**#include**<stdio.h>

**#include** <evmomapl138\_gpio.h>

**#include** "evmomapl138\_psc.h"

//#include "soc\_OMAPL138.h"

**#include** "evmomapl138.h"

//#include "lcdkOMAPL138.h"

/\* HW Macros \*/

**#include** "types.h"

/\* System Defines \*/

**#include** "evmomapl138\_sysconfig.h"

////////////////////////////////////////////////////////////////////////////////////////

/\* Switch Configuration \*/

/\* GP0[1] to GP0[4] is mapped to SW1[3:4] on OMAPL138/C6748 LCDK boards \*/

/\* Pin Multiplexing bit mask to select GP0[1] to GP0[4] pin. \*/

**#define** PINMUX1\_GPIO0\_1\_ENABLE (SYSCFG\_PINMUX1\_PINMUX1\_27\_24\_GPIO0\_1\_SYSCFG\_PINMUX1\_PINMUX1\_27\_24\_SHIFT)

**#define** PINMUX1\_GPIO0\_2\_ENABLE (SYSCFG\_PINMUX1\_PINMUX1\_23\_20\_GPIO0\_2\_SYSCFG\_PINMUX1\_PINMUX1\_23\_20\_SHIFT)

**#define** PINMUX1\_GPIO0\_3\_ENABLE (SYSCFG\_PINMUX1\_PINMUX1\_19\_16\_GPIO0\_3\_SYSCFG\_PINMUX1\_PINMUX1\_19\_16\_SHIFT)

**#define** PINMUX1\_GPIO0\_4\_ENABLE (SYSCFG\_PINMUX1\_PINMUX1\_15\_12\_GPIO0\_4\_SYSCFG\_PINMUX1\_PINMUX1\_15\_12\_SHIFT)

/\* LED Configuration \*/

/\* GP6[12], GP6[13], GP2[12] and GP0[9] is mapped to D4, D5, D6, D7 LEDs on OMAPL138/C6748 LCDK boards \*/

/\* Pin Multiplexing bit mask to select GP6[12] pin. \*/

**#define** PINMUX13\_GPIO6\_12\_ENABLE (SYSCFG\_PINMUX13\_PINMUX13\_15\_12\_GPIO6\_12\_SYSCFG\_PINMUX13\_PINMUX13\_15\_12\_SHIFT)

/\* Pin Multiplexing bit mask to select GP6[13] pin. \*/

**#define** PINMUX13\_GPIO6\_13\_ENABLE (SYSCFG\_PINMUX13\_PINMUX13\_11\_8\_GPIO6\_13\_SYSCFG\_PINMUX13\_PINMUX13\_11\_8\_SHIFT)

/\* Pin Multiplexing bit mask to select GP2[12] pin. \*/

**#define** PINMUX5\_GPIO2\_12\_ENABLE (SYSCFG\_PINMUX5\_PINMUX5\_15\_12\_GPIO2\_12\_SYSCFG\_PINMUX5\_PINMUX5\_15\_12\_SHIFT)

/\* Pin Multiplexing bit mask to select GP0[9] pin. \*/

**#define** PINMUX0\_GPIO0\_9\_ENABLE (SYSCFG\_PINMUX0\_PINMUX0\_27\_24\_GPIO0\_9\_SYSCFG\_PINMUX0\_PINMUX0\_27\_24\_SHIFT)

**void** **PinMuxSetup\_leds**(**void**)

{

**unsigned** **int** savePinmux = 0;

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX13 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(13)) & ~(SYSCFG\_PINMUX13\_PINMUX13\_15\_12));

/\* Setting the pins corresponding to GP6[12] in PINMUX13 register.\*/

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(13)) =(PINMUX13\_GPIO6\_12\_ENABLE | savePinmux);

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX13 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(13)) & ~(SYSCFG\_PINMUX13\_PINMUX13\_11\_8));

/\* Setting the pins corresponding to GP6[13] in PINMUX13 register.\*/

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(13)) =(PINMUX13\_GPIO6\_13\_ENABLE | savePinmux);

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX5 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(5)) & ~(SYSCFG\_PINMUX5\_PINMUX5\_15\_12));

/\* Setting the pins corresponding to GP2[12] in PINMUX5 register.\*/

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(5)) =(PINMUX5\_GPIO2\_12\_ENABLE | savePinmux);

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX0 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(0)) &~(SYSCFG\_PINMUX0\_PINMUX0\_27\_24));

/\* Setting the pins corresponding to GP0[9] in PINMUX0 register.\*/

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(0)) =(PINMUX0\_GPIO0\_9\_ENABLE | savePinmux);

}

**void** **PinMuxSetup\_switches**(**void**)

{

**unsigned** **int** savePinmux = 0;

/\* Setting the pins corresponding to GP0[1] in PINMUX1 register.\*/

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX1 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) &~(SYSCFG\_PINMUX1\_PINMUX1\_27\_24));

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) =(PINMUX1\_GPIO0\_1\_ENABLE | savePinmux);

/\* Setting the pins corresponding to GP0[2] in PINMUX1 register.\*/

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX1 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) &~(SYSCFG\_PINMUX1\_PINMUX1\_23\_20));

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) =(PINMUX1\_GPIO0\_2\_ENABLE | savePinmux);

/\* Setting the pins corresponding to GP0[3] in PINMUX1 register.\*/

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX1 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) &~(SYSCFG\_PINMUX1\_PINMUX1\_19\_16));

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) =(PINMUX1\_GPIO0\_3\_ENABLE | savePinmux);

/\* Setting the pins corresponding to GP0[4] in PINMUX1 register.\*/

/\*

\*\* Clearing the bit in context and retaining the other bit values

\*\* in PINMUX1 register.

\*/

savePinmux = (HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) &~(SYSCFG\_PINMUX1\_PINMUX1\_15\_12));

HWREG(SOC\_SYSCFG\_0\_REGS + SYSCFG0\_PINMUX(1)) =(PINMUX1\_GPIO0\_4\_ENABLE | savePinmux);

}

////////////////////////////////////////////////////////////////////////////////////////

//static void Delay(volatile unsigned int delay)

**int** **main**(**void**)

{

/\* The Local PSC number for GPIO is 3. GPIO belongs to PSC1 module.\*/

PSCModuleControl(SOC\_PSC\_1\_REGS, HW\_PSC\_GPIO, PSC\_POWERDOMAIN\_ALWAYS\_ON,

PSC\_MDCTL\_NEXT\_ENABLE);

/\* Pin Multiplexing of pins GP0[1] to GP0[4] of GPIO Bank 2 for DIP SWITCHEs in OMAPL138 LCDK board \*/

PinMuxSetup\_switches();

/\* Pin Multiplexing of pins GP6[12], GP6[13], GP2[12], GP0[9], for LEDs in OMAPL138 LCDK board \*/

PinMuxSetup\_leds();

/\* 2,3,4,5 is the GPIO no for GP0[1] to GP0[4]; Refer page no 901 in OMAPL138/C6748 TRM \*/

/\* SWITCHEs SETUP \*/

/\* Sets the pin 2 (GP0[1]) as input.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 2, GPIO\_DIR\_INPUT);

/\* Sets the pin 3 (GP0[2]) as input.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 3, GPIO\_DIR\_INPUT);

/\* Sets the pin 4 (GP0[3]) as input.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 4, GPIO\_DIR\_INPUT);

/\* Sets the pin 5 (GP0[4]) as input.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 5, GPIO\_DIR\_INPUT);

/\* LEDs SETUP \*/

/\* Sets the pin 109 (GP6[12]) as output.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 109, GPIO\_DIR\_OUTPUT);

/\* Sets the pin 110 (GP6[13]) as output.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 110, GPIO\_DIR\_OUTPUT);

/\* Sets the pin 45 (GP2[12]) as output.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 45, GPIO\_DIR\_OUTPUT);

/\* Sets the pin 10 (GP0[9]) as output.\*/

GPIODirModeSet(SOC\_GPIO\_0\_REGS, 10, GPIO\_DIR\_OUTPUT);

**while**(1)

{

/\* 2,3,4,5 is the GPIO no for GP0[1] to GP0[4]; Refer page no 901 in OMAPL138/C6748 TRM \*/

**if** (GPIOPinRead(SOC\_GPIO\_0\_REGS, 2))

{

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 109, GPIO\_PIN\_LOW);

**printf**("S1 is not pressed\n");

}

**else**

{

**printf**("S1 is pressed\n");

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 109, GPIO\_PIN\_HIGH);

}

**if** (GPIOPinRead(SOC\_GPIO\_0\_REGS, 3))

{

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 110, GPIO\_PIN\_LOW);

**printf**("S2 is not pressed\n");

}

**else**

{

**printf**("S2 is pressed\n");

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 110, GPIO\_PIN\_HIGH);

}

**if** (GPIOPinRead(SOC\_GPIO\_0\_REGS, 4))

{

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 45, GPIO\_PIN\_LOW);

**printf**("S3 is not pressed\n");

}

**else**

{

**printf**("S3 is pressed\n");

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 45, GPIO\_PIN\_HIGH);

}

**if** (GPIOPinRead(SOC\_GPIO\_0\_REGS, 5))

{

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 10, GPIO\_PIN\_LOW);

**printf**("S4 is not pressed\n");

}

**else**

{

**printf**("S4 is pressed\n");

GPIOPinWrite(SOC\_GPIO\_0\_REGS, 10, GPIO\_PIN\_HIGH);

}

}

}

////////////// Errors ///////////////////////////////////////////////////////////////////////////

