

Software Delays and Quantization

1. Software delay for 1 second:

Use the delay subroutine from question 2 in problem set 1 develop a 1 second timer delay. The program should set the pins P1.4 to P1.7 of Port-1 for about 1 second and then clear these off.

Write the main routine which should:

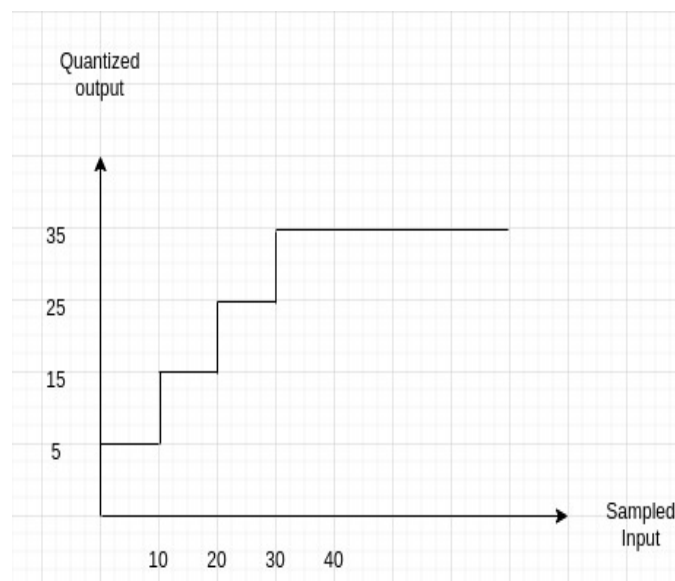
- Set the pins P1.4 to P1.7 of Port-1.
- Enter a loop which calls the delay routine in each iteration
- When the loop terminates, Clear the pins P1.4 to P1.7 of Port-1

Adjust the number of iterations in the main routine loop and the delay count to produce a total delay of nearly 1 second.

2. Quantization:

In order to convert an analog signal to a digital signal, the 3 steps to be followed are sampling, quantization and encoding. Assume that a sampled signal (consisting of 8 samples in the range 0 to 40) is present. Write an assembly code to quantize the signal into 4 discrete levels. Follow the below scheme.

```
if (sample >= 0 and sample < 10):
    output 5
else if (sample >= 10 and sample < 20):
    output 15
else if (sample >= 20 and sample < 30):
    output 25
else:
    output 35
```



(Note that these values are given in decimal. You need to use equivalent hexadecimal values in your code)

```

// -- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL QUANT
HERE: SJMP HERE
ORG 130H
// *****

QUANT:
// ADD YOUR CODE HERE
RET
END

```

- The inputs samples should be present in locations 60H to 67H.
- The quantized output samples should be present in memory locations 70H to 77H.
- The command to enter samples to data memory is as follows.

E char I:60h = 14h,69h,26h,5bh,7fh,1ah,00h,05h

Check the following cases:

- Input samples : 11H, 2AH, 01H, 3FH, 18H, 1CH, 0EH, 06H.
Output samples : 0FH, 23H, 05H, 23H, 19H, 19H, 0FH, 05H.
- Input samples : 00H, 09H, 0AH, 13H, 14H, 1DH, 1EH, FFH.
Output samples : 05H, 05H, 0FH, 0FH, 19H, 19H, 23H, 23H.