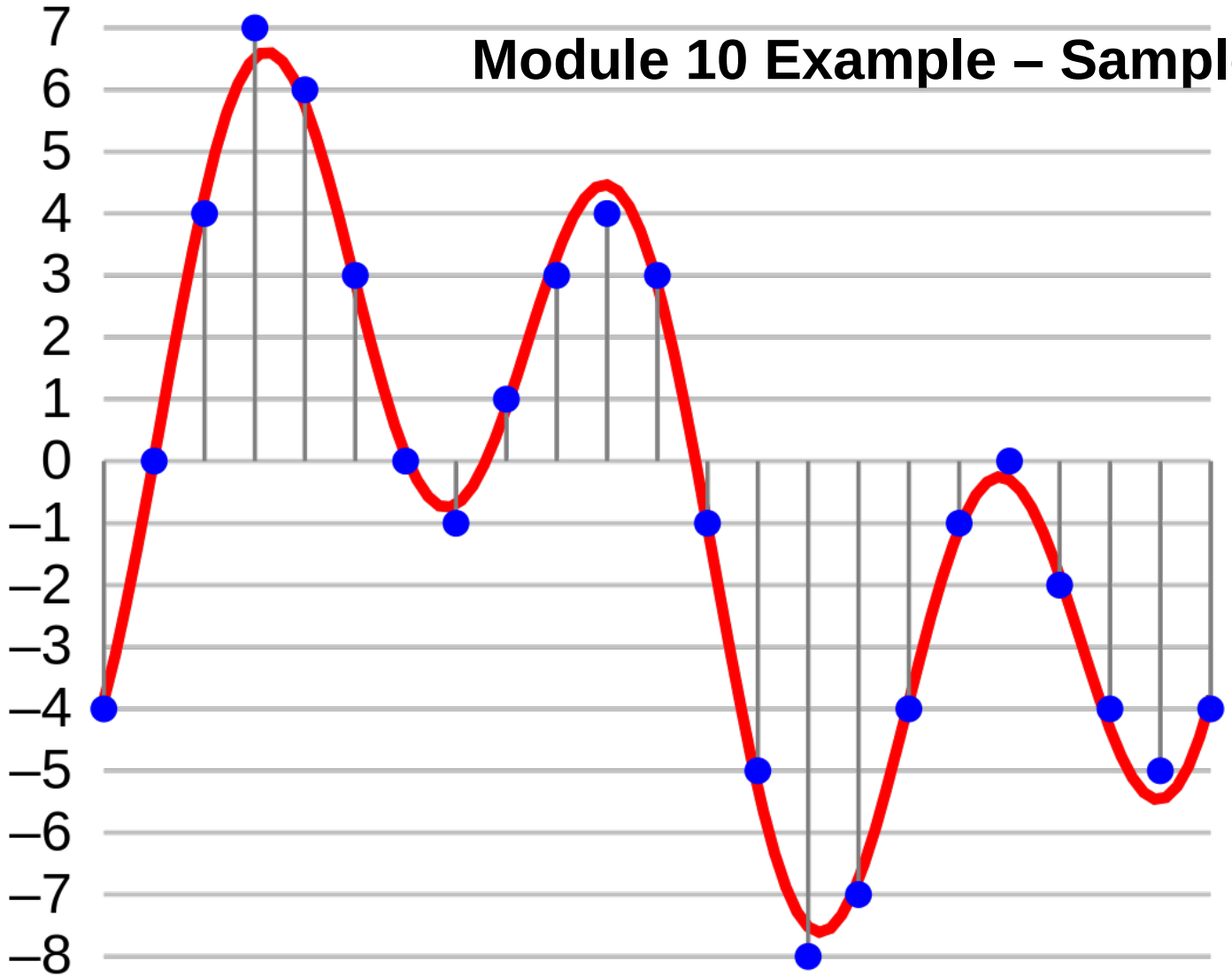
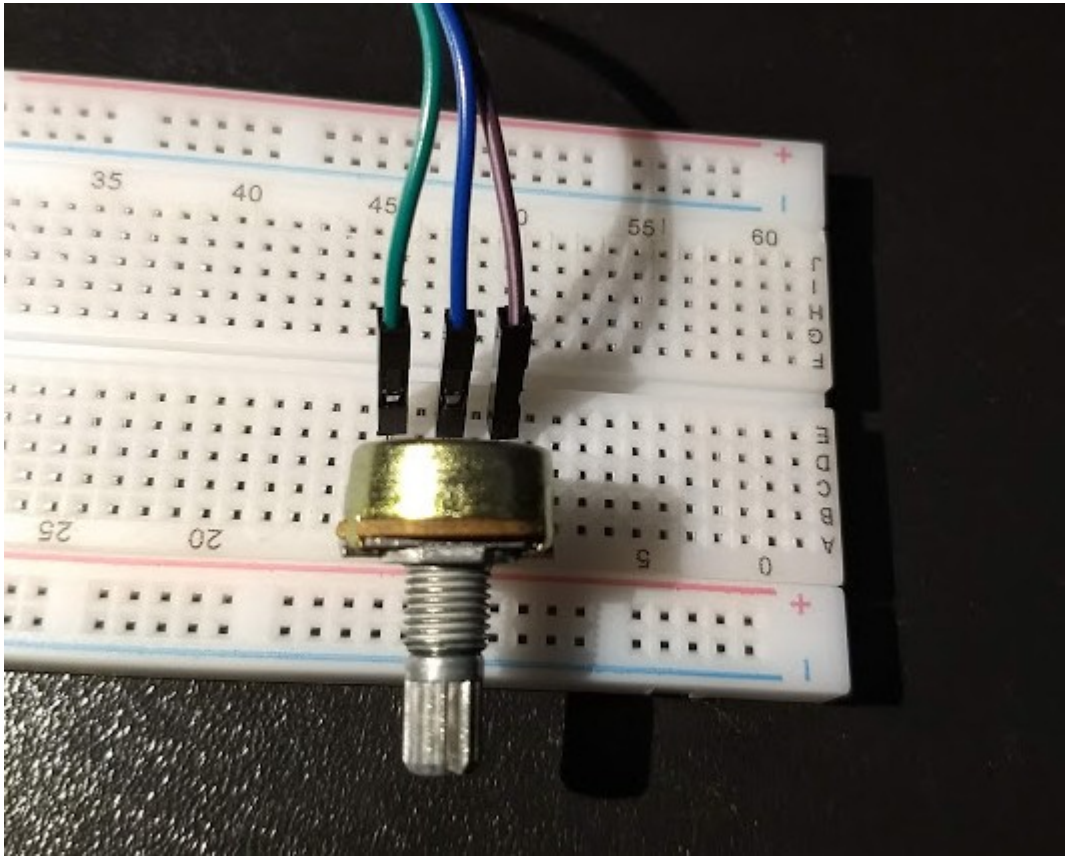
Module 10 Example – Sample an ADC channel

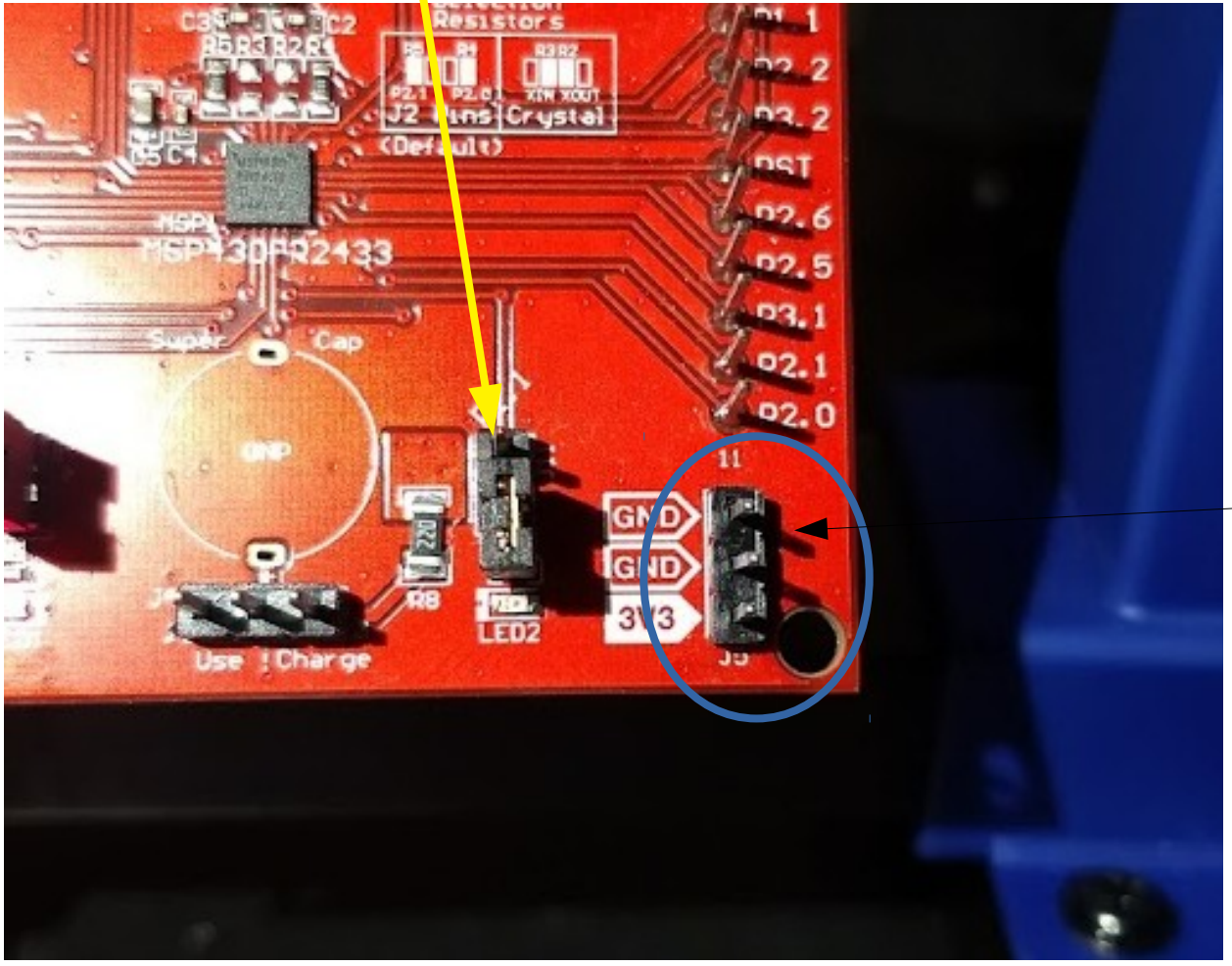


Mount potentiometer from Starter Kit onto Breadboard and insert a wire for each pin



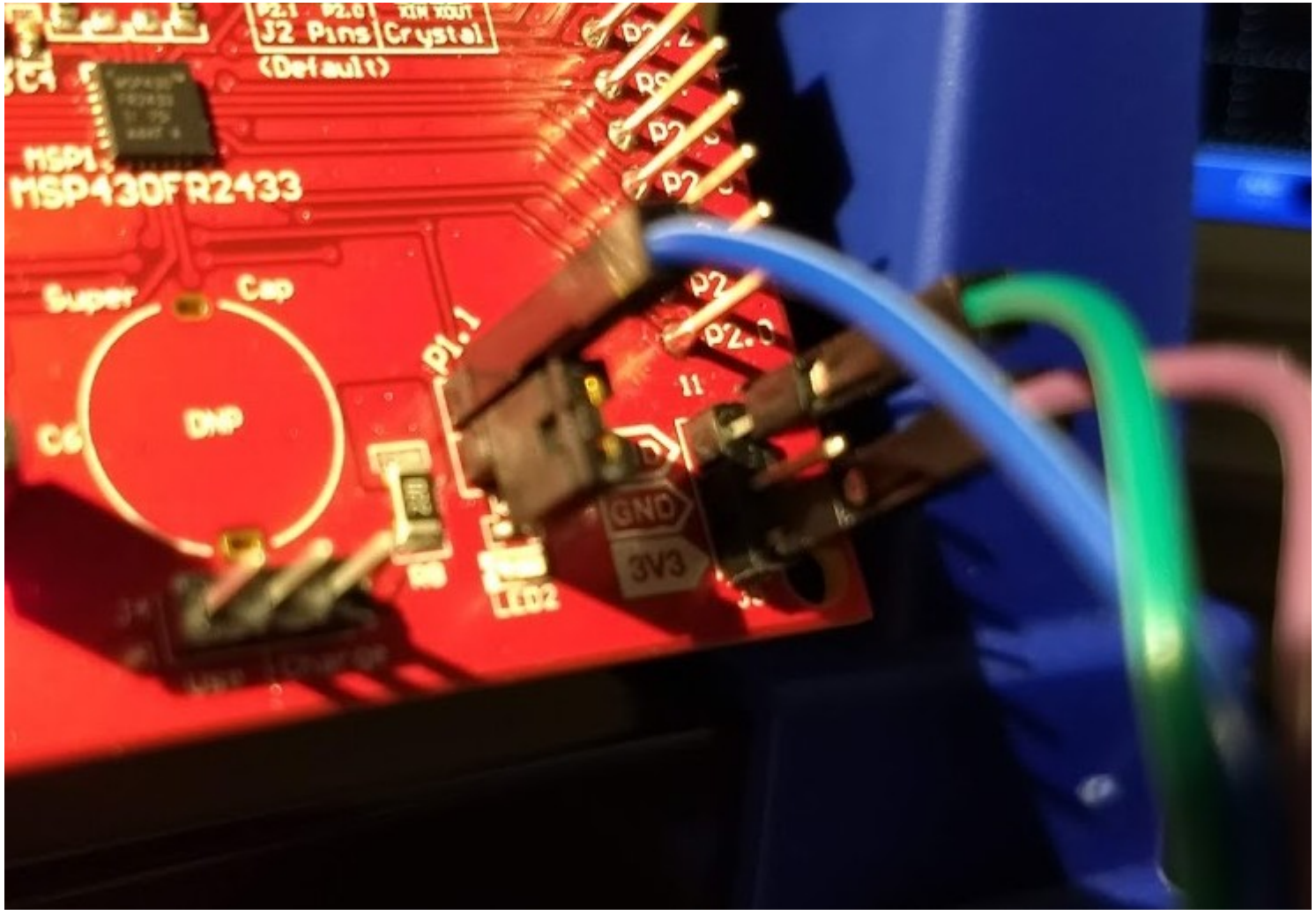
Wires on the ends go to GND and 3V3 on the board, center wiper goes to P1.1

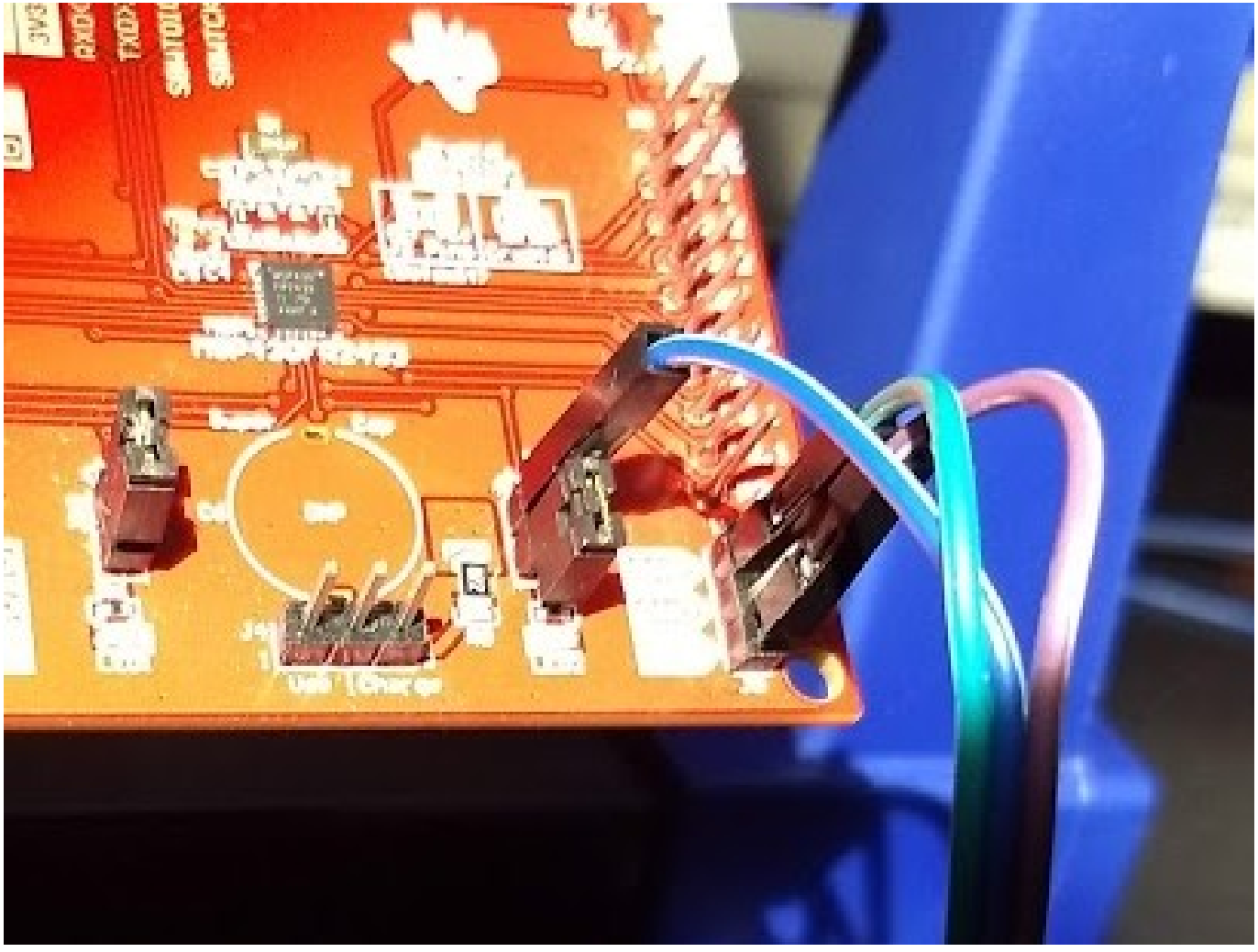
P1.1 Analog A1 Input (also Green LED) – remove jumper expose Upper pin



Ground
And +3V

Connections for Potentiometer – Center connect to P1.1, ends to GND and 3V3





Another view of potentiometer connections to board

```
1
2  //******************************************************************************
3  //  MSP430FR24xx Demo - ADC, Sample A1, AVcc Ref, Set LED if A1 > 0.5*AVcc
4  //
5  //  Description: This example works on Single-Channel Single-Conversion Mode.
6  //  A single sample is made on A1 with default reference to AVcc.
7  //  Software sets ADCSC to start sample and conversion - ADCSC automatically
8  //  cleared at EOC. ADC internal oscillator times sample (16x) and conversion.
9  //  In Mainloop MSP430 waits in LPM0 to save power until ADC conversion complete,
10 //  ADC_ISR will force exit from LPM0 in Mainloop on reti.
11 //  If A1 > 0.5*AVcc, P1.0 set, else reset.
12 //   P1.1 is A1, jumper for Green LED will force 381 result
13 //  ACLK = default REFO ~32768Hz, MCLK = SMCLK = default DCODIV ~1MHz.
14 //
15 //              MSP430FR2433
16 //           -----------------
17 //       /|\|                 |
18 //        | |                 |
19 //        --|RST              |
20 //          |                 |
21 //      >---|P1.1/A1      P1.0|--> LED
22 //
23 //
24 //  Wei Zhao   Texas Instruments Inc.
25 /*
26  *   sketch_ADC_UART_243301.ino
27  *   modified by HW -
28  *   place potentiometer    GND
29  *                          P1.1  - remove Green LED jumper
30  *                          V+3
31  *
32  */
33
34 //******************************************************************************
```

‘Set Up part of the program’

```
34  //*******************************************************************************
35  #include <msp430.h>
36
37  unsigned int ADC_Result;
38
39  int main(void)
40  {
41      WDTCTL = WDTPW | WDTHOLD;                       // Stop WDT
42
43      // Configure GPIO
44      P1DIR |= BIT0;                                  // Set P1.0/LED to output direction
45      P1OUT &= ~BIT0;                                 // P1.0 LED off
46
47      // Configure ADC A1 pin
48      SYSCFG2 |= ADCPCTL1;
49
50      // Disable the GPIO power-on default high-impedance mode to activate
51      // previously configured port settings
52      PM5CTL0 &= ~LOCKLPM5;
53
```

1

2

3

4

5

```
54    //   Clock System Setup  ACLK = 32786, MCLK = SMCLK = 1MHz
55     __bis_SR_register(SCG0);                          // disable FLL
56    CSCTL3 |= SELREF__REFOCLK;                        // Set REFOCLK as FLL reference source
57    CSCTL0 = 0;                                       // clear DCO and MOD registers
58    CSCTL1 &= ~(DCORSEL_7);                           // Clear DCO frequency select bits first
59    CSCTL1 |= DCORSEL_3;                              // Set DCOCLK = 8MHz
60    CSCTL2 = FLLD_1 + 121;                            // FLLD = 1, DCODIV = 4MHz
61    __delay_cycles(3);
62    __bic_SR_register(SCG0);                          // enable FLL
63    while(CSCTL7 & (FLLUNLOCK0 | FLLUNLOCK1));        // Poll until FLL is locked
64    CSCTL4 = SELMS__DCOCLKDIV | SELA__XT1CLK;         // set ACLK = XT1 = 32768Hz,
65                                                      //DCOCLK as MCLK and SMCLK source
66    CSCTL5 |= DIVM1;
```

6

```
68    // Configure ADC10
69    ADCCTL0 |= ADCSHT_2 | ADCON;                      // ADCON, S&H=16 ADC clks
70    ADCCTL1 |= ADCSHP;                                // ADCCLK = MODOSC; sampling timer
71    ADCCTL2 |= ADCRES;                                // 10-bit conversion results
72    ADCMCTL0 |= ADCINCH_1;                            // A1 ADC input select; Vref=AVCC
73    ADCIE |= ADCIE0;                                  // Enable ADC conv complete interrupt
74
```

‘Loop’ part of the program – Get a sample and wait for arrival

```
75
76      while(1)
77      {
78          ADCCTL0 |= ADCENC | ADCSC;                  // Sampling and conversion start
79          __bis_SR_register(LPM0_bits | GIE);         // LPM0, ADC_ISR will force exit
80  // this line below waits until Return-From-Interrupt
81          if (ADC_Result < 0x1FF)
82              P1OUT &= ~BIT0;                         // Clear P1.0 LED off
83          else
84              P1OUT |= BIT0;                          // Set P1.0 LED on
85          __delay_cycles(500000);                     // Delay 0.5 second
86      }
87  }
```

```
89  // ADC interrupt service routine
90  #pragma vector=ADC_VECTOR
91  __interrupt void ADC_ISR(void)
92  {
93      switch(ADCIV)
94      {
95          case ADCIV_NONE:
96              break;
97          case ADCIV_ADCOVIFG:  // memory overflow - ADCMEM0 overflow
98              break;
99          case ADCIV_ADCTOVIFG:  // ADC conversion-time-overflow
100             break;
101         case ADCIV_ADCHIIFG:   // Comparator above upper threshold
102             break;
103         case ADCIV_ADCLOIFG:   // Comparator below lower threshold
104             break;
105         case ADCIV_ADCINIFG:   // Comparator inside window
106             break;
107         case ADCIV_ADCIFG:    // Conversion complete
108             ADC_Result = ADCMEM0;
109             __bic_SR_register_on_exit(LPM0_bits);           // Clear CPUOFF bit from LPM0
110             break;
111         default:
112             break;
113     }
114 }
115
```