LAB REPORT - 3

EMBEDDED SYSTEMS DESIGN

Shrinitihi Venkatesan
F’22

ECEN 5613 **Lab #3 Signoff Sheet** **Fall 2022**

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: SHRINITHI VENKATESAN

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: SHRINITHI

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: 115200
- ☑ Knows how to use SDCC [IDE or make optional]

Manas MD 10/22/22
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

Manas MD 10/22/22
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

SSharcade 10/28/22
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	☐	☐	☐	☑	☐
Hardware physical implementation	☐	☐	☐	☐	☐
Part 1 Required Elements functionality	☐	☐	☐	☐	☐
Sign-off done without excessive retries		☐	☑		
Student understanding and skills	☐	☐	☐	☑	☐
Overall Demo Quality (Part 2 elements)	☐	☐	☐	☑	☐

FOR INSTRUCTOR USE ONLY Part 3 Elements	Not Applicable	Below Expectation	Meets Requirements	Exceeds Requirements	Outstanding
Part 3 Required Elements functionality	☐	☐	☐	☑	☐
Supplemental Elements functionality	☐	☐	☐	☐	☐
Student understanding and skills	☐	☐	☐	☐	☐
Overall Demo Quality (Part 3 elements)	☐	☐	☐	☑	☐

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

Lab 3 Part 1 & 2 Comments

(+) Neat Schematic
(+) Pantheon functional.
(+) Memory map verified using Pantheon.
(+) Good knowledge of terminal emulator.
(+) Heap code tested for all cases
(+) Hex dump verified.
(+) No. memory segmentation
(+) Debug port functional.

Lab 3 Part 3 Signoff

(+) ARM code functional.
  (+) PWM increases & decreases correctly
  (+) 'A' & 'B' character functionality works
  (+) LED brightness varies. (Supplemental)

(+) 8051 Supplemental Part
  (+) PWM functional (X2 mode)
  (+) Low freq & high freq modes work
  (#) Goes into Idle mode (ext interrupt not configured)
  (#) Watchdog resets the system (not fed by timer).
  (+) High speed mode works.
(+) Challenge on Heap analysis done.

LAB3_PART1;

PAULMON21

The Paulmon code was edited to supplement the changes required for AS31 and the following code was used to fill the required memory space between 0x0400 0x7FFF of the XRAM.

COM6 - PuTTY

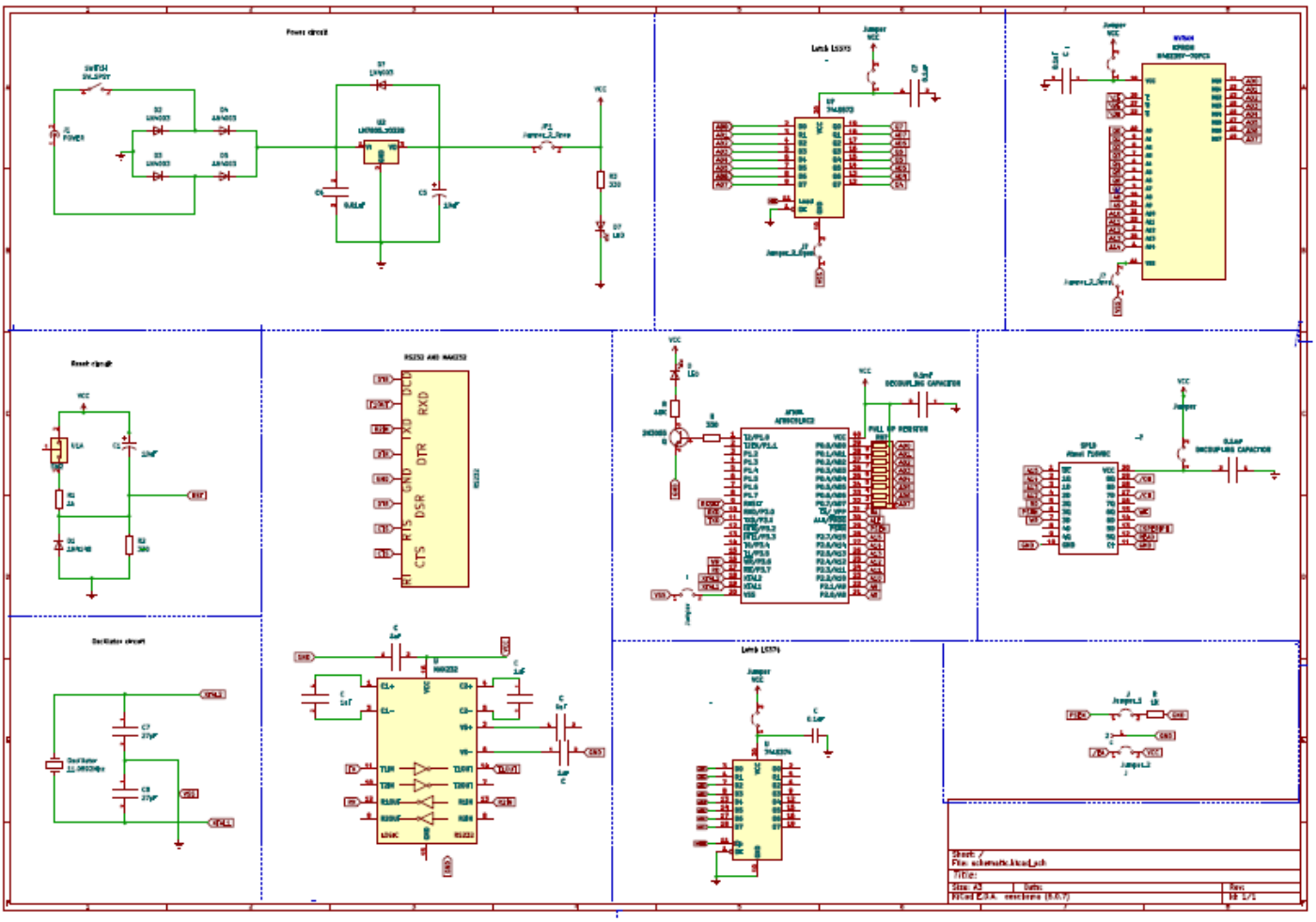
```
DATA        8051 External Memory Editor, Paul Stoffregen, 1996
ADDR: +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   ASCII EQUIVILANT
------------------------------------------------------------------------
7F00: AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA   ****************
7F10: AA AA AA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

COM6 - PuTTY

```
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
------------------------------------------------------------------------
0200: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0210: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0220: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0230: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0240: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0250: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0260: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0270: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0280: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
0290: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
02A0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
02B0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
02C0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
02D0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
02E0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
02F0: 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55   UUUUUUUUUUUUUUUU
------------------------------------------------------------------------
^E-Edit  ^G=Goto  ^C=Code  ^D=Data  ^L=Redraw  ^Q=Quit
```

Schematic and SPLD Code:

From the previous labs, only changes made was that the Write Enable(WE) was connect to write pin of SPLD. and the same changes were added to the schematic as well.



SDCC:

Basic SDCC functionalities were learnt and the settings to use the Code Blocks for the SDCC compiler was configured for the upcoming lab programs.

LAB3_PART2:

An User Interface was created using SDCC programming and was flashed into the 8051 through the RS232 module and the following functionalities were verified by the TA's.

Entering characters and storing in buffer 0:

```
|*********************************************|

Enter the character
B
The ASCII of the input character is 66
The character is stored in the Buffer 0
Count = 2
|*********************************************|
|  Choose a character from the below options   |
|  A-Z  | Character to store in the buffer     |
|  +    | Allocate a new buffer                |
|  -    | Delete a buffer                      |
|  ?    | Display the heap report              |
|  =    | Display contents of Buffer_0         |
|  @    | Free all the buffers                 |
|*********************************************|

Enter the character

The ASCII of the input character is 13
|*********************************************|
|  Choose a character from the below options   |
|  A-Z  | Character to store in the buffer     |
|  +    | Allocate a new buffer                |
|  -    | Delete a buffer                      |
|  ?    | Display the heap report              |
|  =    | Display contents of Buffer_0         |
|  @    | Free all the buffers                 |
|*********************************************|

Enter the character
+
The ASCII of the input character is 43
Allocating a new buffer
Enter a heap size between 30 and 300
 digit_1 = 0
 digit_2 = 4
 digit_3 = 5

New heap size entered is 45
You have given a valid heap size

New heap size entered is 45
```

```
The ASCII of the input character is 13
|*************************************************|
|  Choose a character from the below options      |
|  A-Z  | Character to store in the buffer        |
|  +    | Allocate a new buffer                   |
|  -    | Delete a buffer                         |
|  ?    | Display the heap report                 |
|  =    | Display contents of Buffer_0            |
|  @    | Free all the buffers                    |
|*************************************************|

Enter the character
+
The ASCII of the input character is 43
Allocating a new buffer
Enter a heap size between 30 and 300
 digit_1 = 0
 digit_2 = 4
 digit_3 = 5

New heap size entered is 45
You have given a valid heap size

New heap size entered is 45

Memory Allocation Successful for Buffer 2

Buffer 2 allocated of size 45

Address of buffer 2 is 0x2FE
```

```
_
|  Choose a character from the below options    |
|  A-Z  | Character to store in the buffer      |
|  +    | Allocate a new buffer                 |
|  -    | Delete a buffer                       |
|  ?    | Display the heap report               |
|  =    | Display contents of Buffer_0          |
|  @    | Free all the buffers                  |
|***********************************************|

Enter the character
-
The ASCII of the input character is 45
Please enter the number of buffer to delete
 digit_1 = 0
 digit_2 = 2

The buffer number provided to delete is 2

 Deleting buffer 2 in progress

 Buffer 2 is free
|***********************************************|
|  Choose a character from the below options    |
|  A-Z  | Character to store in the buffer      |
|  +    | Allocate a new buffer                 |
|  -    | Delete a buffer                       |
|  ?    | Display the heap report               |
|  =    | Display contents of Buffer_0          |
|  @    | Free all the buffers                  |
|***********************************************|
```

```
Enter the character
?
The ASCII of the input character is 63

 <<<<<<HEAP REPORT of Buffer_0>>>>>>

Starting Address of Buffer_0 is 0x7A

Ending Address of Buffer_0 is 0x1BA

Buffer Size of Buffer_0 320

Stored characters in buffer = 2

Free Spaces in buffer 318

 <<<<<<HEAP REPORT of Buffer_1>>>>>>

Starting Address of Buffer_1 is 0x1BC

Ending Address of Buffer_1 is 0x2FC

Buffer Size 320
```

```
Enter the character
S
The ASCII of the input character is 83
The character is stored in the Buffer 0
Count = 3
|*********************************************|
|  Choose a character from the below options    |
|  A-Z  | Character to store in the buffer      |
|  +    | Allocate a new buffer                 |
|  -    | Delete a buffer                       |
|  ?    | Display the heap report               |
|  =    | Display contents of Buffer_0          |
|  @    | Free all the buffers                  |
|*********************************************|

Enter the character
D
The ASCII of the input character is 68
The character is stored in the Buffer 0
Count = 4
|*********************************************|
|  Choose a character from the below options    |
|  A-Z  | Character to store in the buffer      |
|  +    | Allocate a new buffer                 |
|  -    | Delete a buffer                       |
|  ?    | Display the heap report               |
|  =    | Display contents of Buffer_0          |
|  @    | Free all the buffers                  |
|*********************************************|

Enter the character
=
The ASCII of the input character is 61

Contents in Buffer 0
0x7A>> 41 42 53 44|*********************************************|
```

```
Enter the character
@
The ASCII of the input character is 64

Free Buffer 0

Buffer 0 will now become empty

Freeing Buffer 1

Buffer 1 is empty

Freeing buffer_2

Buffer 2 freed

Freeing buffer_3

Buffer 3 freed

Freeing buffer_4

Buffer 4 freed
|*********************************************|
|  Choose a character from the below options    |
|  A-Z  | Character to store in the buffer      |
|  +    | Allocate a new buffer                 |
|  -    | Delete a buffer                       |
|  ?    | Display the heap report               |
|  =    | Display contents of Buffer_0          |
|  @    | Free all the buffers                  |
|*********************************************|

Enter the character
?
The ASCII of the input character is 63

Buffer 0 is empty , so no report is available!

Buffer 1 is empty , so no report is available!

All Buffers created are empty, so no heap report available!
```

Virtual Debug Port:

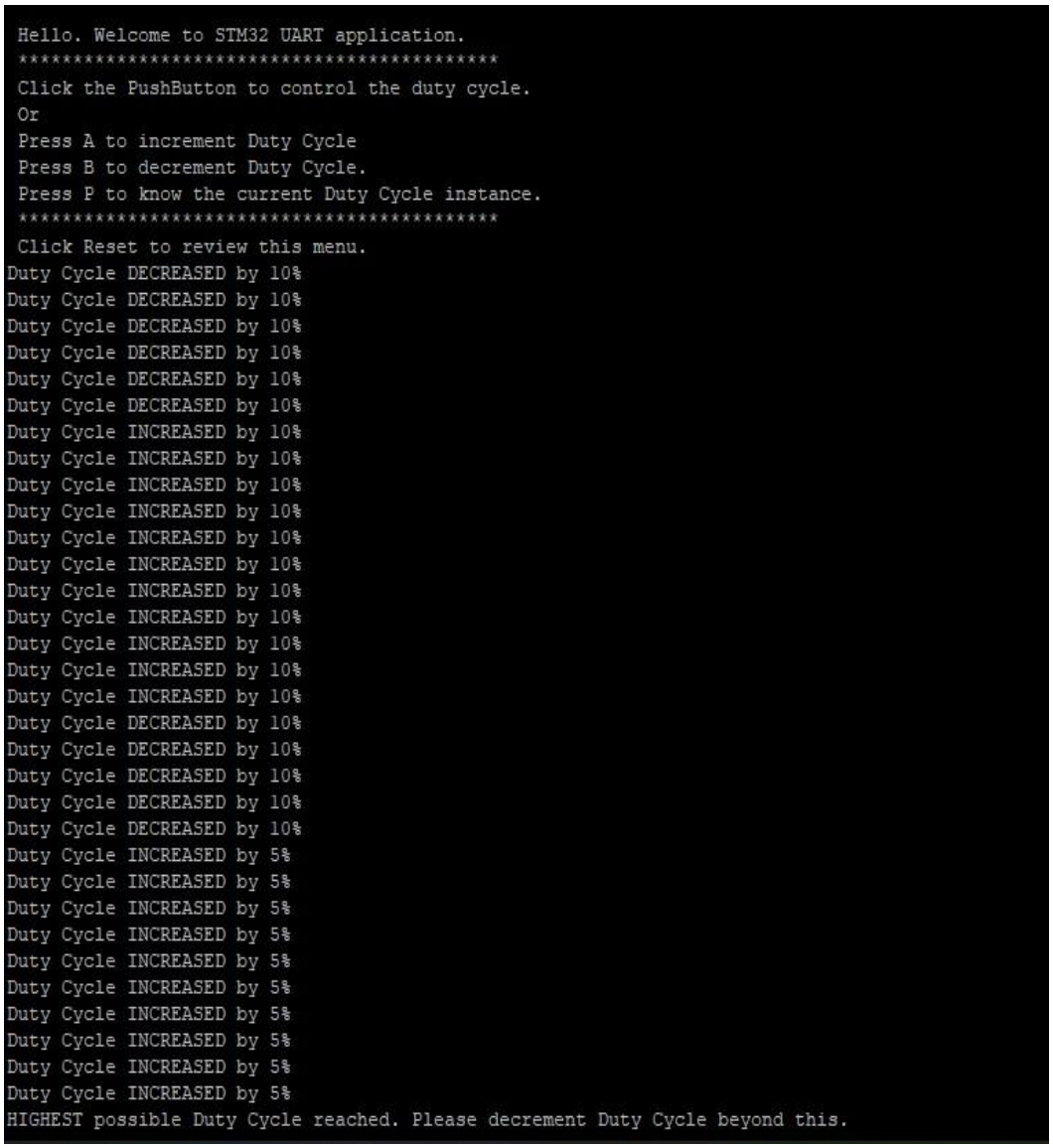
A debug Port was enabled and respective value was assigned to respective points of debug and the following logic was analyzed through LogicPort.

0x52 was assigned to ‘+’ of the UI program and was added to the debug port address 0x7f. It is observed that when the WE signal is low, the write trigger is enabled and the write happens.

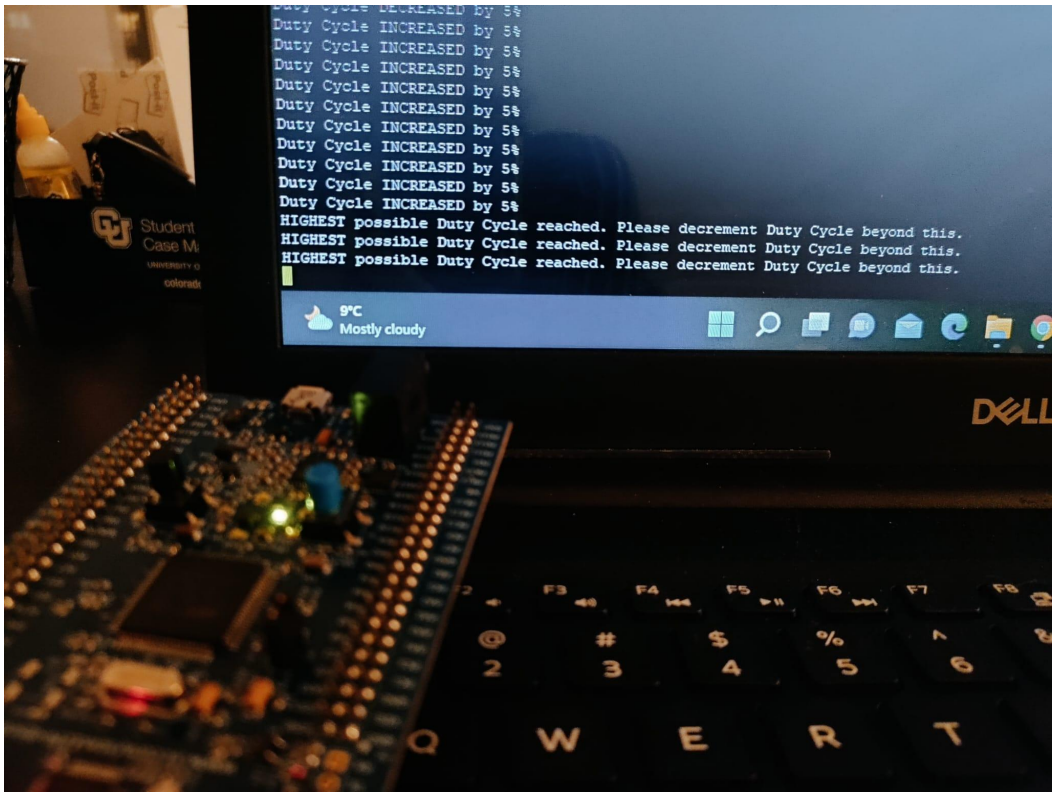
LAB3_Part3:

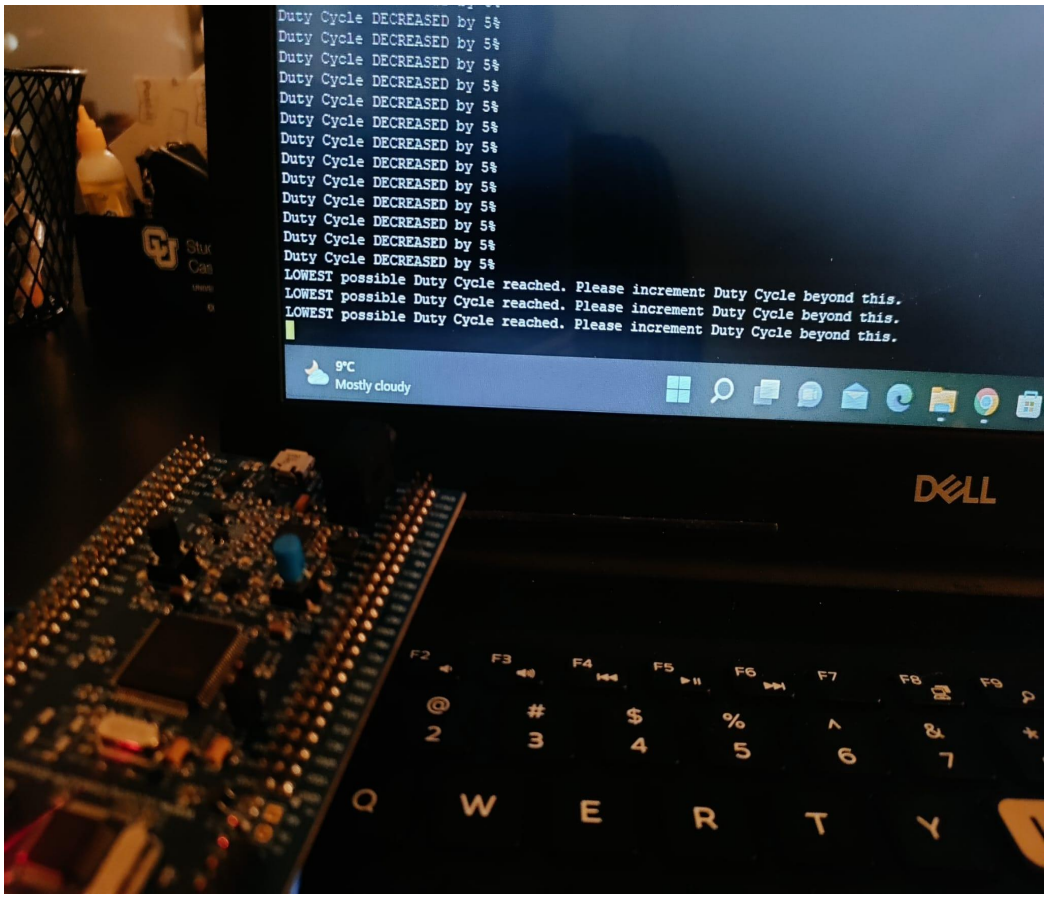
ARM_UI functionality for STM32

The UI was created using STM32 and USART functions were configured. The PWM was modified according to the program requirements and the LED pin 12 (green) was configured to show the output of the pwm.



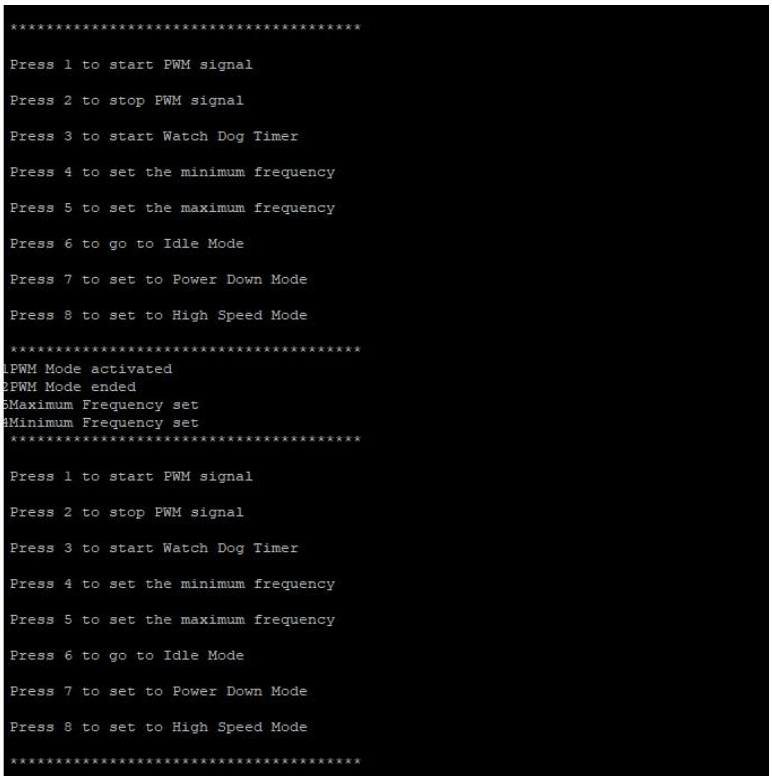
```
HIGHEST possible Duty Cycle reached. Please decrement Duty Cycle beyond this.
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
LOWEST possible Duty Cycle reached. Please increment Duty Cycle beyond this.
LOWEST possible Duty Cycle reached. Please increment Duty Cycle beyond this.
Duty Cycle is at this instance is:
000
Duty Cycle INCREASED by 10%
Duty Cycle is at this instance is:
009
Duty Cycle INCREASED by 10%
Duty Cycle INCREASED by 10%
Duty Cycle INCREASED by 10%
Duty Cycle INCREASED by 10%
Duty Cycle INCREASED by 10%
Duty Cycle INCREASED by 10%
Duty Cycle is at this instance is:
070
Duty Cycle INCREASED by 10%
Duty Cycle INCREASED by 10%
Duty Cycle INCREASED by 10%
Duty Cycle DECREASED by 10%
Duty Cycle DECREASED by 10%
Duty Cycle DECREASED by 10%
Duty Cycle is at this instance is:
```


Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
Duty Cycle INCREASED by 5%
HIGHEST possible Duty Cycle reached. Please decrement Duty Cycle beyond this.
HIGHEST possible Duty Cycle reached. Please decrement Duty Cycle beyond this.
HIGHEST possible Duty Cycle reached. Please decrement Duty Cycle beyond this.
9°C
Mostly cloudy
DELL

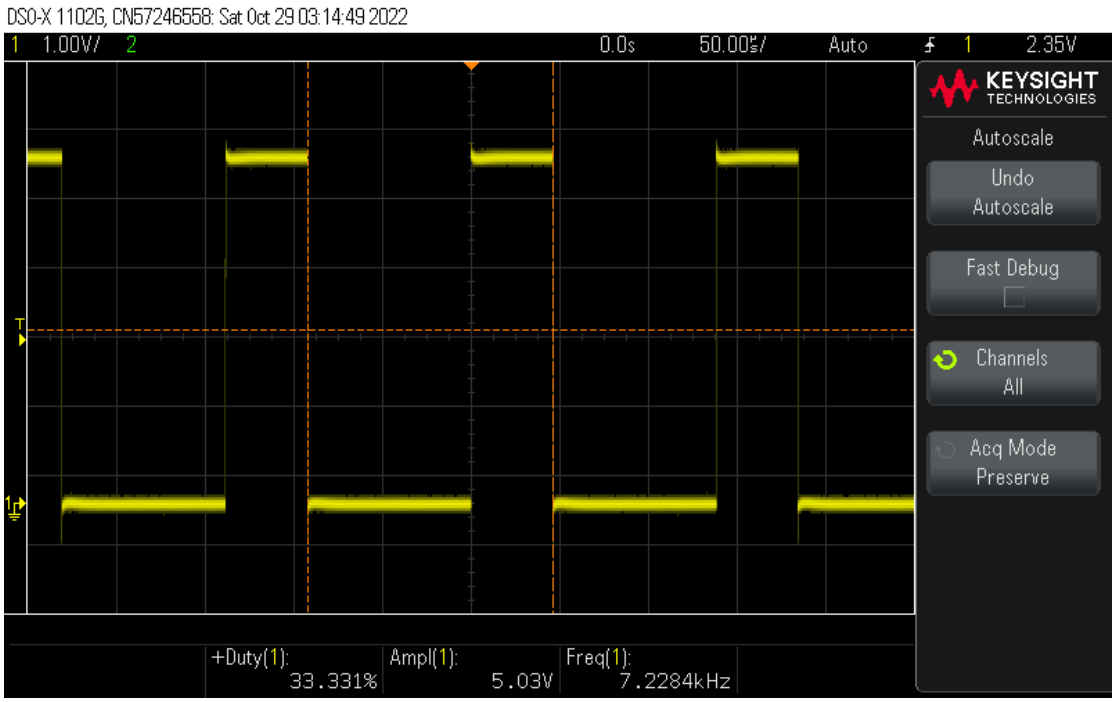

Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
Duty Cycle DECREASED by 5%
LOWEST possible Duty Cycle reached. Please increment Duty Cycle beyond this.
LOWEST possible Duty Cycle reached. Please increment Duty Cycle beyond this.
LOWEST possible Duty Cycle reached. Please increment Duty Cycle beyond this.
9°C
Mostly cloudy
DELL

SUPPLEMENT_8051:

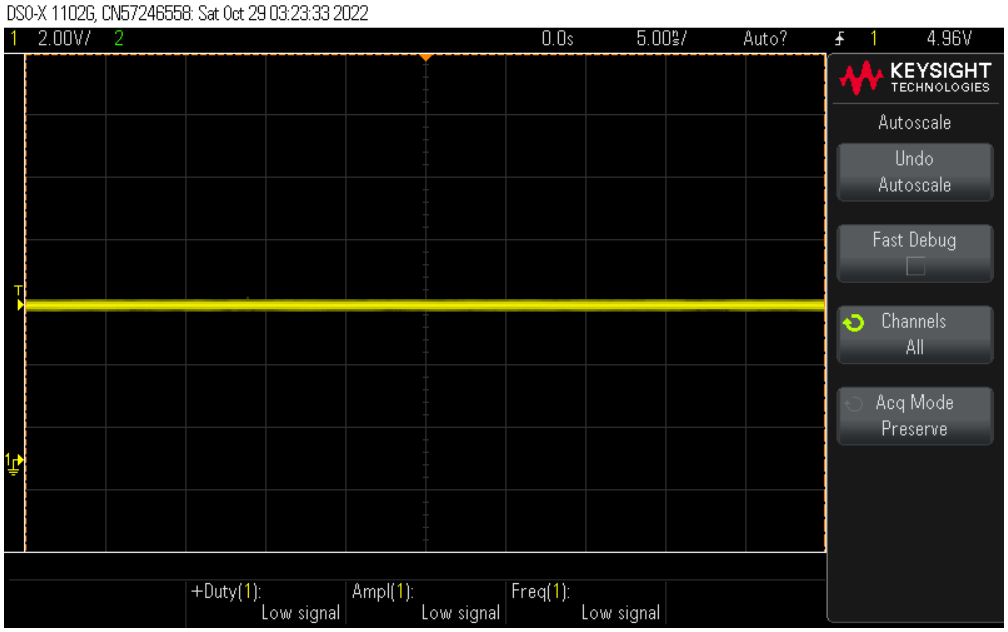
8051 was configured to use various PCA modes and following outputs were verified by the TA.



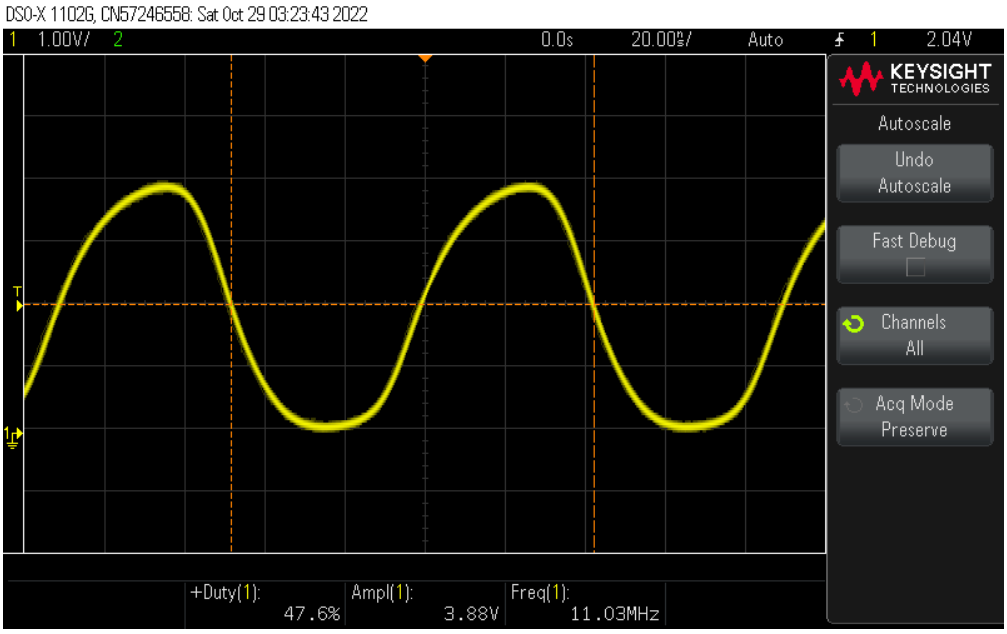
Minimum Frequency



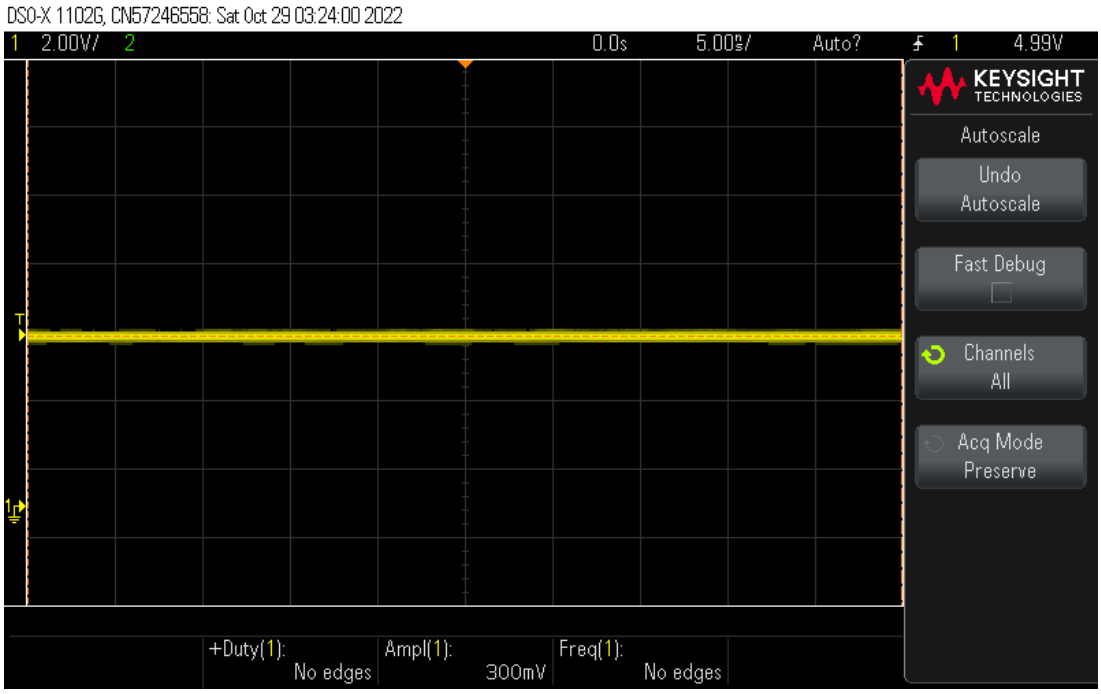
IDLE Mode



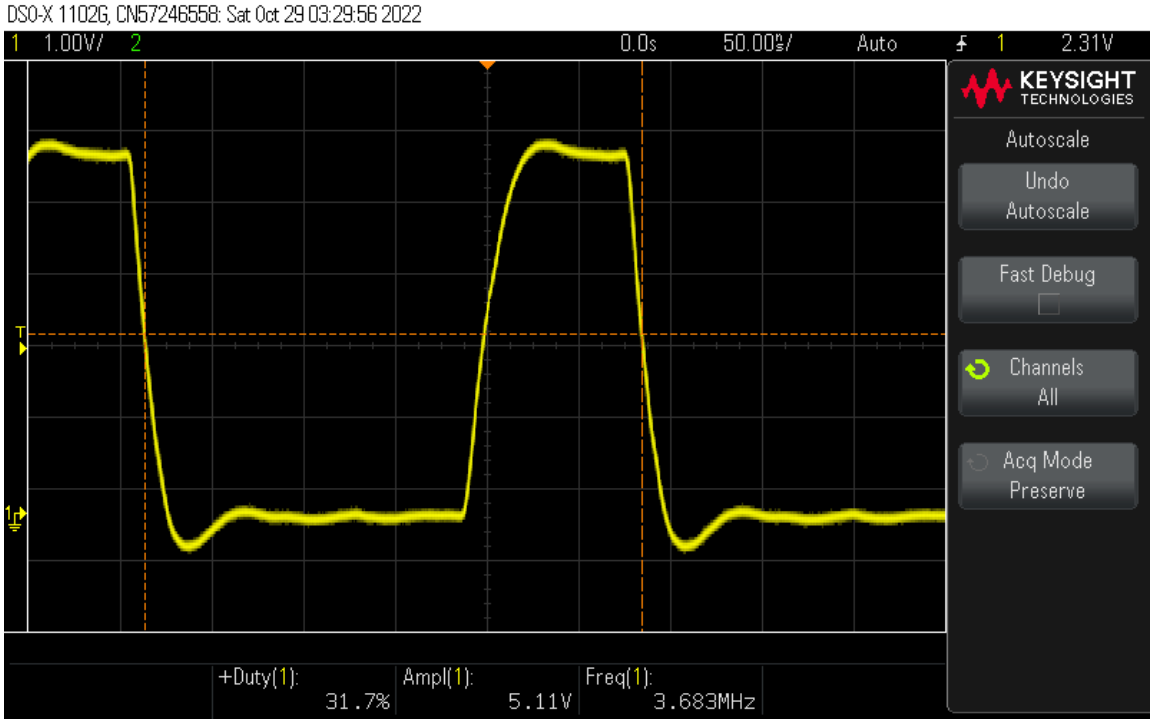
X2 signal of PWM



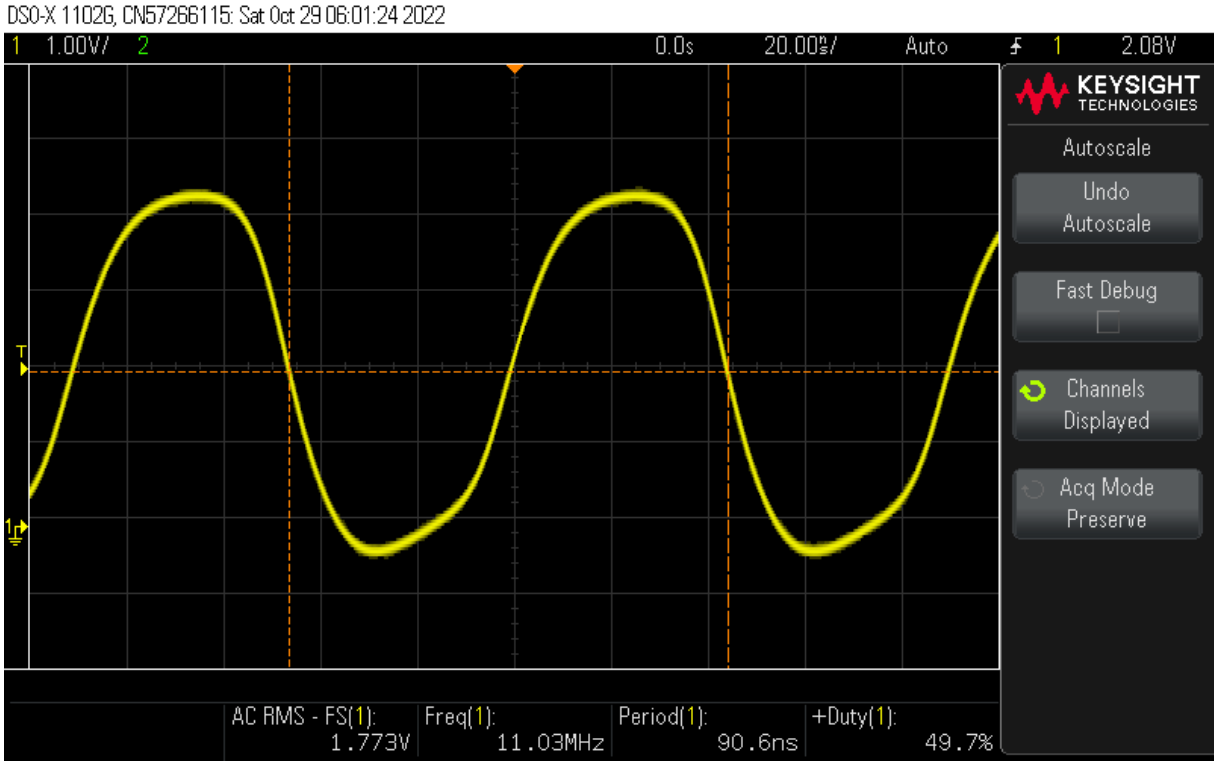
WATCHDOG Timer mode.



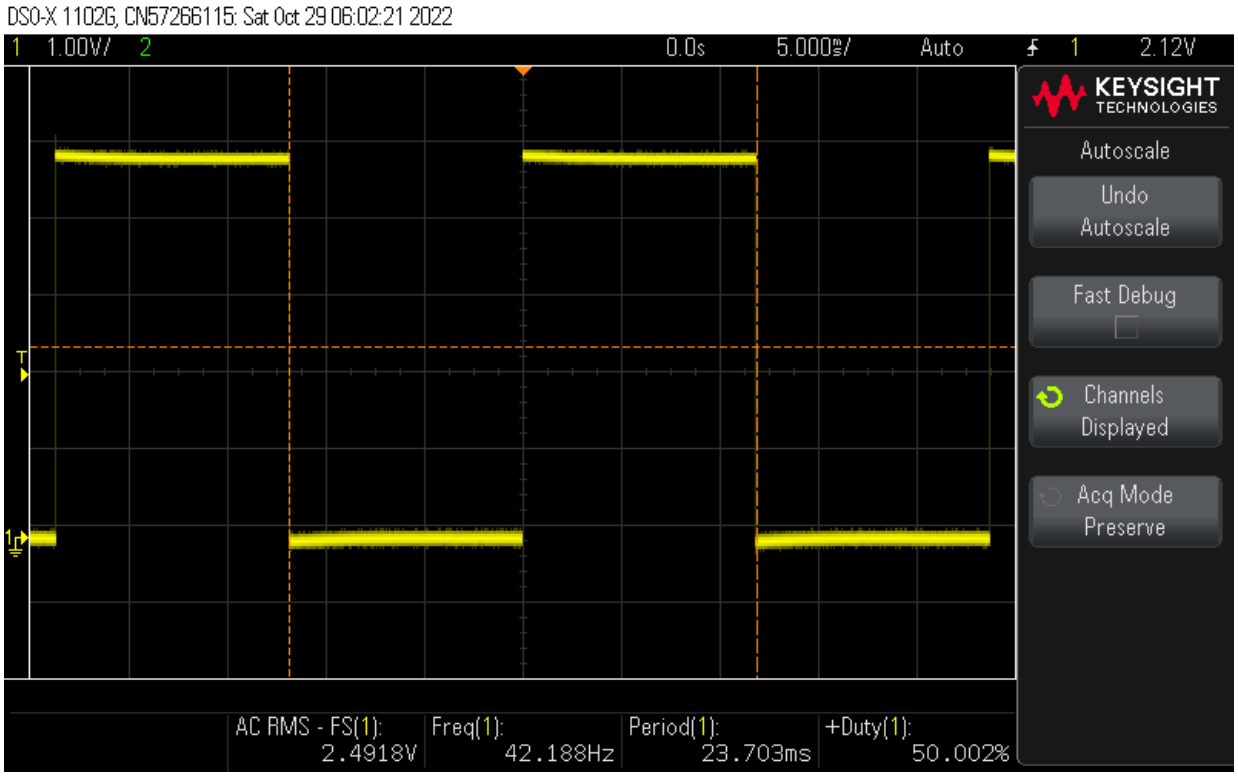
Maximum Frequency.



25% duty cycle:



High speed mode:



QUESTIONS:

a) What operating system (including revision) did you use for your code development?

• Windows 10

b) What compiler (including revision) did you use?

i) Small Device C Compiler (8051)

• SDCC 4.2.0

c) What exactly (include name/revision if appropriate) did you use to build your code (what IDE, make/makefile, or command line)?

• IDE : CodeBlocks 20.03

• IDE for STM32: STM32CUBEMX

• Command line: Putty (Release 0.77)

d) Did you install and use any other software tools to complete your lab assignment?

• PUTTY: Command line

• PAULMON2 is a user-friendly monitor program, used on 8051 microcontroller, which help debugging the program. I have used it to run my code and to decode the heap memory.

• KiCad 6.0 is a software suite used for Electronic Design Automation (EDA). I have developed schematic using the same.

• FLIP: Flip helps in-system programming of flash devices through RS232, USB or CAN. I have flashed my .hex files to 8051 board using the same

e) Did you experience any problems with any of the software tools? If so, describe the problems

• Initially I had trouble using Flip, it was giving a timeout error.

* Later I had issues with my SDCC UI Programming, which I sorted but it took me a lot of time to configure the UART programming. I was lagging behind in this lab, but I also took my time to understand the lab components clearly at my own pace.