

```

1  /*****
2  Test Program
3  Project : Tst_UartMenu
4  Version : 1
5  Date   : 18.12.2017
6  Author  : R.Oliva
7  Company : L&R Ing
8  Comments:
9  Test UART0_Driver Operations with Menu (read/write)
10
11
12  Chip type           : ATmega1284P
13  Program type        : Application
14  AVR Core Clock frequency: 14,745600 MHz
15  Memory model        : Small
16  External RAM size    : 0
17  Data Stack size     : 4096
18  **
19  ** Version 19.12.2017
20  ** Call Simplified ExternalTimer Interrupt, pin_change_isr3()
21  ** based on 0.5Hz signal 19.12.2017
22  ** External_Timer_Init();
23  *****/
24
25
26  #include <megal284p.h>
27  #include <delay.h>
28  #include <string.h>
29  #include <stdio.h>
30  #include <stdarg.h>
31  #include <stdlib.h>
32  #include <string.h>
33  #include <io.h>
34
35
36  // Added for LCD- 19-12-2017
37  #include "..\..\LCD\LCD4x20(2010)\inc\lcd_cl2_3.h"
38
39  // Added for TWI - 18-12-2017
40  #include "..\..\TWI\TWI12\inc\twi-cl2_12.h"
41
42  // Testing UART0_DR1 - 18-12-2017
43  #include "..\inc\Uart0_dr1.h"
44
45  // Added for testing - Initialize_CL2_Simple()
46  #include "Tst_UartMenu_main.h"
47
48  // Added for testing Menus
49  #include "Basic_Menu.h"
50
51
52  // PB.0 down --> initialize RTC 19.11.17
53  #define KBD_LEFT_ARROW      PINB.0
54
55  /* OLD definitions..USART Baud rate */
56  #define BAUD_RATE 19200
57  #define BAUD_INIT (_MCU_CLOCK_FREQUENCY_/(BAUD_RATE*16L)-1)
58
59
60  // Made Global 18.12.17 to control WDOG..
61  bit WD_ON_Flag = 0;    // Default pets the dog
62  // ****
63  // ** For EXT_IRQ (INT3) with 1.0Hz from CKOUT
64  // ** Added 19.12.2017
65  // ****
66  unsigned char sm2_1sec = 0;    // 12.2017 - 1Second @0.5sec interval count 1-3
67  unsigned char Flag_Sec_Change = 0; // For second change signalling 19.12.17
68  int8_t rval = 0;    // Debug CKOUT
69  int8_t rval1 = 0;
70
71  /*****
72  **
73  ** Initialize_CL2_simple() Depends on processor used and board.
74  ** Simple Version TAKES AWAY Interrupts / modified 19.11.2017
75  ** For LCD and RTC testing..
76  ** Current for Megal284 uses AtMegal284P and CL2bm1

```

```

77  ** COM0 at 19200,N,8,1 - traditional
78  ** COM1 not used
79  ** Port C and one pin PortD used for LCD.
80  ** Timer0 not used,
81  ** Timer1 not used
82  ** ADC not set
83  **
84  *****/
85
86  void Initialize_CL2_simple(void)
87  {
88      CLI();          /* disable all interrupts */
89
90
91      // Input/Output Ports initialization
92
93      // Port A initialization
94      // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
95      // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
96      PORTA=0x00;
97      DDRA=0x00;
98
99      // Port B initialization
100     // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
101     // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
102     PORTB=0x00;
103     DDRB=0x00;
104
105     // Port C initialization
106     // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
107     // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
108     // Debug: set Bit C.7 to 1 3.2.09
109     // v0.3 6-2-09 Set PC.2-7 as outputs - For LCD..
110     PORTC=0x00;
111     DDRC=0xFC;
112
113     // Port D initialization
114     // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
115     // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
116     // v0.3 6-2-09 Set Pd.7 as output - For LCD 0x80
117     // V1284p-7 Set Pd.5./6 to outputs too for OLED and BacklightLCD 24.7.10 0xE0
118     PORTD=0x00;
119     //DDRD=0x80;
120     DDRD=0xE0;
121
122     // Following from SDCard4.c initialization..19.11.2017
123
124     // Timer/Counter 0 initialization
125     // Clock source: System Clock
126     // Clock value: Timer 0 Stopped
127     // Mode: Normal top=0xFF
128     // OCOA output: Disconnected
129     // OC0B output: Disconnected
130     TCCR0A=(0<<COM0A1) | (0<<COM0A0) | (0<<COM0B1) | (0<<COM0B0) | (0<<WGM01) | (0<<WGM00);
131     TCCR0B=(0<<WGM02) | (0<<CS02) | (0<<CS01) | (0<<CS00);
132     TCNT0=0x00;
133     OCR0A=0x00;
134     OCR0B=0x00;
135
136     // Timer/Counter 1 initialization
137     // Clock source: System Clock
138     // Clock value: Timer1 Stopped
139     // Mode: Normal top=0xFFFF
140     // OC1A output: Disconnected
141     // OC1B output: Disconnected
142     // Noise Canceler: Off
143     // Input Capture on Falling Edge
144     // Timer1 Overflow Interrupt: Off
145     // Input Capture Interrupt: Off
146     // Compare A Match Interrupt: Off
147     // Compare B Match Interrupt: Off
148     TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
149     TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
150     TCNT1H=0x00;
151     TCNT1L=0x00;
152     ICR1H=0x00;

```

```

153 ICR1L=0x00;
154 OCR1AH=0x00;
155 OCR1AL=0x00;
156 OCR1BH=0x00;
157 OCR1BL=0x00;
158
159 // Timer/Counter 2 initialization
160 // Clock source: System Clock
161 // Clock value: Timer2 Stopped
162 // Mode: Normal top=0xFF
163 // OC2A output: Disconnected
164 // OC2B output: Disconnected
165 ASSR=(0<<EXCLK) | (0<<AS2);
166 TCCR2A=(0<<COM2A1) | (0<<COM2A0) | (0<<COM2B1) | (0<<COM2B0) | (0<<WGM21) | (0<<WGM20);
167 TCCR2B=(0<<WGM22) | (0<<CS22) | (0<<CS21) | (0<<CS20);
168 TCNT2=0x00;
169 OCR2A=0x00;
170 OCR2B=0x00;
171
172 // Timer/Counter 3 initialization
173 // Clock source: System Clock
174 // Clock value: Timer3 Stopped
175 // Mode: Normal top=0xFFFF
176 // OC3A output: Disconnected
177 // OC3B output: Disconnected
178 // Noise Canceler: Off
179 // Input Capture on Falling Edge
180 // Timer3 Overflow Interrupt: Off
181 // Input Capture Interrupt: Off
182 // Compare A Match Interrupt: Off
183 // Compare B Match Interrupt: Off
184 TCCR3A=(0<<COM3A1) | (0<<COM3A0) | (0<<COM3B1) | (0<<COM3B0) | (0<<WGM31) | (0<<WGM30);
185 TCCR3B=(0<<ICNC3) | (0<<ICES3) | (0<<WGM33) | (0<<WGM32) | (0<<CS32) | (0<<CS31) | (0<<CS30
);
186 TCNT3H=0x00;
187 TCNT3L=0x00;
188 ICR3H=0x00;
189 ICR3L=0x00;
190 OCR3AH=0x00;
191 OCR3AL=0x00;
192 OCR3BH=0x00;
193 OCR3BL=0x00;
194
195 // Timer/Counter 0 Interrupt(s) initialization
196 TIMSK0=(0<<OCIE0B) | (0<<OCIE0A) | (0<<TOIE0);
197
198 // Timer/Counter 1 Interrupt(s) initialization
199 TIMSK1=(0<<ICIE1) | (0<<OCIE1B) | (0<<OCIE1A) | (0<<TOIE1);
200
201 // Timer/Counter 2 Interrupt(s) initialization
202 TIMSK2=(0<<OCIE2B) | (0<<OCIE2A) | (0<<TOIE2);
203
204 // Timer/Counter 3 Interrupt(s) initialization
205 TIMSK3=(0<<ICIE3) | (0<<OCIE3B) | (0<<OCIE3A) | (0<<TOIE3);
206
207 // External Interrupt(s) initialization
208 // INT0: Off
209 // INT1: Off
210 // INT2: Off
211 // Interrupt on any change on pins PCINT0-7: Off
212 // Interrupt on any change on pins PCINT8-15: Off
213 // Interrupt on any change on pins PCINT16-23: Off
214 // Interrupt on any change on pins PCINT24-31: Off
215 EICRA=(0<<ISC21) | (0<<ISC20) | (0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
216 EIMSK=(0<<INT2) | (0<<INT1) | (0<<INT0);
217 PCICR=(0<<PCIE3) | (0<<PCIE2) | (0<<PCIE1) | (0<<PCIE0);
218
219 // USART0 initialization
220 // Communication Parameters: 8 Data, 1 Stop, No Parity
221 // USART0 Receiver: On
222 // USART0 Transmitter: On
223 // USART0 Mode: Asynchronous
224 // USART0 Baud Rate: 19200
225 UCSR0A=(0<<RXC0) | (0<<TXC0) | (0<<UDRE0) | (0<<FE0) | (0<<DOR0) | (0<<UPE0) | (0<<U2X0) |
(0<<MPCM0);
226 UCSR0B=(0<<RXCIE0) | (0<<TXCIE0) | (0<<UDRIE0) | (1<<RXEN0) | (1<<TXEN0) | (0<<UCSZ02) | (0
<<RXB80) | (0<<TXB80);

```

File: C:\cvavr328\Work3\CL2\CL2\_Drivers\UART0\TST\_MENU\_PRJ\Tst\_UartMe

```
227 UCSROC=(0<<UMSEL01) | (0<<UMSEL00) | (0<<UPM01) | (0<<UPM00) | (0<<USBS0) | (1<<UCSZ01) | (
1<<UCSZ00) | (0<<UCPOL0);
228 UBRROH=0x00;
229 UBRROL=0x2F;
230
231 /* initialize the USART0 TX, 8N1, Baud rate: 19200 */
232 UCSROA=0;
233 UCSROB=1<<TXEN0;
234 UCSROC=(1<<UCSZ01)|(1<<UCSZ00);
235 UBRROH=BAUD_INIT>>8;
236 UBRROL=BAUD_INIT&0xFF;
237
238 // USART1 initialization
239 // USART1 disabled
240 UCSR1B=(0<<RXCIE1) | (0<<TXCIE1) | (0<<UDRIE1) | (0<<RXEN1) | (0<<TXEN1) | (0<<UCSZ12) | (0
<<RXB81) | (0<<TXB81);
241
242 // Analog Comparator initialization
243 // Analog Comparator: Off
244 // The Analog Comparator's positive input is
245 // connected to the AIN0 pin
246 // The Analog Comparator's negative input is
247 // connected to the AIN1 pin
248 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<
ACIS0);
249 ADCSRB=(0<<ACME);
250 // Digital input buffer on AIN0: On
251 // Digital input buffer on AIN1: On
252 DIDR1=(0<<AIN0D) | (0<<AIN1D);
253
254 // ADC initialization
255 // ADC disabled
256 ADSCRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1)
| (0<<ADPS0);
257
258
259 // This kept from old Initialize_CL2() function..
260 // *****
261 // Two Wire Bus initialization
262 // Bit Rate:
263 // 17.2.09 Changed TWBR to 0x0c for 184kHz..
264 // if XTAL = 7.3728e6 means:
265 // Bit Rate: 184.320 kHz
266 // ** NOTE: For Bit Rate: 115.200 kHz
267 // ** then CV-Wiz selectede TWBR=0x18;
268 // TWBR=0x0C;
269 // Two Wire Bus Slave Address: 0h
270 // General Call Recognition: Off
271 // TWAR=0x00;
272 // Generate Acknowledge Pulse: Off
273 // TWI Interrupt: OFF
274 // TWCR=0x05 - would be for TWI IRQ ON;
275 // TWCR=0x04; // Same as v32..
276 // TWSR=0x00;
277
278 // Two Wire Bus initialization for XTal 14.756MHz 22.3.2012
279 // Bit Rate: 184.320 kHz
280 TWBR=0x20;
281 // Two Wire Bus Slave Address: 0h
282 // General Call Recognition: Off
283 TWAR=0x00;
284 // Generate Acknowledge Pulse: Off
285 // TWI Interrupt: Off
286 TWCR=0x04;
287 TWSR=0x00;
288
289
290 // Watchdog Timer initialization - CV-Wiz (07.2010)
291 // Watchdog Timer Prescaler: OSC/2k
292 // Watchdog Timer interrupt: Off
293 // Re-enable 30-9-2010
294 // b4 WDCE, b3 WDE
295 // b5 b2 b1 b0 WDP3-WDP0 Prescaler
296 #pragma optsize-
297 #asm("wdr")
298 WDTCR=0x18; // 0001 1000
299 //WDTCR=0x08; // 0000 1000 -- Prescaler in 16ms
```

File: C:\cvavr328\Work3\CL2\CL2\_Drivers\UART0\TST\_MENU\_PRJ\Tst\_UartMe

```
300     WDTCR=0x28;      // 0010 1000  -- Prescaler in 1000 = 4sec
301     #ifdef _OPTIMIZE_SIZE_
302     #pragma optimize+
303     #endif
304
305     // Global enable interrupts
306     // SEI();          /* re-enable interrupts 19.11.2017 */
307
308 }
309
310
311 /*****
312 **
313 ** External_Timer_Init(void) - 19.12.2017
314 ** Uses CKOUT as source on PD.4 pin, CL2bm1
315 ** Used together with pin_change_isr3
316 *****/
317 **/
318
319 void External_Timer_Init(void){          // 19.12.2017 CKOUT driven Interrupt
320     // External Interrupt(s) initialization
321     // INT0: Off
322     // INT1: Off
323     // INT2: Off
324     // Interrupt on any change on pins PCINT0-7: Off
325     // Interrupt on any change on pins PCINT8-15: Off
326     // Interrupt on any change on pins PCINT16-23: Off
327     // Interrupt on any change on pins PCINT24-31: On
328     EICRA=0x00;
329     EIMSK=0x00;
330     PCMSK3=0x10;
331     PCICR=0x08;
332     PCIFR=0x08;
333 }
334
335 /*****
336 **
337 ** Simplified External PD.4 interrupt ISR (level change, OSC_OUT set to 1Hz)
338 ** Called on each level change or 2Hz (0.5sec) intervals..
339 ** 24.10.2012 Pin change 24-31 interrupt service routine - first phase add
340 ** OLED change on level transition..
341 ** Revision - 19.12.2017
342 **
343 *****/
344
345 interrupt [PC_INT3] void pin_change_isr3(void)
346 {
347     // New values for 2Hz transition calls.. 24.10.2012
348     // Calculations see Cuaderno#2, SoftwPWRC2 Additions 10.2012
349     // 24.10.2012 Counts transitions 1 to 3
350     #define SECOND2_TOUT 3
351
352
353     // First step - toggle LCD every 0.5 Hz
354     if(OLED) OLED=0; // Add a 1 sec life LED.. 24-10-2012
355     else OLED= 1;
356
357     if(WD_ON_Flag == 0){ #asm("wdr")}
358
359     sm2_1sec ++;
360
361     if(sm2_1sec == SECOND2_TOUT) {
362         sm2_1sec = 1;
363         Flag_Sec_Change = 1;//
364     } // All solved in 1sec- interval..
365
366 }
367
368
369
370
371
372 void main(void)
373 {
374     // Declare your local variables here
375     // 19.11.2017 for Display & COM showing..
376     char s[25];
```

File: C:\cvavr328\Work3\CL2\CL2\_Drivers\UART0\TST\_MENU\_PRJ\Tst\_UartMe

```
377 // 19.11.2017 for Display & COM showing..
378 int8_t RTC_result = 0; // Defined for RTC
379 // Local for new functions (Old RTC funtions use int8_t)
380 uint8_t uRTCHour = 0;
381 uint8_t uRTCMin = 0;
382 uint8_t uRTCSec = 0;
383 uint8_t uRTCDay = 0;
384 uint8_t uRTCMonth = 0;
385 uint16_t uRTCYear = 0;
386
387
388 // *****
389 // * Change to same notation in CL2_Basis4
390 // * 19-11-2017
391 #pragma optimize-
392 CLKPR=0x80;
393 CLKPR=0x00;
394 #ifdef _OPTIMIZE_SIZE_
395 #pragma optimize+
396 #endif
397
398
399 // 18.12.2017 Most initialization sent to Initialize_CL2() funtion
400 Initialize_CL2_simple();
401
402 // Call special COM0 init
403 USART0_Init();
404
405 // Call Simplified ExternalTimer Interrupt, pin_change_isr3()
406 // based on 0.5Hz signal 19.12.2017
407 External_Timer_Init();
408
409 // *****
410 // 19.11.2017 - Added - for life detection..
411 // Moved 19.12.2017 to pin_change_isr3
412 // OLED = 1;
413 // delay_ms(1000); // 1sec
414 // OLED = 0;
415
416
417 // COM Port Init
418 // 18.12.2017
419
420
421
422 /* globally enable interrupts */
423 #asm("sei")
424
425 // *****
426 // 16-5-2017 - Added - for life detection..
427 // BACK_LIGHT = 1;
428 // OLED = 1;
429 // delay_ms(500); // 0.5sec
430 // OLED = 0;
431
432 // *****
433 // 19-11-2017 - Added For LCD starting..
434 // *****
435 printf("LCD Access testing..\n\r");
436 init_display();
437 clear_display();
438 set_LCD_cur(0,0);
439 disp_cstr("TestUARTMenu2017"); // Modificación de etiqueta 19.11.17
440 set_LCD_cur(1,0);
441 disp_cstr("*"); // Cambio
442 delay_ms(1000);
443
444 // *****
445 // RTC Initialize -19.11.2017
446 // *****
447 // #define OPTION_CKOUT_01HZ 0x83
448 // set Clockout
449 set_LCD_cur(0,0);
450 rval1 = 4;
451 rval = rtc_set_ckoutfreq(rval1);
452 printf("RTC Ckout result: %d",rval);
453 disp_cstr("CKout_1Hz");
```

```

454     delay_ms(2000);
455
456     printf("RTC inicializando..\n\r");
457     // RTC Read..
458     RTC_result = rtc_get_timeNdate(&RTCHour, &RTCMin, &RTCSec, &RTCDay, &RTCMonth, &RTCYear);
459
460     // clock battery dead.. reset to something sane
461     // Set to error 16.2.09
462     // Back to TWI_SUCCESS 17.2.09
463     // Force initialize.. 4.1.2012
464     // RTC_result = 187;
465     if((RTC_result != TWI_SUCCESS)|(KBD_LEFT_ARROW == 0))
466     {
467         printf("RTC-Now initialized!\n\r");
468         set_LCD_cur(0,0);
469         disp_cstr("RTes RTC Init.. ");
470         // Probable requirement of Variables, not constants in rtc_set_time/date..
471         // Verify 19.11.2017
472         RTCHour = 22;
473         RTCMin = 41;
474         RTCSec = 0;
475         rtc_set_time(RTCHour, RTCMin, RTCSec);
476         delay_ms(1000);
477         RTCDay = 19;
478         RTCMonth = 11;
479         RTCYear = 2017;
480         rtc_set_date(RTCDay, RTCMonth, RTCYear);
481         delay_ms(1000);
482     }
483     else
484     {
485         printf("RTC-OK!\n\r");
486         set_LCD_cur(0,0);
487         disp_cstr("RTC OK read.. ");
488     }
489
490     rtc_get_timeNdate(&RTCHour, &RTCMin, &RTCSec, &RTCDay, &RTCMonth, &RTCYear);
491     // Added Display RTC on LCD
492     sprintf(s,"%02d/%02d/%04d-%02d:%02d:%02d ", RTCDay, RTCMonth, RTCYear, RTCHour, RTCMin, RTCSec);
493     set_LCD_cur(1,0);
494     disp_str(s);
495     puts(s);
496     delay_ms(2000);
497     // Test new functions a) Time
498     RTC_result = (int8_t)(rtc_get_time(&uRTCHour, &uRTCMin, &uRTCSec));
499     printf("RTC new function ReadTime Access: %d", RTC_result);
500     delay_ms(2000);
501     sprintf(s,"Time:%02d:%02d:%02d ", uRTCHour, uRTCMin, uRTCSec);
502     set_LCD_cur(1,0);
503     disp_str(s);
504     puts(s);
505     delay_ms(2000);
506
507     rtc_get_date(&uRTCDay, &uRTCMonth, &uRTCYear);
508     sprintf(s,"Date:%02d/%02d/%04d ", uRTCDay, uRTCMonth, uRTCYear);
509     set_LCD_cur(1,0);
510     disp_str(s);
511     puts(s);
512     delay_ms(2000);
513
514
515
516
517     printf("Testing UART0 Menu Driver 1:\r\n");
518
519     set_LCD_cur(0,0);
520     disp_cstr("Menu Test.. ");
521
522
523     delay_ms(2000); // 2sec
524     // *****
525
526     while (1)
527     {
528         // Place your code here
529         // 16-5-2017 - Added - for Incorrect Mounting..
530         // if(WD_ON_Flag == 0){ #asm("wdr")} sent to ISR 19.12.2017

```

File: C:\cvavr328\Work3\CL2\CL2\_Drivers\UART0\TST\_MENU\_PRJ\Tst\_UartMe

```
531      // OLED = 1;      // sent to ISR..19.12.2017
532      // delay_ms(500); // 0.5 blinking
533      // OLED = 0;
534      delay_ms(50);      // Minimal delay 19.12.17
535      Check_UART0_Menu(MSG_opt);
536
537      if (Flag_Sec_Change == 1){
538          Flag_Sec_Change = 0;
539          RTC_result = (int8_t)(rtc_get_time(&uRTCHour, &uRTCMin, &uRTCSec));
540          sprintf(s, "Time:%02d:%02d:%02d  ", uRTCHour, uRTCMin, uRTCSec);
541          set_LCD_cur(1, 0);
542          disp_str(s);
543      }
544
545      } // en while(1)
546  }
547
```