Var Sizes; char = 1; short = 2; int = 4; long = 4; float = 4; pointer = 4; long long = 8; double = 8
Units: 10^{-3} = m (mili); 10^{-6} = μ (micor); 10^{-9} = n (nano) || 10^{3} = K (kilo); 10^{6} = M (Mega)
Calculate Baud Rate: Fbrd = 16,000,000 / (16 * BaudRate) //Node, 16 or 8 depending on UART_CTL High speed en
REGISTER DESCRIPTIONS!!!!!!
Reg 1; GPIO_PORTX_DATA_R (data) – offset 0x000, bit 31:8 reserved, bit 7:0 DATA RW
Reg 2; GPIO_PORTX_DIR_R (direction) – offset 0x400, bit 31:8 reserved, bit 7:0 RW, 0 = in, 1 = out
Reg 10; GPIO_PORTX_AFSEL_R (alternative function select) – offset 0x420, bit 7:0 RW, 0 = GPIO, 1 = alternative hardware control
Reg 18; GPIO_PORTX_DEN_R (digital enable) – offset 0x51C, bit 31:8 reserved, bit 7:0 RW, 0 = disabled, 1 = enabled
Reg 21; GPIO_PORTX_AMSEL_R (analog mode select) – offset?, bit 7:0, 0 = disabled, 1 = enabled
Reg 22; GPIO_PORTX_PTCL_R (port control, 8 pins) – 31:28 RW pin 7, 27:24 pin 6, 23:20 pin 5, 19:16 pin 4, 15:12 pin 3, 11:8 pin 2, 7:4 pin 1, 3:0 pin 0
Gen Purpose in/out run mode clock control
SYSCTL_RCGCGPIO_R = 0b0000_0000 – bit 5-0 = port F-A respectively
GPIO order: SYSCTL_RCGCGPIO; GPIO_PORTX_AFSEL; GPIO_PORTX_PCTL; GPIO_PORTX_DEN; GPIO_PORTX_DIR

UART REGISTERS: (in enable order but first disable then enable the uart)
SYSCTL_RCGCUART_R – UART 5-0 = bit 5-0
Reg 8; UARTX_CTL_R - bit 15 enable clear to send, 14 en request to send, 11 when bit 14 clear this is on U1RTS; 9 Receive En; 8 TX en 5 High speed en (0 = 16; 1 = 8) (used to divide baud Rate); 4 TXRIS in UARTRIS = 1 then TXRIS sent after stop bits; 0 UART en
UARTX_IBRD_R = Integer Baud Rate
Reg 5; UARTX_IBRD_R = ibrd; //sets int portion of IBRD
Reg 6; UART_FBRD_R = (int)(fbrd * 64 + 0.5); //This sets float part, have to convert first with the * 64 + 0.5
Reg 7; UART_LCRH_R – 7 stick parity select (usually 0); 6:5 length 5-8 bits; 4 FIFO En; 3 Two Stop bit en; 2 Even Parity en; 1 Parity en
UARTX_CC_R = 0x0 //use system clock
UART interrup ORDER: UART_CTL_R (disable); UART_ICR_R (clear intr);
UART_IM_R (mask); BIND UART INTR; UART_CTL_R(re-enable)

ADC Registers: 12-bit precision; 8 Digital comparators
ADCEN – (adc enable) bit 7:0 0 = not ADC, 1 = ADC
Reg 1: ADC_ACTSS_R – bit 16 ADC Busy; 3:0 Sample Sequence 3-0 en
Reg 2: ADC_RIS_R – bit 16 Intr status; 3:0 SS 3-0 Raw Intr Status
Reg 3: ADC_IM_R – bit 19:16 Digital Comp Itr SS 3-0; 3:0 SS 3-0 intr mask
Reg 4: ADC_ISC_R (intr status & clear) – 19:16 DCIS SS 3-0; 3:0 Status & clr
Reg 6: _EMUX_R (event mux sel) – Event Mux 3-0 takes ½ byte each
Reg 15: ADCSSMUX0-2 (SS input mux) – 31:28 8^{th} sample; 27:24 7^{th} sample; …; 3:0 1^{st}
Reg 29-30: ADCSSCTL1-2 (SS control) – 15 TS3; 14 IE3; 13 END3; D3; ….; D0
ADCSSFIFOn – SS Result FIFO – bit 11:0 data for SSn
ADCSAC – Sample Average Control 2:0 AVG (0x0 = no oversample, 0x1 = 2x, 0x2 = 4x…)
TS = temp sensor; 3 = 4^{th} sample; IE = interrupt enable; END = End of Sequence; D = differential input;
ADC_CC_R = 0x0 //clock config
SYSCTL_RCGCADC_R //Gate Control
PWM = Pulse Width / Period
ORDER: SYSCTL_GPIO; SYSCTL_ADC; GPIO_AFSEL; GPIO_DIR; GPIO_DEN: GPIO_AMSEL; GPIO_ADCCTL; ADCx_ACTSS; ADCx_EMUX; ADCx_SSMUXx; ADCx_SSCTLx; ADCx_SAC; ADCx_IM; ADCx_ISCL

TIMER REGISTERS
GPTMCTL (TIMERx_CTL_R) – 14 invert Output; 13 ADC Trigger; 11:10 0=pos edge, 1=neg edge, 2=resv, 3 = both edge; 9 Stall En; 8 TimeB en Repeats pattern on bit 6;; GPTMCFG (TIMERx_CFG_R) – 2:0 0= concat mode (16/32 use 32, 32/64 use 64) 1 = RTC concat; 4 = split into two tim
GPTMTnMR (TIMERx_TnMR_R – 1:0 0=reserve, 1=One-Shot Mode, 2=Periodic Mode, 3=Capture Mode
GPTMTnILR (TIMERx_TnILR_R) 31:0 Holds upper bound for counting up, initial value for counting down
GPTMIMR (TIMERx_IMR_R) – 16 Write update error mask; 11 TBMIM (match intr); 10 CBEIM (Capture Mode Event Intr Msk); 9 CBMIM (Cap Mode Match Inr mask); 8 TBTOIM (Time-out intr msk);; 4 bit pattern repeats for Timer A on bit 4:0
GTPMMIS (TIMERx_MIS_R) same register as GPTMIMR but holds status of unmasked interrupts
GTPAMICR (TIMERx_ICR_R) same register as GPTMIMR but write 1 to bit to clear desired interrupt
GPTMTnMATCHR (TIMERx_TnMATCHR_R) – 31:0 RW this value is compared to GPTBR to determine match events
TIMERx_TnPR_R 15:0 RW Prescale amount for timer (either 8 or 16 bit depending on timer 16 or 32 bit)
ORDER: SYCTL_TIMER, SYSCTL_GPIO; GPIO_DEN; GPIO_AFSEL; GPIO_PCTL; GPIO_DIR; TIMER-CTL; TIMER_ CFG; TIMER_TnMR; TIMER_TnILR; TIMER_PS (if needed); TIMER_IMR; TIMER_CTL (re-enable)