* **Datasheet TM4C123GH6PM: Pg. 796**
  + **RCGC0:** pg. 455
    - Bits 8-9: Maximum sampling rate
  + **AMSEL**: pg. 684
    - Bits 0-3: Activates analog function for the pin
  + **SSPRI**: pg. 838
    - Sets the priority of the ADC sequencer
  + **EMUX**: pg. 830
    - Bits 12-15: Specify how the ADC will be triggered
  + **ACTSS**: pg 818
    - Enables/Disables sequencer
  + **SSMUX3**: pg. 872
    - Configures what channel to sample
  + **SSCTL3**: pg. 873
    - Specifies the mode of the ADC sample
* [**Successive approximation:**](https://www.youtube.com/watch?v=mAtri-cYSvk)
  + The **speed** of successive approximation related linearly with its precision in bits
* **Data Acquisition System:** sensor data converted to voltage, sent to the ADC and converted into a digital equivalent which is either sent for processing, logged, or used for a function.
  + **TM4C ADC**
    - **Grader uses ADC1 and PD3** for volt meter and oscilloscope
    - **Labs use ADC0, sequencer 3, and PE2** for labs 14 and 15
  + **ADC Parameters:**
    - **Precision**: 12bits = 4096 alternatives (points of measurements in a plot).
    - **Range**: The min to maximum voltage (0v to 3v3).
    - **Resolution**: Smallest change in voltage that can be reliably detected.
      * Resolution = Range / Precision
      * Resolution = 3v3 – 0v / 4096
      * Resolution = 0.7mV
  + **Limitations in DAQ sampling:**
    - Amplitude
    - Amplitude range
    - Time quantization
    - Time interval
  + **Figure of merit:**
    - Precision (number of bits)
    - Speed (how fast we can sample)
    - Power (how mucy energy does it take to operate)
  + Cost of the ADC is a function of the number and quality of the internal components

**ADC Initialization Ritual:** Example for setting up PortE as an analog input

* + Steps 1 - 5: GPIO PortE port configuration
    - Clock
    - DIR
    - ALT
    - DEN
    - AMSELECT
  + Step 6: ADC Clock
    - RCGC0 (bit 16)
  + Step 7: ADC sample speed
    - RCGC0 (bit 8-9)
  + Step 8: ADC Sequencer Priority
    - SSPRI (bit 12-13)
  + Step 9: Disable ADC during configuration
    - ACTSS (bit 3)
  + Step 10: Select ADC trigger type
    - EMUX (bit 12-15)
  + Step 11: Select ADC channel (0-11)
    - SSMUX (bit 0 – 3)
  + Step 12: Configure the sample control bits
    - SSCTL (bit 0 – 3)
  + Step 13: Enable ADC after configuration
    - ACTSS (bit 3)

NOTES:

**SSMUX3** is a 4 bits wide register (the other 28 bits are reserved). The value set on those 4 bits indicates which analog input is sampled for ADC: 0x01 for AIN1, 0x02 for AIN2 etc.

If you want to configure AIN1 as ADC input, you must set those 4 bits to **0001**2.

With **ADC0\_SSMUX3\_R |= 0x00000001**, and looking to the least significant nibble:

1. Initial value of ADC0\_SSMUX3\_R is 0xHHHHHHHH or bb......bbbbbbbb2, where **b** means 0 or 1 and **H** is any value from 0 to F
2. ADC0\_SSMUX3\_R |= 0x000000001 → (binary) bb......bbbbbbbb2  | 00....000000012 → **bb.....bbbbbbb1**2
3. Final value in binary of **ADC0\_SSMUX3\_R**  will be **bb.....bbbbbbb1**2 and only if b3==0 and b2==0 and b1==0 you can get AIN1. So this is the wrong way to configure SSMUX3.

Now let's see what we get with **ADC0\_SSMUX3\_R = (ADC0\_SSMUX3\_R&0xFFFFFFF0)+1**

1. Initial value of ADC0\_SSMUX3\_R is 0xHHHHHHHH or bb......bbbbbbbb2, where **b** means 0 or 1 and **H** is any value from 0 to F
2. (ADC0\_SSMUX3\_R&0xFFFFFFF0) → 0xHHHHHHH0 (clears the least significant nibble)
3. (ADC0\_SSMUX3\_R&0xFFFFFFF0)+1 → 0xHHHHHHH0+1 → 0xHHHHHHH1 or in binary **bb....bbbb0001**2
4. Final value in binary of **ADC0\_SSMUX3\_R** is **bb....bbbb0001**2 wich means AIN1.

**ADC Sample Capture Procedure:**

