

// C programming of LPC1768 ADC to convert analog to digital value and send the converted value on UART1 for display

```
#include <lpc17xx.h>
#include <stdio.h>
#include <string.h>
```

```
int result;
char value[5];
```

```
void uart1_init(void);
void delay(uint32_t);
void adc_init(void);
```

// Function to initialize UART1

```
void uart1_init(void)
{
    LPC_SC->PCONP |= (1 << 4); //enable power to UART1
    LPC_PINCON->PINSEL4 |= (2 << 0) | (2 << 2); //Pin P2.0 used as TXD1 and Pin P2.1 used as RXD1
    LPC_UART1->FCR = 0x07; //enable different FIFO
    LPC_UART1->LCR = 0x83; // 8 bits, no Parity, 1 Stop bit
    LPC_UART1->DLL = 0x12; //115200 Baud Rate @ 25.0 MHZ PCLK
    LPC_UART1->FDR = 0x21; //FR 1,507, DIVADDVAL=1, MULVAL=2
    LPC_UART1->DLM = 0x0; //High divisor latch = 0
    LPC_UART1->LCR = 0x03; //DLAB = 0
}
```

// Delay Function

```
void delay(uint32_t i)
{
    uint32_t x;
    for(x=0; x<=i; x++);
}
```

// Function to initialize ADC

```
void adc_init(void)
{
    LPC_PINCON->PINSEL1 |= (1 << 14); //P0.23 as AD0.0
    LPC_SC->PCONP |= (1 << 12); // power for ADC peripheral
    LPC_ADC->ADCR |= (1 << 21); //To make ADC operational
    LPC_ADC->ADCR |= (1 << 0); //select AD0.0 channel
    LPC_ADC->ADCR |= (1 << 8); //set the ADC Clock Divider as 1.
}
```

//Function to write character to Serial Port

```
int uart1_txdchar (int c)
{
    while (!(LPC_UART1->LSR & 0x20)); // Wait till character send
    LPC_UART1->THR = c;
    return (c);
}
```

//Function to send string on Uart1

```
void uart1_sendstring( char *ch)
{
    while(*(ch) != '\0')
    {
        uart1_txdchar(*ch); // send each character at a time
        ch++;
    }
}
```

```
int main()
{

    uart1_init(); // Initialization of UART1 at 115200 baud rate
    adc_init(); // Initialization of ADC

    while(1)
    {
        LPC_ADC->ADCR |= (1<<24); // Start the conversion
        while((LPC_ADC->ADDR0 & (1<<31)) == 0); // wait till conversion will complete
        result=((LPC_ADC->ADGDR>>4) & 0xffff); // 12 bit result

        uart1_sendstring("result= "); // display the result on uart1
        sprintf(value, " %d\n\r", result);
        uart1_sendstring(value);
    }
}
```