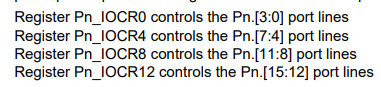
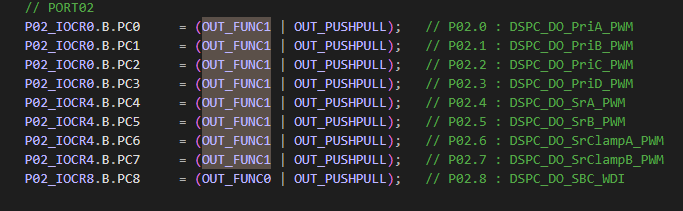
How to Configure Port



Example:



Port 2 Pin 0 hence IOCR0

Port 2 Pin 8 hence IOCR8

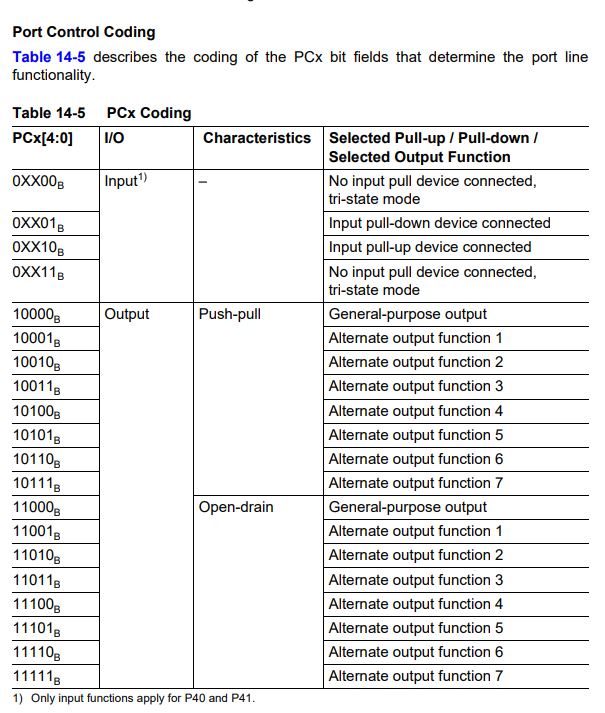
P02\_IOCR8.B.PC8 MEANS Port 2 Pin 8 configured using IOCR 8

The Below Table describe about the configuration of the IOCR registers

OUTFUN0

OUTFUN1

THESE are macro with value as said in the table (see below) which is defined in the code



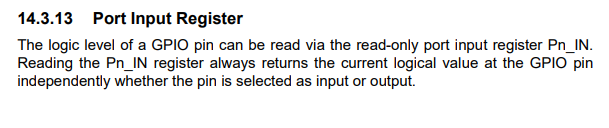
SO

P02\_IOCR0.B.PC3    = (OUT\_FUNC1 | OUT\_PUSHPULL);   // P02.3 : DSPC\_DO\_PriD\_PWM

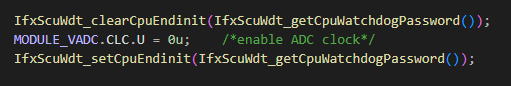
Here this means Port 2 Pin 3 which is configured using IOCR0 is made as Push-Pull Output and it is configured to use Alternate Output Function 1 (The macro OUT\_FUNC1 is having value 0x11 IN CODE)

So what is that Alternate Function 1 you need to see table of port2 pin 3





**ADC Configuration**

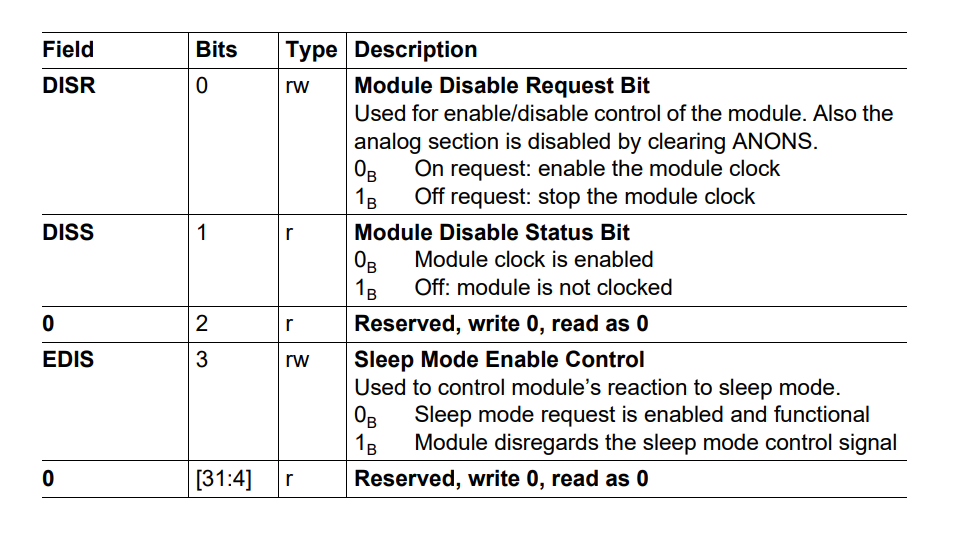


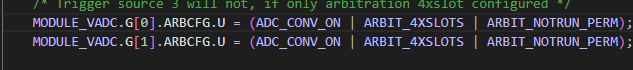
Here the CLC register is made 0 to enable the ADC module

See the below description of the Reg to understand the configuration

Clear CPU End Init is made to write the protected Registers

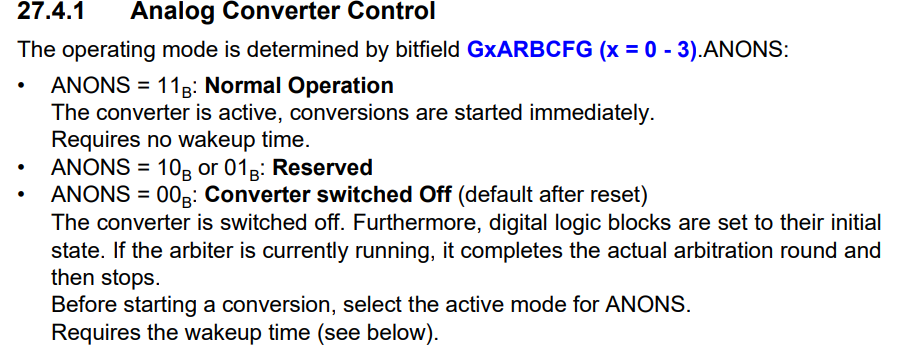
allowing write access to protected CPU registers. This is often necessary before modifying certain CPU registers.



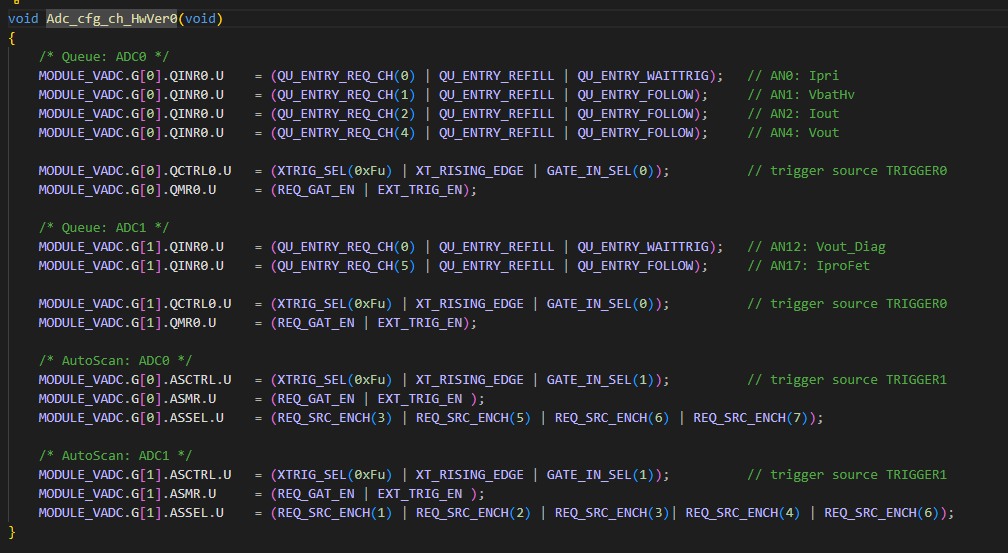


Here there are 2 ADC converter ie Group 1 and 2

Both have 14 analog input channel each and 12 bit resolution



See page 27-83 84 85

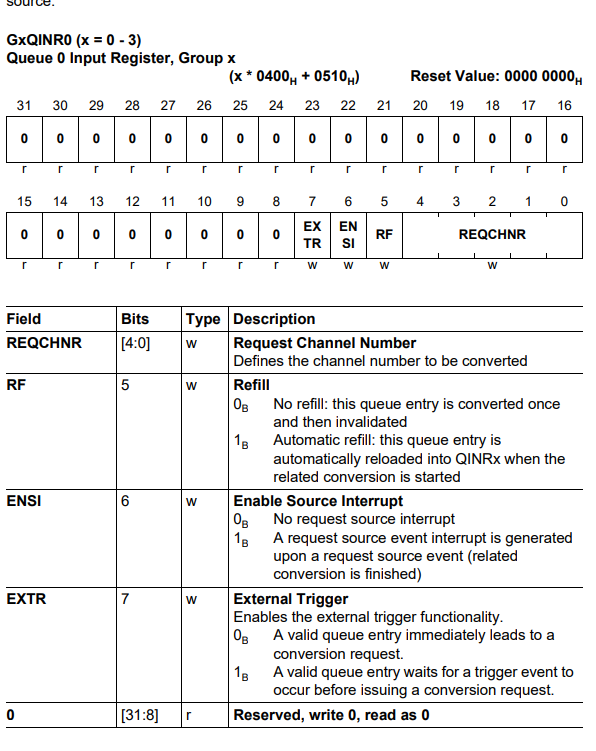


MODULE\_VADC.G[0].QINR0.U    = (QU\_ENTRY\_REQ\_CH(0) | QU\_ENTRY\_REFILL | QU\_ENTRY\_WAITTRIG);

HERE ADC Group 0 Queue 0 input Register

The Queue Input Register is the entry point for conversion requests of a queued request source

Queue conversion and Auto scan conversion of channels



**Power Supply To MC**

All internal supplies except analog domain (VAREF & VDDM) are supplied by the EVRs. The analog supply domain is separated from the main EVR supply domain and can be supplied by separate external regulators or trackers

It is possible to have a unified supply scheme with a 3.3V ADC domain (VDDM / VAREFx = 3.3 V) and the remaining system also running on 3.3 V supply (VDDP3 = 3.3 V). Alternatively it is also possible to have a mixed supply scheme with a 5 V ADC domain (VDDM / VAREFx = 5 V) and the remaining system running on 3.3 V supply (VDDP3 = 3.3 V

**Here we have in LVDC**

**Mixed supply scheme with a 5 V ADC domain (VDDM / VAREFx = 5 V) and the remaining system running on 3.3 V supply (VDDP3 = 3.3 V)**

**GTM Module**

TOM Module

TOM 0 and TOM1

16 channel each (0 to 15)

#define TOM\_CH\_CFG(TOMx\_CHx, Cnt, Period, Cmd, Ctrl)      \

              { GTM\_##TOMx\_CHx##\_CN0.U  = Cnt;    \

                GTM\_##TOMx\_CHx##\_SR0.U  = Period; \

                              GTM\_##TOMx\_CHx##\_SR1.U  = Cmd;    \

                              GTM\_##TOMx\_CHx##\_CTRL.U = Ctrl; }

**(**Channel , Count,Period, cmd, control )

TOM 0

Channel 3 -> Debug Pin of B01 P 33.7

Channel 6 -> Debug Pin of B02 P23.1

Channel 7 -> Generate Trigger to other channel Fast Interrupt (15 us ) (

TOM0\_CH7 -> GTM\_TOM0\_IRQ3 -> SRC\_GTMTOM03

ADC Group 0 and 1 trigger signal 0

->P13.2

Channel 8 -> Driver A -> P02.0

Channel 9 -> Driver B -> P02.1

Channel 10 -> Driver C -> P02.2

Channel 11 -> Driver D -> P02.3

Channel 12-> Driver SRA -> P02.4

Channel 13 -> Driver SRB->P02.5

Based on HW cfg version 1 or 0 14 and 15 channel are configured differently for clamp

Here HWVer 0 is defined

Channel 14 -> driver clamp SRA ->P02.6

Channel 15 -> driver clamp SRB -> P02.7

TOM 1

Channel 0 -> SBC WDG -> P02.8

Channel 5 -> driver Pri E -> P20.10

Channel 11 -> ADC Autoscan source trigger - ADC group 0 and 1 trigger signal 1

Isr 100 us

Channel 12 -> Isr 1 ms

Isr1ms Interrupt: TOM1\_CH12 -> GTM\_TOM1\_IRQ6 -> SRC\_GTMTOM16

Channel 13 -> Debug Pin of C0 -> P20.9

Channel 14 -> Isr 10ms

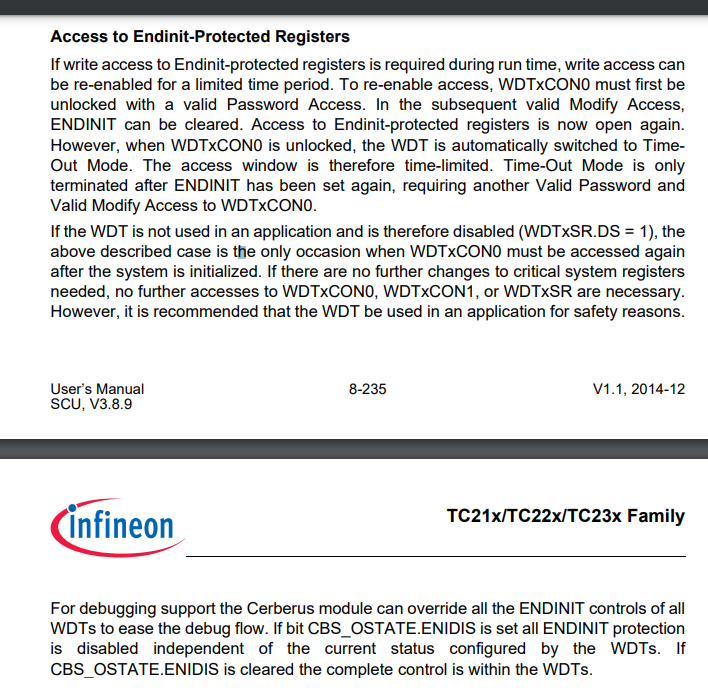
Isr10ms Interrupt: TOM1\_CH14 -> GTM\_TOM1\_IRQ7 -> SRC\_GTMTOM17

**CPU WDT and Endinit Protection**

We Are not using the internal WDT , only the external wdt provided by TLF SBC

The SCU registers to control WDT and Endinit protection is the same

Here we are using the endinit protection not the wdt

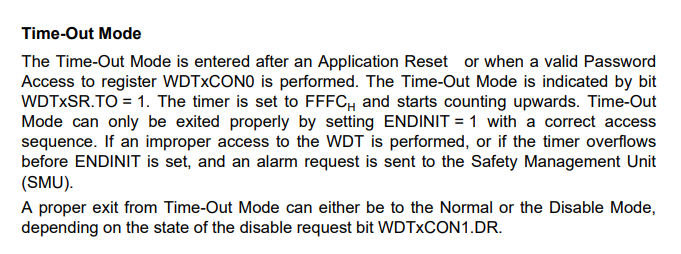


We are clearing the ENDINIT bit before modifying the ENDINIT Protected registers

And after modifying the Registers we are again setting the ENDINIT Bit

There is a timer value set soon after the ENDINIT bit is cleared and before the timer expiring we need to set the ENDINIT Bit back

**About the timer**



WDTCPU0CON0 is for Access Checks, Password, ENDINIT Prot related

WDTCPU0CON1 is related to internal CPU WDG which is disabled in our case