**THUMB-FACT**

```
NAME    main
PUBLIC  __iar_program_start
           SECTION datas:CONST(2)
num
dc32 0x05
fact
      ds32 1
        SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
ldr r6,num
ldr r0,=thumbcode
movlr,pc
bx r0
str r6,fact
stop    b stop
 code16
thumbcode
mov r4,r6
loop    sub r4,#1
beq here
mul r6,r4
b loop
here
bxlr
        END
```

**THUMB-SUM OF SQ**

```
NAME    main

PUBLIC  __iar_program_start
        SECTION datas:CONST(2)
num
        dc32 0x0A
sum
ds32  1

        SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
ldr r6,num
ldr r0,=thumbcode
movlr,pc
bx r0
str r1,sum
stop   b stop
  code16
thumbcode
mov r4,r6
mov r1,#0x00
loopmul r6,r4
add r1,r1,r6
sub r4,#1
mov r6,r4
bne loop
herebxlr
end
```

### THUMB-LAR SMALL

```
NAME    main
PUBLIC  __iar_program_start
        SECTION datas:CONST(2)
num
        dc32 0x0A,0x02,0x06,0x04,0x00
numend
dc32  0x00
len
        dc32 (numend-num)/4
        SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
ldr r1,=num
ldr r2,len
sub r2,r2,#1
ldr r3,num
ldr r4,num
ldr r0,=thumbcode
movlr,pc
bx r0
stop b stop
  code16
thumbcode
loop    add r1,#4
ldr r5,[r1]
cmp r5,r3
bplsubss
mov r3,r5
subsscmp r5,r4
bmi loop1
mov r4,r5
loop1   sub r2,#1
cmp r2,#0
bne loop
bxlr
end
```

### 64 BIT ADD/SUB

```
NAME main
```

```
   PUBLIC __iar_program_start
   SECTION .data:CONST:NOROOT(2)
first: DC32 0xAAAAAAAA,0x22222222
second: DC32 0x11111111,0x11111111
sum: DS32 0x04
difference: DS32 0x04
  SECTION .intvec:CODE:NOROOT(2)
main
__iar_program_start
  ADR R0,first
  ADR R1,second
  ADR R9,sum
  ADR R8,difference
  LDMIA R0,{R2,R3} ;Load first operand into R2 and R3
  LDMIA R1,{R4,R5} ;Load second operand into R4 and R5
  ADDS R6,R3,R5 ;Add lower 32 bits
  ADC R7,R2,R4 ;Add upper 32 bits
  STMIA R9,{R6,R7} ;Store sum
  SUBS R6, R3,R5 ;Subtract lower 32 bits
  SBC R7,R2,R4 ;Subtract upper 32 bits
  STMIA R8,{R6,R7} ;Store difference
stop B stop
 END
```

## BUBBLE SORT

```
NAME    main
PUBLIC  __iar_program_start
      SECTION .data:CONST:NOROOT (2)
mydata:
      DC32 0X99999999,0X33333333,0X44444444,0XAAAAAAAA,0XFFFFFFFF
mydataend:
      DC32 0x00
length:
      DC32 (mydataend-mydata)/4
      SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
MAIN
      LDR R0,length
      SUB R0,R0,#1
      MOV R5,R0
      LDR R1,=mydata
LOOP2   MOV R0,R5
      MOV R2,R1
      ADD R2,R2,#4
L1      LDR R3,[R1]
      LDR R4,[R2]
      CMP R3,R4
      BLO LOOP1 (BHS-DESCENDING)
      SWP R3,R4,[R1]
      STR R3,[R2]
```

```
LOOP1   ADD R2,R2,#4
        SUBS R0,R0,#1
        BHI L1
        ADD R1,R1,#4
        SUBS R5,R5,#1
        BGT LOOP2
stop    B    stop
        END
```

**NCR**

```
NAME   main
PUBLIC  __iar_program_start
        SECTION .data:CONST:NOROOT (2)
n       DC32  0xa
r       dc32  0x3
quotientdcd 0x0,0x0,0x0
remaindcd 0x0
        SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
ldr r0,n
bl fact           ;N! IS FOUND
mov r10,r6 ;r10=n!
ldr r0,r
bl fact          ; R! IS FOUND
mov r1,r6 ;r1=r!
ldr r5,n
ldr r8,r
sub r5,r5,r8   ;r5=n-r
mov r0,r5
bl fact          ; (N-R)! IS FOUND
mov r5,r6  ;r5=(n-r)!
mul r9,r5,r1 ;r5=r!*(n-r)!
bl div
stop         b stop
fact           EOR R6,R6,R6
            MOV R6,R0
            MOV R4,R6
LOOP           SUBS R4,R4,#1
            MULNE R7,R6,R4
            MOV R6,R7
            BNE LOOP
mov PC,LR
divmov r0,r10
mov r1,r9
mov r3,#0
loop
cmp r1,#0
beq err
cmp r0,r1
```

```
blt done
add r3,r3,#1
sub r0,r0,r1
b loop
err
mov r3,#0xffffffff
done
ldr r2,=remain
ldr r4,=quotient
str r0,[r2]
str r3,[r4]
movpc,lr
       END
```

**FACT**

```
NAME    main
PUBLIC  __iar_program_start
       SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
mov r0,#10
mov r1,r0
loop    subs r1,r1,#1
mulne r2,r0,r1
mov r0,r2
bne loop
stop    B       stop
       END
```

**PACKED**

```
NAME main
  PUBLIC __iar_program_start
  SECTION .data:CONST:NOROOT(2)
num DC16 0x1234
result DS32 0x00
  SECTION .intvec:CODE:NOROOT(2)
__iar_program_start
       ADR R0,num
       MOV R5,#4
       MOV R9,#12
       MOV R1,#0
;MOV R2,R1
       MOV R10,#10 ;Multiply by 10
       MOV R11,#0x0F ;To mask
loop            MOV R7,R1
               MUL R1,R7,R10
               LDRH R8,[R0]
               ROR R8,R8,R9
```

```
                AND R4,R8,R11
                ADD R1,R1,R4
                SUBS R9,R9,#4
                SUBS R5,R5,#1
                BNE loop
                STR R1,result
stop B stop
 END
```

**UNPACKED**

```
NAME main
   PUBLIC __iar_program_start
   SECTION .data:CONST:NOROOT(2)
num DC8 0x01,0x02,0x03,0x04
result DC8 0x00
  SECTION .intvec:CODE:NOROOT(2)
__iar_program_start
main
  ADR R0,num
  MOV R5,#4
  MOV R1,#0
  MOV R2,R1
  MOV R10,#10
loop     MOV R7,R1
         MUL R1,R7,R10 ;10(a+(10(b+(10(c+10d)))));
         LDRB R4,[R0],#1
         ADD R1,R1,R4
         SUBS R5,R5,#1 ;Count
         BNE loop
         STR R1,result
stop B stop
   END
```

**STRING PAL**

```
NAME main
       PUBLIC __iar_program_start
       SECTION .data:CONST:NOROOT(2)
string DCB 'radar'
endDCb 0x00
length DCD end-string
       SECTION .intvec:CODE:NOROOT(2)
__iar_program_start
main
ldr R5,length
ldr R0,=string
mov R1,R0
add R1,R1,R5
    SUB R1,R1,#1
loop CMP R1,R0
```

```
    BMI PALI
    LDRB R2,[R0],#1
    LDRB R3,[R1],#-1
    CMP R2,R3
    BEQ loop
NOTPALI MOV R10,#0
STOP B STOP
PALI MOV R10,#0XFF
stop    b stop
end
```

**STRING REV CPY**

```
NAME    main
PUBLIC  __iar_program_start
   SECTION .data:CONST:NOROOT (2)
str dc8 'hello'
stre dc8 0x0
len dc32 (stre-str)
copy ds32 0x20
rev ds32 0x20
   SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
                ldr r0,=str
                ldr r1,len
                ldr r3,=copy
                ldr r4,=rev
loop    ldrb r5,[r0],#1
                strb r5,[r3],#1
                strb r5,[r4],#-1
                subs r1,r1,#1
                bne loop
                ldr r1,len
                sub r1,r1,#1
stop    b stop
        END
```

**String copy and reverse**

```
  NAME    main
PUBLIC  __iar_program_start
   SECTION .data:CONST:NOROOT (2)
str dc8 'hello'
stre dc8 0x0
len dc32 (stre-str)
copy ds32 0x20
rev ds32 0x20
   SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
```

```
                ldr r0,=str
                ldr r1,len
                ldr r3,=copy
                ldr r4,=rev
mov r2, r0
add r2 ,r2, r1
loop    ldrb r5,[r0],#1
                strb r5,[r3],#1
                strb r5,[r4],#-1
ldrb r5,[r2],#1
strb r5,[r4],#-1
                subs r1,r1,#1
                bne loop
                ldr r1,len
                sub r1,r1,#1
stop    b stop
        END
```

**INTERFACING**

**DAC**

**//Sine**

```
#include "lpc214x.h"
#include "stdint.h"
voiddelay_ms(uint16_t j)
{
uint16_tx,i;
        for(i=0;i<j;i++)
for(x=0; x<6000; x++);
}
int main (void)
{
uint16_t value;
uint16_t i = 0;
uint16_t
sintable[64]={512,562,611,660,707,753,796,836,873,907,937,963,984,1001,
1013,1021,1023,1021,1013,1001,984,963,937,907,873,836,796,753,707,660,6
11,562,512,461,412,363,316,270,227,187,150,116,86,60,39,22,10,2,0,2,10,
22,39,60,86,116,150,187,227,270,316,363,412,461 };
        PINSEL1 = 0x00080000; IO0DIR = 0xFFFFFFFF;
        while(1){
        while(i<64)
        {
        DACR= (sintable[i]*50);
        delay_ms(1);
        i++;
        }
        i=0;
}
}
```

**//triangle**

```
#include "lpc214x.h"
#include "stdint.h"
int main (void)
{
uint16_t value;
uint16_t i = 0;
        PINSEL1 = 0x00080000;
        IO0DIR = 0xFFFFFFFF;
        while(1)
        {
        i=0;
        while(i!=1023)
        {
        DACR=i*50;
        i++;
        }
        i=1023;
        while(i!=0)
        {
        DACR=i*50;
        i--;
        }
        }
}
```

```
//SAWTOOTH
#include "lpc214x.h"
#include "stdint.h"
int main (void)
{
uint16_t value;
uint16_t i = 0;
        PINSEL1 = 0x00080000;
        IO0DIR = 0xFFFFFFFF;
        while(1)
        {
        i=0;
        while(i!=1023)
        {
        DACR=i*50;
        i++;
        }
        DACR=0;
        }
}
```

```
//SQUARE
#include "lpc214x.h"
#include "stdint.h"
voiddelay_ms(uint16_t j)
{
uint16_tx,i;
        for(i=0;i<j;i++)
for(x=0; x<6000; x++);
}
int main (void)
{
uint16_t value;
uint16_t i = 0;
        PINSEL1 = 0x00080000;
        IO0DIR = 0xFFFFFFFF;
        while(1)
        {
        DACR=1023*50;
        delay_ms(10);
        DACR=0;
        delay_ms(10);
        }
}
```

**KEYPAD**

```
#include "lpc214x.h"
#include "stdint.h"
unsignedinti,delay_ms,segval;
unsigned char index, lcdval,row,keyscan,keyret,keynum=0, keypress,scanret = 0xFF;
unsigned char seg7[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67,0x77,0x7c,0x39,0x5e,0x79,0x71,0x00,0x00,0x0
};
unsigned char scan[]    = {0xEF,0xDF,0xBF,0x7F,0x00} ;
unsigned char keycode[]  =
{0xEE,0xED,0xEB,0xE7,0xDE,0xDD,0xDB,0xD7,0xBE,0xBD,0xBB,0xB7,0x7E,0x7D,0x7B,0x77,0x00};
voidInitLPC(void)
{
        PINSEL0 = 0x00L;
        IO0DIR = 0XFFFFFFF0;
}
void Delay(unsigned intdms)
{       delay_ms = dms;
        while(delay_ms> 0)
                delay_ms--;
}
voidGetKey()
{
    row=0;
    while(1)
    {
        IO0CLR = 0xFF;
        row&= 0x3;
        keyscan=scan[row];
        IO0SET = keyscan;
        Delay(2);
        keyret = IO0PIN;
       if (keyscan != keyret)
          break;
        row++;
    }
for(i=0;i<0x10;i++)
    {
        if(keycode[i]==keyret)
```

```
                    keynum=i;
        }
        IO0CLR = 0xFF00;
        segval = seg7[keynum];
        segval<<= 8;
        IO0SET = segval;
}
void main(void)
{
        InitLPC();
        index=0;
              while(1)
            GetKey();
}
```

**LCD**

**//SINGLE LINE**

```
#include "lpc214x.h"
#include "stdint.h"
unsignedint cmd8[] = {0X38,0x38,0x0E,0x02,0x01,0x00};
unsigned intmsg[] = {'H','e','l','l','o',0x20,'R','I','T',0x20,0x00};
unsignedintlcdval,index,delay_ms;
voidInitLPC(void)
{
PINSEL0 = 0x00L;
IO0DIR = 0XFFFFFFFF;
}
void Delay(unsigned intdms)
{
delay_ms = dms;
while(delay_ms> 0)
{
delay_ms--;
}
}
Void InitLCD()
{
index=0;
lcdval=cmd8[index];
while(lcdval !=0x0)
{
IO0SET = lcdval;
lcdval |= 0x400;
IO0SET = lcdval;
Delay(500);
IO0CLR=0xFFFF;
index++;
lcdval=cmd8[index];
}
}
```

```
voidShowMsg()
{
index=0;
lcdval=msg[index];
while(lcdval !=0x0)
{
IO0SET = lcdval;
lcdval |= 0x500;
IO0SET = lcdval;
Delay(500);
IO0CLR=0xFFFF;
index++;
lcdval=msg[index];
}
}
void main(void)
{
InitLPC();
while(1)
{
InitLCD();
ShowMsg();
Delay(5000);
}
}
```

**//DOUBLE LINE**

```
#include "lpc214x.h"
#include "stdint.h"
unsignedint cmd8_one[] = {0x38,0x0E,0x02,0x01,0x00};
unsignedint cmd8_two[]= {0X38,0x0E,0x02,0xC0,0x00};
unsigned intmsg_one[] = {'H','e','l','l','o',0x20,'R','I','T',0x20,0x00};
unsigned intmsg_two[] = {'L','i','n','e',0x20,'T','W','O',0x20,0x00};
unsignedintlcdval,index,delay_ms;
voidInitLPC(void)
{
PINSEL0 = 0x00L;
IO0DIR = 0XFFFFFFFF;
}
void Delay(unsigned intdms)
{
delay_ms = dms;
while(delay_ms>0)
delay_ms--;
}
voidInitLCD_one()
{
index=0;
lcdval=cmd8_one[index];
while(lcdval !=0x0)
{
```

```
IO0SET = lcdval;
lcdval |= 0x400;
IO0SET = lcdval;
Delay(500);
IO0CLR=0xFFFF;
index++;
lcdval=cmd8_one[index];
}
}
voidInitLCD_two()
{
index=0;
lcdval=cmd8_two[index];
while(lcdval !=0x0)
{
IO0SET = lcdval;
lcdval |= 0x400;
IO0SET = lcdval;
Delay(500);
IO0CLR=0xFFFF;
index++;
lcdval=cmd8_two[index];
}
}
voidShowMsg_one()
{
index=0;
lcdval=msg_one[index];
while(lcdval !=0x0)
{
IO0SET = lcdval;
lcdval |= 0x500;
IO0SET = lcdval;
Delay(500);
IO0CLR=0xFFFF;
index++;
lcdval=msg_one[index];
}
}
voidShowMsg_two()
{
index=0;
lcdval=msg_two[index];
while(lcdval !=0x0)
{
IO0SET = lcdval;
lcdval |= 0x500;
IO0SET = lcdval;
Delay(500);
IO0CLR=0xFFFF;
index++;
```

```
lcdval=msg_two[index];
}
}
void main(void)
{
InitLPC();
while(1)
{
InitLCD_one();
ShowMsg_one();
InitLCD_two();
ShowMsg_two();
}
}
```

**LED**

**//<u>COUNTER</u>**

```
#include "lpc214x.h"
#include "stdint.h"
unsignedintdelay_ms,led_val;
unsigned char index;
voidInitLPC(void)
{
        PINSEL0 = 0x00L;
        IODIR = 0XFFFFFFFF;
}
void Delay(unsigned intdms)
{
        delay_ms = dms;
        while(delay_ms> 0)
                delay_ms--;
}
main()
{
    index=0;
    InitLPC();
        while(1)
        {
         index&= 0xffffffff;
         led_val = index++;
        IOSET =led_val;
        Delay(200000);
        IOCLR=0xFFFFFFFF;
        }
}
```

**//<u>BLINKING</u>**

```
#include "lpc214x.h"
```

```
#include "stdint.h"
unsignedintdelay_ms,led_val;
unsigned char index;
unsigned
intmvright[]={0x80808080,0x40404040,0x20202020,0x10101010,0x08080808,0x04040404,0x02020
202,0x01010101,0x00};
voidInitLPC(void)
{
        PINSEL0 = 0x00L;
        IODIR = 0XFFFFFFFF;
}
void Delay(unsigned intdms)
{
        delay_ms = dms;
        while(delay_ms> 0)
                delay_ms--;
}
main()
{
        index=0;
        InitLPC();
        while(1)
        {
        index&= 0x7;
        led_val = mvright[index++];
        IOSET =led_val;
        Delay(20000);
        IOCLR=0xFFFFFFFF;
        }
}
```

**7-SEG**

**//0000-FFFF**

```
include "lpc214x.h"
#include "stdint.h"
#define IO1      0x10000
#define IO2      0x20000
#define IO3      0x40000
#define IO4      0x80000
#define IOX      0xF0000
#define IOXcl    0xFFFFF
int count=0000;
unsignedint d1,d2,d3,d4;
unsigned char seg[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67,0x77,0x7c,0x39,0x5e,0x79,0x71,0x00};
voidinit_gpio()
{
        PINSEL0 = 0x00000000;
        PINSEL1 = 0x00000000;
        PINSEL2 = 0x00000000;
```

```
            IO0DIR = 0XFFFFFFFF;
            IO1DIR = 0XFFFFFFFF;
}
void delay()
{
  int c = 50000;
  while(c)
    c--;
}
voidshow_disp()
{
            //Digit 1
            d1 = count & 0x000F;
            IO0CLR = IOXcl;
            IO0SET = seg[d1];
            IO1SET = IOX;
            IO1CLR = IO1;
            delay();
            IO1SET = IOX;

            //Digit 2
d2 = count & 0x00F0;
            d2 >>= 4;
            IO0CLR = IOXcl;
            IO0SET= seg[d2];
            IO1SET = IOX;
            IO1CLR = IO2;
            delay();
            IO1SET= IOX;

            //Digit 3
            d3 = count & 0x0F00;
            d3 >>= 8;
            IO0CLR = IOXcl;
            IO0SET= seg[d3];
            IO1SET = IOX;
            IO1CLR = IO3;
            delay();
             IO1SET = IOX;

            //Digit 4
            d4 = count & 0xF000;
            d4>>= 4;
            IO0CLR = IOXcl;
            IO0SET = seg[d4];
            IO1SET = IOX;
            IO1CLR = IO4;
            delay();
            IO1SET = IOX;

}
```

```
int main( void )
{
  init_gpio();
  while(1)
  {
          show_disp();
          count++;
          count&= 0xFFFF;
  }
}
```

**//0-F**

```
#include "lpc214x.h"
#include "stdint.h"
#define IO1    0x10000
#define IOX    0xF0000
#define IOXcl  0xFFFFF
int count=0x0;
unsignedint d0;
unsigned char
seg[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67,0x77,0x7c,0x39
,0x5e,0x79,0x71,0x00};
voidinit_gpio()
{
      PINSEL0=0x00000000;
      PINSEL1=0x00000000;
      PINSEL2=0x00000000;
      IO0DIR=0xFFFFFFFF;
      IO1DIR=0xFFFFFFFF;
}
void delay()
{
      int c=250000;
      while(c)
      c--;
}
voidshow_disp()
{
      //Digit 0
      d0=count& 0x0F;
      IO0CLR=IOXcl;
      IO0SET=seg[d0];
      IO1SET=IOX;
      IO1CLR=IO1;
      delay();
      IO1SET=IOXcl;
}
int main(void)
{
      init_gpio();
      while(1)
      {
        show_disp();
        count++;
        count&=0xF;
        }
}
```