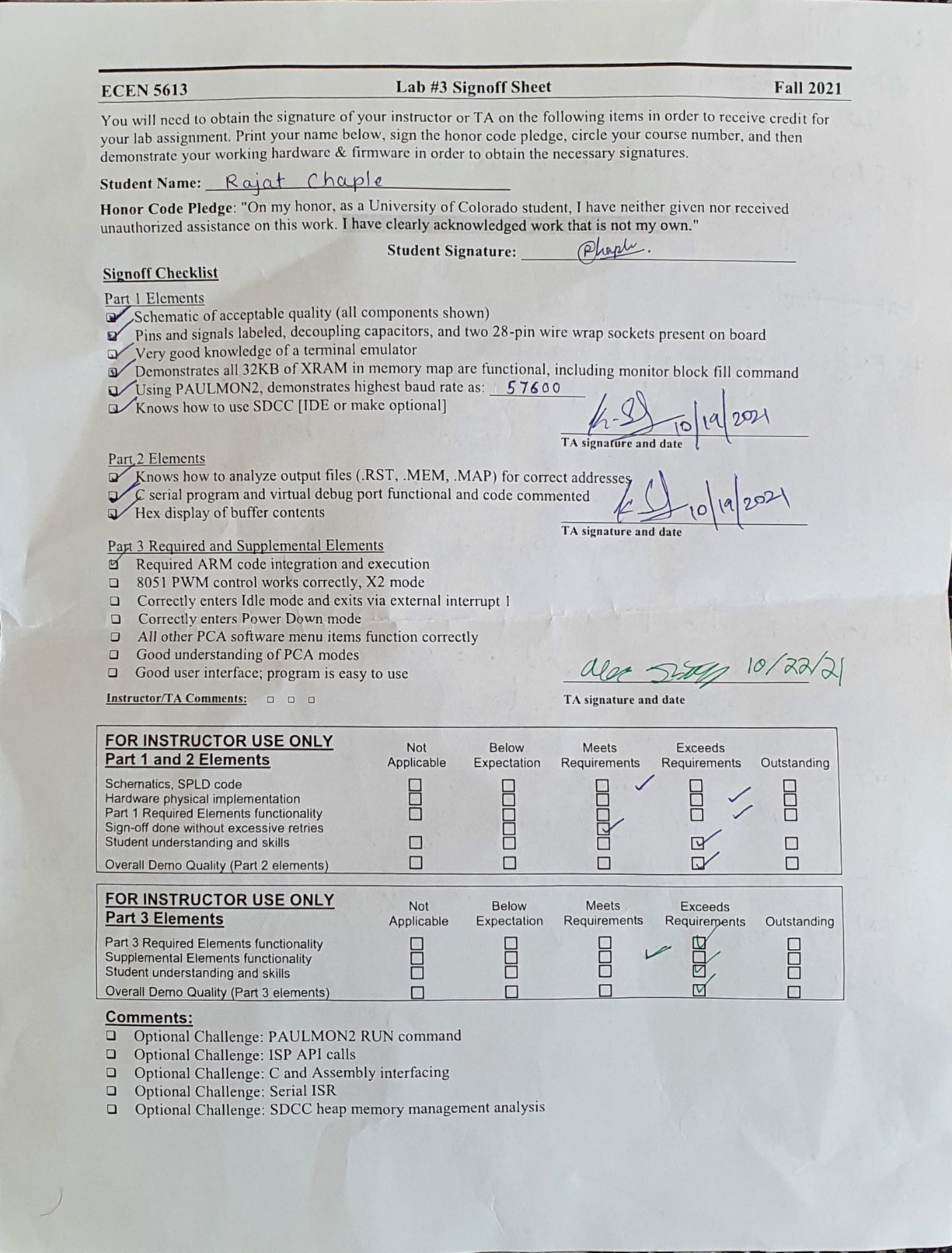
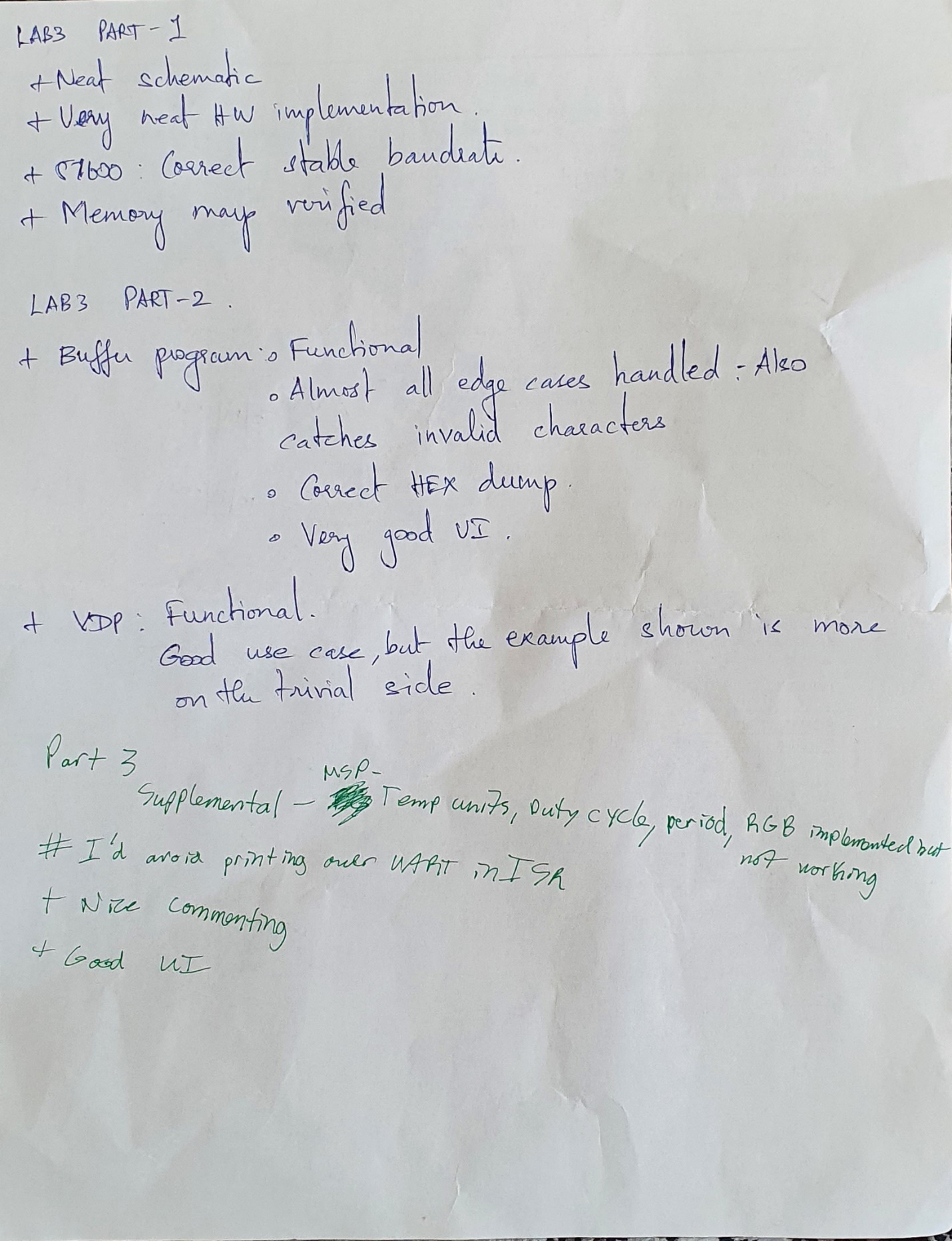
**ESD 5613 Lab3 Submission**

**(Rajat Chaple)**

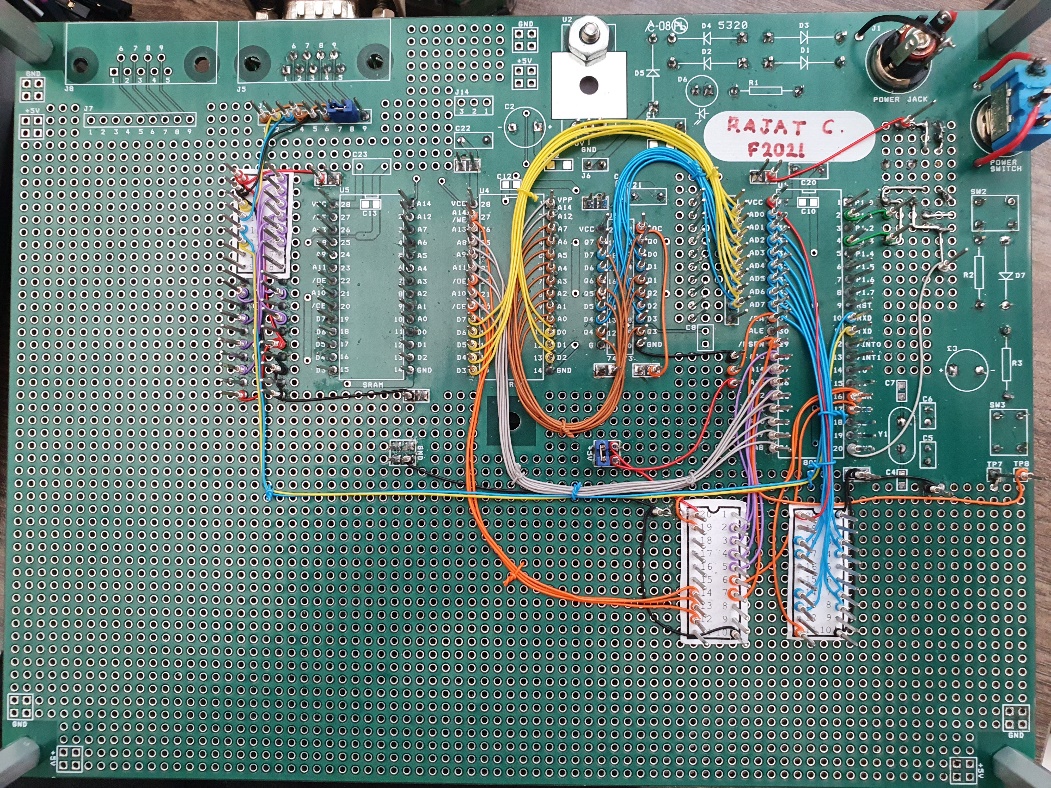


****

**PCB Layout**

A picture containing text, electronics, circuit

Description automatically generated



**Part 1:**

1. Decoding logic:

Table

Description automatically generated

For address space: 0000h – 7FFFh

1. Programming using batchisp

Text

Description automatically generated

1. Paulmon2

Application code starts at 2000h

Text

Description automatically generated

Text

Description automatically generated

Only memory written is : 0000h to 7FFFh Text

Description automatically generated

Text

Description automatically generated

1. **SDCC to generate hex file**

**Written a program to print out following and toggle a pin:**

Table

Description automatically generated with medium confidence

Diagram

Description automatically generated

**Mem file : indicates memory structure**

Table

Description automatically generated with low confidence

**Ihx file : application starts at 2000**

A screenshot of a computer

Description automatically generated with medium confidence

**Rst file : includes object addresses and function addresses**

Table

Description automatically generated with medium confidence

**Maximum supported baud rate**

57600

Measured as follow:

1. Verified that paulmon2 setting is autobaud
2. Changed tera-term baud rate for different values starting from 9600 to 115200
3. On each baud rate change, restarted the device and started paulmon using ‘Enter’ key
4. Garbage data was observed for baud rate > 57600. For these values Bit error rate reached beyond acceptable value.

**Bit rate measured at 9600 bps**

A screenshot of a computer

Description automatically generated with medium confidence

**Part 2**

**Command line processing**

1. Command line processor log is attached below. It carries out following operations
2. Creating buffer0 and buffer1 at the start
3. Adding buffers with ‘+’
4. Removing buffer with ‘-‘
5. Query ‘?’
6. Hexdump ‘=’
7. Clearing buffers ‘@’
8. Stored data ‘a’ to ‘z’



1. All corner cases are also handled. Where,
2. Characters entered instead of numbers
3. Memory not allocated in multiples of 16 for buffer 0 & 1
4. Memory allocated for remaining buf isn’t within 30 to 300 bytes
5. Attempted to malloc more memory than heap allows



**Virtual Debug port**

Virtual debug port is developed and used to debug and verify that received characters are right.

Debug memory location is 0xFFFE where received characters are being written to.

Following is the screenshot of logic analyzer displaying echoed character.

A screenshot of a computer

Description automatically generated

**Part 3**

Command line processing using MSP 432

Implemented

1. Echo functionality
2. Interrupt driven Tx Rx
3. Circular buffer used
4. Generated PWM signal on pin P7.6 of which ON time is varied by integrating two switches. Left switch increments duty cycle by 10% and right switch reduces
5. Integrated temperature sensor
6. PWM duty cycle varied based on temperature rise or fall
7. Following commands were implemented (This includes supplemental 1 element)
8. P: Print Duty cycle
9. T: Read Temperature
10. D: Duty change
11. U: Changing Unit
12. S: Setting up PWM
13. B: Blinking spectrum PWM LEDs

Commands in the screenshot explains what waveform is for:

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated

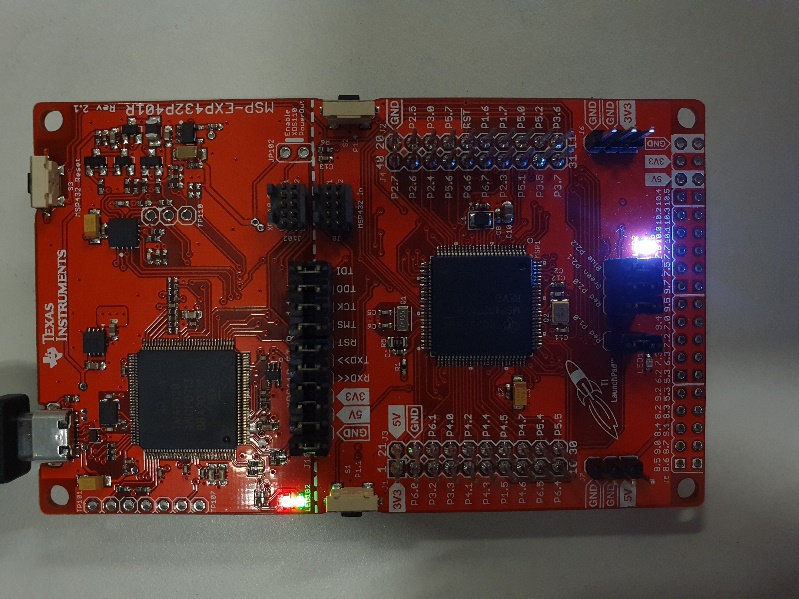
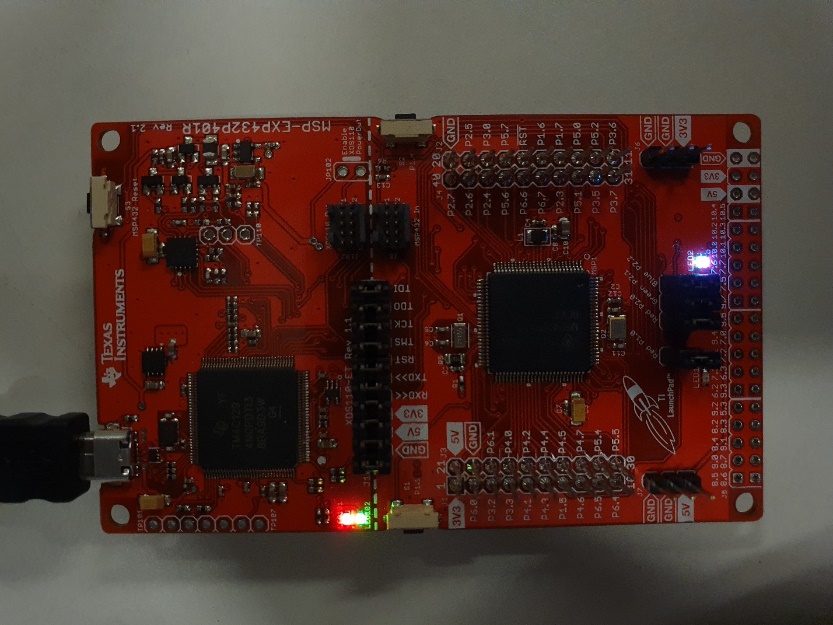
Temperature caused change in Duty cycle

A screenshot of a computer

Description automatically generated with medium confidence

RGB Spectrum

Left image shows white light with lower duty cycle than the right image



**Supplemental 2**

Graphical user interface, text

Description automatically generated

PWM on pin 1.4 (channel 1)

Graphical user interface, diagram

Description automatically generated

Max frequency CKRL

Graphical user interface, diagram

Description automatically generated

Min frequency (CKRL)

Graphical user interface

Description automatically generated

Idle mode – PSEN and ALE high

Graphical user interface

Description automatically generated

Power down mode – PSEN Low

Graphical user interface, diagram

Description automatically generated

**Challenge (15: Interrupt based UART)**

Implemented interrupt driven Tx and Rx on MSP432.

Implemented FIFO for asynchronous queuing and dequeuing.

In case of Polled implementation,

1. Microcontroller needs to be awake to keep polling through flags to detect whether characters have received or sent successfully.
2. Controller is busy looking for any sent or received characters which is wasteful.

In case of Interrupt based implementation,

1. Controller can go to low power modes and can wake up when UART interrupt is received
2. MCU processing power can be utilized in completing more useful work.

**[Part 3 Required Element1] As part of your submission, provide answers to the following questions:**

1. What operating system (including revision) did you use for your code development?
   1. Windows 10 OS Build 19042.1288
2. What compiler (including revision) did you use?
   1. Gcc cross compiler
3. What exactly (include name/revision if appropriate) did you use to build your code (what IDE, make/makefile, or command line)?
   1. Code composer studio 10.4.0
4. Did you install and use any other software tools to complete your lab assignment?
   1. Tera term
   2. Analog Discovery Digilent waveform
5. Did you experience any problems with any of the software tools? If so, describe the problems.
   1. No Problems