|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Uhrstatus** | | | | | |  | **spistatus** | | | | | | | **busstatus** | | |
| SYNC\_FIRSTRUN | SYNC\_NULL | SYNC\_WAIT | SYNC\_READY | SYNC\_OK | SYNC\_CHECK |  | ACTIVE\_BIT | SPI\_SHIFT\_IN\_OK\_BIT | STARTDATEN\_BIT | ENDDATEN\_BIT | SUCCESS\_BIT | LB\_BIT | HB\_BIT | TWI\_CONTROLBIT | SPI\_SENDBIT | WEB\_CONTROLBIT |
|  |  |  |  |  |  | if (SPI\_CONTROL\_PORTPIN & (1<< SPI\_CONTROL\_CS\_HC)) // CS ist HI, SPI beendet  {  SPI\_CONTROL\_PORT |= (1<<SPI\_CONTROL\_MISO); // MISO ist HI in Pausen    #pragma mark PASSIVE  if (spistatus &(1<<ACTIVE\_BIT)) // CS beendet. ACTIVE\_BIT ist NOCH gesetzt: Slave ist NEU passiv geworden. Aufraeumen, Daten uebernehmen  {  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  /\*  Eine Uebertragung hat stattgefunden.  Die out-Daten sind auf dem Webserver.  Die in-Daten vom Webserver sind geladen.  Sie muessen noch je nach in\_startdaten ausgewertet werden.  \*/    // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  wdt\_reset();  SPI\_Call\_count0++;  // Eingang von Interrupt-Routine, Daten von Webserver  //err\_clr\_line(1);  err\_gotoxy(19,0);  err\_putc(' '); // SUCCESS-Marke weg    // in lcd verschoben  //lcd\_clr\_line(3);  lcd\_gotoxy(0,3);    // Eingang anzeigen  lcd\_puts("iW \0");  lcd\_puthex(in\_startdaten);  lcd\_putc(' ');  lcd\_puthex(in\_hbdaten);  lcd\_puthex(in\_lbdaten);  lcd\_putc(' ');  uint8\_t j=0;  for (j=0;j<1;j++)  {  //lcd\_putc(' ');  //lcd\_puthex(inbuffer[j]);  //err\_putc(inbuffer[j]);  }  //lcd\_putint(outbuffer[8]);  //lcd\_putc(' ');  lcd\_putint(SPI\_Call\_count0);    lcd\_gotoxy(12,2);  lcd\_putint2(inbuffer[16]);// stunde  lcd\_putint2(inbuffer[17]);// minute  //lcd\_putint2(inbuffer[18]);// wochentag  //lcd\_gotoxy(0,1);  //lcd\_putc('A');  // Start, RTC setzen mit Zeit vom Webserver  if ((uhrstatus & (1<<SYNC\_NULL)) && (!(uhrstatus & (1<<SYNC\_FIRSTRUN) )) )    {  uint8\_t res = 0;  res=rtc\_write\_Control(1);  delay\_ms(30);  //lcd\_putc('B');  // stunde, minute, sekunde  res=rtc\_write\_Zeit(inbuffer[16],inbuffer[17],0);// uint8\_t stunde, uint8\_t minute, uint8\_t sekunde  delay\_ms(30);  //lcd\_putc('C');  if (res)  {  err\_gotoxy(11,0);  err\_puts("Z-\0");  err\_putint1(res);  std = inbuffer[16];  Zeit.stunde = inbuffer[16];  min = inbuffer[17];  Zeit.minute = inbuffer[17];  //err\_putc('\*');  }  else {  err\_gotoxy(11,0);  err\_puts("Z+ \0");  }  //lcd\_putc('D');  // Datum: 1 = Montag  res=rtc\_write\_Datum(inbuffer[18],inbuffer[19],inbuffer[20],18);// uint8\_t wochentag, uint8\_t tagdesmonats, uint8\_t monat, uint8\_t jahr  delay\_ms(30);  //lcd\_putc('E');  if (res)  {  //err\_gotoxy(14,0);  //err\_puts(" ");    err\_gotoxy(15,0);  err\_puts("D-\0");  err\_putint1(res);  Zeit.wochentag = inbuffer[18];    Zeit.kalendertag = inbuffer[19];  Zeit.kalenderjahr = 18;  /\*  Zeit.minute = RTCdaten[0];  Zeit.stunde = RTCdaten[1];  Zeit.kalendertag = RTCdaten[2];  Zeit.kalendermonat = RTCdaten[3];  Zeit.kalenderjahr = RTCdaten[4];  Zeit.wochentag = RTCdaten[5]-1; // RTC, DCF77: Montag=1; EEPROM: Montag=0  \*/  //err\_putc('\*');  }  else  {  //err\_gotoxy(17,0);  //err\_puts(" ");    err\_gotoxy(15,0);  err\_puts("D+ \0");  }  //lcd\_putc('F');  oldmin = inbuffer[17];  oldstd = inbuffer[16];  oldtag = inbuffer[18];  if (res==0)  {  uhrstatus &= ~(1<<SYNC\_NULL);  }  //lcd\_putc('G');  }    // FIRSTRUN nach erster SPI-Sequenz resetten  uhrstatus &= ~(1<<SYNC\_FIRSTRUN);  OutCounter++;      err\_gotoxy(0,0);  err\_puts(" \0");    err\_gotoxy(19,0);  err\_putc(' ');    // Uebertragung pruefen  if (ByteCounter == SPI\_BUFSIZE-1) // Uebertragung war vollstaendig  {  //lcd\_putc('I');  if (out\_startdaten + in\_enddaten==0xFF)  {  err\_gotoxy(19,0);  err\_putc('+');  spistatus |= (1<<SUCCESS\_BIT); // Bit fuer vollstaendige und korrekte Uebertragung setzen  lcd\_gotoxy(19,0);  lcd\_putc(' ');  //lcd\_clr\_line(2);  // err\_clr\_line(2);  err\_clr\_line(3);    err\_gotoxy(0,3);  err\_puts("OK\0");  err\_puthex(ByteCounter);      err\_putc(' ');  err\_puthex(in\_startdaten);  err\_puthex(in\_enddaten);    err\_putc(' ');  err\_puthex(out\_startdaten);  err\_puthex(out\_enddaten);  err\_putc(' ');  err\_puthex(out\_startdaten + in\_enddaten);  //err\_gotoxy(0,2);  //err\_puthex(in\_lbdaten);  //err\_puthex(in\_hbdaten);  //err\_gotoxy(0,1);  //err\_puthex(loopCounterSPI++);  //err\_puts("OK \0");    //err\_puthex(out\_startdaten + in\_enddaten);  // if (out\_startdaten==0xB1)  {  SendOKCounter++;  }  spistatus |= (1<<SPI\_SHIFT\_IN\_OK\_BIT);  }  else  {  spistatus &= ~(1<<SUCCESS\_BIT); // Uebertragung fehlerhaft, Bit loeschen  err\_putc('-');  //err\_clr\_line(1);  //lcd\_clr\_line(3);    // err\_clr\_line(2);  err\_clr\_line(3);  err\_gotoxy(0,3);  err\_puts("E1\0");  err\_puthex(ByteCounter);    err\_putc(' ');  err\_puthex(in\_startdaten);  err\_puthex(in\_enddaten);  err\_putc(' ');  err\_puthex(out\_startdaten);  err\_puthex(out\_enddaten);  err\_putc(' ');  err\_puthex(out\_startdaten + in\_enddaten);    spistatus &= ~(1<<SPI\_SHIFT\_IN\_OK\_BIT);    {  SendErrCounter++;  }  //errCounter++;  in\_startdaten = DATATASK; // 180414 weiterfahren wie bisher    }    }  else  {  //lcd\_putc('G');  spistatus &= ~(1<<SUCCESS\_BIT); // Uebertragung unvollstaendig, Bit loeschen    //err\_putc('!');  // err\_clr\_line(1);  // err\_clr\_line(2);  err\_gotoxy(0,2);  err\_puts("!E2\0");  err\_putint(ByteCounter);  err\_putc(' ');  err\_puthex(in\_lbdaten);  err\_puthex(in\_hbdaten);  err\_putc(' ');  err\_puthex(in\_startdaten);  err\_puthex(in\_enddaten);  err\_putc(' ');  err\_puthex(out\_startdaten + in\_enddaten);  //delay\_ms(100);  //errCounter++;  IncompleteCounter++;  spistatus &= ~(1<<SPI\_SHIFT\_IN\_OK\_BIT);    in\_startdaten = DATATASK; // 180414 weiterfahren wie bisher    }    // lcd\_gotoxy(6,0);  // lcd\_puts("out");  // lcd\_puthex(out\_startdaten);  //lcd\_puthex(outbuffer[8]);  //lcd\_gotoxy(11, 1); // Events zahelen  //lcd\_puthex(OutCounter);  /\*  lcd\_puthex(SendOKCounter);  lcd\_puthex(SendErrCounter);  lcd\_puthex(IncompleteCounter);  \*/    lcd\_gotoxy(0,2);  lcd\_putc('i');  lcd\_puthex(in\_startdaten);  //lcd\_puthex(complement);    //lcd\_putc(' ');  //lcd\_putc('a');  //lcd\_puthex(out\_startdaten);  //lcd\_puthex(in\_enddaten);  //lcd\_puthex(outbuffer[8]);    lcd\_putc(' ');  lcd\_putc('l');  lcd\_puthex(in\_lbdaten);  lcd\_putc(' ');  lcd\_putc('h');  lcd\_puthex(in\_hbdaten);  //out\_hbdaten++;  //out\_lbdaten--;        /\*  lcd\_gotoxy(0,0);  lcd\_puthex(inbuffer[9]);  lcd\_puthex(inbuffer[10]);  lcd\_puthex(inbuffer[11]);  lcd\_puthex(inbuffer[12]);  lcd\_puthex(inbuffer[13]);  \*/  //lcd\_gotoxy(13,0); // SPI - Fehler zaehlen  //lcd\_puts("ERR \0");  //lcd\_gotoxy(17,0);  //lcd\_puthex(errCounter);      // ACTIVE\_BIT loeschen, wurde    spistatus &= ~(1<<ACTIVE\_BIT); // Bit 0 loeschen, wurde nach 'neu aktiv' gesetzt    // Bits im Zusammenhang mit der Uebertragung zuruecksetzen. Wurden in ISR gesetzt  spistatus &= ~(1<<STARTDATEN\_BIT); // Bit 1 loeschen  spistatus &= ~(1<<ENDDATEN\_BIT); // Bit 2 loeschen  spistatus &= ~(1<<SUCCESS\_BIT); // Bit 3 loeschen  spistatus &= ~(1<<LB\_BIT); // Bit 4 loeschen  spistatus &= ~(1<<HB\_BIT); // Bit 5 loeschen      // aufraeumen  out\_startdaten=0x00;  out\_hbdaten=0;  out\_lbdaten=0;  for (i=0;i<SPI\_BUFSIZE;i++)  {  outbuffer[i]=0;  }    /\*  lcd\_gotoxy(0,0); // Fehler zaehlen  lcd\_puts("IC \0");  lcd\_gotoxy(2,0);  lcd\_puthex(IncompleteCounter);  lcd\_gotoxy(5,0);  lcd\_puts("TW \0");  lcd\_gotoxy(7,0);  lcd\_puthex(TWI\_errCounter);    lcd\_gotoxy(5,1);  lcd\_puts("SE \0");  lcd\_gotoxy(7,1);  lcd\_puthex(SendErrCounter);  \*/  } // if Active-Bit    #pragma mark HomeCentral-Tasks    } |  |  |  |  |  |  |  |  |  |  |