**MC= TM4C1294NCPDT**

Manufacturer: Texas Instruments

Sub-Architecture: Cortex-M4F

**Family Name:** Tiva C

**No. of Bits:** 32bit

* 120MHz 32-bit ARM Cortex-M4 CPU with floating point
* 1MB Flash
* 256KB SRAM
* 6KB EEPROM

SRAM offset – 0x2000.0000 with 256mb

**programing style, blinking led program**

* stdint.h:- c99 standard for interger variables.
* stdbool.h:- c99 for bool definetions.
* hw\_memap.h:- macro defines the memory map of tiva series. this defines the pheriperial base adddress and the location of the gpio port. GPIO\_PORTN\_BASE.
* hw\_types.h:- define common types and macros.
* sysctl.h:- defines macro for system control api of driver lib. it include the functions SysCtlClockSet() and SysCtlClockGet().
* gpio.h:- defines and macros for the gpio api of driver lib. it includes the api function such as   
   GPIOPinTypeGPIOOutput and GPIOPinWrite.
* uint8\_t a=1; a is the 8bit integer variable.
* code outline: int main(void).

**Clock setup:-**

* **PLL:** pahse locked loop is a control system that gentrates a output signal whose phase is related to the phse of the input signal. it is used to achieve comp sysnc.
* Configure the system clock run with 25mhz on the main oscr, driving the pll at 480mhz.
* The pll is conneted to the 10bit divider. the division is calculated by SysCtlClockFreqSet().
* The main clock 25Mhz drives the ehternet and the main processor.

Gpio setup.

Registers

* This particular processor has 22 registers the general purpose registers are from R1 to R12

**http://www.aravindev.com/tutorial-1--blink.html**  **-- reference**

* clock for the controller should be set by calling the SysCtlClockSet(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHz|SYSCTL\_OSC\_MAIN);

We need to generate a system clock of 40 MHz. We know that the input clock frequency after system prescale is 200 MHz. Now we need to prescale this by 5 which is done my the macro : SYSCTL\_SYSDIV\_5. There are many more prescales available which you are free to use. But make sure that the generated system clock frequency is less than 80 MHz.  
SYSCTL\_USE\_PLL - this macro is used to specify that we are using the PLL to generate the 400 MHz clock frequency.  
SYSCTL\_XTAL\_16MHz - This says to the uC that we give the input to the PLL from an external crystal(XTAL) of frequency 16 MHz.  
SYSCTL\_OSC\_MAIN - This denotes that we are using an external main oscillator ( Again represents the 16 MHz crystal ).

* By default all the peripherals are disabled. we will have to enable it before we use it.
* before calling any pheriperal driverlib function we must enable the clock to the pheriperal pin. is we fail to so so isr error would be genreted.

SysCtlPeripheralsEnable(SYSCTL\_PERIPH\_GPION); --> GPIO N port selected.

* When the board is powered on the default all peripherals are disabled, thus the about function is used to enable the peripheral port selected.

**Setting the data direction**

GPIOPinTypeGPIOOutput(base\_add,pins)

GPIOPinTypeGPIOOutput(GPIO\_PORTN\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1).

* The above function tells the data direction of the gpio pin, whether the pin behaves as input or output.

GPIOPinTypeGPIOInput(GPIO\_PORTN\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1).

* the pins are configured as bit banding i.e pins to be configured is suppose to be mentioned seperatly when we configure the ports. The function we use is the above functions.
* if we want to configure two pins in a particular port then we must use the logical OR operater in the pin mode parameter of the function.

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_1,2);

* This fuction is used to write data to the pin selected. ex:- consider we wnt to set the pin3 of any port to high, so that the data must be 0b0001000. i.e 16 in decimal, since second parameter contains only the gpio\_pin\_3 then 0b11111111 can also be set instead of 16 since it is bit banding.

**Delay function**

* The delay can be created by SysCtlDelay() function.
* void delay\_ms(int del)     //generates delay in milliseconds  
  {  
  del = (SysCtlClockGet()/3.0)\*del/1000.0;  
  SysCtlDelay(del);  
  }

* On board leds are connected to the gpiof port i.e pin 2,4,8.

**GPIO PIN**

* A gpio pin can be either acted as a gpio pin as well as a peripheral pins
* Note that all pin cannot act as the peripheral featured pins, it which it is only a gpio pin .
* the dirvers are at driverlib/gpio.h

http://embeddedgurus.com/state-space/tag/arm-cortex-m/