Gajanan#17

Installation directory of codecomposer : /home/ganraj/ti/ccs1250

0. Look at the section 5 : experimental evaluation in the 1st paper link :

https://dl.acm.org/doi/pdf/10.1145/3508352.3549451

1. LED blinking code of MSP430
2. To test timing of LeNet-5 in MSP430.

Prerequisites :

Proxy login : : https://proxy62.iitd.ac.in/cgi-bin/proxy.cgi

1. Understand about <https://www.youtube.com/watch?v=k5FgMRrRm-c>
2. https://www.youtube.com/watch?v=vn7TqU89e9o&list=PLRqKd7sGGKMTB5egcnNuR\_SJjMgzFntKj&index=2

For Lab PC :

Workspace : /home/ganraj/workspace\_v12

For my PC :

Workspace : /home/ganraj/ti/ccs1250

Problem occuring :

error initializing emulator msp-fet430uif is already in use . I am clearing the msp430 board . I am updating the system. Facing issues with proxy I guesss. Anyways doing just the thing to clear the msp430 board. Still the error persists : seeing manuel : <https://www.ti.com/lit/ug/slau678c/slau678c.pdf?ts=1701161921322&ref_url=https%253A%252F%252Fwww.google.com%252F>

Now I am installing the drivers for it. TI ACCOUNT : [boradeganraj@gmail.com](mailto:boradeganraj@gmail.com), Gajanan#17

Installation directory of drivers : /opt/ti/fetdrivers . This is for your pc too.

SO I DONOT KNOW WHAT I DID BUT IT GOT WORKING CORRECTLY : SO BASICALLY I USED THE RESOURCE : <https://www.youtube.com/watch?v=H1HToCzku9Y>

[](https://www.youtube.com/watch?v=H1HToCzku9Y)

I IMPORTED THE OUT OF BOX BLINK CODE . NOW LET ME CHECK THE OTHER THINGS TOO. Basically you just install test packages which is being given to you by TI. And then I found the code of LED blink and I found it and I worked it out.

---> last thing which I did today is about the suming of the elements of the array.

After coming to room and eating chicken lababdar with oil :

1. [phdschooltpc/msp430-xor: Machine learning example running on MSP430. (github.com)](https://github.com/phdschooltpc/msp430-xor)
2. [lucianobajr/internet-of-things: :robot: Internet Of Things (CCF 351) (github.com)](https://github.com/lucianobajr/internet-of-things)
3. [CS249r :: Tiny Machine Learning (Fall 2022, Grad. Seminar) - Copy Final Projects (google.com)](https://sites.google.com/g.harvard.edu/tinyml/finalprojects?authuser=0#h.42412o3r6elt)
4. [Machine Learning for Microcontroller-Class Hardware: A Review (arxiv.org)](https://arxiv.org/ftp/arxiv/papers/2205/2205.14550.pdf)

Day #2

1. Code for finding the time :
   1. #include <msp430.h>
   3. void main(void)
   4. {
   5. // Stop the Watchdog Timer
   6. WDTCTL = WDTPW + WDTHOLD;
   8. // Enable the cycle counter
   9. \_\_bis\_SR\_register(SCG0);
   11. // Your code or function to be timed goes here
   12. for (int i = 0; i < 1000; ++i) {
   13. // Do some computation
   14. }
   16. // Disable the cycle counter
   17. \_\_bis\_SR\_register(SCG1);
   19. // Read the cycle count
   20. unsigned int cycles = \_get\_CYCLES();
   22. // Print the result
   23. printf("Cycle count: %u\n", cycles);
   25. while (1);
   26. }

Now for my device MSP430fr5994 : THE CLOCK FREQUENCY IS 16-MHz system clock. But this code did not work facing compiling issues.

1. Bhaiya itna aasan bhi nahi hai . I have to learn about timers in msp430 and all. <https://www.youtube.com/playlist?list=PL643xA3Ie_EuHoNV7AgvJXq-z1hrE8vsm> Need to look at this video series.
2. Talked to nivedita mam, now able to print hello world.
3. #include <msp430.h>
4. #include <stdio.h>

7. int main(void)
8. {
9. WDTCTL = WDTPW | WDTHOLD; // Stop WDT
11. configureUART();
13. // Configure GPIO
14. P1OUT &= ~BIT0; // Clear P1.0 output latch for a defined power-on state
15. P1DIR |= BIT0; // Set P1.0 to output direction
17. PM5CTL0 &= ~LOCKLPM5; // Disable the GPIO power-on default high-impedance mode
18. // to activate previously configured port settings
20. // ------------------- STARING TIMER ----------------------//
21. // Set up Timer A
22. volatile unsigned long start\_time, end\_time;
23. TA0CTL = TASSEL\_2 + MC\_0 + TACLR; // Use the SMCLK, stop the timer, and clear it
24. TA0CCR0 = 0xFFFF; // Set the timer's upper limit
26. // ------------------- MEMORY ALLOCATION STARTED --------------------------//
27. volatile char count = 5, index, sum=0;
28. volatile char num[] = {1,2,3,4,5};
29. index = 0;
30. while(index < count){
32. sum += num[index];
33. index++;
34. }
36. end\_time = TA0R;
38. // Calculate and print the elapsed time
39. unsigned long elapsed\_time = end\_time - start\_time;
41. printf("Hello world!");

44. return 0;
45. /\*
46. while(1)
47. {
48. P1OUT ^= BIT0; // Toggle LED
49. //\_\_delay\_cycles(2);
50. }
51. \*/
52. } ---> working
53. We have to see time via register values only.
    1. // Setup Timer\_A
    2. TA0CTL = TASSEL\_2 + MC\_1 + ID\_3; // SMCLK, Up mode, divide by 8
    3. TA0CCR0 = 2000; // Set the timer period for 1 ms (16 MHz / 8 / 2000 = 1 ms)
54. **TASSEL\_2 (Clock Source):**
    1. TASSEL\_2 selects SMCLK (Sub-System Master Clock) as the clock source for Timer\_A. SMCLK is often derived from the main system clock.
55. **MC\_1 (Mode Control - Up mode):**
    1. MC\_1 sets the timer to Up mode, where the timer counts up to the value in TA0CCR0 and then resets. This is a basic mode for generating a periodic interrupt or measuring time intervals.

**30/11/23**

1. **WHOLE information about the clock : https://chat.openai.com/share/09ea156a-f421-4f47-a3ab-028f4e9fdbfd**
2. The volatile keyword tells the compiler that the variable's value may change at any time without any action being taken by the code the compiler finds nearby. This prevents the compiler from optimizing away reads or writes to that variable, ensuring that the variable is always read from or written to memory, and not cached in a register.
3. <https://www.youtube.com/watch?v=i66oTB8LPvw&list=PLRqKd7sGGKMTB5egcnNuR_SJjMgzFntKj&index=6> --- > learn about the serial communication between the device and the pc.
4. see basically I am using msp430fr5994 And my code is hangling on the declarations : volatile unsigned int matrix1[size][size]; volatile unsigned int matrix2[size][size]; volatile unsigned int result[size][size]; . i DONOT KNOW WHAT TO Do :
   1. The MSP430 series typically has limited RAM. If the matrices are too large, they may be causing a stack overflow. Consider allocating the matrices in a different memory segment, such as using malloc to allocate memory from the heap.
5. <https://github.com/danaolcott/MSP430> --> for using python/java with code composer
6. <https://github.com/ticepd/msp430-examples>
7. Python + codecomposer : <https://github.com/cetic/python-msp430-tools/blob/master/doc/README-msp430-bsl.txt>
8. I have this working code now :
9. #include <msp430.h>
10. #include <stdio.h>
11. #include <stdlib.h>
12. #include <string.h>
14. int main(void)
15. {
16. WDTCTL = WDTPW | WDTHOLD; // Stop WDT
18. // Configure GPIO
19. P1OUT &= ~BIT0; // Clear P1.0 output latch for a defined power-on state
20. P1DIR |= BIT0; // Set P1.0 to output direction
22. PM5CTL0 &= ~LOCKLPM5; // Disable the GPIO power-on default high-impedance mode
23. // to activate previously configured port settings
25. // Setup Timer\_A
26. TA0CTL = TASSEL\_2 + MC\_1; // SMCLK, Up mode (no input divider)
27. TA0CCR0 = 16000; // Set the timer period for 1 ms (16 MHz / 16000 = 1 ms)
29. // Start Timer\_A
30. TA0CTL |= TACLR; // Clear Timer\_A
31. TA0CTL |= MC\_1; // Start Timer\_A in Up mode

34. /\* -------------------- INTITAL TESTED CODE FOR ADDITION AND DIVISION -----------------------------
35. volatile int a = 3, b=5;
36. volatile int c = a/b;
37. -----------------------------------------------------------------------------------------------------\*/
39. int size = 2;
40. for (;size < 1000;size++){
42. int \*a = (int \*)malloc(size \* size \* sizeof(int));
43. int \*b = (int \*)malloc(size \* size \* sizeof(int));
44. int \*result = (int \*)malloc(size \* size \* sizeof(int));
45. int i = 0;
46. for (; i < size; i++){
47. int j = 0;
48. for (j = 0; j < size; j++){
49. \*(a + i\*size + j) = 1;
50. \*(b + i\*size + j) = 2;
51. \*(result + i\*size + j) = 0;
52. }
53. }
55. i=0;
56. for (;i<size;i++){
57. int j = 0;
58. for (;j<size;j++){
59. int k = 0;
61. \*(result + i \* size + j) = 0;
62. for (;k<size;k++){
64. \*(result + i \* size + j) += \*(a + i \* size + k) \* \*(b + k \* size + j);
65. volatile int res = \*(result + i\*size +j);
66. }
67. }
68. }
70. free(a);
71. free(b);
72. free(result);
74. }
76. // Stop Timer\_A
77. TA0CTL &= ~MC\_1; // Stop Timer\_A
79. // Calculate and print elapsed time
80. volatile unsigned int elapsed\_time = TA0R;
82. /\*
84. // Assuming a 16 MHz system clock and a divide-by-8 prescaler
85. double clockFrequency = 16000000.0; // 16 MHz
86. double prescaler = 8.0; // Divide by 8

89. // Calculate the time period of one Timer\_A count in seconds
90. double timerPeriod = 1.0 / (clockFrequency / prescaler);
92. // Convert Timer\_A counts to actual time in seconds
93. volatile double actualTimeInSeconds = elapsed\_time \* timerPeriod;
94. \*/

97. }

Now look at this source for energy plotting : <https://webmail.iitd.ac.in/src/webmail.php>

01/12/23 :

Flash memory

Flash memory is used to store the programs and constant variables

The size of Flash memory in MSP430F2003 is 1 KB

The size of Flash memory in MSP430F2003 is 2 KB

RAM

RAM is used to store the temporary data (Read/Write memory)

The size of RAM in MSP430F2003 and F2013 is 128 Bytes

Basic Clock system :

The clock module provides the CLK for CPU and peripherals. Three clock signals are

available from the basic clock module:

MCLK : Master clock is used by CPU and system

SMCLK : Subsystem Master clock is distributed to high speed peripherals

ACLK : Auxiliary clock is also distributed to low speed peripherals

IMp : <https://maheshelectronics.files.wordpress.com/2020/05/mpmc_material345-units.pdf>

<https://github.com/phdschooltpc/msp430-xor> --> for neural network to run onto msp board

<https://github.com/leleonardzhang/Capuchin> --> Important

<https://github.com/tejaskannan/budget-rnn> --> above and this paper is for running neural network into the board.

Updates : matrix multiplication is not working fine for 3\*3 matrix . I have used malloc but it is not happeniing.

<https://e2e.ti.com/support/microcontrollers/msp-low-power-microcontrollers-group/msp430/f/msp-low-power-microcontroller-forum/561317/msp430fr5994-writable-memory-above-64k>

2/12/23

<https://github.com/jamy54/Lenet_accelerator> --> **ABSOLUTELY IMP**

***NOT USED :* Memory mapping issue :** [**https://e2e.ti.com/support/microcontrollers/msp-low-power-microcontrollers-group/msp430/f/msp-low-power-microcontroller-forum/63459/msp430f5438-region-text-is-full**](https://e2e.ti.com/support/microcontrollers/msp-low-power-microcontrollers-group/msp430/f/msp-low-power-microcontroller-forum/63459/msp430f5438-region-text-is-full)

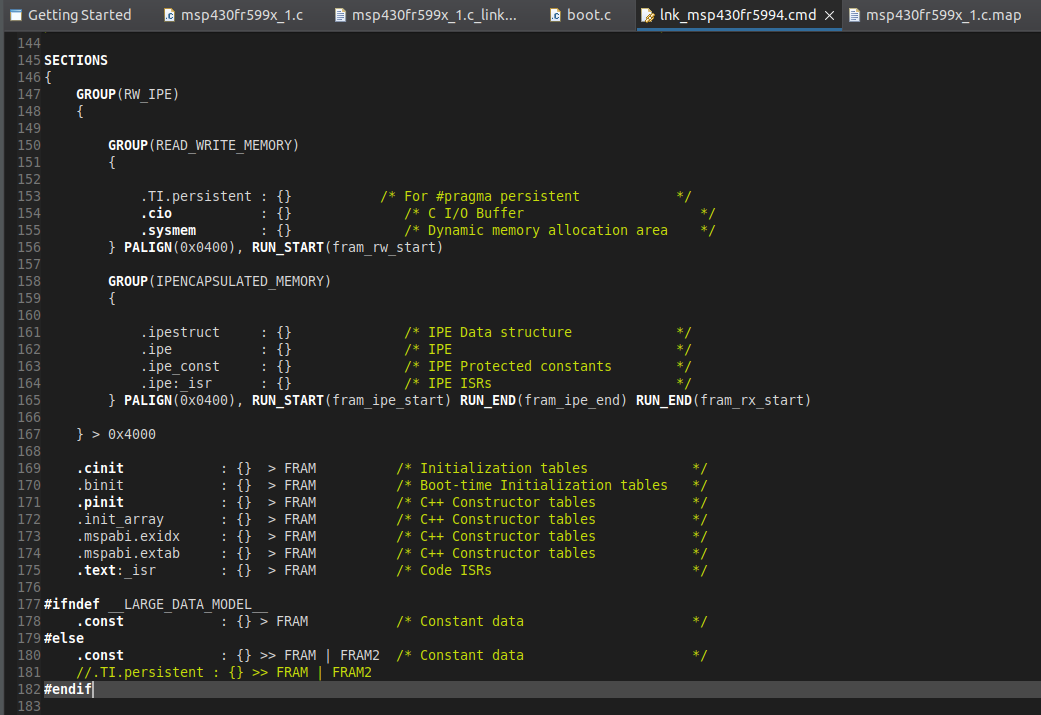
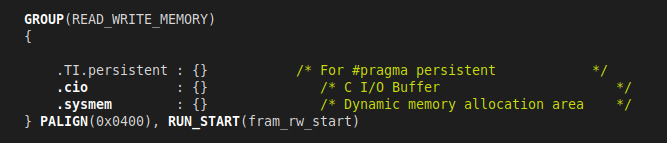
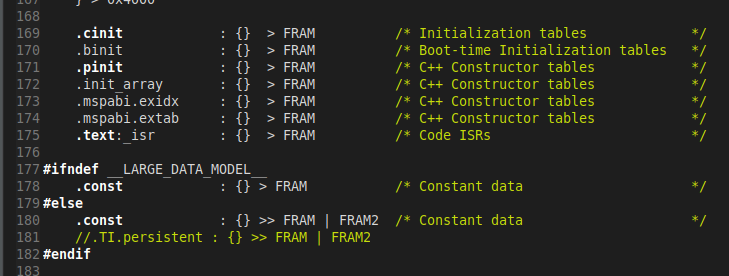
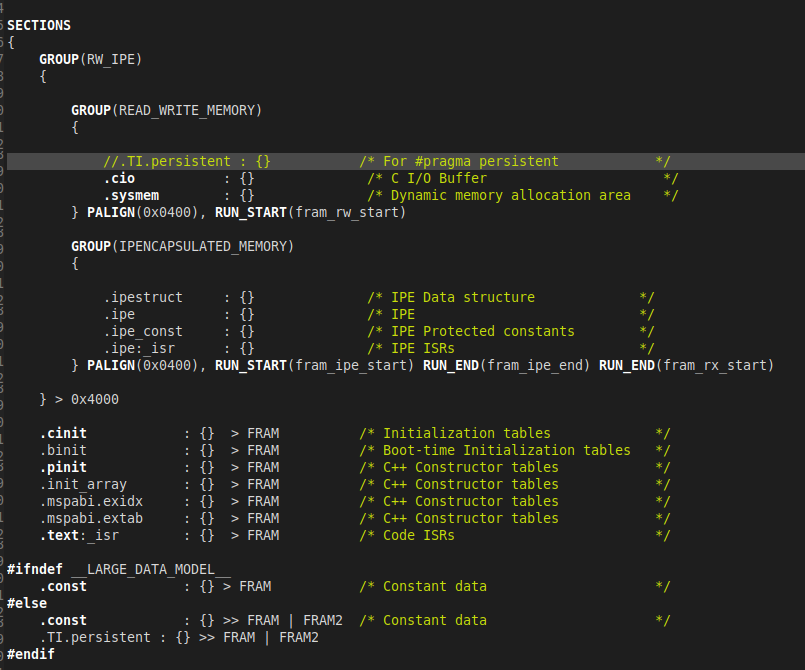
**3/12/2023 ---> holiday**

**04/12/23 :**

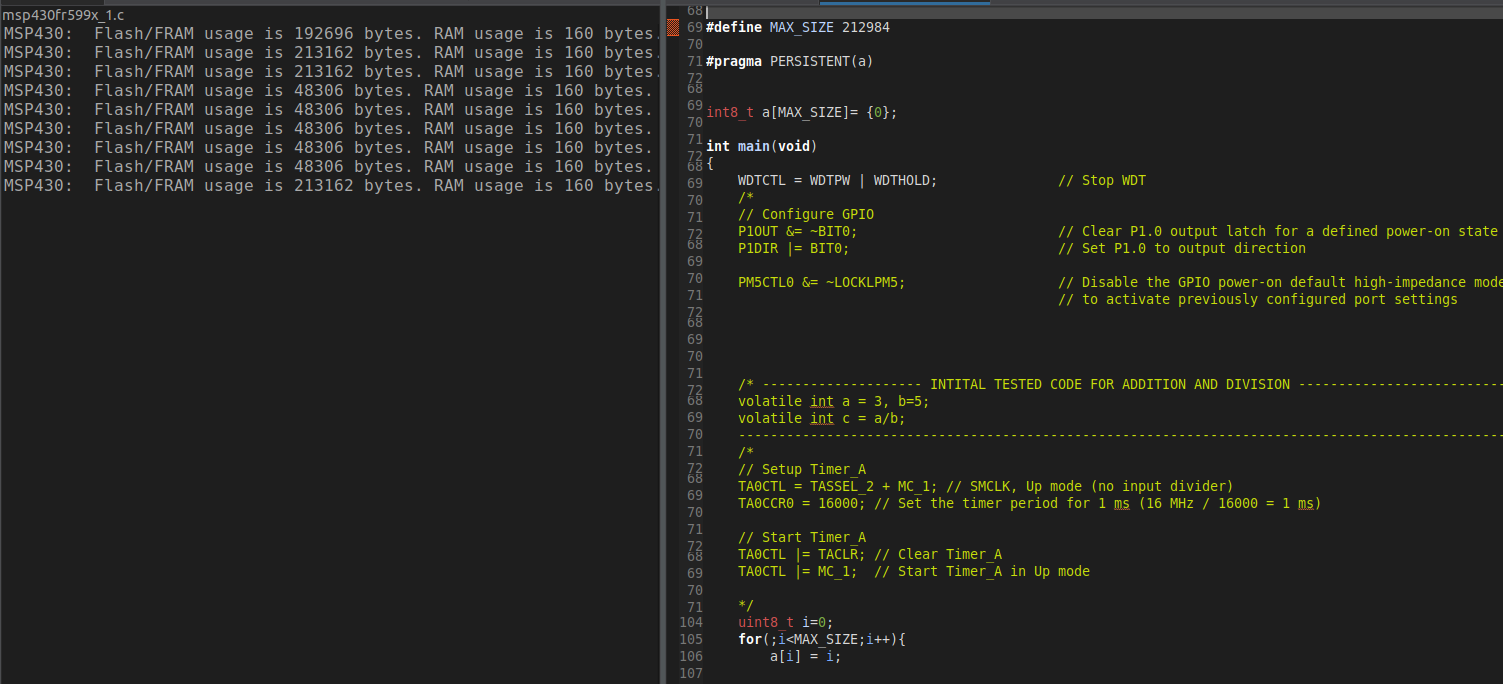
1. **Sir se mila --> board leliye**
2. [MSP430 FRAM Technology – How To and Best Practices (Rev. B)](https://www.ti.com/lit/an/slaa628b/slaa628b.pdf?ts=1701719984341&ref_url=https%253A%252F%252Fmobile-int.aws.ti.com%252F) --> look here for FRAM

05/12/23 : CGPA CAME AND MOVIE ANIMAL SEEN

06/12/23 :

1. Memory allocation limit exceed : how ? --> <https://e2e.ti.com/support/microcontrollers/msp-low-power-microcontrollers-group/msp430/f/msp-low-power-microcontroller-forum/813093/ccs-msp430fr5994-how-to-solve-the-problem-array-is-too-large>
2. In Code Composer Studio (CCS), a .cmd file is a linker command file. It is used by the linker during the build process to specify the memory layout of your program. The linker command file contains directives and commands that instruct the linker on how to organize and allocate the various sections of your program in the target memory.
3. 
4. Here to use only FRAM data -->
   1. 
   2. BUT IF WE NEED TO ALLOCATE THE DATA TO FRAM2 : THEN COMMENT THIS LINE OF .TI.persistent : {} and then add : at 181 line : 
   3. Here the line is commented but you have to uncomment it --> for FRAM2 utilisation. Notice by this line, your data will now go to FRAM2 only and not in FRAM.
   4. 

so this is the final configuration.

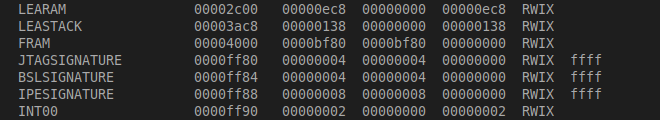


And in memory map the details are :

Address start length used ununsed

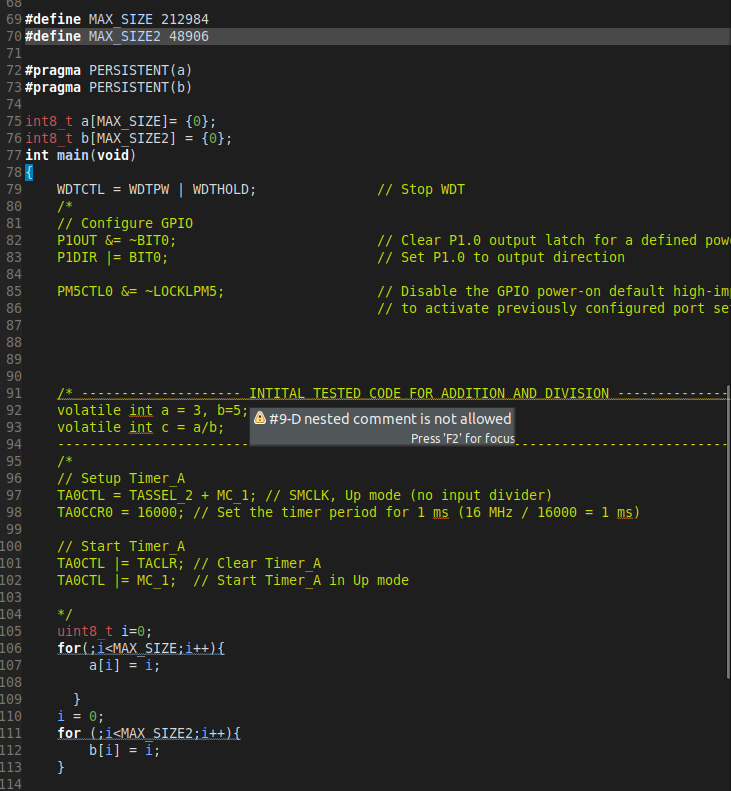
So in fram2, I am using 212984 bytes of data as I have used uint8\_t . You can see in above photo. And In FRAM, I am using 48128 bytes of data.

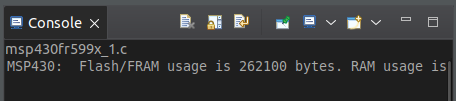
Summarising : adding .TI.persistent : {} >> FRAM | FRAM2 will first try to allocate memory in fram and if not then it will try to allocate memory in fram2.



Wow, total utilization of fram as you can see above. With total size of array as 48996. Here but the problem is if I increase the size of 48997, then the allocation will happen in fram2. And the allocation in fram will not happen here in this case.

I CAN 2 ARRAYS TO use total space :



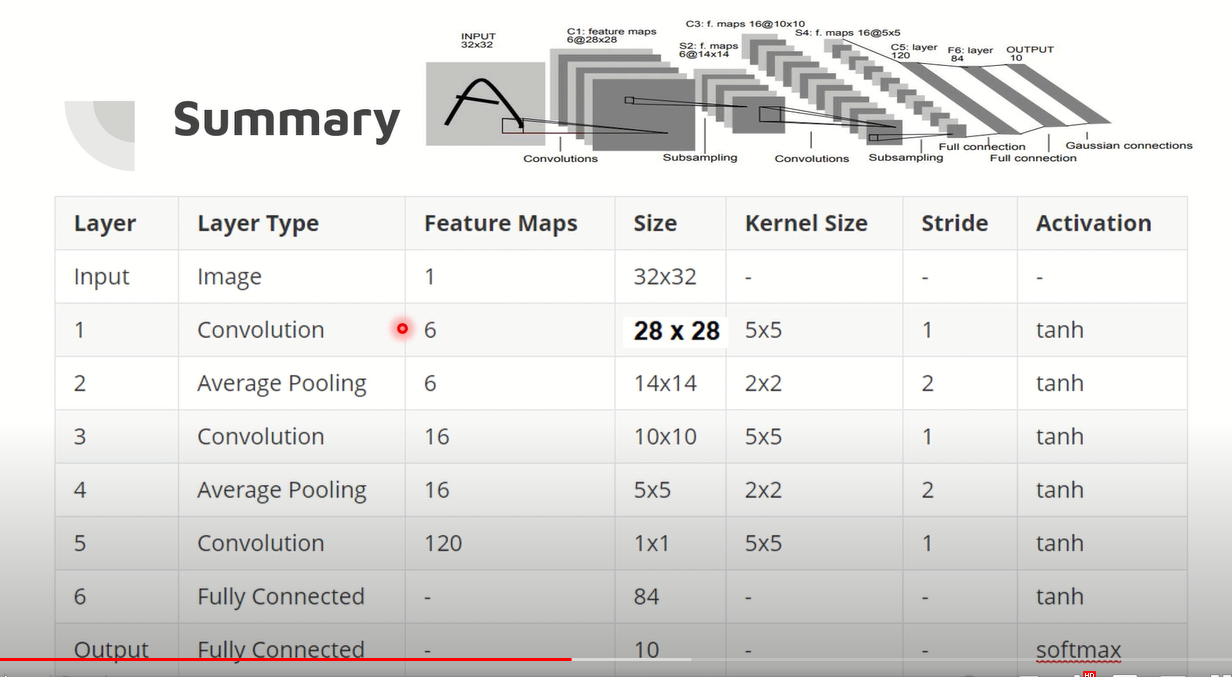
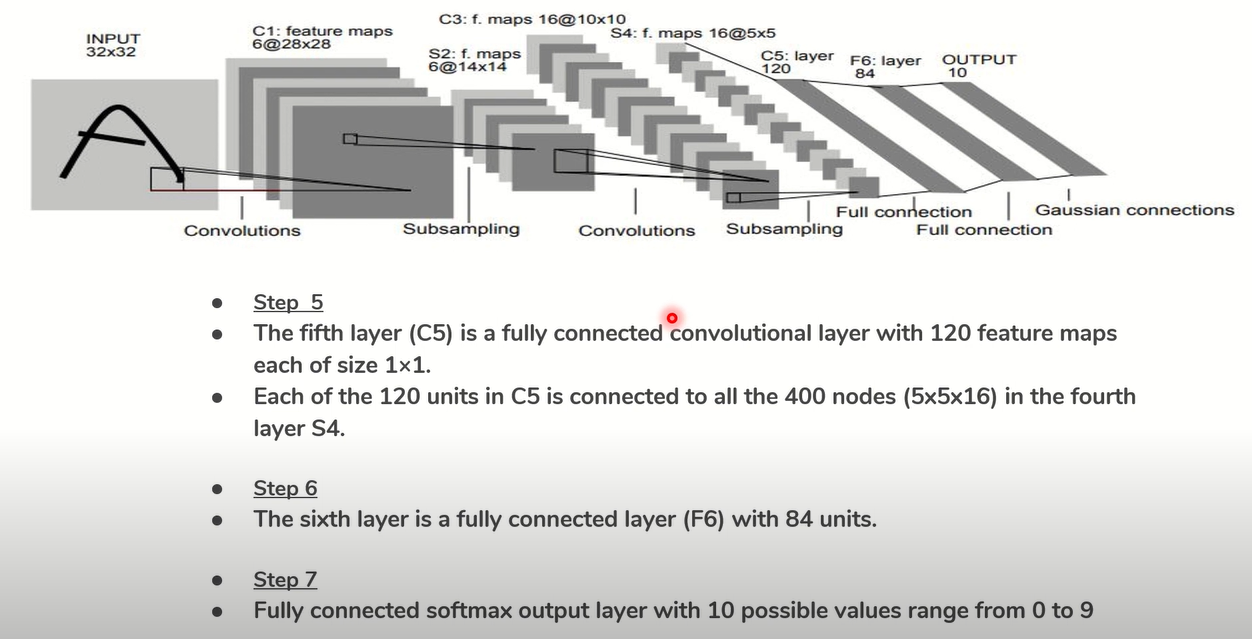
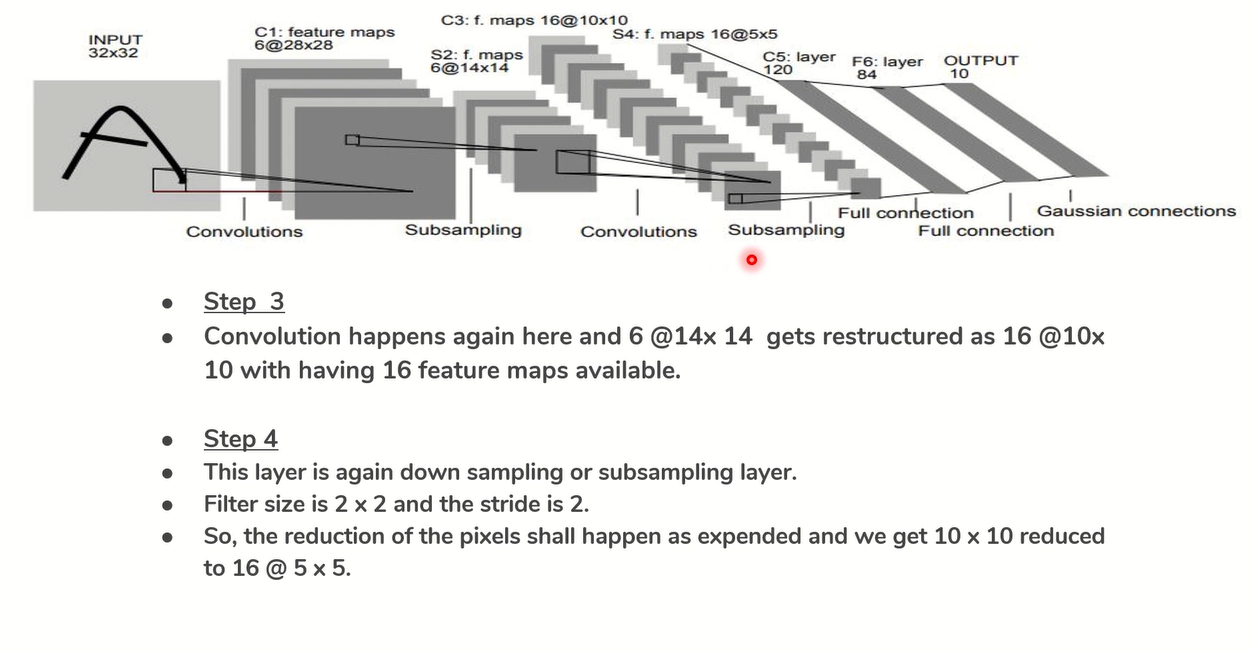
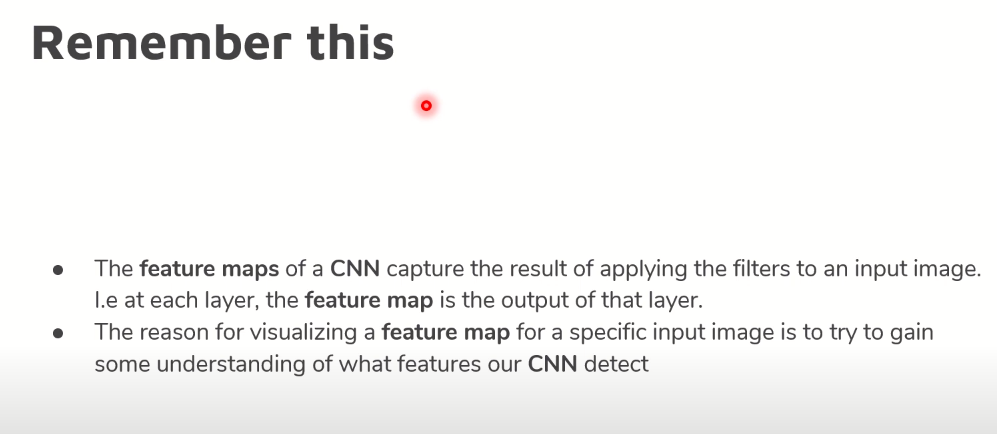
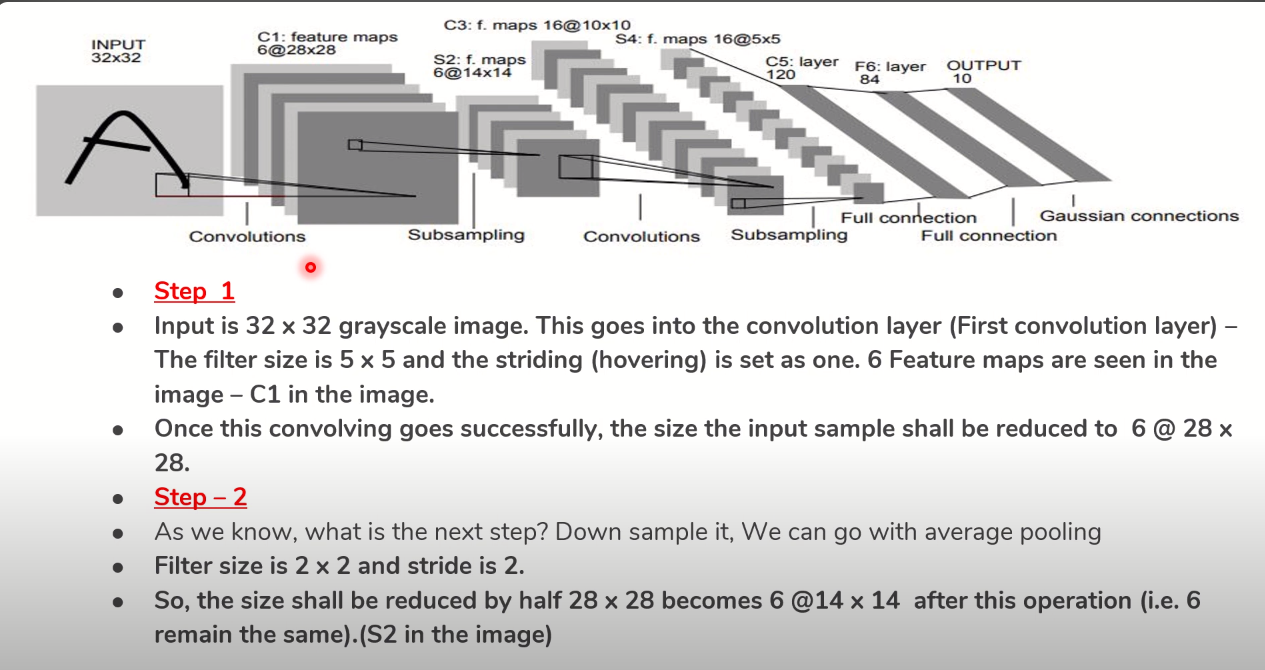


See so maximum size of one array is 212984 bytes which can be allocated in fram2 and in other fram, I can accommodate 48906 bytes.

Accha here the thing is I am not using RAM or LERAM. Ok now I want to run TinyML here. Time taken badme dekhta hu. ---> sarangi sir bole.

7/12 - 13/12/23 :

* + 1. [Neural networks - YouTube](https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi)
    2. [Convolutional Neural Networks: CNN Basics - YouTube](https://www.youtube.com/playlist?list=PLxlkzujLkmQ8Ov11Xm-K913C7uNSfr5Rp) -- > till LeNet
    3. Above are done watching. That is learned basics of Neural network and CNN.
    4. [8. LeNet CNN - Complete Description and Implementation with Code. (youtube.com)](https://www.youtube.com/watch?v=PiF0l6xif-k) ---> watching now



Code implementation of LeNet :

[LeNet with TensorFlow. LeNet is considered to be the ancestor… | by mrgrhn | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/lenet-with-tensorflow-a35da0d503df)