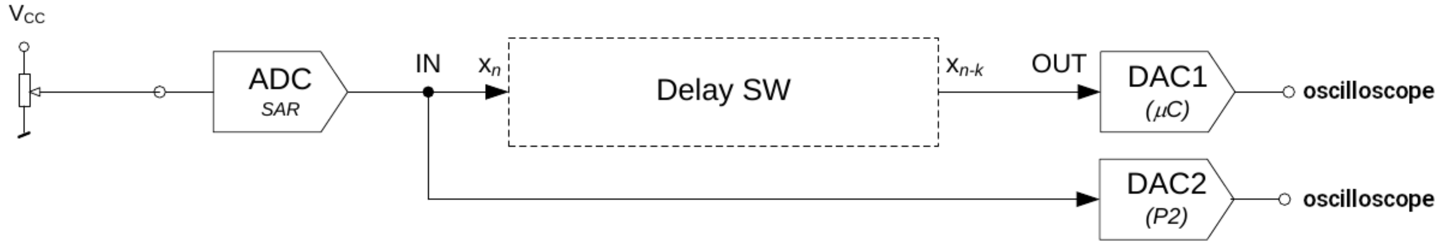
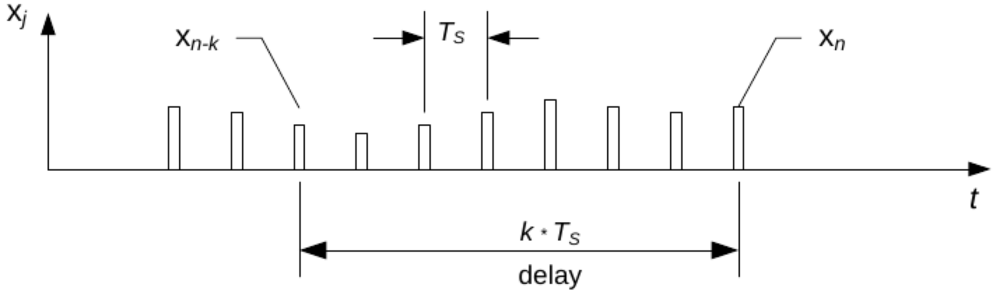
Task

Delaying Analog Signals Using Tables

Below is a basic block diagram of the system and the principle of delaying a quantized analog signal.



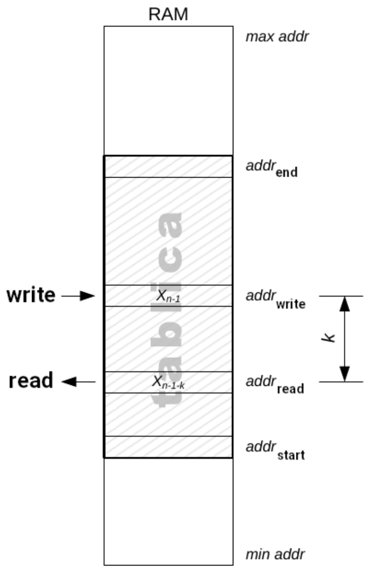
Block Diagram of the Delay System



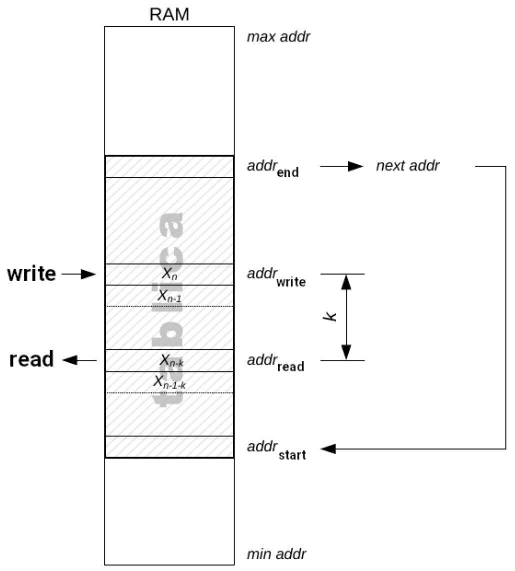
Samples $\mathbf{X_j}$ of the analog signal $\mathbf{X}$

The implementation of a FILO (First-In-Last-Out) register, which stores incoming samples and ensures the desired delay $k * T_s$ (where T_s is the sampling period), can be done in various ways. Two approaches are outlined below:

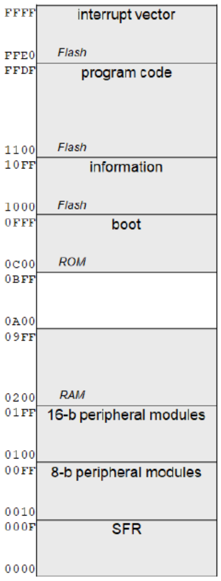
- consistently shifting the contents of memory cells forming the FILO register – not recommended
- placing the currently sampled data in a table and addressing the appropriate reading of the samples that were previously stored in the table. The principle of this method is shown below – the address for writing the currently sampled value and reading the delayed (previously entered) sample is incremented in each cycle. This ensures constant k, and therefore a constant delay time.



n-1 moment (previous)



n moment (current)



MSP430F169 memory map

The table is placed in RAM and indirect addressing should be used for addressing it. When placing the table in the address range of RAM, **ensure there is no conflict with the area occupied by the *stack***.

Write a program to delay the analog signal:

- delay in the range of 0.5–1.0 seconds
- the program should be implemented exclusively using interrupts from Timer_A (not using the resources of the main program)
- suggested table size - 256 samples