```
------ SYSTICK TIMER ------
```

- * The basis of SysTick -->
 - 1) A 24-bit down counter that runs at the bus clock frequency
 - 2) Max.RELOAD VALUE = 0x00FFFFFF OR 2^24 1
- * SysTick --> simple counter used to
 - 1) Create time delays
 - 2) Generate periodic interrupts.
- * Intialization
 - 1) Clear Enable to stop counter during initialization
 - 2) Specify the RELOAD value
 - 3) Clear the counter via NVIC ST CURRENT R
 - 4) Set NVIC ST CTRL R
 - --> CLK_SRC (Bit 2) = 1 (If bus clock is the only option)
 - --> INTEN = 0 (No Interrupts)
 - --> Enable (Bit 0) = 1 (To enable counter)
- * COUNT BIT
- --> BIT 16 IN NVIC_ST_CTRL_R
- --> Acts like a FLAG When the CURRENT value counts down from 1 to 0

```
* * UART Baud Rate Generation

• BRD = IBRD + FBRD

• BRD = IVARTSysClb / (ClbDiv * Baud Rate)
```

* UART Setup

- 1. Enable clock using RCGCUART
- Enable GPIO Clock (SYSCTL_RCGCGPIO_R)
- 3. Disable UART by clearing UARTEN in UARTX_CTL_
- 4. Write the values for baud rate in UARTBRD and UARTFBR
- bits, and parity enable and type) to UARTLCRH

Select Word Length 5 bits --> 00

8 bits --> 11

- 7. Set GPIO AFSEL, PCTL, and DEN

* IMPORTANT REGISTERS

2)	UARTx_CTL_R	enable	TX(bit	8)	RX(bit	9)	Clock(bit	0)

3) UARTx_RSR_R --> Contain the error bits that occurs when receiving

4) UARTx_FR_R --> CONTAINS THE STATUS OF FIFO FOR TX AND RX

TXFE RXFF TXFF RXFE

* * IMPORTANT NOTES:

- 1) UARTx_FR_R --> 0x90 TXFE (BIT 7) & RXFE (BIT 4) IS ENABLED
 2) UARTx CTL R --> 0x0300 TXE (BIT 8) & RXE (BIT 9) IS ENABLED
- While transmitting and receiving using FIFO enabled , the Data Register holds:
- --> 12 bits in Receiving (including ERROR)

*
* Usage >>>
1) They are used in printers to move paper and print heads
2) Stepper motors are used in applications where precise positioning is more important than high RPM, high torque, or high efficiency
3) When interfacing with any microcontroller,it is represented by 4 bits AS OUTPUT (DIR = 1)
* Operation of Bipolar Stepper Motor >>>
• To move a bipolar stepper, we reverse the direction of current through one (not both) of the coils
• To move it again, we reverse the direction of current in the other coil.
• Let the direction of the current be signified by up and down.
• To make the current go up, the microcontroller outputs a binary 01 to the interface.
• To make the current go down, it outputs a binary 10.
• Since there are 2 coils, four outputs will be required (e.g., 0101 means up/up).
• To spin the motor, we output the sequence 0101, 0110, 1010, 1001… over and over. Each output causes the motor to rotate a fixed angle.
• To rotate the other direction, we reverse the sequence (0101, 1001, 1010, 0110).
• the amount of rotations caused by each current reversal is a FIXED ANGLE depending on number of teeth of the magnet
* Calculations >>>
• Step Angle = 360 / (2 * no. Of Teeth)> for illustrations
• Number of Steps per rotation = 360 / Step angle
• SPEED of Stepper Motor = 1000 * 60 * (1 / NumberOfStepsPerRotation) * (1 / delayBetweenSteps)
Notes : delay btn. steps in millis Unit of Speed is RPM (Rotations per minute)

```
*----- PORTF -----
*1) Two build in Switches(sw)
```

--> PF0 & PF4

--> INPUT (DIR = 0 ? INPUT : OUTPUT;) $GPIO_PORTF_DIR_R = 0x0E$

by setting GPIO_PORTF_PUR_R = 0x 11

*2) Three LEDS

TO MAKE OUTPUT : (DIR = 1 ? OUTPUT: INPUT ;) $GPIO_PORTF_DIR_R = 0x 0E$

*3) REGISTERS OF PORT

GPIO_PORTF_DIR_R --> Direction register to specify which pins are input(0) / output(1)

GPIO_PORTF_AFSEL_R --> To activate Alternative Function SELect Used with PCTL --> (PORT CONTROL) Each pin has 4 bits in PCTL register

GPIO_PORTF_DEN_R --> To use pins as Digital input or output (Digital ENable)

GPIO_PORTF_AMSEL_R --> To use pin as Analog input (Analog Mode SELect)

SYSCTL_RCGCGPIO_R --> To enable port clock It's a register of 8 bits and each bit represents a port

will wait for its status bit in the PRGPIO to be TRUE GPIO_PORTF_CR_R --> (GPIO Commit) SET which bits are commited to changes

SYSCTL_PRGPIO_R --> Because it takes time for the clock to stabilize, we

TO NMI AND JTAG/SWD DEBUG HARDWARE

*4) GPIO ACTIVATION

TO ACTIVATE ANY GPIO PORT FOR DIGITAL I/O WE NEED TO DO 7 STEPS:

- Activate the clock RCGCGPIO and wait for its status bit in PRGPIO
- Disable the analog function AMSEL_R
- 4. Disable alternate function AFSEL_R 5. Enable digital port DEN_R
- 6. Clear PCTL_R to select digital function (4 bits/pin)
- 7. Set direction register DIR_R (0 for In, 1 for out)