**Interrupts I**

*// The interrupt handler for the for SysTick interrupt.*

static void SysTickIntHandler(void)

{

*// Initiate a conversion*

    ADCProcessorTrigger(ADC0\_BASE, 3);

    adcSampleCount++;

}

*// The handler for the ADC conversion complete interrupt.*

*// Writes to the circular buffer.*

void ADCIntHandler(void)

{

    uint32\_t ulValue;

*// Get the single sample from ADC*

    ADCSequenceDataGet(ADC0\_BASE, 3, &ulValue);

*// Place it in the circular buffer (advancing write index)*

    writeCircBuf(&intBuffer, ulValue);

*// Clean up, clearing the interrupt*

    ADCIntClear(ADC0\_BASE, 3);

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Initialisation functions for the clock (incl. SysTick), ADC, display*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

void initIntHandler(void)

{

*// Set up the period for the SysTick timer.  The SysTick timer period is*

*// set as a function of the system clock.*

    SysTickPeriodSet(SysCtlClockGet() / SAMPLE\_RATE\_HZ);

*// Register the interrupt handler*

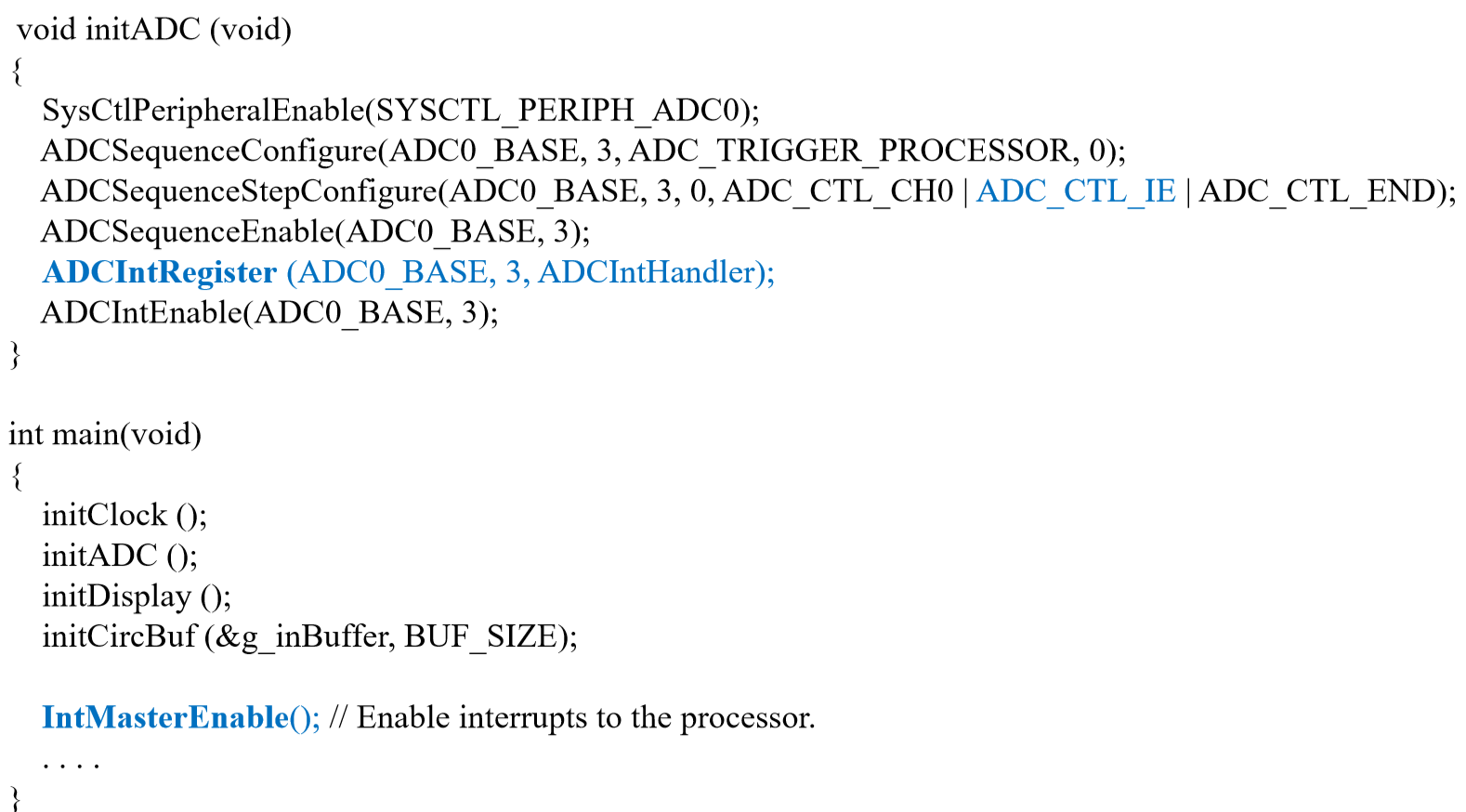
    SysTickIntRegister(SysTickIntHandler);

*// Enable interrupt and device*

    SysTickIntEnable();

    SysTickEnable();

}



Initializes

Passing in a Function pointer ADCIntHandler() in to ADCIntRegister(…)