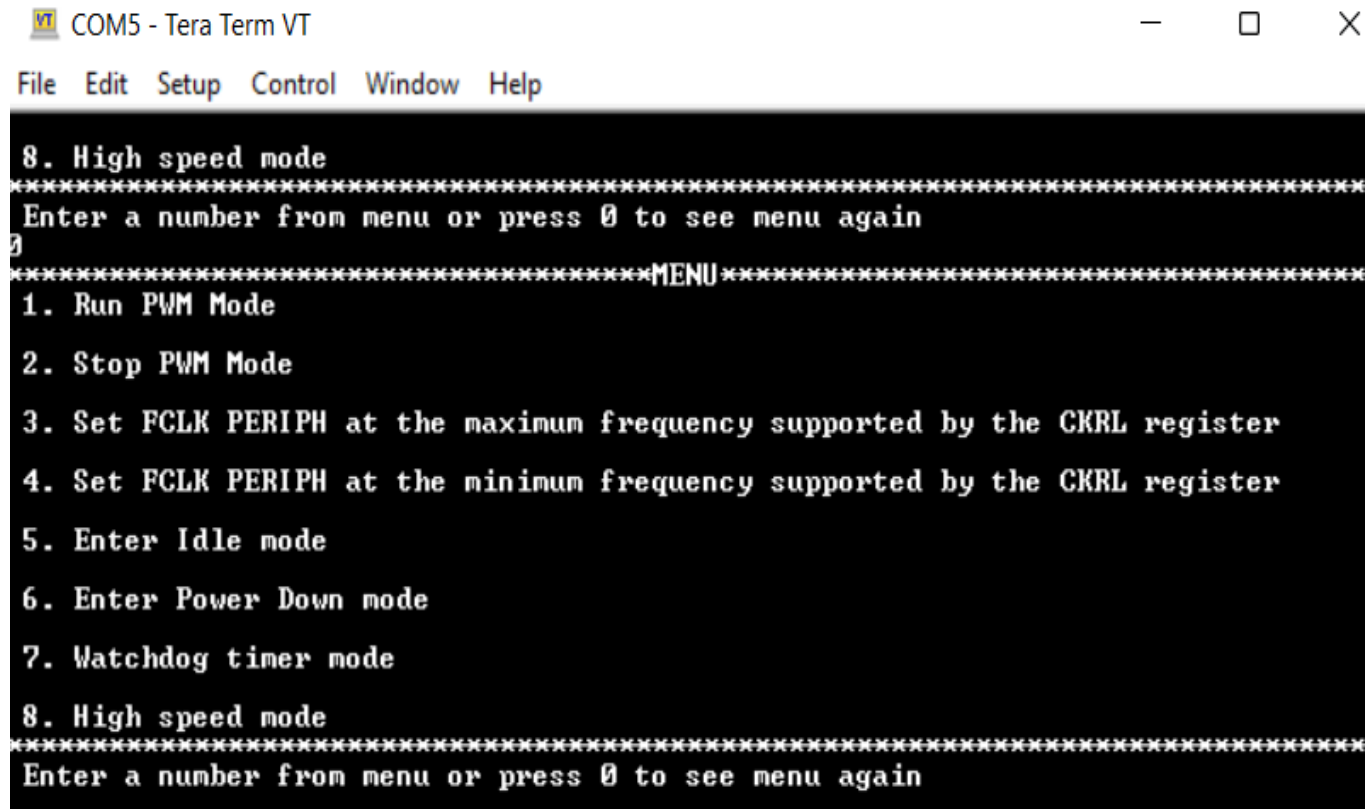
- For user button press there will be a 10% of increment in duty cycle till 100% and then decrement for further press in steps of 10% till 0% and it stays there.

```
VT COM13 - Tera Term VT
File Edit Setup Control Window Help
duty cycle is 85
B
duty cycle is 80
Button debounced and processed.
Button debounced and processed.
Button debounced and processed.
P
duty cycle is 90
Button debounced and processed.
Button debounced and processed.
P
duty cycle is 70
Button debounced and processed.
P
duty cycle is 60
Button debounced and processed.
Button debounced and processed.
Button debounced and processed.
Button debounced and processed.
Button debounced and processed.
Button debounced and processed.
P
duty cycle is 0
B
duty cycle is 0
B
duty cycle is 0
Button debounced and processed.
P
duty cycle is 10
```

- Here the user button debouncing is avoided by using a timer interrupt which creates a delay of 300ms where it stays in interrupt only without considering further more presses and does not modify the duty cycle.

Supplemental Part 3: PCA MODES OF 8051

- Menu for choosing PCA modes where it displays the number of available modes that you have implemented for the user to choose. And by giving 0 character as input, It will display the menu again. Here the attached output of terminal emulator.

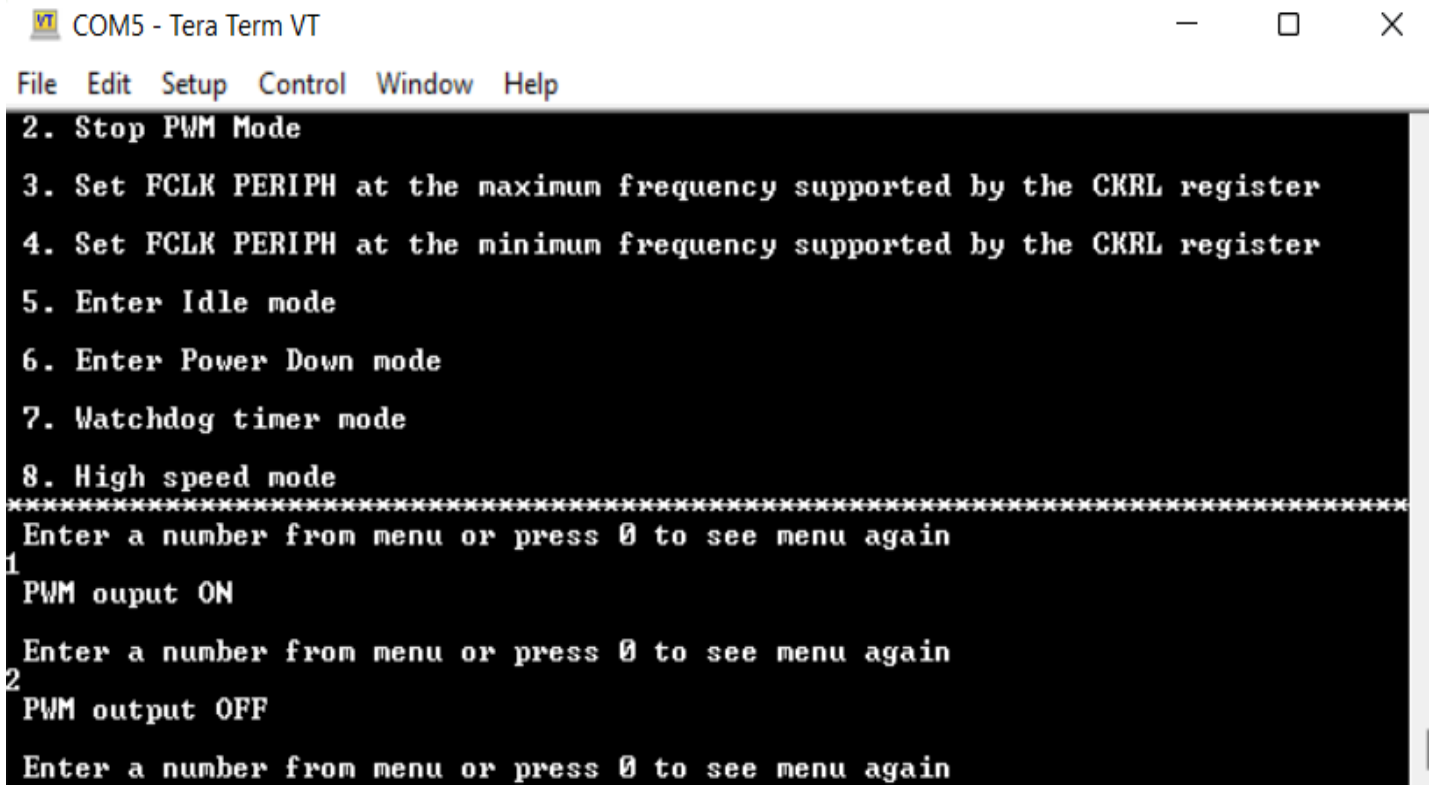


VT COM5 - Tera Term VT

File Edit Setup Control Window Help

```
8. High speed mode
*****
Enter a number from menu or press 0 to see menu again
0
*****MENU*****
1. Run PWM Mode
2. Stop PWM Mode
3. Set FCLK PERIPH at the maximum frequency supported by the CKRL register
4. Set FCLK PERIPH at the minimum frequency supported by the CKRL register
5. Enter Idle mode
6. Enter Power Down mode
7. Watchdog timer mode
8. High speed mode
*****
Enter a number from menu or press 0 to see menu again
```

- 1 gives the PWM mode on, which is implemented in module 1 where we have to check for p1.4 pin of 8051 for pwm with a duty cycle of 33%. Here the attached terminal output for it.



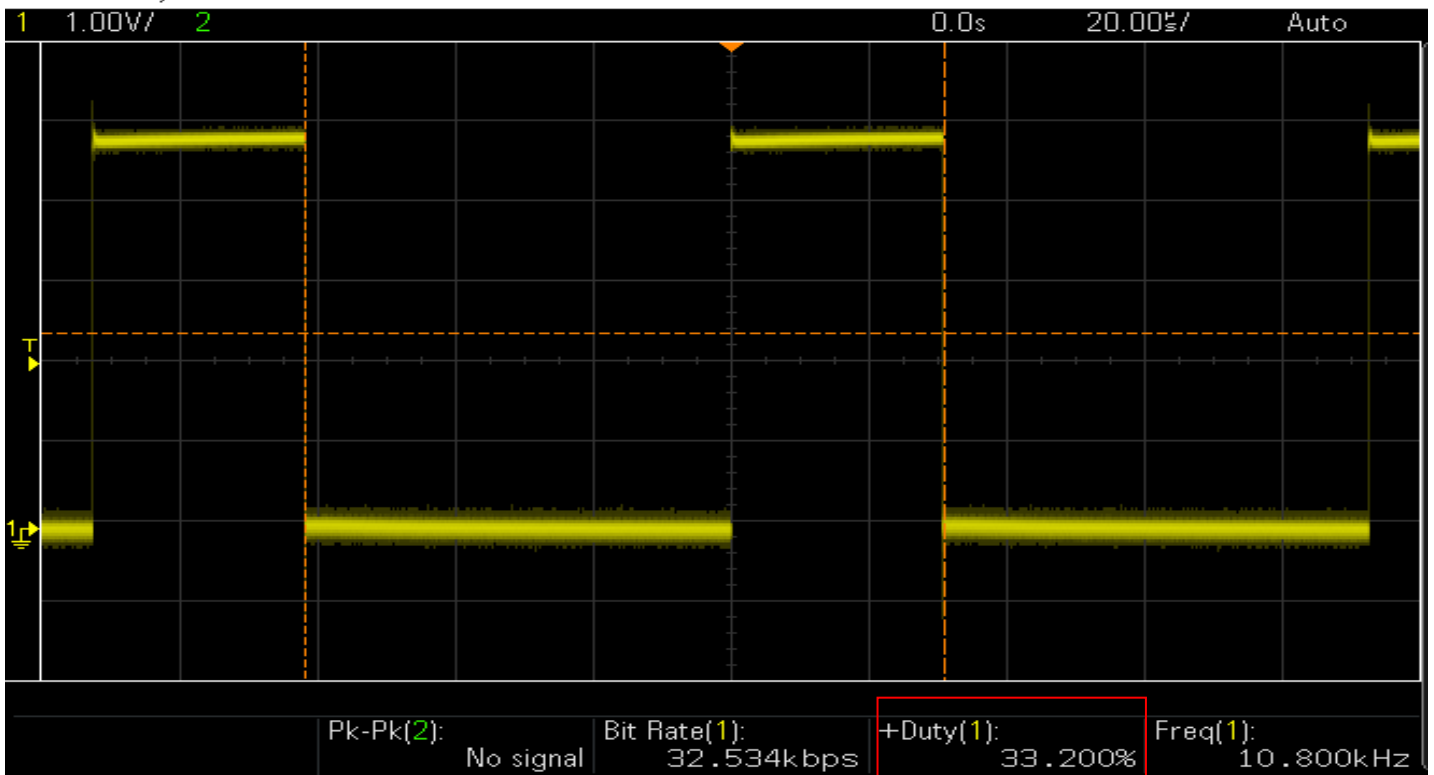
VT COM5 - Tera Term VT

File Edit Setup Control Window Help

```
2. Stop PWM Mode
3. Set FCLK PERIPH at the maximum frequency supported by the CKRL register
4. Set FCLK PERIPH at the minimum frequency supported by the CKRL register
5. Enter Idle mode
6. Enter Power Down mode
7. Watchdog timer mode
8. High speed mode
*****
Enter a number from menu or press 0 to see menu again
1
PWM ouput ON
Enter a number from menu or press 0 to see menu again
2
PWM output OFF
Enter a number from menu or press 0 to see menu again
```

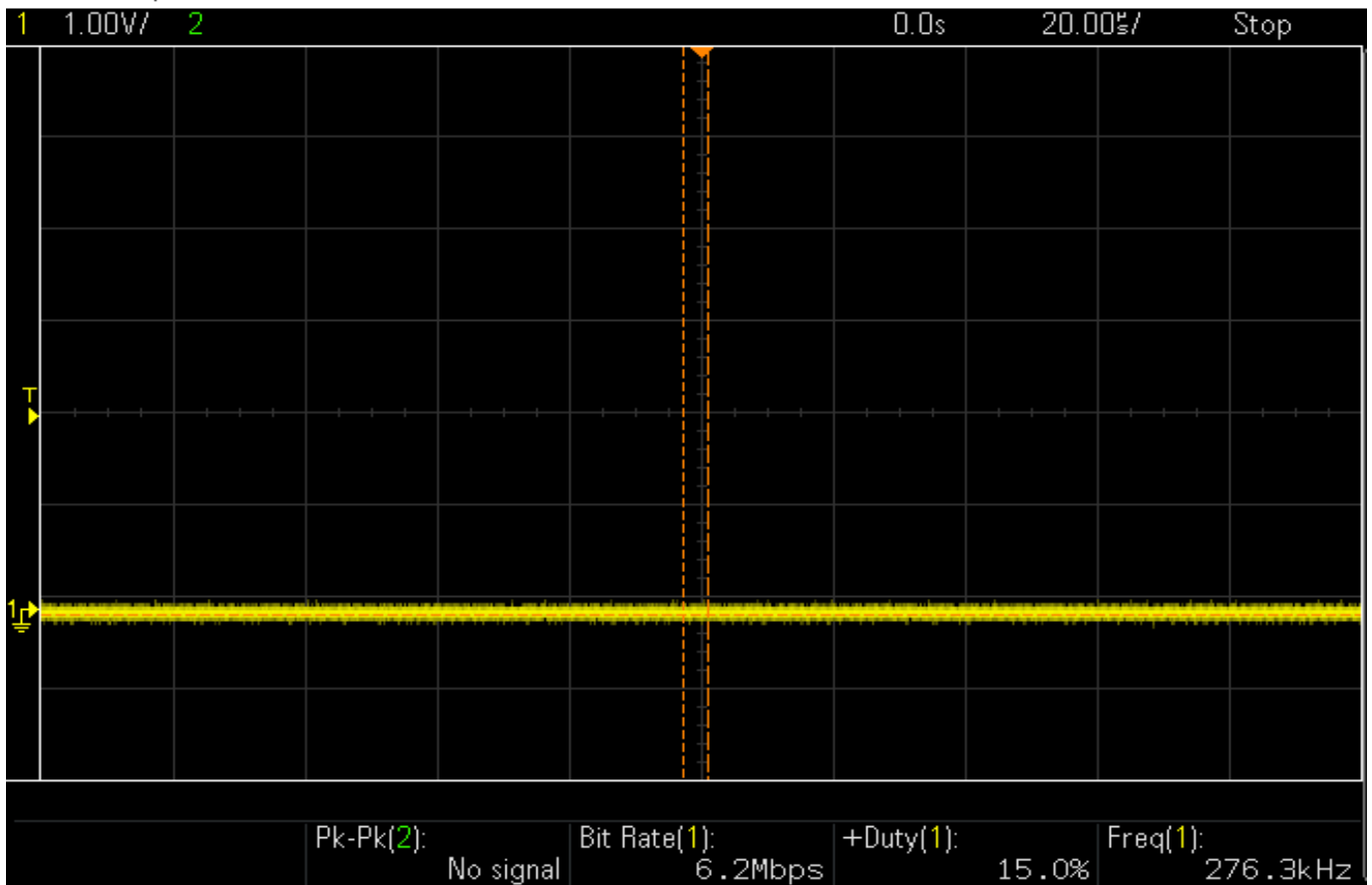
- The oscilloscope output of p1.4 pin of 8051 with a duty of 33% is attached in PWM mode On.

DSO-X 1102G, CN57246526: Sat Oct 28 03:52:34 2023



- The oscilloscope output of p1.4 pin of 8051 with a duty of 100% is attached in PWM mode OFF. This gives when you give 2 as input in terminal

DSO-X 1102G, CN57246526: Sat Oct 28 04:07:08 2023



- Input 3 and 4 to the terminal emulator gives runs the ALE at maximum and minimum frequency which is basically 3.6MHz and 7.6KHz. Here the output of terminal emulator is attached for 3 and 4 inputs.

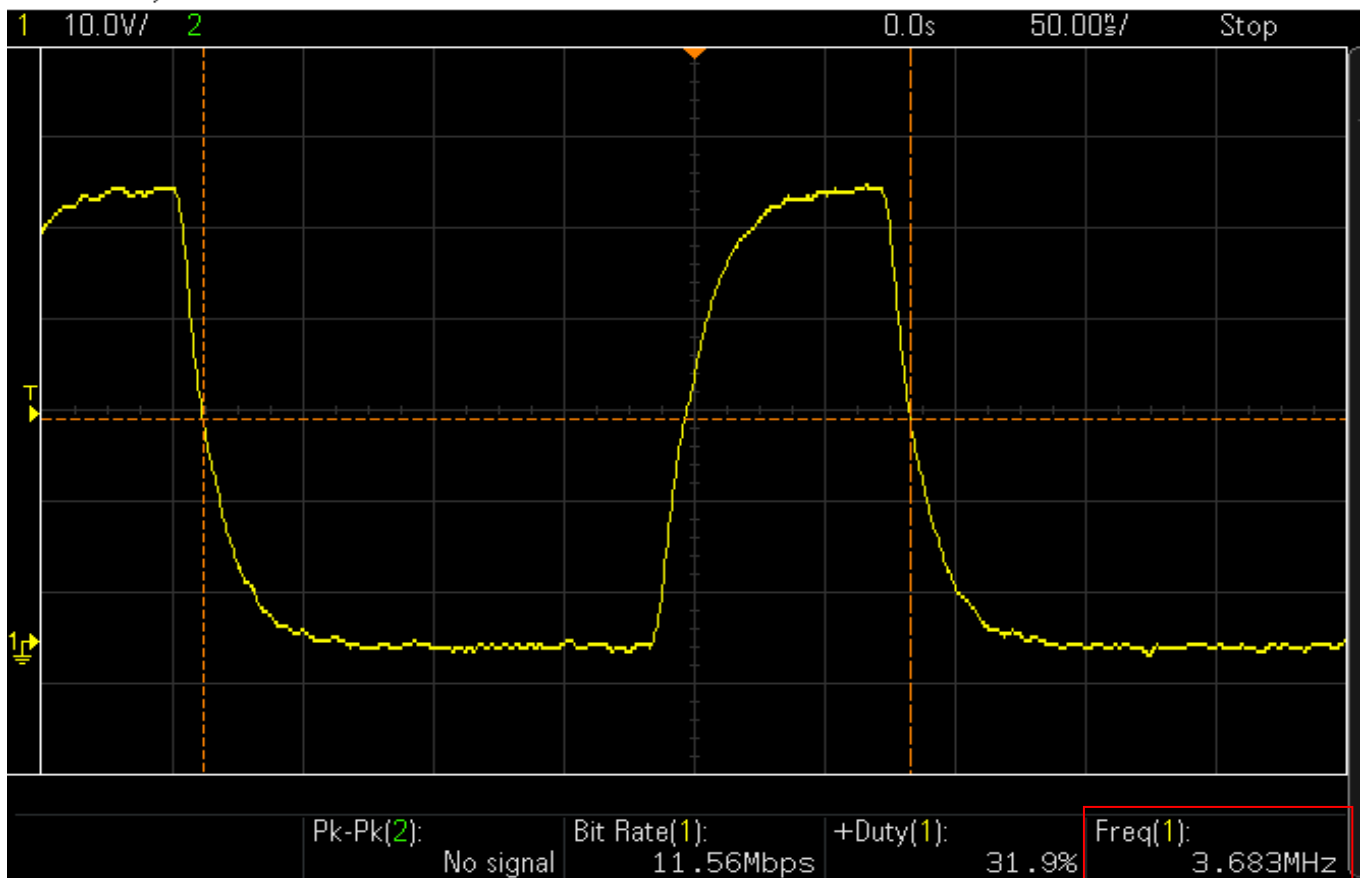
```

COM5 - Tera Term VT
File Edit Setup Control Window Help
5. Enter Idle mode
6. Enter Power Down mode
7. Watchdog timer mode
8. High speed mode
*****
Enter a number from menu or press 0 to see menu again
1
PWM output ON
Enter a number from menu or press 0 to see menu again
2
PWM output OFF
Enter a number from menu or press 0 to see menu again
4
Peripheral clock at Minimum Frequency
Enter a number from menu or press 0 to see menu again
3
Peripheral clock at Maximum Frequency

```

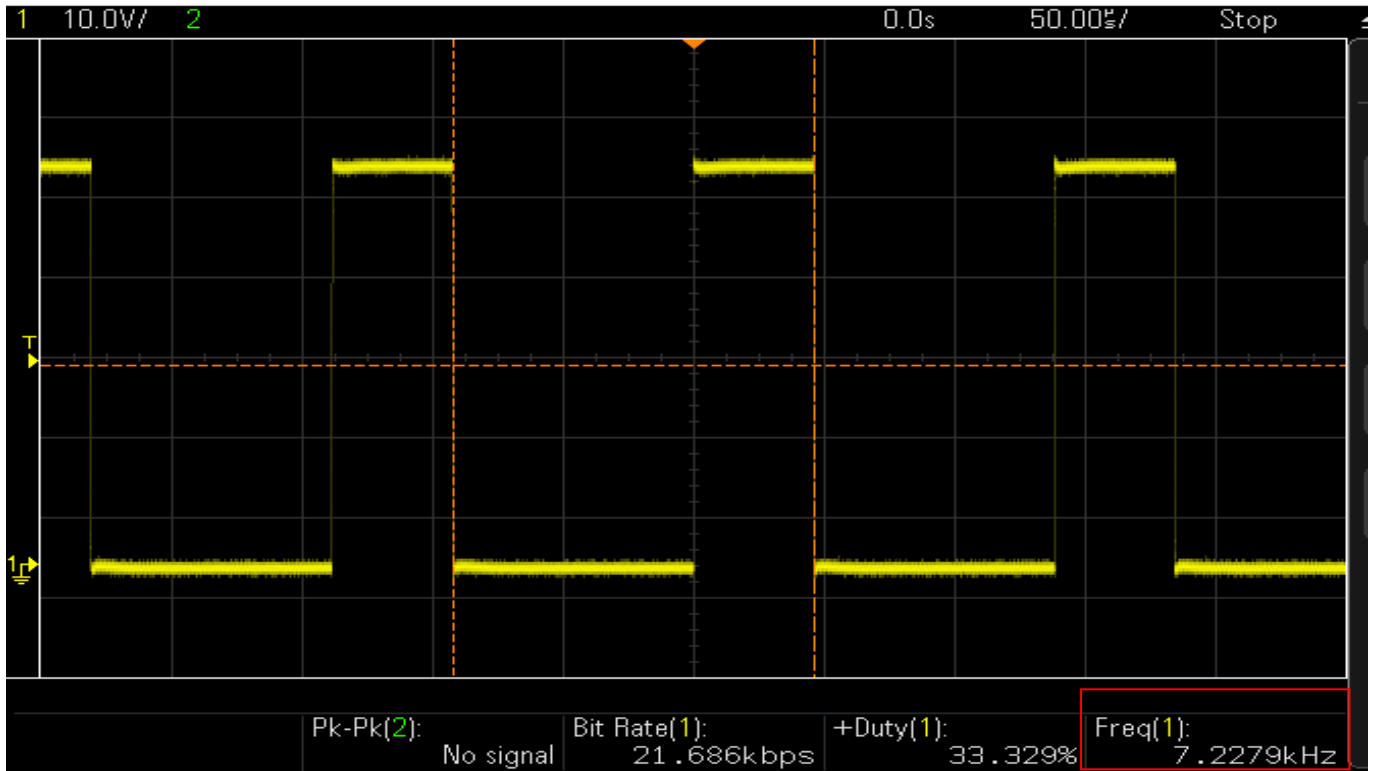
- The oscilloscope output for this 3 and 4 inputs are attached here. The maximum possible frequency seen at ALE is shown below.

DSO-X 1102G, CN57246526: Mon Oct 30 06:26:55 2023



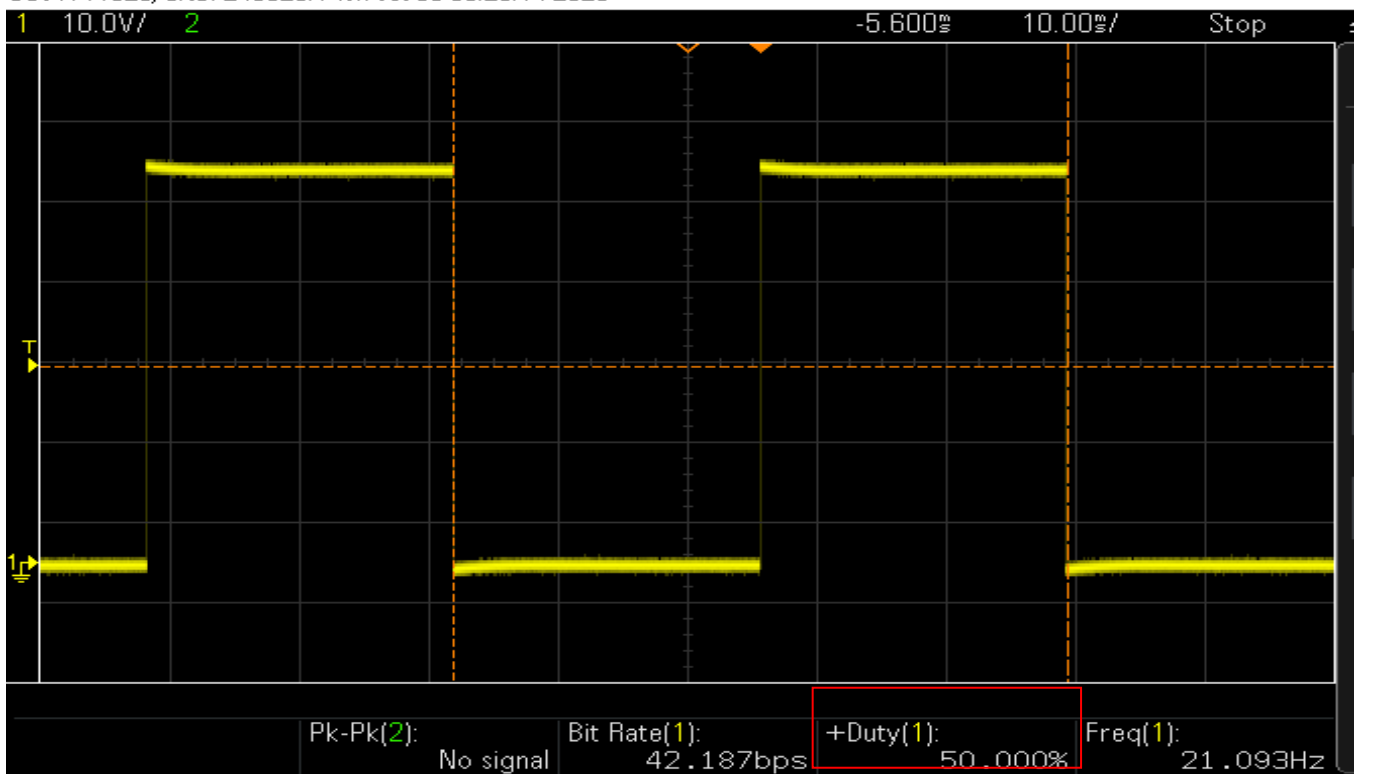
- The minimum possible frequency seen at ALE is shown below.

DSO-X 1102G, CN57246526: Mon Oct 30 06:27:23 2023



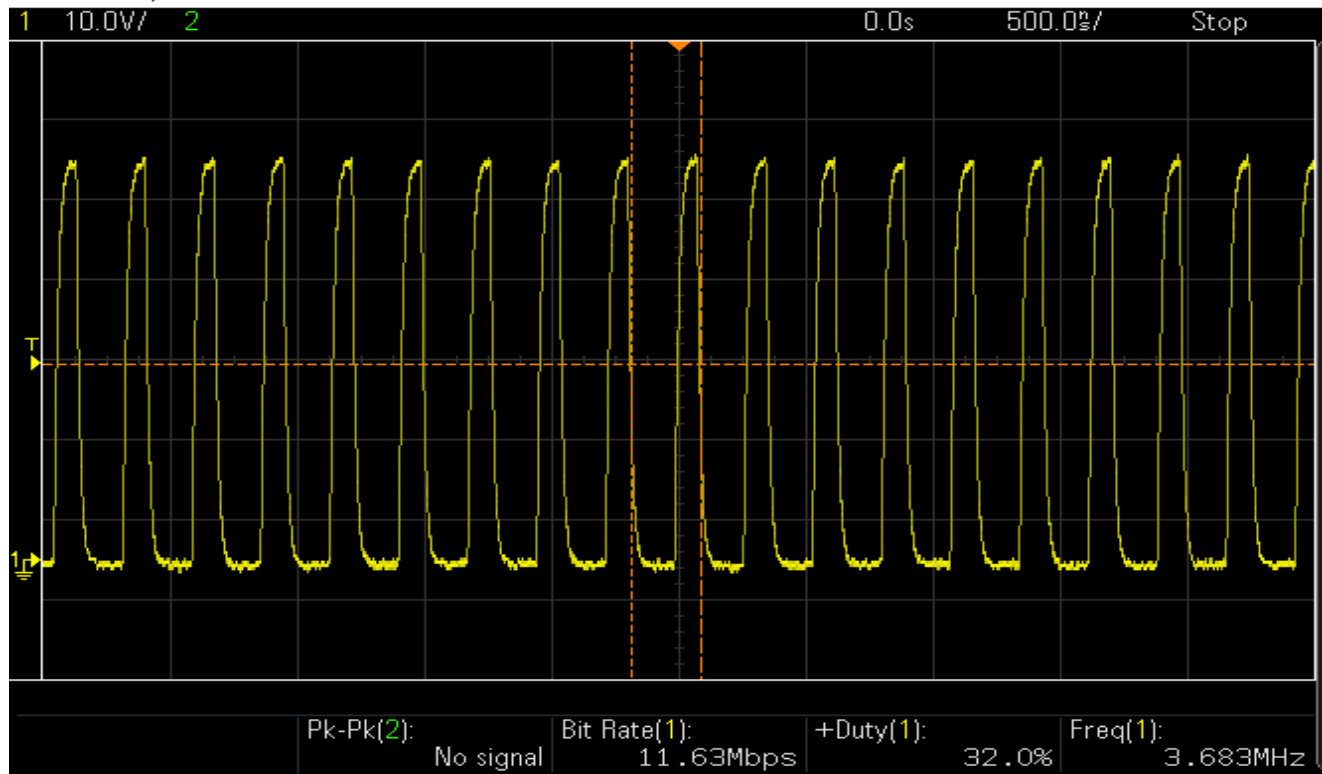
- The output of High speed is give below which is module 0 i.e p1.3 of 8051 pin with a duty cycle of 50% at that pin

DSO-X 1102G, CN57246526: Mon Oct 30 06:29:14 2023



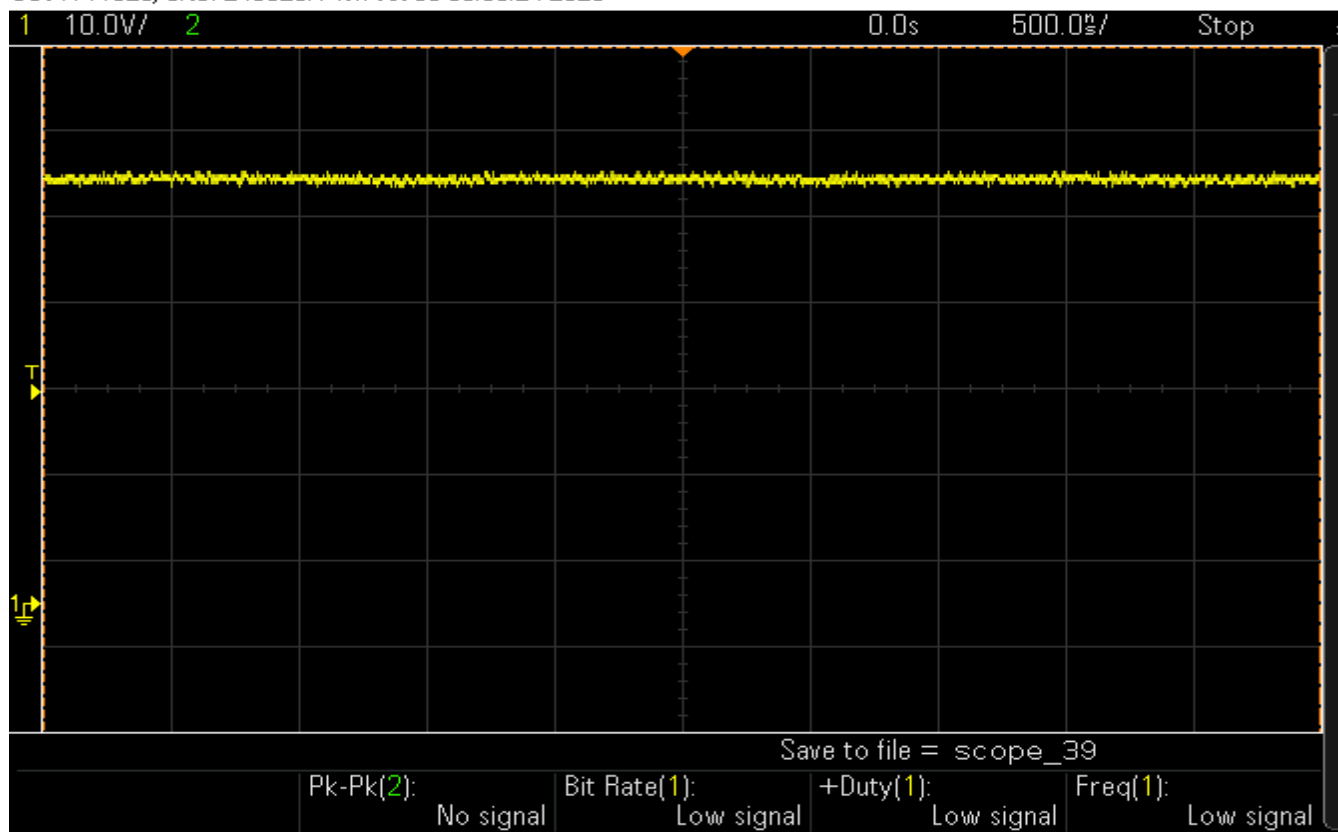
- IDLE mode output is given with oscilloscope which stops the toggling of ALE , to come out of IDLE mode we should raise INTO which is configured in code. Here the below is before entering IDLE mode.

DSO-X 11026, CN57246526: Mon Oct 30 06:29:53 2023

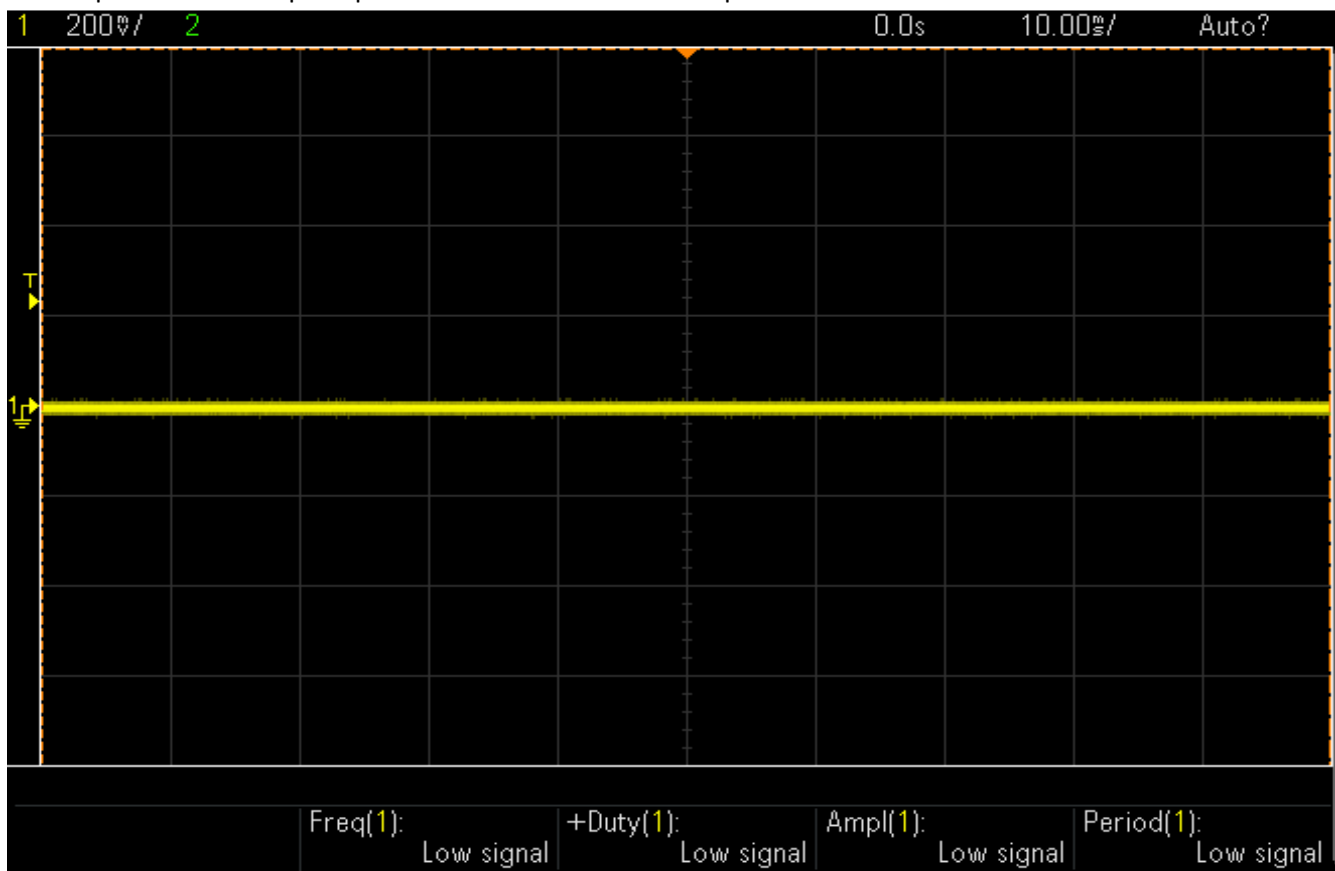


- This is when it entered IDLE mode which stopped toggling the ALE. And then It comes out after touching the INTO pin of 8051

DSO-X 11026, CN57246526: Mon Oct 30 06:30:24 2023



- Output of oscilloscope at power down mode. ALE low in power down mode



- Output of oscilloscope in watch dog mode. Which is checked by a flag whether watch dog is ON or OFF.

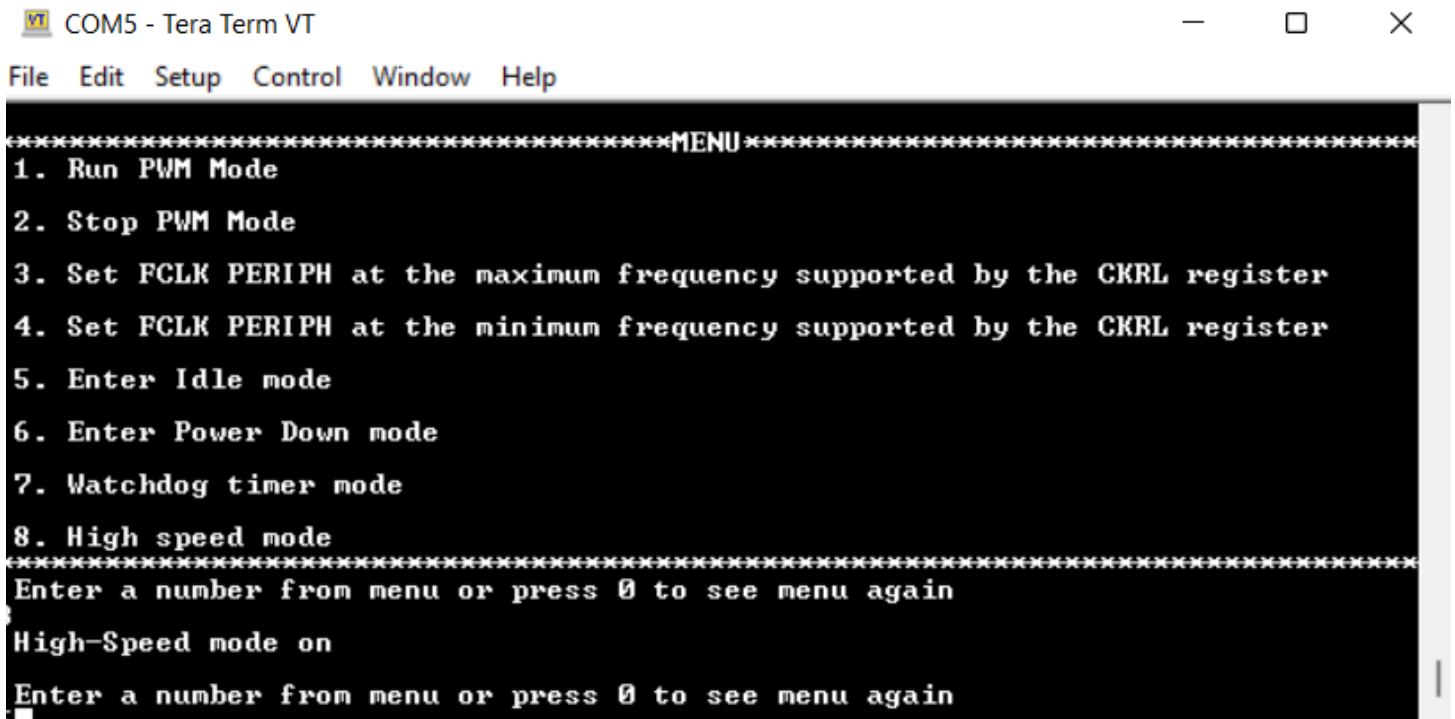
```

COM5 - Tera Term VT
File Edit Setup Control Window Help

*****MENU*****
1. Run PWM Mode
2. Stop PWM Mode
3. Set FCLK PERIPH at the maximum frequency supported by the CKRL register
4. Set FCLK PERIPH at the minimum frequency supported by the CKRL register
5. Enter Idle mode
6. Enter Power Down mode
7. Watchdog timer mode
8. High speed mode
*****
Enter a number from menu or press 0 to see menu again
?
Watch-dog mode
watchdog timer is on
Enter a number from menu or press 0 to see menu again

```


- Outputs of all inputs for all modes in terminal emulator. This is for High speed mode



VT COM5 - Tera Term VT

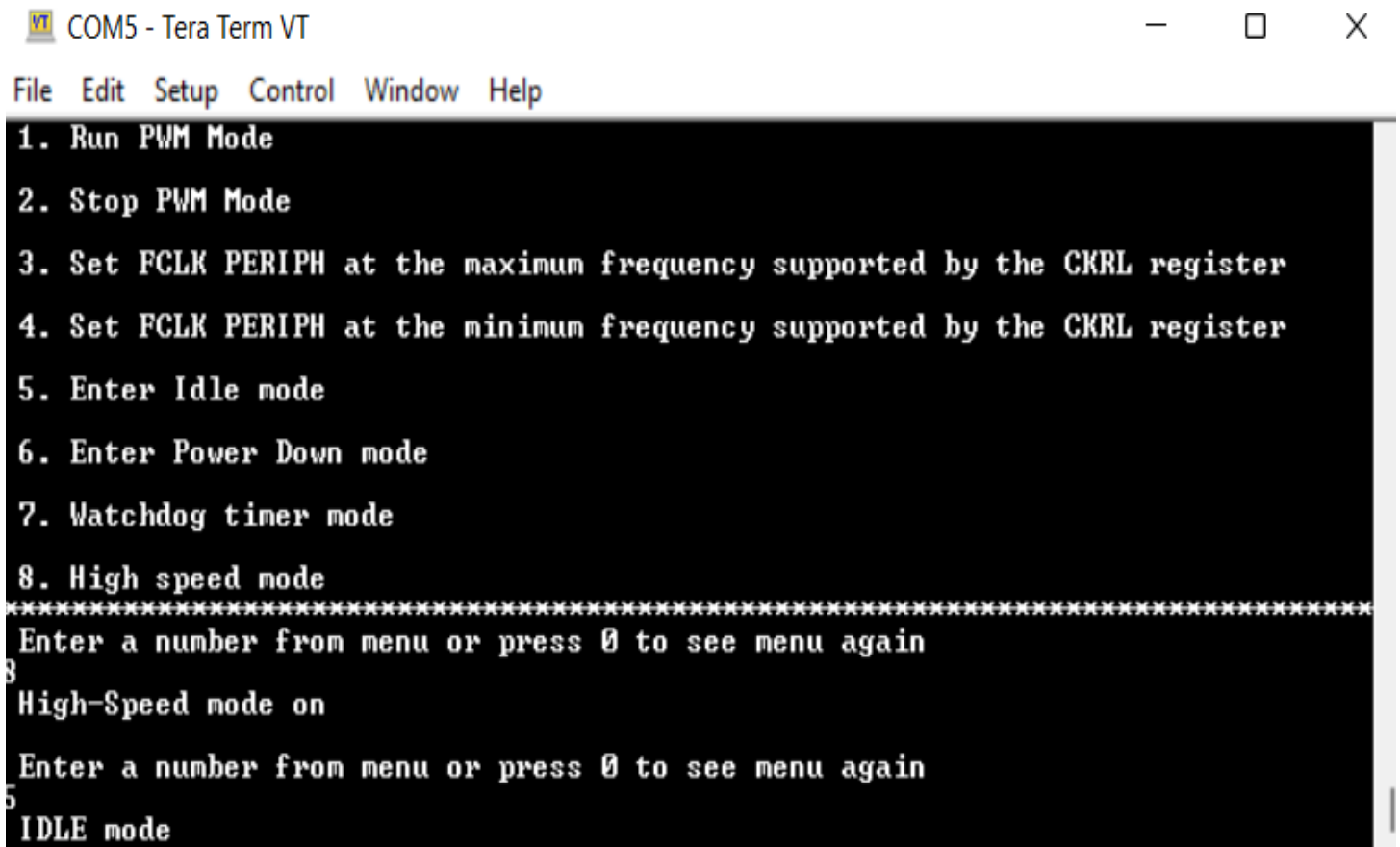
File Edit Setup Control Window Help

```

*****MENU*****
1. Run PWM Mode
2. Stop PWM Mode
3. Set FCLK PERIPH at the maximum frequency supported by the CKRL register
4. Set FCLK PERIPH at the minimum frequency supported by the CKRL register
5. Enter Idle mode
6. Enter Power Down mode
7. Watchdog timer mode
8. High speed mode
*****
Enter a number from menu or press 0 to see menu again
High-Speed mode on
Enter a number from menu or press 0 to see menu again

```

- This is for IDLE mode



VT COM5 - Tera Term VT

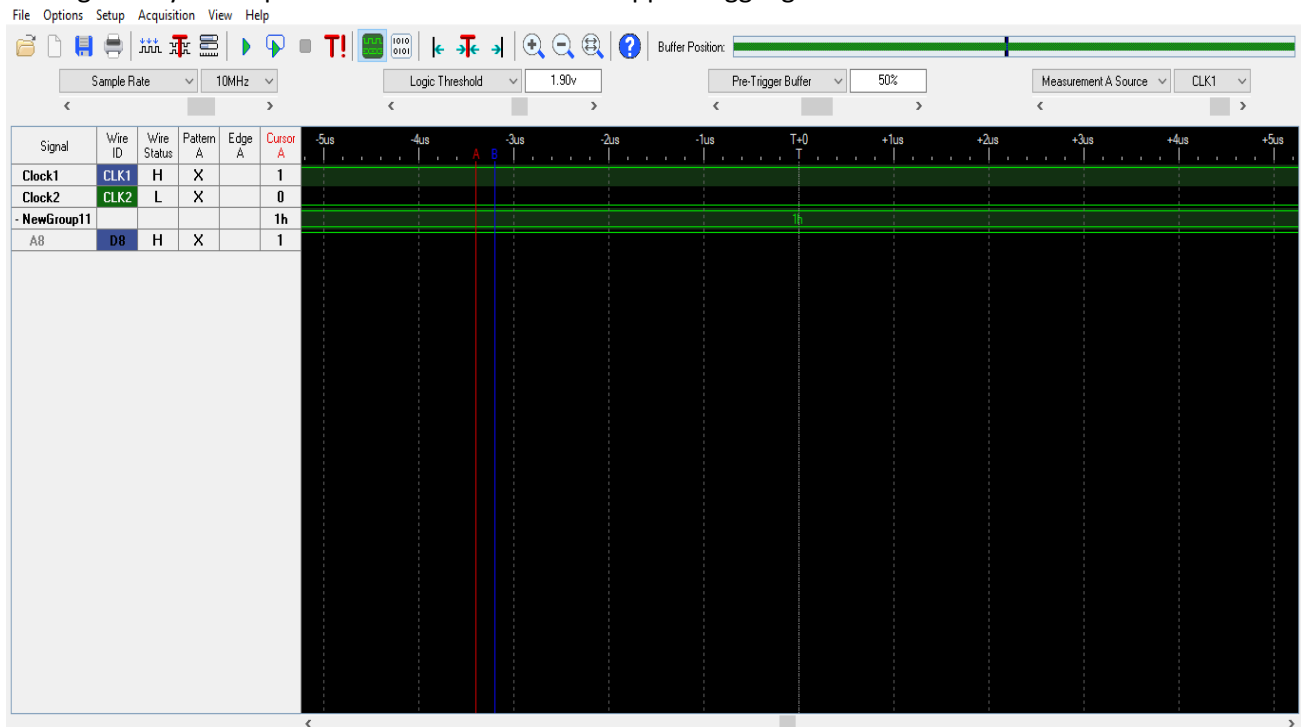
File Edit Setup Control Window Help

```

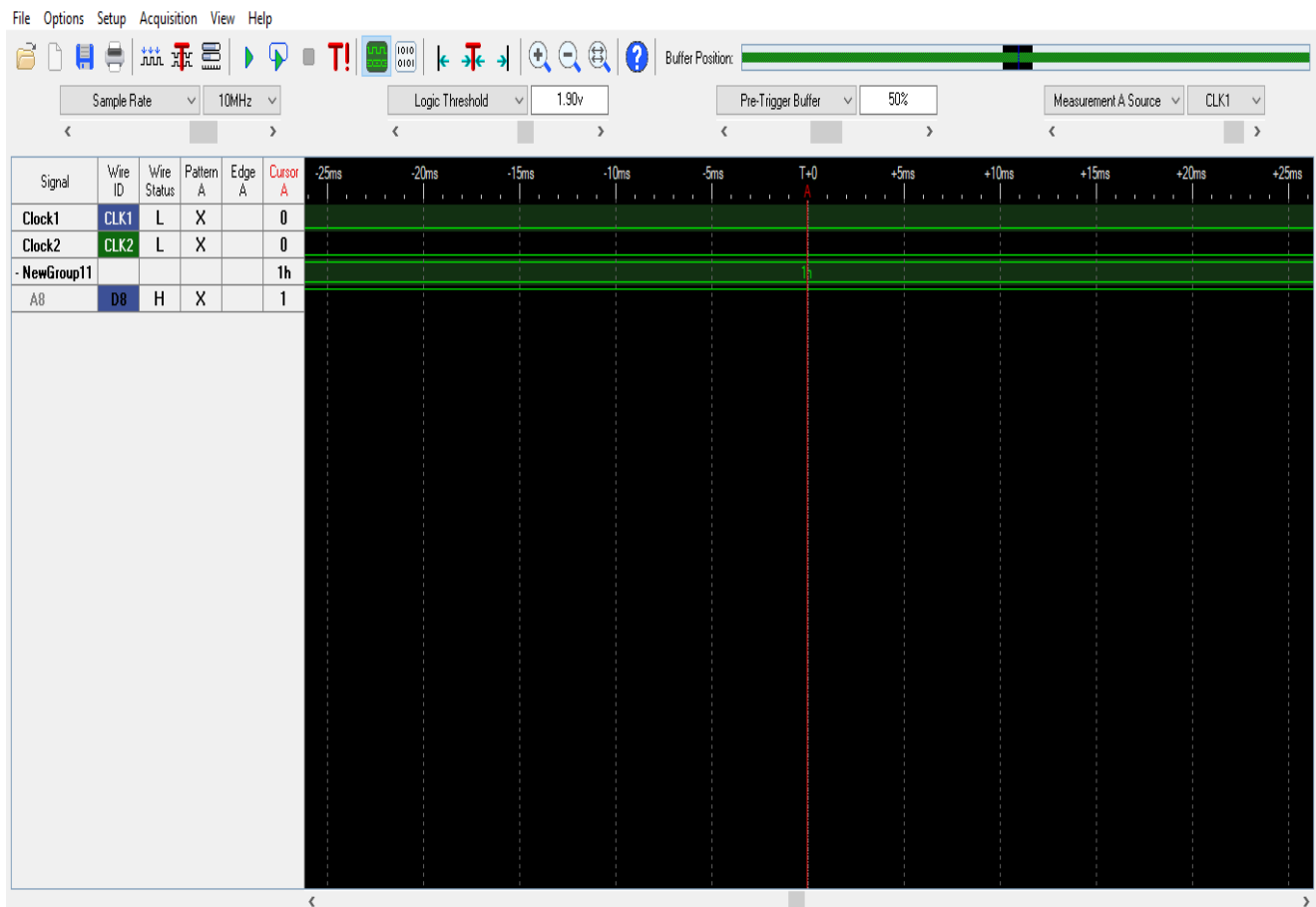
1. Run PWM Mode
2. Stop PWM Mode
3. Set FCLK PERIPH at the maximum frequency supported by the CKRL register
4. Set FCLK PERIPH at the minimum frequency supported by the CKRL register
5. Enter Idle mode
6. Enter Power Down mode
7. Watchdog timer mode
8. High speed mode
*****
Enter a number from menu or press 0 to see menu again
High-Speed mode on
Enter a number from menu or press 0 to see menu again
IDLE mode

```

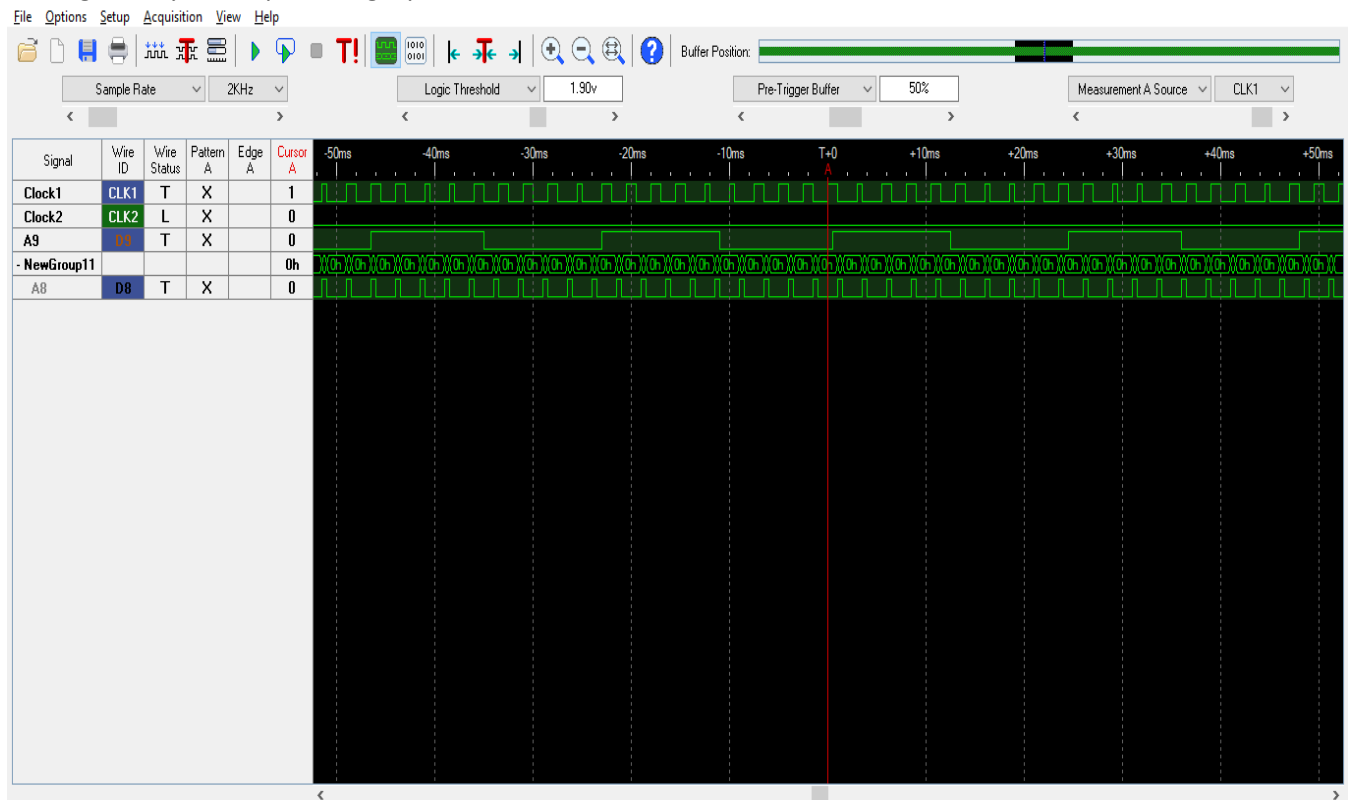
- Logic analyzer outputs for IDLE mode. Which stopped toggling of ALE in this mode.



- Logic analyzer output of power down mode.



- Logic analyzer output of highspeed mode.



- High-Speed Output Mode: In this mode, the PCA module can generate high-speed pulse signals with varying duty cycles, making it useful for tasks such as generating precise timing signals for applications like motor control or frequency synthesis.
- Pulse Width Modulator (PWM) Mode: The PCA module can function as a PWM generator, allowing it to produce a square wave output with a programmable duty cycle. PWM is commonly used for tasks like controlling the speed of motors and regulating the brightness of LEDs.
- Watchdog Timer Mode (Module 4): Module 4 of the PCA can be configured as a watchdog timer. This timer is designed to monitor the operation of the microcontroller and reset it if it fails to provide periodic "petting" signals. It enhances system reliability by preventing lockups or software glitches from causing system failures.
- All these modes are observed in oscilloscopes and logic analyzers.

OPTIONAL CHALLENGE PART3: Buffer with heap size of 5600

- Creation of buffer 0 and buffer 1 with 2400 bytes. And buffer 2 and buffer 3 with 200 and 300 bytes using + character.

```
PAULMON2 Loc:7FFF > Jump to memory location
Jump to memory location <7FFF>, or ESC to quit: 2000
running program:

***** HELLO :) *****

Enter a number between 32 & 4800 for buffer size
2400
The input number is:2400
Start Address of buffer0 = 0x3
Start Address of buffer1 = 0x965
Buffer_0 and Buffer_1 successfully created

OPTIONS TO CHOOSE
You can enter characters to be stored in buffer 0
Enter + to create a new buffer of size between 20 and 400 bytes
Enter - to delete a buffer.
Enter ? to generate a heap report.
Enter = to display current contents of buffer 0.
Enter @ to free all buffers and start program again.

*****THANK YOU*****
+

Creates a new buffer
Enter the new buffer size between 20 and 400
200
Successful allocated memory for Buffer_2
Buffer_2 has allocated a size of 200
Start Address of buffer_2 = 0x0
+

Creates a new buffer
Enter the new buffer size between 20 and 400
300
Successful allocated memory for Buffer_3
Buffer_3 has allocated a size of 300
Start Address of buffer_3 = 0x0
```

- Deletion of buffer 2 with – character and its heap report.

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
300
Successful allocated memory for Buffer_3
Buffer_3 has allocated a size of 300
Start Address of buffer_3 = 0x0
Enter a valid buffer number
2
Deleting buffer 2
Buffer 2 is Free
?
***** REPORT OF HEAP *****
Buffer 0
Its Start Address = 0x3
Its Ending Address = 0x963
Its Size = 2400
Current Storage characters in buffer 0 = 0
Current Free Spaces in buffer = 2400
-----
Buffer 1
Its Start Address = 0x965
Its Ending Address = 0x12C5
Its Size = 2400
Storage characters in buffer =0, because storage characters are @ buffer 0
Current Free Spaces in buffer = 2400
-----
Buffer 3
Start Address = 0x1391
Ending Address = 0x14BD
Buffer Size = 300
Storage characters in buffer = 0
Free Spaces in buffer = 300
-----
Number of storage characters = 0
Total number of characters received = 4
Total number of buffers that were allocated since the start of the program = 4
Total storage characters stored since last '?' = 0
```

- Creating one more buffer with size of 100 bytes.

```

COM5 - Tera Term VT
File Edit Setup Control Window Help
+
Creates a new buffer
Enter the new buffer size between 20 and 400
100
Successful allocated memory for Buffer_4
Buffer_4 has allocated a size of 100
Start Address of buffer_4 = 0x0
?
***** REPORT OF HEAP *****
Buffer 0
Its Start Address = 0x3
Its Ending Address = 0x963
Its Size = 2400
Current Storage characters in buffer 0 = 0
Current Free Spaces in buffer = 2400
-----

Buffer 1
Its Start Address = 0x965
Its Ending Address = 0x12C5
Its Size = 2400
Storage characters in buffer =0, because storage characters are @ buffer 0
Current Free Spaces in buffer = 2400
-----

Buffer 3
Start Address = 0x1391
Ending Address = 0x14BD
Buffer Size = 300
Storage characters in buffer = 0
Free Spaces in buffer = 300
-----

Buffer 4
Start Address = 0x12C7
Ending Address = 0x132B
Buffer Size = 100
Storage characters in buffer = 0

```

- Creation of buffer with 210 bytes of size.

```

COM5 - Tera Term VT
File Edit Setup Control Window Help
Enter the new buffer size between 20 and 400
210
Successful allocated memory for Buffer_6
Buffer_6 has allocated a size of 210
Start Address of buffer_6 = 0x0
?
***** REPORT OF HEAP *****
Buffer 0
Its Start Address = 0x3
Its Ending Address = 0x963
Its Size = 2400
Current Storage characters in buffer 0 = 0
Current Free Spaces in buffer = 2400
-----

Buffer 1
Its Start Address = 0x965
Its Ending Address = 0x12C5
Its Size = 2400
Storage characters in buffer =0, because storage characters are @ buffer 0
Current Free Spaces in buffer = 2400
-----

Buffer 3
Start Address = 0x1391
Ending Address = 0x14BD
Buffer Size = 300
Storage characters in buffer = 0
Free Spaces in buffer = 300
-----

Buffer 6
Start Address = 0x14BF
Ending Address = 0x1591
Buffer Size = 210
Storage characters in buffer = 0
Free Spaces in buffer = 210

```


- Creation of buffer with 800 bytes. Whose allocation will be failed because of heap memory insufficiency.

```

-----
Buffer 3
Start Address = 0x1391
Ending Address = 0x148D
Buffer Size = 300
Storage characters in buffer = 0
Free Spaces in buffer = 300
-----

Number of storage characters = 0
Total number of characters received = 16
Total number of buffers that were allocated since the start of the program = 7
Total storage characters stored since last '?' = 0
-----

+
Creates a new buffer
Enter the new buffer size between 20 and 400
800
Memory Allocation for Buffer_7 Failed
Press '+' to try again

```

The above are required heap buffer with size of 5600 outputs.

- What operating system (including revision) did you use for your code development?
Windows 11
- What compiler (including revision) did you use?
SDCC 4.1.0
- What exactly (include name/revision if appropriate) did you use to build your code (what IDE, make/makefile, or command line)?
Code blocks(SDCC) for AT89C51 along with STM32 IDE
- Did you install and use any other software tools to complete your lab assignment?
NO
- Did you experience any problems with any of the software tools? If so, describe the problems
No

LAB 3 Learnings:

1. Memory allocation and error handling.
2. Compiling with SDCC.
3. Writing interrupt handler for AT89C51 in C.
4. Designing a user Interface using UART.
5. Paulmon2 and methods to use it
6. Analyze the output files of SDCC
7. X2 mode, PCA, Idle and power down mode in AT89C51.
8. Learn more about logic analyzer and how to debug using it
9. SDCC Heap analysis
10. PAULMON as an on-chip debugger