```c
int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );

    printf("\n\r-------------------------------------------------------");
    printf("\n\r                ADC Sequencer Demo                     ");
    printf("\n\r-------------------------------------------------------\n\r");

    TC0_TimerStart();

    ADC0_Enable();

    DMAC_ChannelCallbackRegister(DMAC_CHANNEL_0, DMAC_Callback, 0);
    DMAC_ChannelTransfer(DMAC_CHANNEL_0, (const void *)&ADC0_REGS->ADC_RESULT, &adc_count, sizeof(adc_count));

    while ( true )
    {
        /* Maintain state machines of all polled MPLAB Harmony modules. */
        SYS_Tasks ( );

        if(transferDone == true)
        {
            transferDone = false;

            for(int i = 0; i<3;i++)
            {
               inp_voltage[i] = (float)adc_count[i] * ADC_VREF / 4095U;
            }

            printf("\rADC0 Count = 0x%x, ADC0 Voltage = %d.%02d V, "
                   "ADC1 Count = 0x%x, ADC1 Voltage = %d.%02d V, "
                   "ADC2 Count = 0x%x, ADC2 Voltage = %d.%02d V ",
                   adc_count[0], (int)inp_voltage[0], (int)((inp_voltage[0] - (int)inp_voltage[0])*100.0),
                   adc_count[1], (int)inp_voltage[1], (int)((inp_voltage[1] - (int)inp_voltage[1])*100.0),
                   adc_count[2], (int)inp_voltage[2], (int)((inp_voltage[2] - (int)inp_voltage[2])*100.0));

            DMAC_ChannelTransfer(DMAC_CHANNEL_0, (const void *)&ADC0_REGS->ADC_RESULT,
              &adc_count, sizeof(adc_count));
        }
    }
    /* Execution should not come here during normal operation */

    return ( EXIT_FAILURE );
}
```