Mikroprozessorsysteme
Praktikum 6

Wasserspender

Von: Aria Gholami & Yousef Ebrahimzadeh Attari

Inhaltsverzeichnis

Inhalt

1 Installationsanleitung

1.1 Kopieren

Kopieren Sie bitte die folgenden Dateien auf Ihr System:

Ordner „h“:

- aic.inc & aic.h
- ebi.inc
- pio.inc & pio.h
- pmc.inc & pmc.h
- ser_io.h
- std_c.h
- tc.h
- ustart.inc & ustart.h

Ordner „boot“:

- boot.S
- boot_flash.S
- boot_ice.S
- swi.S

Ordner „Termin 6“:

- boot_ice.o
- makefile
- ser_io.o & ser_io.S
- seriell.c & seriell.h & seriell.o & seriell.S
- swi.c & swi.o
- Termin6.c

Und stellen Sie bitte sicher, dass auf Ihren System unter dem Verzeichnis (/opt/arm-eb63-elf/lib/gcc/arm-eb63-elf/4.4.6/libgcc.a) die Datei libgcc.a vorhanden ist.

1.2 Einrichten des Projektes

1.) Starten Sie auf Ihren Computer das Terminal

2.) Tippen Sie ,,snavigator'' ein

3.) Wählen Sie im Source Navigator ,,Browse''

4.) Wählen Sie in dem von Ihnen gewählten Verzeichnis: ,,Termin6/Termin6.proj'' aus

5.) Wechseln Sie wieder zum Terminal

6.) Tippen Sie ,,minicom'' ein

7.) wechseln Sie wieder zu dem Source Navigator

8.) Wählen Sie nun im Reiter: ,,Tools/Debug'' aus

9.) Wählen Sie als die zu Debuggende Datei ,,/Termin6Komplett.elf'' aus in dem von Ihnnen gewählten Pfad aus

10.) Wählen Sie im Reiter ,, File/Connect to Target'' aus

11.) Wählen Sie bei:

- Target: ,,Remote/TCP''

- Hostname: ,,141.100.52.XX'', fügen Sie bei XX bitte noch die Adresse ihres Systems ein

- Port: 2001

12.) Bestätigen Sie ihre Auswahl mit ,,OK''

13.) Klicken Sie nun auf das Feld links ,,Run'' und danach ,,Continue'', um das System zu starten(Stellen Sie sicher, dass keine Breakpoints gesetzt wurden)

1.3Projekt beschädigt/verändert

1.) Starten Sie auf Ihrem System die Kommandozeile

2.) Tippen Sie ,,snavigator'' ein

3.) Wählen Sie im Source Navigator ,,New Project...''

4.) Wählen Sie:

- Project File: ,,.../Termin6/Termin6.proj'' in dem von Ihnen gewählten Verzeichnis

- Add Directory: ,,.../Termin6'' in dem von Ihnen gewählten Verzeichnis

5.) Überschreiben Sie die vorhandene Dateien

6.) Wählen Sie im Reiter ,,Tools/make'' und erzeugen Sie die ,,Termin6Komplett.elf'' neu

1.4 User Definierte Werte Einstellen

Daten, die Sie verändern können:

- BAUDRATE: Berechnen Sie anhand der vorgegebenen Formel/Beispiel aus dem Kommentar ,,//define US_BAUD = (CLOCK_SPEED / (16 * (BAUD)) //25 Mhz / (16 * 38400) = 40,x -> 41 -> hex: 0x29'' in ,,seriell.c'' und tragen Sie den hexadezimalen Wert eine Codezeile tiefer bei #define US_BAUD'' ein
- ABFÜLLMENGE: ändern Sie in der Datei ,,Termin6Komplett.c'' den Wert von ,,voll'' auf den gewünschten Wert in Gramm

2 Benutzerhandbuch

2.1 Vorwort

In dieser Dokumentation wird Ihnen eine detaillierte Erklärung zur Funktion und Bedienung des Projekts „Ausschankstation“ zur Verfügung gestellt. Das Evaluationsboard, welches hierfür verwendet wird, ist das AT91EB63 von Atmel. Als Programmiersprachen kommen Assembler und Hochsprache C zum Einsatz. Das Ziel dieses Projektes ist es eine automatisierte Abfüllanlage zu programmieren, die eine zuvor vorgegebene Menge Wasser in einen Becher abfüllt.

Durch zwei Tasten am Mikrocontroller soll der Abfüllvorgang bedient werden. Diese Tasten können die Waage kalibrieren (SW1) und den Abfüllvorgang starten(SW2). Ein Terminal auf dem Bildschirm dient zum Informationsaustausch mit dem Benutzer.

2.2 Behälter wiegen

Das Wiegen beginnt, nachdem Sie einen Behälter auf die Waage gestellt haben und anschließend den Taster SW1 betätigt haben.

Folgende Fälle können eintreten:

- Behälter ist zu schwer: Das System gibt die Meldung aus, dass der Behälter zu schwer ist und weist darauf hin, dass ein passender Behälter aufgestellt werden soll.
- Kein Behälter auf der Waage: Das System gibt die Meldung aus, dass kein Behälter aufgestellt ist und weist darauf hin, dass ein passender Behälter aufgestellt werden soll.
- Behälter entspricht den Auflagen: Der Behälter wird gewogen und das Gewicht wird in Gramm auf dem Terminal angezeigt.

2.3 Behälter mit Wasser füllen

Der Abfüllvorgang wir nach dem Aufstellen eines geeigneten Bechers und dem Betätigen des Tasters SW2 gestartet. Es wird die von Ihnen vorgegebene Menge in Gramm abgefüllt.

Falls währrend dem Abfüllvorgang der Becher entnohmen oder auch angehoben wird, wird der Abfüllvorgang sofort unterbrochen und das System zeigt eine Fehlermeldung.

2.4 Behälter entnehmen

Das System meldet Ihnen die Abgeüllte Menge in Gramm und das der Behälter entnommen werden kann. Wenn Sie den Behälter entnommen haben, startet das System neu.

3 Programmanalyse

3.1 Termin6Komplett.c

int main(void): Als aller Erstes wird die Peripherie freigeschaltet und initialisiert. Anschließend wird zur Begrüßung ein Text ausgegeben. Danach wird die serielle Schnittstelle USART0 initialisiert und bereitgestellt. In Folge dessen bekommt der Benutzer eine ausführliche Anweisung zur Benutzung unserer Ausschankstation. Bevor die Waage anfängt zu wiegen, wird der Counter für die Waage initialisiert, angemacht und gestartet. Falls KEY1 gedrückt wird, wird sofort die Masse des Bechers erechnet und ausgegeben. Danach kann man KEY2 drücken, um Wasser in den Behälter zu geben. Davor wird überprüft, ob die Masse des Bechers sich geändert hat bzw. Ob sich die Masse des Bechers im Fehlertoleranzbereich noch verhält. Falls nicht wird die Pumpe gestoppt und eine Fehlermeldung ausgegeben. Falls doch beginnt der Pumpvorgang. Jedoch kann es zu Komplikationen kommen:

- Becherr wird weggenommen -> Pumpvorgang wird abgebrochen und eine Fehlermeldung wird ausgegeben
- Becher wird angehoben(Gewich ist um 15% geändert) -> Pumpe wird abgebrochen und eine Fehlermeldung wird ausgegeben
- das Gewicht des Bechers wird zu groß ->Pumpvorgang wird abgebrochen und eine Fehlermeldung wird ausgegeben

Damit es zu keinen Komplikationen kommt, wird der Becher regelmäßig gewogen, das Gewicht gespeichert und auf dem Terminal ausgegeben. Zum Schluss wird das Gewicht nochmal angezeigt. Der Benutzer wird anschließend aufgefordert den Becher wegzunehmen. Nachdem der Benutzer dies gemacht hat, wird das System resetet und das System startet neu.

void init_messung(void): Die Clock wird angeschaltet und gestartet

float MessungDerMasse(void): Periodendauer1(abs(capturediff1)) und Periodendauer2(abs(capturediff2)) werden berechnet und daraus wird das Gewicht des Bechers berechnet: m = 2000* ((Periodendauer1/Periodendauer2)-1)-0)

int MasseMitGenauigkeit(void): Masse des Bechers wird 3 mal berechnet, damit der Wert des Bechers genau bestimmt werden kann.

void pump_stop(void): Timer wird initialisiert und angeschaltet.

void pump_start(void): Mode vom Timer, Register A und B vom Timer werden initialisiert. Anschließend wird der Zaehler resetet und der Timer gestartet

void intOutput(int var): wandelt Gewicht des Bechers vom Integer zu char[] und gibt dieses char[] nacheinander auf dem Terminal aus.

3.2 seriell.c

int init_ser(): initialisiert die serielle Schnittstelle USART0

char putch(char Zeichen): Gibt, wenn möglich, ein Zeichen auf die serielle Schnittstelle aus und liefert das Zeichen wieder zurück, wenn eine Ausgabe nicht möglich ist.

3.3 swi.c

void SWIHandler(): Ruft anhand der Nummer des Software Interrupts die init_ser( ), putch ( ) oder getch ( ) auf .

3.4 ser_io.S

inits(): ruft init_ser mit einen Software Interrupt aus.

puts(char*): Sendet ein character Array/ string via putc( ) bis zur NULL terminierung an die Serielle Schnittstelle.

putc(char*): sendet einen character an die Serielle Schnittstelle. Sollte der Empfänger beschäftigt sein, wird der Vorgang wiederholt.

3.5 Code

3.5.1 Termin6Komplett.c:

```
void pump_stop(void)
{
  StructPIO* piobaseA = PIOA_BASE;        // Base address PIOA (for pump pins)
  piobaseA->PIO_CODR  = (1<<PIOTIOA3);     // Clear (set to low) to prevent permanent high
  piobaseA->PIO_PER   = (1<<PIOTIOA3);   // Port enable (switch to PIOB)
}
void init_messung(void){
    pmcbase->PMC_PCER = 0x06f80;
    piobaseA->PIO_PDR = 0x090;
    tcbase4->TC_CCR   = TC_CLKDIS;
    tcbase5->TC_CCR   = TC_CLKDIS;
    tcbase4->TC_CMR   = TC4_INIT;            //Channel Mode Register = siehe define
    tcbase5->TC_CMR   = TC5_INIT;            //Channel Mode Register = siehe define
    tcbase4->TC_CCR   = TC_CLKEN;
    tcbase5->TC_CCR   = TC_CLKEN;
    tcbase4->TC_CCR   = TC_SWTRG;
    tcbase5->TC_CCR   = TC_SWTRG;
    piobaseA->PIO_PDR = 0x090;
}
```

```
float MessungDerMasse(void){
    volatile int    captureRA1;
    volatile int    captureRB1;
    volatile int    capturediff1;
    volatile int    captureRA2;
    volatile int    captureRB2;
    volatile int    capturediff2;
    volatile float  Periodendauer1;
    volatile float  Periodendauer2;
    int temp = 0;
    temp = tcbase4->TC_SR & 0x40; // vor SWTRG TC_SR(timer control status register) lesen
    temp = tcbase5->TC_SR & 0x40;
    tcbase4->TC_CCR    =    TC_SWTRG;
    while (!( tcbase4->TC_SR & 0x40));
        captureRA1    = tcbase4->TC_RA;
        captureRB1    = tcbase4->TC_RB;
        capturediff1    =    abs(captureRB1) - abs(captureRA1);
        Periodendauer1 = abs(capturediff1);    // Zeit in us
    tcbase5->TC_CCR    =    TC_SWTRG;
    while (!( tcbase5->TC_SR & 0x40));
        captureRA2    = tcbase5->TC_RA;
        captureRB2    = tcbase5->TC_RB;
        capturediff2    =    abs(captureRB2) - abs(captureRA2);
        Periodendauer2 = abs(capturediff2);    // Zeit in us
    return (c1* ((Periodendauer1/Periodendauer2)-1)-c2 );
}
int MasseMitGenauigkeit(void){
    float erg = MessungDerMasse();
    erg = erg + MessungDerMasse();
    erg = erg + MessungDerMasse();
    return ((erg/3)+0.5);
}
// Timer3 initialisieren
void pump_start( void )
{
    StructTC* timerbase3  = TCB3_BASE;        // Basisadressse TC Block 3
    StructPIO* piobaseA   = PIOA_BASE;        // Basisadresse PIO A
    timerbase3->TC_CCR = TC_CLKDIS;            // Disable Clock
    // Initialize the mode of the timer 3
    timerbase3->TC_CMR =          //Channel Mode Register
       TC_ACPC_CLEAR_OUTPUT  |    //ACPC    : Register C clear TIOA
       TC_ACPA_SET_OUTPUT    |    //ACPA    : Register A set TIOA
       TC_WAVE               |    //WAVE    : Waveform mode
       TC_CPCTRG             |    //CPCTRG  : Register C compare trigger enable
       TC_CLKS_MCK1024;           //TCCLKS  : MCKI / 1024 = 24 414,0625
    // Initialize the counter:
    timerbase3->TC_RA = 244;
    timerbase3->TC_RC = 488;
    // Start the timer :
    timerbase3->TC_CCR = TC_CLKEN ;        // CCR = Channel Control Register -> Clock enable
    timerbase3->TC_CCR = TC_SWTRG ;        // Z????hler reset, Clock starten
    piobaseA->PIO_PDR  = (1<<PIOTIOA3) ;
    piobaseA->PIO_OER  = (1<<PIOTIOA3) ;
    piobaseA->PIO_CODR = (1<<PIOTIOA3) ;    // (Clear Output Data Register)
}
```

```
void intOutput (int var)
{
    long int a = 10;
    int b = 0;
    char carray[22];
    if(var < 0){
        var = var * -1;
        carray[0]='-';
        b=b+1;
        }
    while((var % a) != var){
    a=a*10;
    }
    a=a/10;
    while(a>0){            // var stellenweise in das array schreiben
    carray[b]=((var/a)+48);
    var = var - ((var/a)*a);
    a = a/10;             // a um eine Gr????enordnung verringern
    b = b+1;             // n??chste Stelle des arrays
    }
    carray[b]='\0';                 //array "abschlie??en"
    puts(carray);            //array ausgeben
}
```

```
int main(void)
{
    pump_stop();
    int test1 = 0;
    int test = 50;
    // Serielle initialisieren
    inits();
    puts("Herzlich willkommen zu unserer Ausschankstation. Bitte befolgen Sie den unten
stehenden Anweisungen.\n");
    while(1){
    pump_stop();                              //falls die Pumpe noch lauuft
    int fehlertoleranz = 7;
    int voll = 50;                             // Welche Menge soll abgefuellt werden!!!
    puts("Bitte stellen Sie den Becher auf die Waage und druecken Sie die Taste 1! \n");
    init_messung();
    while(piobaseB->PIO_PDSR & KEY1){}                                  //polling (KEY1)
    int becher = 0;
    becher = MasseMitGenauigkeit();                                     //Bechergewicht messen
    puts("Der Becher hat folgende Masse (in Gramm): ");
    intOutput(becher);
    puts("\n");
    puts("Dr??cken Sie SW2 um den Becher zu bef??llen. \n")  ;
    while(piobaseB->PIO_PDSR & KEY2){}                                  //polling (KEY2)
             if (MasseMitGenauigkeit() < becher+fehlertoleranz && MasseMitGenauigkeit() >
becher-fehlertoleranz){
            pump_start();
            int alteMasse = becher;
            int neueMasse = becher;
            int testMasse = becher;
            int ausgabe = 1;
            while ((neueMasse - becher) < voll){
              test1++;
              if(test1%test==0){
                    if(testMasse>=neueMasse){
                    pump_stop();
                    puts("keine Gewichtszunahme !! \n");
                    break;
                    }
                 testMasse = neueMasse;
               }
                alteMasse = neueMasse;
               neueMasse = MasseMitGenauigkeit(); //Becher wird mit Inhalt gemessen
               if((neueMasse-becher) >= ausgabe){
                    puts(" im Becher sind ");
                    intOutput(neueMasse-becher);
                    puts(" Gramm \n");
                    ausgabe = neueMasse-becher+1;
                }
             if (neueMasse < 1 || (alteMasse+6) < neueMasse || (alteMasse-6) > neueMasse){
                    pump_stop();
                    puts("Es kam zu einen Fehler an der Gewichtsmessung \n");
                    break;
              }
            }
            pump_stop();
            puts("Vorgang abgeschlossen \n\n");
        } else{
            pump_stop();
            puts("Es kam zu einen Fehler an der Gewichtsmessung \n");
        }
    }
    return 0;
}
```

3.5.2 seriell.c:

```
int init_ser()
{
	StructPIO* piobaseA = PIOA_BASE;
	StructPMC* pmcbase  = PMC_BASE;
	StructUSART* usartbase0 = USART0;
	pmcbase->PMC_PCER = 0x4;   // Clock f?r US0 einschalten
	piobaseA->PIO_PDR = 0x18000;      // US0 TxD und RxD
	usartbase0->US_CR = 0xa0;  // TxD und RxD disable
	usartbase0->US_BRGR = US_BAUD;    // Baud Rate Generator Register
	usartbase0->US_MR = 0xc0;  // Keine Parit?t, 8 Bit, MCKI
	usartbase0->US_CR = 0x50;  // TxD und RxD enable
	return 0;
}

char putch(char Zeichen)
{
	StructUSART* usartbase0 = USART0;
	if( usartbase0->US_CSR & US_TXRDY )
	{
		usartbase0->US_THR = Zeichen;
	}
	else
	{
		Zeichen = 0;  // wenn keine Ausgabe
	}
	return Zeichen;
}
```

3.5.3 swi.c:

```
void SWIHandler()
{
	register char *reg_r0 asm ("r0");
	register unsigned int *reg_14 asm ("r14");
	switch( *(reg_14 - 1) & 0x00FFFFFF)
	{
		case 0x100:
			init_ser();
			break;
		case 0x200:
			*reg_r0 = putch(*reg_r0);
			break;
		case 0x300:
			*reg_r0 = getch();
			break;
	}
}
```

3.5.4 ser_io.S:

```
@ Funktion
	.text
	.align 2
	.global	inits
	.type	inits,function
inits:
	swi	0x100
	bx	lr	@ R?cksprung
```

```
@ Funktion
	.text
	.align 2
	.global	putc
	.type	putc,function
putc:

	mov	r1, r0		@ Zeichen nach r1
	ldr	r0, =Zeichen	@ Zeiger holen
	str	r1, [r0]		@ Zeichen unter Zeiger ablegen
	mov r6, r1			@ UPDATE: zum retten des Zeichens, falls der
Sender besch?ftigt ist
	swi	0x200		@
	ldr	r1, =Zeichen	@ Zeiger holen
	ldr	r0, [r1]		@ Zeichen aus Zeiger holen
	cmp r0, #0			@ UPDATE: Falls der Sender besch?ftigt ist,
kommt eine Null
	mov r0, r6			@ UPDATE: Speicher das Zeichen zur?ck, damit
es wieder vhd. ist falls der Sender besch?ftigt war
	beq putc			@ UPDATE: Falls Senden fehlgeschlagen: try
again
	bx	lr

@ Funktion
	.text
	.align 2
	.global	puts
	.type	puts,function
puts:
	stmfd sp!,{lr}	@ Retten der Register

// Hier mu? Ihr Code eingef?gt werden.
	mov r4, r0			@ Speichere Adresse in R4
loopPuts:
	ldrb r5, [r4], #1		@ Lade char in r4
	cmp r5, #0xa		@ Vergleiche ob Zeichen gleich Newline
	bne endNewLine			@ Falls nicht, springe an das Ende der Newline
Routine
	mov r0, #0xd		@ Setze ein Carriage Return
	bl putc				@ Schreibe das Carriage Return
endNewLine:
	 mov r0, r5			@ schreibe das Line Feed zur?ck in R0
	cmp r0, #0			@ Vergleiche ob das Zeichen "0" ist
	beq endPuts			@ Falls "0" springe an das Ende der Funktion
	bl putc				@ Falls nicht Null, schreibe das Zeichen/
den Line Feed
	b loopPuts			@ Falls nicht Null, wiederhole den Vorgang

endPuts:
	ldmfd sp!,{pc}			@ R?cksprung

	.data
Zeichen:	.word 0
.end
```