MSP430 中 Timer_A and Timer_B 的 UART 运用
Uart application of Timer_A and Timer_B in MSP430

丁鹏飞 徐国军
Ding Pengfei Xu Guojun

(西安邮电学院 044# 西安 710061)
(Xi'an Institue of Post and Telecomunication,Xi'an,710061)

摘 要

自 1996 年 MSP430 十六位单片机问世以来，它的低功耗性能及丰富的片内资源受到各方面的好评，本文针对 MSP430F13x 及 MSP43014X 系列单片机中的定时器进行介绍,利用定时器 A(Timer_A)和定时器 B(Timer_B)中的捕获比较寄存器来开发多个串行通信口，使十六位单片机在通信领域发挥更大的潜力。

关键词 Timer_A Timer_B 捕获/比较寄存器 UART 口

一、 概述

MSP430 具有丰富的外围模块，如 MSP430F149 就包含：12 位 A/D，精密模拟比较器，硬件乘法器，2 组频率可达 8MHZ 的时钟模块，2 个可以实现异步、同步及多址访问的串行通行接口，采用了超低功耗技术，可以进行在线调试与编程，其指令周期可达 125ns。MSP430F14X 系列目前在市场的售价大约为 60 元人民币，与其它单片机相比，MSP430 具有更高的性价比和优越性，适合做测控、通讯等嵌入系统。本文介绍利用 MSP430 中的定时器解决多串口通信。

目前美国德州仪器所出的所有 FLASH 单片机都含有 Timer_A，而在 MSP430F13X 系列和 MSP43014X 系列中既含有 Timer_A 也含有 Timer_B,在 F13X 中有一个带有 3 个捕获/比较模块的 Timer_B，在 F14X 中有一个带有 7 个捕获/比较模块的 Timer_B,它们均是扩展 UART 口的核心。

Timer_A 和 Timer_B 都是非常有用的定时器，Timer_A 具有以下特点：

（1） 十六位计数器，有四种工作模式（停止、增计数、连续、增/减计数）
（2） 可以选择计数器时钟源(外设的，内置的快慢速均可)
（3） 三个具有可配置输入端的捕获/比较寄存器(具有自动锁存功能)
（4） 可用于串行通信

与 Timer_A 相比，Timer_B 可进行 8、10、12、16 位计数，但在 Timer_B 中未实现锁存功能。Timer_A 和 Timer_B 可支持同时进行的多时序控制、多个捕获/比较功能、多种输出波形（PWM 波形）,也可以是上述功能的组合。

二、 可行性分析

首先，收和发是针对不同的定时器，它们的中断源不同，中断标志会记住不同中断。其次，同一定时器的不同捕获寄存器（不包括 CCR0）的中断标志有优先级、共用一个中断向量的中断标志，中断向量寄存器（TAIV 或 TBIV）用于确定产生中断请求的中断源，所以当出现同发或者同收现象时，可以根据中断向量寄存器中的内容来确定具体的中断操作，在进入一个中断后，中断向量寄存器会保存另外中断源的中断向量，直到前一中段结束而执行该中断，即不会丢失另外的中断。

基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

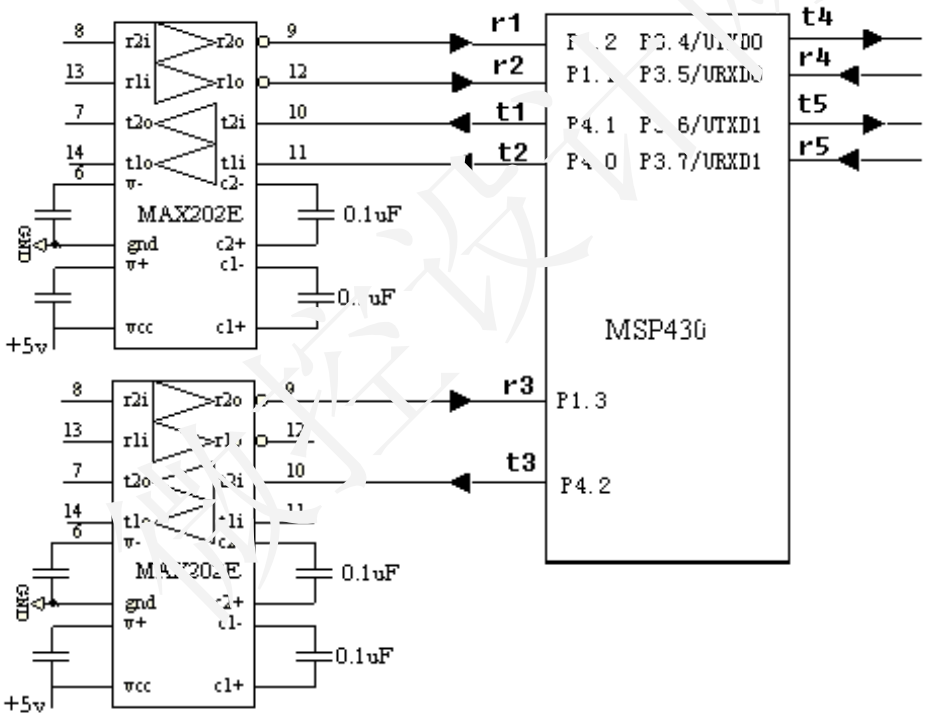
三、 系统原理

计算机进行串行通信的高电平为**-5~-15V**，低电平为 **5~15V**，而从 **MSP430** 单片机输出的信号的高电平为 **3V**，低电平 **0V**，要实现 **MSP430** 与计算机通信必须进行电平转换。实现这一转换，可以利用 **MAX202E** 来实现。

发送特性的实现是用比较功能将数据从输出单元的引脚移出，波特率是用定时器定时产生中断来实现的。

接收特性的实现是利用定时器的捕获功能实现接收数据的检测，当检测到起始位时，将定时器设置为比较模式，接收的位被 **EQUx** 信号自动锁存。这样就不会因 **CPU** 无法及时响应而造成接收错，提高了接收控制的灵活性，从而 **CPU** 因不需等待接收而提高其利用率。

以 **MSP430F149** 为例，**Timer_A** 的捕捉/比较寄存器具有将输入信号锁存的功能，故利用它实现输入信号的准确接收；而 **Timer_B** 捕捉/比较寄存器不具有将输入信号锁存的功能，不宜将其作为串口的输入，本例利用它控制串口的输出。因 **Timer_A** 只有三个捕捉/比较寄存器，故利用它们可以开发出三个 **UART** 口，加上其本身的两个 **UART** 口，**MSPF149** 最终可有五个 **UART** 口，硬件电路简图如图（**1**）（其中，**UTXD1**、**URXD1** 的电平转换与其它几个 **UART** 的 硬件连接 一样）



图（1）

四、主程序工作过程及软件框图

用 **Timer_A** 和 **Timer_B** 要实现多个 **UART** 口，合理选取 **Timer_A** 和 **Timer_B** 的工作模式非常重要。经分析，我们选取 **Timer_A** 和 **Timer_B** 工作在连续记数模式。在主程序中，需要发送数据时，由主程序调用发送子程序（其主要作用是初始化用于发送数据的 **Timer_B**，经过初始化以后，由定时器定时产生中断来发送数据）。而接收模块初始化以后，接收数据的起始位的检测由定时器的捕获功能自动实现，不需软件控制，当捕获器检测到起始位以后由接收中断服务子程序实现数据的接收，其软件框图如下：

基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

以 Timer_A 的 CCR0 的接收、Timer_B 的 CCR0 的发送为例，其中断服务子程序分别如下：

1、 发送中断服务子程序：

```
TX_LOOP
       ADD    #Bitime,&TBCCR0
       BIT     #0001H,R5
       JNZ     TX1
       BIC     #OUT,&TBCCTL0
       JMP     TX2
TX1    BIS     #OUT,&TBCCTL0
TX2    SUB    #01H,R6
       JNZ     TX3
       BIC     #CCIE,&TBCCTL0
TX3    RRA    R5
RETI
```

2、 接收中断服务子程序：

```
RX_LOOP
      CMP    #0009H,R8
      JNZ     RX_START
      BIC     #CAP,&CCTL0
      MOV    &TAR,&CCR0
      ADD    #Bitime1_5,&CCR0
      MOV    #0000H,R7
      DEC     R8
      JMP     RXEND
RX_START
      BIT     #SCCI,&CCTL0
      RRC.B   R7
      SUB     #01H,R8
      JNZ     RXEND1
      BIC     #CCIE,&CCTL0
RXEND1
      ADD    #Bitime,&CCR0
RXEND
      RETI
```

利用 Timer_A 和 Timer_B 实现 UART 功能的完整调试程序见附页：（在程序中，考虑到接收与发送的多个 UART 的中断具有相似之处，我们在程序中仅对两个发送口和两个接收口进行了调试，经调试，证实了利用 Timer_A 和 Timer_B

基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

可以实现多个 UART 口功能。

五、对 Timer_A 和 Timer_B 实现的 UART 口的实时性、误差分析：

1. 接收和发送的实时性：

在发送数据时，如果定时器的计数时钟频率比较高，例如，当定时器的计数时钟频率为 8MHZ，波特率为 9600 时，接收/发送相邻比特的间隔为 833 个时钟周期，即使延时几十个时钟周期（按理论，只要小于 250 个时钟周期即可）发送下一个比特也能正确接收该字节。在接收数据时，由于定时器具有自动锁存功能，对数据位的接收只要在下一位数据到来之前接收即可。由此可见，用定时器实现的 UART 口完全可以实现实时性，这完全可以用于对实时性要求比较高的系统。

2. 接收和发送一个字节的 CPU 占用率：

由于是利用中断来唤醒接收和发送中断服务的，在接收和发送的位间隔时间中 CPU 可以继续执行其他程序。CPU 占用率可用以下公式计算：

CPU 占用率=（发送或接收一位总的占用时钟周期*波特率）/CPU 频率

经计算，发送和接收一个字节的 CPU 占用率（以最高时钟频率 8MHZ、波特率为 9600 为例）分别约为：3.31%和 3.39%。

3. 时间误差分析：

对于一般的 UART 口来说，由于字节间存在累积误差，因此，对波特率的要求较高，需要时钟频率非常接近波特率的整数倍。而实际波特率往往是通过时钟分频得到，无法产生精确的波特率，不可避免地会扩大累积误差。对于利用定时器实现的 UART 口来说，在发送或接收下一个字节时，定时器重新开始该字节内的计数，因此，字节之间不存在累积误差，仅存在字节内的累积误差，因此，这大大降低了时钟接近波特率的整数倍的要求，在定时器的计数时钟频率比较高（为波特率的几百倍时）时，时钟的选择将不受限制。

在定时器实现的 UART 口中，当定时器的计数时钟频率比较高时，字节内产生的累积误差也非常小。例如，当定时器的计数时钟频率为 8MHZ，波特率为 9600 时，要求定时器每计数 833.333 时发送一个比特，然而，定时器只能计整数，因此选择每计数 833 时发送一个比特。在这种情况下，每一位所产生的误差只有千之几，这样的误差较一般的 UART 口来说是非常小的。

4. 与一般的 UART 口的性能比较：

UART 口的实现在各种不同类型的微处理机系统中都不同，可能是用通用的 I/O 端口以等待的方式用软件实现位的处理，这样的处理方式需要极大的 CPU 开销，因此增大了功耗，也降低了 CPU 的可用性。而用定时器实现的 UART 口，在发送或接收时的位间隔期间，CPU 可以继续执行其他的程序，加上 MSP430 单片机自身的特点，它甚至可以在超低功耗模式下接收和发送。

六、结束语

MSP430 单片机的定时器功能强大，中断源较多，可以任意嵌套，它给工程开发人员提供了较多的选择余地。上面仅介绍了用 TIMER_A、TIEMR_B 来扩充 UART 口，开发人员只要熟悉定时器的工作原理，就可以根据自己的需要，从

基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

多种方法中选出最优方案。MSP430 单片机所具有的较高性价比及优越性必将使其成为众多单片机中出色的一员。

参考文献

1. 胡大可．《MSP430 系列 FLASH 型超低功耗 16 位单片机》．北京航空航天大学出版设，2001
2. 《MSP430 系列软件用户指南》．利尔达（中国）电子有限公司
3. 《MSP430 ASSERMBLER,LINKER,AND LIBRARIAN PROMGRAMING》．利尔达（中国）电子有限公司
4. 邬宽明．《单片机外围器件使用手册－数据传输接口器件分册》．北京航空航天大学出版设

完整的调试程序源代码

```
#include   "msp430x14x.h"
       ORG     0FFFEH
       DW      MAIN           /*设置上电起始地址*/
       RSEG    UDATA0
       DS      0
Bitime      EQU    209H    /* 发送比特的时间间隔，时钟平率为 5.030MHZ，波特率为 9600 比特/秒*/
Bitime1_5   EQU    30EH    /* 起始位与第一个比特的时间间隔*/
TX_Count    EQU    000AH   /*发送数据的计数值*/
RX_Count    EQU    0009H   /*接收数据的计数值*/
CHECKDATA   EQU    0300H   /*存储通道 1（CCR0）接收数据的起始地址*/
CHECKDATA1 EQU    0500H   /*存储通道 1（CCR1）接收数据的起始地址*/
       RSEG    CSTACK         /*定义堆栈段*/
       DS      0
       RSEG    CODE           /*代码段开始*/
       DS      0
MAIN
     MOV      #(WDTPW+WDTHOLD),&WDTCTL   /* 关闭看门狗*/
INITSYS
       MOV      #SFE(CSTACK),SP
       MOV.B    #(RSEL2+RSEL1+RSEL0),&BCSCTL1     /*设置基础时钟模块*/
       MOV.B    #(DCO2+DCO1+DCO0),&DCOCTL
       BIC.B    #OSCOFF,SR
       PUSH     #0FFFFH                                /*基础时钟的调整时钟*/
LOOP
       DEC      0(SP)
       JNZ      LOOP
       MOV.B    #SELM_1,&BCSCTL2                  /* 选择系统时钟源*/
SetupTA         MOV    #(TASSEL_2+MC1),&TACTL      /*设置定时器 A 的时钟源及工作模式*/
                MOV    #(TBSSEL_2+MC1),&TBCTL      /*设置定时器 B 的时钟源及工作模式*/
SetupC0         MOV    #OUT,&TBCCTL0                /*通道 1 发送空闲数据*/
                MOV    #OUT,&TBCCTL1                /*通道 2 发送空闲数据*/
Setupp1_2       MOV.B    #00H,&P1DIR   /*设置定时器 A 和定时器 B 输入/出数据的管脚*/
                BIS.B    #06H,&P1SEL   /*P1.1 和 P1.2 用于接收数据*/
```


基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

```
                BIS.B   #03H,&P4DIR   /*P4.0 和 P4.1 用于发送数据*/
                BIS.B   #03H,&P4SEL
Delay           PUSH    #00FFH        /*用于延时*/
Delay2          DEC     0(SP)
                JNZ     Delay2
                INCD    SP
                MOV     #GIE,SR       /*开中断*/
/*--------------调试发送、接受模块的功能----------------------*/
        MOV     #CHECKDATA,R14     /*设置接收通道 1 数据的起始地址*/
        MOV     #CHECKDATA1,R4     /*设置接收通道 2 数据的起始地址*/
        MOV     #35H,R5            /*R5、R9 用于存储发送的数据*/
        MOV     #39H,R9            /*发送数据可以放于 RAM 中，或用变量代替 R5，R9*/
        MOV     #00H,R10
        CALL    #RX_Ready          /*开启接收端口*/
        CALL    #TX_Byte           /*发送 2 个通道的可以同时开启，但这里先开启通道 1*/
TEST_TX12
        CMP     #00H,R6            /*判断通道 1 发送是否完毕，可以在中断服务程序中设置发送完标
志*/
        JNZ     TEST_TX22
        MOV     #0035H,R5          /*发送的数据没有设置奇偶校验位，但可以根据情况设置*/
        CALL    #TX_Byte           /*发送完毕继续发送以便于调试发送数据是否正确*/
TEST_TX22
        CMP     #00H,R10           /*判断通道 2 发送是否完毕，可以在中断服务程序中设置发送完标志
*/
        JNZ     TEST_RX12
        MOV     #0039H,R9          /*发送完毕继续发送以便于调试发送数据是否正确*/
        CALL    #TX_Byte1
        JMP     TEST_RX12
TEST_RX12
        CMP     #00H,R8            /*判断通道 1 数据是否接收完*/
        JNZ     TEST_RX22
        MOV     R7,R15             /*将接暂时数据存储于 R15 中，也可以利用 RAM 来存储*/
        MOV     #RX_Count,R8       /*接收控制*/
        MOV     #(CCIE+CAP+CCIS_0+CM_2+OUT),&CCTL0   /*准备接受下一个数据*/
LOAD    MOV.B   R15,0(R14)     /*将接收数据存储在 RAM 中，用于调试接收是否正确，可以奇偶校验来
判断*/
        INC     R14            /*R14 用于存储接收数据存放的地址，可以用变量代替*/
TEST_RX22
        CMP     #00H,R12           /*判断通道 2 数据是否接收完*/
        JNZ     TEST_TX12
        MOV     R11,R13            /*将接暂时数据存储于 R13 中，也可以利用 RAM 来存储*/
        MOV     #RX_Count,R12   /*接收控制*/
        MOV     #(CCIE+CAP+CCIS_0+CM_2+OUT),&CCTL1
LOAD1 MOV.B   R13,0(R4)        /*将接收数据存储在 RAM 中，用于调试接收是否正确，可以奇偶校验来
判断*/
        INC     R4             /*R4 用于存储接收数据存放的地址，可以用变量代替*/
```


基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

```
        JMP      TEST_TX12
/*-----------调试代码段结束------*/

/***********发送数据子程序*****************/
TX_Byte                                  /*通道 1 的发送中断服务子程序*/
      MOV      #TX_Count,R6              /*R6 用于存放发送的比特数，可以用变量代替*/
      BIS     #0FF00H,R5                /*用于设置停止位*/
      MOV      #(CCIE+CLLD_3),&TBCCTL0   /*设置定时器 B 的 CCR0 的工作模式*/
      MOV      #Bitime , &TBCCR0         /* 设置第一位的中断时间*/
      ADD      &TBR,&TBCCR0
      BIC     #OUT,&TBCCTL0             /*发送起始位*/
      ADD      #Bitime , &TBCCR0         /* 下一位的接收时间*/
      RET
/*CCR1，CCR2 的中断共用一个地址，如果 CCR2 用于模块功能，必须首先判断是否为 CCR1 产生的中断
*/
TX_Byte1                                 /*通道 2 的发送中断服务子程序*/
      MOV      #TX_Count,R10             /*发送控制，R10 用于存储通道 2 的发送比特位的个数*/
      BIS     #0FF00H,R9
      MOV      #(CCIE+CLLD_3),&TBCCTL1   /*设置定时器 B 的 CCR1 的工作模式*/
      MOV      #Bitime , &TBCCR1         /* 设置第一位的中断时间*/
      ADD      &TBR,&TBCCR1
      BIC     #OUT,&TBCCTL1             /*发送起始位*/
      ADD      #Bitime , &TBCCR1         /* 下一位的接收时间*/
      RET
/*---------准备接受数据-----------------------------*/
RX_Ready
      MOV      #(TASSEL_2+MC_2),&TACTL   /*设置定时器 A 的工作模式*/
      MOV      #RX_Count,R8      /*R8，R12 用于存储接收的比特数，可以用变量代替*/
      MOV      #RX_Count,R12                    /*接受控制*/
      MOV      #(CCIE+CAP+CCIS_0+CM_2+OUT),&CCTL0 /* 设置定时器 A 的 CCR0 的工作模式*/
      MOV      #(CCIE+CAP+CCIS_0+CM_2+OUT),&CCTL1  /* 设置定时器 A 的 CCR1 的工作模式*/
      MOV      #0000H,R11                       /* R11 用于存储 CCR1 接受的数据*/
      MOV      #0000H,R7                        /* R7 用于存储通道 1 接受的数据*/
      RET
/*---------------发送中断服务子程序（CCR0）--------------------*/
TX_LOOP                          /*通道 1 的发送中断服务子程序*/
      ADD     #Bitime , &TBCCR0
      BIT     #0001H,R5          /*判断发送位为 1 还是 0*/
      JNZ     TX1
      BIC     #OUT,&TBCCTL0     /*发送起始位*/
      JMP      TX2
TX1   BIS     #OUT,&TBCCTL0
TX2   SUB      #01H,R6           /*发送比特数减 1*/
      JNZ     TX3
      BIC     #CCIE,&TBCCTL0
TX3   RRA      R5                /*发送数据存储于 R5 中，可以用变量代替，必须与主程序中的设置一
```


基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

```
致*/
        RETI
/*----------------发送中断服务子程序（CCR1）----------------------*/
TX_LOOP1                            /*通道 2 的发送中断服务子程序*/
        ADD     #Bitime , &TBCCR1
        BIT     #0001H,R9           /*判断发送位为 1 还是 0*/
        JNZ     TX11
        BIC     #OUT,&TBCCTL1       /*发送起始位*/
        JMP     TX21
TX11 BIS    #OUT,&TBCCTL1
TX21 SUB    #01H,R10                /*发送比特数减 1*/
        JNZ     TX31
        BIC     #CCIE,&TBCCTL1
TX31 RRA    R9                      /*发送数据存储于 R9 中，可以用变量代替，必须与主程序中的设置一致
*/
        BIC     #CCIFG,&TBCCTL1     /*中断完，关闭中断标志位*/
        RETI
/*----------------接受中断服务子程序（CCR0）----------------------*/
RX_LOOP                             /*通道 1 的接收中断服务子程序*/
        CMP     #0009H,R8           /*判断接收位数*/
        JNZ     RX_START
        BIC     #CAP,&CCTL0         /*接收到起始位，将定时器设置为比较模式*/
        MOV     #Bitime_5, &CCR0    /* 设置第一位的接收时间*/
        ADD     &TAR,&CCR0
        MOV     #0000H,R7           /*将接收存储器清除，准备接收数据*/
        DEC     R8
        JMP     RXEND
RX_START
        BIT     #SCCI,&CCTL0        /*判断接收位为是 0 还是 1*/
        RRC.B   R7
        SUB     #01H,R8
        JNZ     RXEND1
        BIC     #CCIE,&CCTL0        /*接收完，关闭中断允许位，开启在主程序*/
        JMP     RXEND
RXEND1
        ADD     #Bitime , &CCR0     /     * 设置下一位的接收时间*/
RXEND
        RETI
/*----------------接受中断服务子程序（CCR1）----------------------*/
RX_LOOP1                            /*通道 1 的接收中断服务子程序*/
        CMP     #0009H,R12          /*判断接收位数*/
        JNZ     RX_START1
        BIC     #CAP,&CCTL1         /*接收到起始位，将定时器设置为比较模式*/
        MOV     #Bitime_5, &CCR1    / * 设置第一位的接收时间*/
        ADD     &TAR,&CCR1
        MOV     #0000H,R11          /*将接收存储器清除，准备接收数据*/
```


基于模拟前端信号处理与控制技术的专业论坛、网站. [MSP430 单片机、虚拟综合示波器] 电话:0755-81808400

```
        DEC     R12
        JMP     RXEND10
RX_START1
        BIT     #SCCI,&CCTL1            /*判断接收位为是 0 还是 1*/
        RRC.B   R11
        SUB     #01H,R12
        JNZ     RXEND11
        BIC     #CCIE,&CCTL1            /*接收完，关闭中断允许位，开启在主程序*/
RXEND11
        ADD     #Bitime , &CCR1         /       * 设置下一位的接收时间*/
RXEND10
        BIC     #CCIFG,&CCTL1           /*中断完，关闭中断标志位*/
        RETI
/*----------列中断向量表----------------*/
        COMMON INTVEC
         ORG    TIMERB0_VECTOR
          DW      TX_LOOP
         ORG    TIMERB1_VECTOR
          DW      TX_LOOP1
         ORG    TIMERA0_VECTOR
          DW      RX_LOOP
         ORG    TIMERA1_VECTOR
          DW      RX_LOOP1
         END
```

说明：该程序用于测试利用定时器实现 **UART** 口的可行性，接收数据直接通过存储器所接收的数据来判断接收的正确性，在整个调试过程中，我们用超级终端随机发送数据，利用示波器观察 **UART** 发送的数据，通过在线调试来观察接收的数据是否正确。因此，没有考虑寄存器的有限性、数据的奇偶位。在实际运用系统中，可以用存储器来代替某些寄存器，也可以扩充存储空间和判断奇偶位。

通过反复的调试，用定时器扩充的 **UART** 口可以正确无误的进行接收/发送。由于该程序仅用于调试用定时器扩充的 **UART** 口的可行性，在实际运用中可以根据具体情况做相应改动。

利用定时器扩充的另外的一个 **UART** 口与其它两个类似，在扩充另外一个 **UART** 口时，必须考虑 **CCR1** 与 **CCR2** 共用一个中断向量的中断标志。

作者简介： 丁鹏飞 男，西安邮电学院电子与信息工程系本科生
徐国军 男，西安邮电学院通信工程工程系本科生
电话号码： 丁鹏飞（**029-5384124**），徐国军（**029-5384044**）