座位号：_____

姓名：_____

绝密★启用前

2024年4月高等教育自学考试全国统一命题考试

微型计算机原理与接口技术

（课程代码　02205）

注意事项：
1．本试卷分为两部分，第一部分为选择题，第二部分为非选择题。
2．应考者必须按试题顺序在答题卡（纸）指定位置上作答，答在试卷上无效。
3．涂写部分、画图部分必须使用2B铅笔，书写部分必须使用黑色字迹签字笔。

第一部分　选择题

一、单项选择题：本大题共15小题，每小题1分，共15分。在每小题列出的备选项中只有一项是最符合题目要求的，请将其选出。

1．对于八位二进制原码 $[10000001]_{原}$，下面选项中与之不相等的是
A．$[11111111]_{补}$　　B．-1D
C．129D　　D．$[11111110]_{反}$

2．已知'A'的ASCII码的十进制值是65，语句“printf("%o", 'A');”的正确输出是
A．A　　B．65
C．41　　D．101

3．如有定义：“char str[] = {"abc"};”，则str[3]的值是
A．'b'　　B．'c'
C．'\0'　　D．'d'

4．执行下面语句后，k和j的值分别是
```
for ( k = 'a'; k <= 'f'; k += 2)
    for (j = 0; j < k – 'a'; ++j)
        if  (j % 2)  break;
```
A．'c', 0　　B．'e', 1
C．'f', 0　　D．'g', 1

5．如有定义
```
union sample {
    unsigned char a;
    struct {
      unsigned char a1:3;
      unsigned char a2:2;
    } b;
} x;
x.a = 0x61;
```
则x.b.a1的值是
A．0　　B．1
C．2　　D．3

6．执行下列语句时，输出的行数是
```
int s = 1;
while (s < 10) {
    s *= 2;
    if (s % 4 != 0) continue;
    printf("%d\n",  s);
}
```
A．1　　B．2
C．3　　D．4

7．ARM指令集中所有ARM指令的长度均为
A．1个字节　　B．2个字节
C．3个字节　　D．4个字节

8．ARM处理器通过Store指令，将数据从寄存器保存到
A．高速缓存　　B．存储器
C．寄存器　　D．硬盘

9．ARM处理器运行模式中，系统模式数量有
A．1种　　B．2种
C．6种　　D．7种

10．ARM处理器运行模式中，每种异常模式都有
A．1个SPSR　　B．2个SPSR
C．3个SPSR　　D．4个SPSR

11．ARM 处理器发生复位后，强制 PC（R15）取指并执行的地址是

A．0x4000 0100　　B．0x0000 0101

C．0x0000 0000　　D．0x4000 0101

12．LPC2138 微控制器中，每个作为 GPIO 功能的引脚受若干寄存器的控制，其中不包括

A．IOxDIR　　B．IOxSET

C．IOxPIN　　D．IOxCLR

13．LPC2138 的向量中断控制器（VIC）中，具有可动态分配优先级的数量是

A．4 个　　B．8 个

C．16 个　　D．32 个

14．异步串行通信的传输线路的形式中，不存在的形式是

A．半单工通信　　B．单工通信

C．半双工通信　　D．全双工通信

15．在 A/D 转换中，输入模拟信号中最高频率分量为 5kHz，为了保证采样信号不失真，则最大采样周期为

A．0.1ms　　B．0.2ms

C．0.4ms　　D．1ms

第二部分　非选择题

二、填空题：本大题共 14 小题，每小题 2 分，共 28 分。

16．冯·诺依曼结构中的 CPU 由_____和_____组成。

17．如有定义“int a[] = {1, 4, 9, 3, 2};”，用表达式_____可获取数组 a 占用的存储空间字节数，用表达式_____可获取数组元素个数。

18．如果 x = 3，y = 0，执行表达式 ((x −= 3) && (y += 4))? ++x : −−y 后，x 的值是_____，y 的值是_____。

19．以下函数交换两个整型变量的值。如变量 x=2，y=3，执行 swap(&x, &y)后，x 的值为 3，y 的值为 2。请填空：

```
void swap(int *a, int *b)
{    int  k= *a;
     _____;
     _____;
}
```


20．以下递归函数显示一个由 * 组成的三角形。调用 tri(4, 0)将输出如下的三角形：

```
   *
  ***
 *****
*******
```

第一个参数表示行数，第二个参数表示最下面一行前的空格数。请填空：

```
void tri( int m, int n){
    int k;
    if ( m == 1 ) {
        for (k = 0; k < n; ++k) printf(" ");
        printf("*\n") ;
        return;
    }
    _____;
    for (k = 0; k < n; ++k) printf(" ");
    for (k = 0; _____ ; ++k)   printf("*");
    printf("\n");
}
```

21．某个栈的结点类型 Node 定义如下：

```
struct Node {
      int   data;
      struct Node *next;
};
```

栈顶指针是 top。系统中所有空闲单元被链成一个链接栈，栈顶指针为 topFree。top 和 topFree 都是外部变量。以下是出栈函数的实现，如果栈为空，返回 0，否则返回 1，并将出栈元素存放在 data 指向的单元中。请填空：

```
int pop(int *data) {
    Node   *p = top;
    if   (top == NULL)   return 0;   /*  出栈失败   */
    top = _____;
    *data = p->data ;
    p->next = _____;
    topFree = p;
    return 1;   }
```

22．ARM 指令执行过程的 3 个步骤分别是取指、_____和执行。

23．在 ARM 处理器中，当处理器认为当前指令未定义时，会产生_____异常中断。

24．在 ARM 体系的存储器中，按照大端字节顺序将数据 0x12345678 存入地址 0x40001000 中，则地址 0x40001001 中存入的字节数据为十六进制数_____。

25．LPC2138 微控制器的 GPIO 寄存器中，对 IOSET 寄存器写入 0，对应引脚输出_____。

26．LPC2138 微控制器的引脚设置为 GPIO 工作方式时，指令“IO0DIR = 0x00000010;”将引脚_____配置为输出。

27．向量中断控制器（VIC）是_____和 ARM 内核之间的桥梁。

28．异步串行通信中，传输的每个字符的数据位最少为_____位。

29．一个 10 位的 A/D 转换器，满量程电压为 5V，则其分辨率为_____mV（计算结果保留 2 位小数）。

三、改错题：本大题共 4 小题，每小题 2 分，共 8 分。每小题只有一处错误或不妥，请指出，并将其改正。

30．以下程序输出字符串 str 的内容，但执行结果有时正确，有时不正确。请指出错误并改正。

```
int main(){
    char   str[4] = "abc0";
    printf("%s\n", str);
    return 0;
}
```

31．函数 judge 判断参数 ch 的内容是否是小写字母。如调用 judge('n')返回 1，而调用 judge('$')返回 0。请指出错误并改正。

```
int   judge(char   ch) {
    return   ch  -  'a' >= 0 && ch  -  'a' < 026;
}
```

32．以下程序段统计有 size 个元素的数组 a 中值为 3 的元素个数，请指出错误并改正。

```
for (count = i = 0; i < size; ++i)
    if   (a[i] = 3)   ++count;
```

33．下面语句段输出整数 n 的所有因子并统计因子个数，请指出错误并改正。

```
for (k = 1, count = 0;   k <= n; ++k)
    if ( n % k == 0)
    printf("%d   ", k);
    ++count;
```


四、程序阅读题：本大题共 4 小题，每小题 4 分，共 16 分。

34．写出下列程序的执行结果。

```
#include <stdio.h>
int main(){
    int a[] = {2, 5, 2 };
    int ch, avg = 0;
    for (ch = 0; ch < 3; ++ch) {
        if (ch % 2) avg += ++a[ch];
        else avg += a[ch]++;
        printf("%d\t", a[ch]);
    }
    printf("%.3f \n", (double) avg / 3);
    return 0;
}
```

35．写出下列程序的执行结果。

```
#include <stdio.h>
void f(int a[],int n);
int main(){
    int a[] = {1, 2, 3, 4};
    f(a,4);
    printf("%d\t%d\t%d\t%d\n", a[0], a[1], a[2], a[3]);
    return 0;
}
void f(int a[], int n){
    static int cnt = 0;
    if (n == 1)return;
    a[cnt++] = a[n-1];
    printf("%d\n", cnt);
    f(a, n-1);
}
```

36．写出下列程序的执行结果。

```
void foo(char *p){
    switch(*p-'a') {
        case 1 : case 2:   printf("%c", ++(*p));
```

```
            case 3 : printf("C");
            case 4 : case 5: printf("%c\n", (*p)++); break;
            case 7 : printf("%c", (*p)--);
            default : printf("%c\n", *p);
        }
    }
    int main(){
        char s[] = "abh",   *p;
        for (p = s; *p; ++p)
            foo(p);
        printf("%s\n", s );
        return 0;
    }
```

37．写出下列程序的执行结果。

```
#include <stdio.h>
typedef struct {
    char   data;
    int    next;
} Node;
int show(Node a[], int head);
int main(){
    Node test[] = {{'b',3},{'a', 0},{'d',-1},{'c',2}};
    int k1;
    printf("%d\n",show(test, 1));
    for (k1 = 1; k1 != -1; k1 = test[k1].next)
        test[k1].next = test[test[k1].next].next ;
    printf("%d\n",show(test, 1));
    return 0;
}
int show(Node a[], int head){
    int   k = head,count = 0;
    do {
        printf("%c",a[k].data);
        k = a[k].next;
```



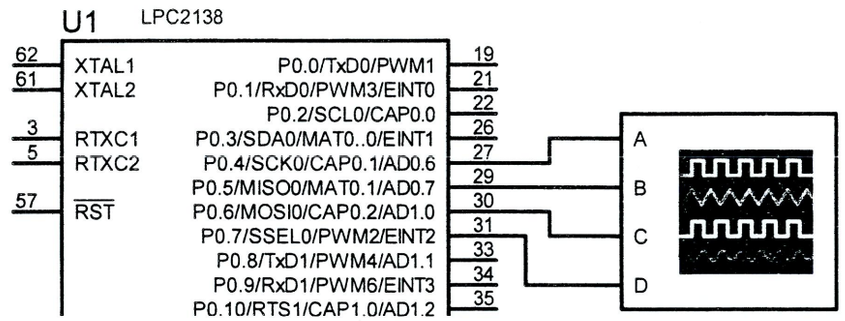
```
        ++count;
    } while ( k != -1);
    printf("\n");
    return count;
}
```

五、程序设计题：本大题共 1 小题，共 8 分。

38．编写一个函数 odd，计算一个正整数的奇数位数字的和，并返回该正整数的位数。例如正整数 123456 是一个 6 位数，奇数位数字分别为 6、4、2，和为 12。若有定义“int sum;”，调用函数 odd(123456, &sum)后，sum=12，返回值是 6。

六、分析题：本大题共 2 小题，每小题 10 分，共 20 分。

39．如题 39 图所示为 LPC2138 微控制器电路，用来输出不同占空比的矩形波形。电路的晶振频率为 24MHz。试阅读下述程序，回答问题，将编号①~⑩处空缺的内容填写在答题卡上。



题 39 图

```
/* Main.c */
#include <LPC2138.h>
#define   POUT    (uint32)((1 << 4) | (1 << 5) | (1 << 6) |(1 << 7))
uint32 ms_count=0;
void   __irq TIMER0_ISR (void)        /*  中断服务程序  */
{
    ms_count++;
    if ((ms_count % 200) == 0)
        IO0PIN = (IO0PIN & 0xFFCF)|(1 << 4);
    if ((ms_count % 200) == 50)   {
        IO0CLR = (1 << 4);
        IO0SET = (1 << 5);
    }
```

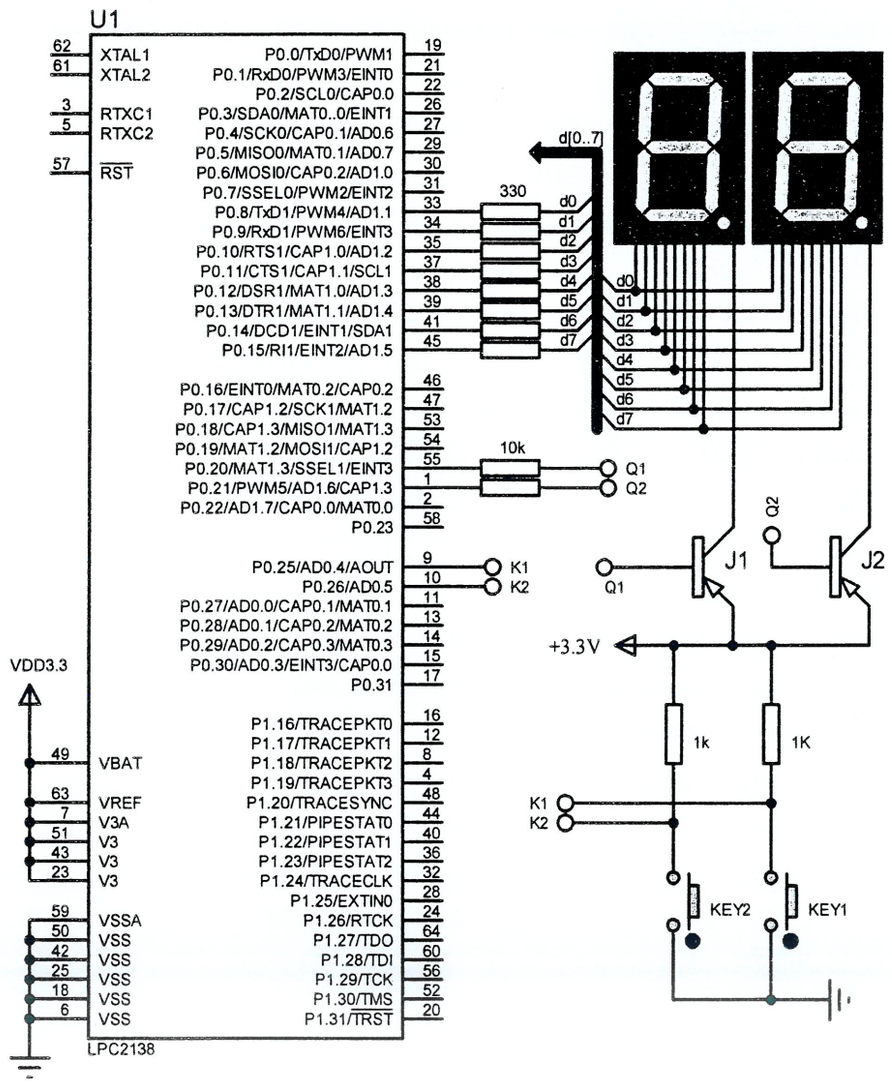
```
    if ((ms_count % 500) == 0)
        IO0PIN = (IO0PIN & 0xFF3F)|(1 << 6);
    if ((ms_count % 500) == 200)
        IO0CLR = (1 << 6);
    if ((ms_count % 500) == 400)
        IO0SET = (1 << 7);
    T0IR = 0x01;
    VICVectAddr = 0x00;
}
void   Time0Init(void)
{
    T0TC     = 0;              /* 定时器 T0 初始化 */
    T0PR = 99;                 /* 设置定时器 0 分频为 100 分频，得 300000Hz */
    T0MCR = 0x03;              /* 匹配通道 MR0 匹配中断并复位 TC */
    T0MR0 = 300;               /* MR0 比较值 */
    T0TCR = 0x03;              /* 启动并复位 T0TC */
    T0TCR = 0x01;              /* 使能定时器 0 */
    VICIntSelect = (1<<4);
    VICIntEnable = (1<<4);
}
int   main(void)
{
    PINSEL0 = 0x00;
    IO0DIR    = POUT;
    Time0Init();
    IO0CLR = POUT;
    while(1);
}
```

(1) 上述程序将编号为___①___的定时器设置为___②___中断；通过将中断选择寄存器的第___③___位设置为___④___来完成该中断的设置。

(2) 程序运行后定时器 0 每隔___⑤___ms 产生一次中断。

(3) 程序运行后 P0.4 引脚输出波形的频率为___⑥___Hz；P0.5 引脚输出波形的占空比为___⑦___；P0.6 引脚输出波形的频率为___⑧___Hz，占空比为___⑨___；P0.7 引脚输出波形的占空比为___⑩___。

40．题 40 图所示为 LPC2138 微控制器构成的键控数码管电路，按键按下后立即释放。试阅读下述程序，回答问题，将编号①~⑩处空缺的内容填写在答题卡上。



题 40 图

```
#include <LPC2138.h>
typedef unsigned long uint32_t;
typedef unsigned char uint8_t;
typedef union{
    uint32_t data;
    struct {
        uint32_t    b0:8, b1:8, b2:8, b3:8;
```

```
    } bytes;      // 按字节操作
    struct {
        uint32_t  d0:1, d1:1, d2:1, d3:1, d4:1, d5:1, d6:1, d7:1;
        uint32_t  d8:1, d9:1, d10:1, d11:1, d12:1, d13:1, d14:1, d15:1;
        uint32_t  d16:1, d17:1, d18:1, d19:1, d20:1, d21:1, d22:1, d23:1;
        uint32_t  d24:1, d25:1, d26:1, d27:1, d28:1, d29:1, d30:1, d31:1;
    } field;      // 按位操作
} reg_bits_t, reg_io_pin_t, reg_io_set_t, reg_io_clr_t, reg_io_dir_t;
#define  rIO0PIN   (*((volatile reg_io_pin_t *) 0xE0028000))
#define  rIO0SET   (*((volatile reg_io_set_t *) 0xE0028004))
#define  rIO0DIR   (*((volatile reg_io_dir_t *) 0xE0028008))
#define  rIO0CLR   (*((volatile reg_io_clr_t *) 0xE002800C))
#define  Q1 20                    // 控制数码管的十位
#define  Q2 21                    // 控制数码管的个位
#define  KEY1  25
#define  KEY2  26
#define  gpio_bit(reg, g, i)   rIO##g##reg.field.d##i
#define  gpio_byt(reg, g, i)   rIO##g##reg.bytes.b##i
#define  set_seg(code)   gpio_byt(PIN, 0, 1) = code;
#define  lighten(bit)   gpio_bit(CLR, 0, bit) = 1;
#define  darken(bit)   gpio_bit(SET, 0, bit) = 1;
uint8_t disp_code[] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};
                                  // 数字 0~9 的显示码
void DelayMs(unsigned int t);     // 延时函数，延时 t ms。
int IsKeyPressed(volatile int key_state);
void InitRegs(void);              // 寄存器初始化，设置引脚 P0.25~P0.26 为 GPIO 输入；
                                  // 设置引脚 P0.8~P0.15, P0.20~P0.21 为 GPIO 输出。
void DisplayNum(int n);
int main(void)
{
    int keypressed, n = 0;
    InitRegs();
    darken(Q2);
    darken(Q1);
```



```
    while(1) {
        keypressed = IsKeyPressed();
        if(keypressed == KEY1)    n = (++n) % 100;
        if(keypressed == KEY2) {
            n += 10;
            n %= 50;
        }
        DisplayNum(n);    //显示计数值
    }
}
void DisplayNum(int n)        //显示数码管的个位和十位
{
    lighten(Q2);
    darken(Q1);
    set_seg(disp_code[n%10]);   //显示个位
    DelayMs(50);
    darken(Q2);
    lighten(Q1);
    set_seg(disp_code[n/10]);   //显示十位
    DelayMs(50);
}
int IsKeyPressed(void) {
    int key;
    if(((gpio_bit(PIN,0,25)==0)|(gpio_bit(PIN,0,26)==0))) {
        DelayMs(10);
        if(((gpio_bit(PIN,0,25)==0)|(gpio_bit(PIN,0,26)==0))) {
            if(gpio_bit(PIN,0,25)==0)    key=(int) KEY1;
            else        key=(int) KEY2;
        }
        while(((gpio_bit(PIN,0,25)==0)|(gpio_bit(PIN,0,26)==0))) ;
    }
    else    key=0x0;
    return key;
}
```

（1）电路中三极管 J1 和 J2 的类型是___①___（PNP/NPN），它们的作用是控制数码管的___②___。电路中两个数码管的类型是___③___（共阴极/共阳极）结构，如果显示码为 0x99，则显示的数字为___④___。

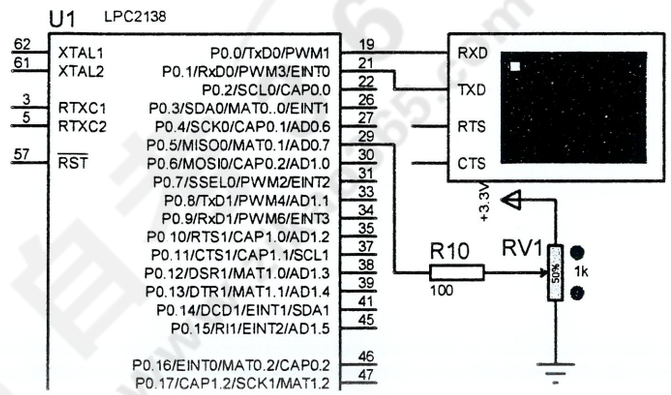
（2）程序运行后，先点按 KEY1 键 21 次，两个数码管从左到右显示的数值为___⑤___，接着点按 KEY2 键 3 次，两个数码管从左到右显示的数值为___⑥___。

（3）程序运行后，两个数码管能够显示的最大数值为___⑦___。

（4）没有按键时，执行函数 IsKeyPressed()后的返回值为___⑧___；点按 KEY1 键，执行函数 IsKeyPressed()后的返回值为___⑨___；点按 KEY2 键，执行函数 IsKeyPressed()后的返回值为___⑩___。

七、应用题：本大题共 1 小题，共 5 分。

41。题 41 图所示为 LPC2138 微控制器的 AD 采样电路，采样结果通过 UART0 输出，UART0 波特率为 9600，8 位数据位，1 位停止位，无奇偶校验。试完善下述程序。将编号①~⑤处空缺的内容填写在答题卡上。



题 41 图

```
/* Main.c */
#include <LPC2138.h>
#include <stdio.h>
typedef union                          // AD 控制寄存器
{
    uint32 data;
    struct{
        uint32 sel      :8;
        uint32 clkdiv   :8;
        uint32 burst    :1;
        uint32 clks     :3;
        uint32          :1;
        uint32 pdn      :1;
        uint32          :2;
        uint32 start    :3;
        uint32 edge     :1;
        uint32          :4;
    } field;
} reg_ad_cr;
typedef union                          //AD 数据寄存器
{
    uint32 data;
    struct{
        uint32          :6;
        uint32 result   :___①___;
        uint32          :8;
        uint32 chn      :3;
        uint32          :3;
        uint32 overrun  :1;
        uint32 done     :1;
    } field;
} reg_ad_gdr;
#define rAD0CR    (*((volatile reg_ad_cr *) 0xE0034000))
#define rAD0GDR (*((volatile reg_ad_gdr *) 0xE0034004))
const uint32 Fpclk = 11059000;
void UART0SendString(char *str);       //串口发送字符串 str
void Delay(uint32 dly);                //延时函数
void UART0Init()
 {
    U0LCR = 0x80;
    U0DLM = 0;
    U0DLL = 72;
    U0LCR = 0x03;
 }
```

```
int main(void)
{
    uint32 adc_data;
    char message[20] = {0};
    reg_ad_cr cr;
    uint8 chn;
    PINSEL0 = 0x00000005;     // 设置 P0.0 和 P0.1 引脚
    UART0Init();
    PINSEL0 |= 0x00000C00;    // 设置 P0.5 引脚
    cr.field.sel = 1<<___②___;
    cr.field.clkdiv = Fpclk/1000000 - 1;
    cr.field.burst = 0;
    cr.field.clks = 0;
    cr.field.pdn = 1;
    cr.field.start = 1;
    rAD0CR = ___③___;
    while(1)
    {
        while(rAD0GDR.field.done == 0);
        adc_data = rAD0GDR.field.result;
        chn = rAD0GDR.field.___④___;
        adc_data = adc_data * 3300;
        adc_data = adc_data / 1024;
        sprintf(message,"AD0.%d:%5d mV\r\n",chn,adc_data);
        UART0SendString(message);
        Delay(10);
        rAD0CR.field.___⑤___ = 1;
    }
}
```