```c
/*
  Initiating ADC Conversion:
    1. Writing to ATD0CTL5
    2. Edge on external trigger
    3. Level on external trigger

  Completed Conversion
    1. Reading ATD0STAT1
    2. Interrupt when complete
*/

#include <hidef.h>      /* common defines and macros */
#include <mc9s12dp512.h>     /* derivative information */
#pragma LINK_INFO DERIVATIVE "mc9s12dp512"

#include "PLL.h"
#include "lcd.h"
#include "Timer.h"
#include <stdio.h>

#include "SCI1.h"
#include "Fifo.h"
#include "Xbee.h"

volatile char whee;

void main(void) {
  int i;
  DDRP |= 0x80;
  PLL_Init(); // 24 MHz
  Timer_Init();
  LCD_Open();
  Fifo_Init();
  SCI1_Init(9600); // SCI output to PC

  asm cli;

    LCD_Clear();

  XBee_Init();
  Timer_Wait10ms(100);
  for(;;) {
    long blah;
    FrameType frame;
    Fifo_Init();
    while(!XBee_RecieveTxFrame(&frame));

    LCD_Clear();
    for(i=5; i<frame.length; i++) {
      if(i == 13) {
        LCD_GoTo(1,0);
      }
      LCD_OutChar(frame.data[i]);
    }
    Fifo_Init();
    PTP ^= 0x80;
  }
}
```

```c
#include "derivative.h"      /* derivative-specific definitions */
#include "SCI1.h"
#include "Timer.h"
#include "XBee.h"

int ok(void) {
  if(SCI1_InChar() != 'O') {
    return 0;
  }
  if(SCI1_InChar() != 'K') {
    return 0;
  }
  if(SCI1_InChar() != 0x0D) {
    return 0;
  }

  return 1;
}


void sendATCommand(char * command) {
  char * temp;
  do {
    temp = command;
    while(*temp) {
      SCI1_OutChar(*temp);
      temp++;
    }
    SCI1_OutChar(0x0D);
    Timer_Wait1ms(20);
  } while(!ok());
}

/*-----------------------XBee_Init---------------------------
  Initialize XBee
  Inputs: none
  Outputs: none */
void XBee_Init(void) {
    SCI1_OutChar('X');
    Timer_Wait10ms(110);

    SCI1_OutString("+++");
    Timer_Wait10ms(110);

    sendATCommand("ATDL4F");
    sendATCommand("ATDH0");
    sendATCommand("ATMY4E");
    sendATCommand("ATAP1");
    sendATCommand("ATCN");
    PTP ^= 0x80;
}

/*-----------------------XBee_RecieveTxFrame------------------
  Receives a frame from data in
  Inputs: None
  Outputs: Input Frame */
int XBee_RecieveTxFrame(FrameType * frame)
{
  short i;
  static short FrameID = 1;

  if(SCI1_InChar() != 0x7E)
  {
    return 0;
  }

  frame->length = SCI1_InChar();
  frame->length <<= 8;
  frame->length += SCI1_InChar();

  for(i = 0; i < frame->length; i++)
  {
    frame->data[i] = SCI1_InChar();
  }

  frame->checkSum = SCI1_InChar();
  frame->frameID = FrameID++;
  return 1;
}
```

```
typedef struct
{
  short length; // Length of data
  char data[30];
  char checkSum;
  short frameID;
} FrameType;



/*-----------------------XBee_Init----------------------------
  Initialize XBee
  Inputs: none
  Outputs: none */
void XBee_Init(void);

/*-----------------------XBee_RecieveTxFrame-------------------
  Receives a frame from data in
  Inputs: None
  Outputs: Input Frame */
int XBee_RecieveTxFrame(FrameType * frame);
```