

```

#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h"
#include "inc/hw_gpio.h"
#include "inc/hw_types.h"
#include "driverlib/gpio.h"
#include "driverlib/sysctl.h"
#include "driverlib/rom.h"
#include "driverlib/rom_map.h"
#include "driverlib/pin_map.h"
#include "driverlib/adc.h"

#include "common.h"
#include "init.h"
#include "init_cpu.h"

```

```

//-----
// GET THE ACTUAL CPU CLOCK RATE |
//-----

```

```
PRIVATE uint uiSysClk = 0;
```

```
uint GetCpuClk(void) { return(uiSysClk); }
```

```

//-----
// INITIALIZE THE CPU |
//-----

```

```
void Init_Cpu(void) {
```

```
    // Run from the PLL at 120 MHz //
```

```
    uiSysClk = MAP_SysCtlClockFreqSet( ( SYSCTL_XTAL_25MHZ |
                                         SYSCTL_OSC_MAIN   |
                                         SYSCTL_USE_PLL     |
                                         SYSCTL_CFG_VCO_480 ),
                                         120000000 );
```

```

MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA); // Turn on port A.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB); // Turn on port B.
// MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC); // Turn on port C.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD); // Turn on port D.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE); // Turn on port E.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); // Turn on port F.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOG); // Turn on port G.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOH); // Turn on port H.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOJ); // Turn on port J.
// MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOK); // Turn on port K.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOL); // Turn on port L.
// MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOM); // Turn on port M.
// MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPION); // Turn on port N.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOP); // Turn on port P.
// MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOQ); // Turn on port Q. **

while(!MAP_SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOP)) { } // Wait till Port P ready.
// Change to Port Q if enabled. **

MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_SSI2); // Enable SSI's 2 and 3.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_SSI3);

MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0); // Enable Debug UART.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_UART4); // Enable MIDI UART.

MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_EEPROM0); // Enable EEPROM.

```

```

MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_USB0);           // Enable USB port.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);           // Enable the A2D ports.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);          // Enable A2D timer.

//////////
//  GPIO A  //
//////////

// A0 In  DebugRxD/   Debug UART 0
// A1 Out DebugTxD/   Debug UART 0
// A2 In  MidiRxD     Midi  UART 4
// A3 Out MidiTxD     Midi  UART 4
// A4 Out LED_1
// A5 Out LED_2
// A6 In  TP0         PU
// A7 In  TP1         PU

// UART0 //

GPIOPinConfigure(GPIO_PA0_U0RX);           // Configure UART 0.
GPIOPinConfigure(GPIO_PA1_U0TX);
MAP_GPIOPinTypeUART(GPIO_PORTA_BASE, PA_DebugRxD_BIT | PA_DebugTxD_BIT);

GPIOPinConfigure(GPIO_PA2_U4RX);           // Configure UART 4.
GPIOPinConfigure(GPIO_PA3_U4TX);
MAP_GPIOPinTypeUART(GPIO_PORTA_BASE, PA_MidiRxD_BIT | PA_MidiTxD_BIT);

// LEDs //

MAP_GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, PA_LED_1_BIT); // LED 1 port output.
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, PA_LED_2_BIT); // LED 2 port output.

// TEST POINTS //

GPIOPinTypeGPIOInput(GPIO_PORTA_BASE, GPIO_PIN_6 | GPIO_PIN_7); // Inputs.
GPIOPadConfigSet( GPIO_PORTA_BASE, // Input Pullups.
                  ( GPIO_PIN_6 | GPIO_PIN_7 ),
                  GPIO_STRENGTH_4MA, // Strength ignored.
                  GPIO_PIN_TYPE_STD_WPU ); // Weak Pullups.

//////////
//  GPIO B  //
//////////

// B0 ?   UsbId
// B1 ?   UsbVbus
// B2 In  SW1
// B3 In  T2         PU
// B4 In  T3         PU
// B5 In  T4         PU

MAP_GPIOPinTypeUSBAnalog(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1);

//////////
//  GPIO C  //
//////////

// C4 In  T5         PU
// C5 In  T6         PU
// C6 In  T7         PU
// C7 In  T8         PU

```

```

////////////////////////////////////
//  GPIO D  //
////////////////////////////////////

// D0 Out  CPU-SPI-1-A0
// D1 Out  CPU-SPI-1-A1
// D2 Out  CPU-SPI-1-A2
// D3 Out  CPU-SPI-1-A3
// D4 In   T9
// D5 In   T10
// D6 ?    UsbEn
// D7 ?    UsbPLft

// SPI-1 ADDRESSES //

MAP_GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, PD_CpuSpi_1_A0_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, PD_CpuSpi_1_A1_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, PD_CpuSpi_1_A2_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, PD_CpuSpi_1_A3_BIT);

// TEST POINTS //

MAP_GPIOPinTypeGPIOInput(GPIO_PORTD_BASE, PD_TP9_BIT | PD_TP10_BIT); // Inputs.
MAP_GPIOPadConfigSet( GPIO_PORTD_BASE, // Input Pullups.
                      ( PD_TP9_BIT | PD_TP10_BIT ),
                      GPIO_STRENGTH_4MA, // Strength ignored.
                      GPIO_PIN_TYPE_STD_WPU ); // Weak Pullups

// USB //

HWREG(GPIO_PORTD_BASE + GPIO_O_LOCK) = GPIO_LOCK_KEY;
HWREG(GPIO_PORTD_BASE + GPIO_O_CR) = 0xFF;
MAP_GPIOPinConfigure(GPIO_PD6_USB0EPEN);
MAP_GPIOPinConfigure(GPIO_PD7_USB0PFLT);
MAP_GPIOPinTypeUSBDigital(GPIO_PORTD_BASE, GPIO_PIN_6 | GPIO_PIN_7);

////////////////////////////////////
//  GPIO E  //
////////////////////////////////////

// E0 In  AD3   Volume Pedal
// E1 In  AD2   Power Filament peak V+
// E2 In  AD1   Low Power B+ Sense
// E3 In  AD0   High Power B+ Sense
// E4 In  AD9   Temperature
// E5 In  AD8   tp-77

GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_0);
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_1);
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_2);
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_4);
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_5);

////////////////////////////////////
//  GPIO F  //
////////////////////////////////////

// F0 In  T13           (Ssi2RxD)
// F1 Out  CPU-SPI-2-DO Ssi3TxD
// F2 In  T14
// F3 Out  CPU-SPI-2-CLK Ssi3Clk
// F4 In  T12

GPIOPinConfigure(GPIO_FF1_SSI3XDAT0);
GPIOPinConfigure(GPIO_FF3_SSI3CLK);
GPIOPinTypeSSI(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_3);

```

```
//////////////////////////////////
//  GPIO G  //
//////////////////////////////////
```

```
// G0 In  T15
// G1 In  T16
// G2 In  T17
// G3 In  T18
// G4 In  T19
// G5 Out  CPU-SPI-1-DO      Ssi2TxD
// G6 Out  CPU-D-CS/
// G7 Out  CPU-SPI-1-CLK    Ssi2Clk
```

```
GPIOPinConfigure(GPIO_PG5_SSI2XDAT0);
GPIOPinConfigure(GPIO_PG7_SSI2CLK);
GPIOPinTypeSSI(GPIO_PORTG_BASE, GPIO_PIN_5 | GPIO_PIN_7);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTG_BASE, PG_CpuSpi_2D_CS_BIT);
```

```
//////////////////////////////////
//  GPIO H  //
//////////////////////////////////
```

```
// H0 In  SW2-0
// H1 In  SW2-1
// H2 In  SW2-2
// H3 In  SW2-3
```

```
MAP_GPIOPinTypeGPIOInput( GPIO_PORTH_BASE,
                           PH_SW2_0_BIT |
                           PH_SW2_1_BIT |
                           PH_SW2_2_BIT |
                           PH_SW2_3_BIT );
```

```
// Inputs.
```

```
MAP_GPIOPadConfigSet( GPIO_PORTH_BASE,
                      ( PH_SW2_0_BIT | PH_SW2_1_BIT |
                        PH_SW2_2_BIT | PH_SW2_3_BIT ),
                      GPIO_STRENGTH_4MA,
                      GPIO_PIN_TYPE_STD_WPU );
```

```
// Input Pullups.
```

```
// Strength ignored.
```

```
// Weak Pullups
```

```
//////////////////////////////////
//  GPIO J  //
//////////////////////////////////
```

```
// J0 Out  CPU-SPI-1-CS/
// J1 Out  CPU-SPI-2-CS/
```

```
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTJ_BASE, PJ_CpuSpi_1A_CS_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTJ_BASE, PJ_CpuSpi_2BCD_CS_BIT);
```

```
//////////////////////////////////
//  GPIO K  //
//////////////////////////////////
```

```
// K0 In  T22
// K1 In  T23
// K2 In  T24
// K3 In  T25
// K4 In  T26
// K5 In  T27
// K6 In  T28
// K7 In  T29
```

```

////////////////////////////////////
//  GPIO L  //
////////////////////////////////////

// L0 Out
// L1 In
// L2 In
// L3 In
// L4 Out
// L5 Out
// L6 UsbDp
// L7 UsbDm

// SPI CONTROL SIGNALS //

MAP_GPIOPinTypeGPIOOutput(GPIO_PORTL_BASE, PL_CpuParSer_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTL_BASE, PL_CpuParOE_BIT);

// USB //

MAP_GPIOPinTypeUSBAAnalog(GPIO_PORTL_BASE, GPIO_PIN_6 | GPIO_PIN_7);

////////////////////////////////////
//  GPIO M  //
////////////////////////////////////

// M0 In
// M1 In
// M2 In
// M3 In
// M4 In
// M5 In
// M6 In
// M7 In

////////////////////////////////////
//  GPIO N  //
////////////////////////////////////

// N0 In
// N1 In
// N2 In
// N3 In
// N4 In
// N5 In

////////////////////////////////////
//  GPIO P  //
////////////////////////////////////

// P0 Out  A0
// P1 Out  A1
// P2 Out  A2
// P3 Out  A3
// P4 Out  A4
// P5 Out  A5

// SPI-2 ADDRESSES //

MAP_GPIOPinTypeGPIOOutput(GPIO_PORTP_BASE, PP_CpuSpi_2_A0_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTP_BASE, PP_CpuSpi_2_A1_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTP_BASE, PP_CpuSpi_2_A2_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTP_BASE, PP_CpuSpi_2_A3_BIT);
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTP_BASE, PP_CpuSpi_2_A4_BIT);

```

```
MAP_GPIOPinTypeGPIOOutput(GPIO_PORTP_BASE, PP_CpuSpi_2_A5_BIT);
```

```
//////////  
//  GPIO Q  //  
//////////
```

```
// Q0 In  
// Q1 In  
// Q2 In  
// Q3 In  
// Q4 In  
// Q5 In  
// Q6 In
```

```
}
```

```
//-----
```