# 15 HARDWARE INSIGHTS: CLOCKS, RESETS and POWER SUPPLY – Code Snips

**Note: These Code Snips are taken straight from the book chapter; i.e. the “Program Examples”. In some cases therefore they are not complete programs.**

/\*Program Example 15.1 Adjusts clock divider through register CCLKCFG,

with trial blinky action \*/

#include "mbed.h" //keep this, as we are using DigitalOut

DigitalOut myled(LED1);

#define CCLKCFG (\*(volatile unsigned char \*)(0x400FC104))

// function prototypes

void delay(void);

int main() {

CCLKCFG=0x00000005; // divider divides by this number plus 1

while(1) {

myled = 1;

delay();

myled = 0;

delay();

}

}

void delay(void){ //delay function.

int j; //loop variable j

for (j=0;j<5000000;j++) {

j++;

j--; //waste time

}

}

****Program Example 15.1: Changing the CPU Clock Divider settings****

/\*Program Example 15.2 Switches off PLL0, with blinky action

\*/

#include "mbed.h"

DigitalOut myled(LED1);

#define CCLKCFG (\*(volatile unsigned char \*)(0x400FC104))

#define PLL0CON (\*(volatile unsigned char \*)(0x400FC080))

#define PLL0FEED (\*(volatile unsigned char \*)(0x400FC08C))

#define PLL0STAT (\*(volatile unsigned int \*)(0x400FC088))

// function prototypes

void delay(void);

int main() {

// Disconnect PLL0

PLL0CON &= ~(1<<1); // Clears bit 1 of PLL0CON, the Connect bit

PLL0FEED = 0xAA; // Feed the PLL. Enables action of above line

PLL0FEED = 0x55; //

// Wait for PLL0 to disconnect. Wait for bit 25 to become 0.

while ((PLL0STAT & (1<<25)) != 0x00);//Bit 25 shows connection status

// Turn off PLL0; on completion, PLL0 is bypassed.

PLL0CON &= ~(1<<0); //Bit 0 of PLL0CON disables PLL

PLL0FEED = 0xAA; // Feed the PLL. Enables action of above line

PLL0FEED = 0x55;

// Wait for PLL0 to shut down

while ((PLL0STAT & (1<<24)) != 0x00);//Bit 24 shows enable status

/\*\*\*\*Insert Optional Extra Code Here\*\*\*\*

to change PLL0 settings or clock source.

\*\*OR\*\* just continue with PLL0 disabled and bypassed\*/

//blink at the new clock frequency

while(1) {

myled = 1;

delay();

myled = 0;

delay();

}

}

void delay(void){ //delay function.

int j; //loop variable j

for (j=0;j<5000000;j++) {

j++;

j--; //waste time

}

}

**P**rogram Example 15.2: Switching off the main PLL****

#define PLL0CFG (\*(volatile unsigned int \*)(0x400FC084))

...

// Set PLL0 multiplier

PLL0CFG = 07; //arbitrary multiply by 8, divide value left at 1

PLL0FEED = 0xAA; // Feed the PLL

PLL0FEED = 0x55;

// Turn on PLL0

PLL0CON |= 1<<0;

PLL0FEED = 0xAA; // Feed the PLL

PLL0FEED = 0x55;

// Wait for main PLL (PLL0) to come up

while ((PLL0STAT & (1<<24)) == 0x00);

// Wait for PLOCK0 to become 1

while ((PLL0STAT & (1<<26)) == 0x00);

// Connect to the PLL0

PLL0CON |= 1<<1;

PLL0FEED = 0xAA; // Feed the PLL

PLL0FEED = 0x55;

while ((PLL0STAT & (1<<25)) == 0x00); //Wait for PLL0 to connect

****Program Example 15.3: Code fragment to set PLL0 multiplier****

/\*Program Example 15.4

Powers down certain elements of the mbed, when not in use.

\*/

#include "mbed.h"

//import next from mbed site

#include "PowerControl/EthernetPowerControl.h"

DigitalOut myled1(LED1);

DigitalOut myled4(LED4);

#define PCONP (\*(volatile unsigned long \*)(0x400FC0C4))

Ticker blinker;

void blink() {

myled1=!myled1;

myled4=!myled4;

}

int main() {

myled1=!myled4;

PHY\_PowerDown(); //\*\*comment this in and out

//Turn all peripherals OFF, except repetitive interrupt timer,

PCONP = 0x00008000; //which is needed for Ticker

blinker.attach(&blink, 0.0625);

while (1) {

wait(1);

}

}

**P**rogram Example 15.4: Switching off unused circuit sections****