

Mikrocontroller-Grundlagen

Teil 20

Anhand mehrerer Beispiele beschreiben wir ausführlich die Anwendungsprogramme zum Auslesen bzw. Beschreiben von EEPROMs

6.10.4 Anwendungsprogramme

Die im folgenden beschriebenen Anwendungsprogramme stützen sich auf die bisher beschriebenen Unterprogramme, die den Schreib- bzw. Lesezugriff auf bzw. von I²C-Teilnehmern unterstützen.

Die Programme erlauben die Messung am I²C-Bus mit einem 2-Kanal-Oszillo-

I²C-Datenübertragung, welches von dem Unterprogramm im Carry-Bit übergeben wird, zeigt der Port-Pin P3.7 an. Dies kann durch das Anstecken der LED-Ausgabe-Platine an dem Port 3 überprüft werden. Die nachfolgende kurze Verzögerung dient lediglich dazu, eine bessere Triggerung des Oszilloskopes zu ermöglichen.

6.10.4.2 Schreiben des EEPROMs

Abbildung 157 zeigt das Testprogramm 50, welches die an dem Port P 1 anliegenden 8Bit-Daten in die Adresse 0 des angeschlossenen EEPROMs schreibt. Dazu ist an dem Port P 1 die Schalterplatine und am Port P 3 die 8Bit-LED-Platine anzuschließen. Nach dem Auslesen des Ports P 1 und Übertragen zum EEPROM wird das Ergebnis am Port P 3.7 und damit an der LED D 7 angezeigt, welche bei erfolgreichem Schreiben gelöscht bleibt.

Im Gegensatz zum Testprogramm 49 erfolgt bei diesem Programm lediglich ein einmaliges Beschreiben des EEPROMs, da die Anzahl der Schreibzyklen begrenzt ist. Eine Ausnahme bilden lediglich die EEPROMs mit Ferro-RAMs. Die mit dem Testprogramm 50 geschriebene Bit-Kombination läßt sich über das Testprogramm 49 wiederum zurücklesen (Umstecken der LED-Anzeigenplatine erforderlich).

6.10.4.3 Lesen und Schreiben des EEPROMs

Abbildung 158 zeigt das Testprogramm 51, welches zunächst den Inhalt der Adresse 0 des EEPROMs ausliest, dessen Binärwert um 1 erhöht, wieder zurückschreibt und über die an P1 angeschlossene LED-Ausgabeplatine anzeigt. Das Ergebnis des

Bild 158: Testprogramm 51, Auslesen und Wieder-Zurückschreiben des EEPROM-Inhalts

```

0651          PRGM49: ;Prüfung und Auslesen des EEPROM's
0651 75B000    MOV     P3, #00      ; Portinhalt löschen
0654 7800     PRGM49W:MOV    R0, #0      ; Adresse im EEPROM 0 gewählt
0656 7900     MOV     R1, #0      ; Bank 0 des EEPROM's adressieren
0658 120135   LCALL  EEPROMR1    ; versucht ein Byte vom EEPROM zu lesen
                                           ; C ist gesetzt, wenn kein EEPROM erkannt
                                           ; wurde
065B F590     MOV     P1, A        ; Ausgabe des gelesenen Bytes auf P1
065D 92B7     MOV     P3.7, C      ; Ausgabe des Acknowledge-Bits auf P3.7
065F 78FF     MOV     R0, #0FFH    ; kurze Verzögerung zum Triggern
0661 D8FE     DJNZ   R0, $        ; Schleife 255 mal durchlaufen
0663 80EF     SJMP  PRGM49W      ; Endlosschleife
  
```

Bild 156: Testprogramm 49, kontinuierliches Auslesen des EEPROMs

```

0665          PRGM50: ;Schreiben eines Bytes in das EEPROM
0665 75B000    MOV     P3, #00      ; Portinhalt löschen
0668 E590     MOV     A, P1        ; Schalterkombination von P 1 lesen
066A 7800     MOV     R0, #0      ; Adresse 0 im EEPROM
066C 7900     MOV     R1, #0      ; Bank 0 des EEPROMs adressieren
066E FA       MOV     R2, A        ; Date von P 1
066F 120118   LCALL  EEPROMW1    ; 1 Byte über die I2C-Schnittstelle zum
                                           ; EEPROM übertragen
0672 92B7     MOV     P3.7, C      ; Ausgabe des Acknowledge-Bits auf P 3.7
0674 80FE     JMP     $
  
```

Bild 157: Testprogramm 50, Beschreiben des EEPROMs

skop, wobei darauf zu achten ist, daß jeweils mit einem 10:1-Tastteiler zu messen ist, da in der Mikrocontroller-Testschaltung als Pull-up-Widerstand lediglich der interne 50k Ω -Widerstand des Mikrocontrollers verwendet wird.

Voraussetzung für den Betrieb der Testprogramme 49 bis 53 ist, daß ein EEPROM des Types 2402-2416 auf der Mikrocontroller-Entwicklungsplatine eingesetzt ist.

6.10.4.1 Auslesen des EEPROMs

Abbildung 156 zeigt das Testprogramm 49, welches von der Adresse 0 des EEPROMs die dort gespeicherten Daten kontinuierlich ausliest und auf dem 8Bit-Port P 1 ausgibt, wo die LED-Anzeigen-Platine anzuschließen ist. Das Resultat der

```

0676          PRGM51: ;Auslesen des EEPROM-Inhalts und um 1 erhöhen
0676 75B000    MOV     P3, #00      ; Portinhalt löschen
0679 7800     MOV     R0, #0      ; Adresse im EEPROM 0 gewählt
067B 7900     MOV     R1, #00     ; Bank 0 des EEPROMs adressieren
067D 120135   LCALL  EEPROMR1    ; versucht ein Byte vom EEPROM zu lesen
                                           ; C ist gesetzt, wenn kein EEPROM erkannt
                                           ; wurde
0680 92B7     MOV     P3.7, C      ; Ausgabe des Acknowledge-Bits auf P3.7
0682 400D     JC     PRGM51F      ; Springe, wenn EEPROM nicht gefunden
0684 04       INC     A            ; Inhalt um 1 erhöhen
0685 F590     MOV     P1, A        ; neuen Inhalt anzeigen
0687 7800     MOV     R0, #00     ; Adresse 0 im EEPROM
0689 7900     MOV     R1, #00     ; Bank 0 des EEPROMs adressieren
068B FA       MOV     R2, A        ; Date laden
068C 120118   LCALL  EEPROMW1    ; 1 Byte über die I2C Schnittstelle zum
                                           ; EEPROM übertragen
068F 92B7     MOV     P3.7, C      ; Ausgabe des Acknowledge-Bits auf P3.7
0691 80FE     PRGM51F: JMP $
  
```

0693	PRGM52:		;8 Byte Daten über Ser ausgeben und anschließend 8 Byte lesen, ;zunächst Init der ser. Schnittst. 9600 Baud, 8 Daten
0693 F110	ACALL	L52INIT	; Init der seriellen Schnittstelle
0695 D1E7	PRGM52S:	ACALL L52REEPR	; Lesen der ersten 8 Byte des EEPROMs ; Rückgabe: C gesetzt, wenn Fehler erkannt
0697 5006	JNC	PRGM52A	; Springe, wenn kein Fehler
0699 743F	MOV	A, #'??	; Fehler beim Zugriff auf das EEPROM
069B F108	ACALL	L52AUSG	; Ausgabe des in A übergebenen Zeichens
069D 80FE	SJMP	\$; Endlosschleife
069F	PRGM52A:		; EEPROM ist erfolgreich ausgelesen worden
069F D1CE	ACALL	L52WSTR	; Ausgabe des 8 Byte Strings ; EEPROM-Inhalt ausgegeben und jetzt warten, ; bis 8 Zeichen empfangen werden
06A1 D1BB	ACALL	L52RSTR	; Einlesen von 8 Zeichen von der ser. Schnittst. ; Es sind 8 Zeichen empfangen worden, die im ; EEPROM abgelegt werden
06A3 D1F6	ACALL	L52WEEPR	; Schreiben der 8 Byte in das EEPROM ; Rückgabe: C gesetzt, wenn Fehler erkannt
06A5 5006	JNC	PRGM52B	; Springe, wenn kein Fehler
06A7 743F	MOV	A, #'??	; Fehler beim Zugriff auf das EEPROM
06A9 F108	ACALL	L52AUSG	; Ausgabe des in A übergebenen Zeichens
06AB 80FE	SJMP	\$; Endlosschleife
06ADD1E7	PRGM52B:	ACALL L52REEPR	; EEPROM ist erfolgreich beschrieben worden ; Lesen der ersten 8 Byte des EEPROM's ; Rückgabe: C gesetzt, wenn Fehler erkannt
06AF 5006	JNC	PRGM52C	; Springe, wenn kein Fehler
06B1 742D	MOV	A, #'-'	; Fehler beim Zugriff auf das EEPROM
06B3 F108	ACALL	L52AUSG	; Ausgabe des in A übergebenen Zeichens
06B5 80F6	SJMP	PRGM52B	; Wiederholung
06B7	PRGM52C:		; kein Fehler
06B7 D1CE	ACALL	L52WSTR	; Ausgabe des 8 Byte Strings
06B9 80DA	SJMP	PRGM52S	;

Bild 159: Testprogramm 52, Übertragung des EEPROM-Inhalts über die serielle Schnittstelle

0030	ZHL1	EQU 030H	; Zähler für die Ausgabe des Strings
0031	ZHL2	EQU 031H	; Zähler für die Ausgabe des Strings
0721	PRGM53:		;permanente Ausgabe der ersten 8 Byte im EEPROM mit Anzeige der ;Fehlversuche beim Zugriff auf das EEPROM. Es können mehrere ;Master quasi-gleichzeitig auf das gemeinsame EEPROM zugreifen, ;zunächst Init der ser. Schnittst. 9600 Baud, 8 Daten
0721 7464	MOV	A, #100	; Ausgabe des Strings nur jeden 10ten Zyklus
0723 F530	MOV	ZHL1, A	; Speichern des Wertes
0725 F531	MOV	ZHL2, A	; Speichern des Wertes
0727 F110	ACALL	L52INIT	; Init der seriellen Schnittstelle
0729 D1E7	PRGM53S:	ACALL L52REEPR	; Lesen der ersten 8 Byte des EEPROMs ; Rückgabe: C gesetzt, wenn Fehler erkannt
072B 500D	JNC	PRGM53A	; Springe, wenn kein Fehler
072D D53008	DJNZ	ZHL1, PRGM53B	; Springe, wenn keine Ausgabe erforderlich
0730 7464	MOV	A, #100	; Ausgabe des Strings nur jeden 10ten Zyklus
0732 F530	MOV	ZHL1, A	; Speichern des Wertes
0734 742D	MOV	A, #'-'	; Fehler beim Zugriff auf das EEPROM
0736 F108	ACALL	L52AUSG	; Ausgabe des in A übergebenen Zeichens
0738 80EF	PRGM53B:	SJMP PRGM53S	; Endlosschleife
073A	PRGM53A:		; EEPROM ist erfolgreich ausgelesen worden
073A D53106	DJNZ	ZHL2, PRGM53C	; Springe, wenn keine Ausgabe erforderlich
073D 7464	MOV	A, #100	; Ausgabe des Strings nur jeden 10ten Zyklus
073F F531	MOV	ZHL2, A	; Speichern des Wertes
0741 D1CE	ACALL	L52WSTR	; Ausgabe des 8 Byte Strings
0743 80E4	PRGM53C:	SJMP PRGM53S	; Endlosschleife

Bild 160: Testprogramm 53, Test der Multi-Master-Fähigkeit des I²C-Busses

Auslese- bzw. Schreibversuches wird wiederum auf dem Port-Pin P 3.7 dargestellt. Auch dieses Programm wird nach dem Reset des Mikrocontrollers nur einmal durchlaufen, um ein häufiges Schreiben zu verhindern.

Das Testprogramm 51 zeigt somit nach jedem Reset des Mikroprozessors über den eingebauten Taster oder nach jedem Einschalten der Spannungsversorgung an P1 einen um 1 erhöhte Bit-Kombination an.

6.10.4.4 Lesen und Schreiben eines 8Bit-Blockes

Abbildung 159 zeigt das Testprogramm 52, welches über die serielle Schnittstelle den Inhalt der ersten 8 Byte des EEPROMs ausgibt, auf die Eingabe von 8 beliebigen Zeichen über die serielle Schnittstelle wartet und anschließend diesen Block in das EEPROM zurückschreibt.

Für die korrekte Funktion des Programmes ist es erforderlich, daß ein PC an die serielle Schnittstelle der Mikrocontroller-Platine angeschlossen ist und auf diesem ein Terminal-Programm mit den Einstellungen für die serielle Schnittstelle von 9.600 Baud, 8 Daten-Bits, 1 Stopp-Bit und keine Parität vorgesehen ist.

Nach dem Reset der Mikrocontroller-Schaltung werden nach der Initialisierung der seriellen Schnittstelle des Mikrocontrollers die ersten 8 Bit des angeschlossenen EEPROMs über die serielle Schnittstelle zum angeschlossenen PC übertragen. Falls der EEPROM-Zugriff nicht möglich ist, überträgt das Programm ein „?“.

Nach erfolgreichem Auslesen des EEPROMs wartet das Programm auf die Übertragung von 8 Zeichen über die serielle Schnittstelle des angeschlossenen PCs. Nach dem Einlesen des letzten Zeichens schreibt das Testprogramm diese in das EEPROM zurück. Sollte hierbei ein Schreibfehler o. ä. auftreten, wird dieses über ein „?“ am PC angezeigt.

Nach dem Schreiben der 8 Bytes zum EEPROM benötigt dieses einige Millisekunden, um die internen Speicherzellen zu beschreiben. Während dieser Zeit antwortet es auf Anfragen vom Master mit einem negativen ACK. Das Testprogramm versucht nun, die 8Byte-Daten vom EEPROM zurückzulesen. Jeder nicht erfolgreiche Versuch wird über die serielle Schnittstelle mit einem „-“-Zeichen quittiert. Das Testprogramm gibt ca. 4 bis 5 „-“-Zeichen aus, bevor der Lesezugriff erfolgreich war.

Bei Einsatz eines Ferro-RAMs ist das EEPROM sofort nach dem Schreiben wieder auslesebereit, womit die Ausgabe der „-“-Zeichen entfällt.

Abbildung 161 zeigt die zum Testprogramm 52 gehörenden Unterprogramme. Im einzelnen übernehmen diese das Einle-

```

06BB      L52RSTR: ;Einlesen von 8 Zeichen von der seriellen Schnittstelle
06BB 7820      MOV    R0, #20H      ; Zeiger laden
06BD 7400      MOV    A, #0        ; Adresse 0 im EEPROM
06BF F6        MOV    @R0, A      ; später ansprechen
06C0 08        INC    R0          ; Zeiger um 1 erhöhen
06C1 7908      MOV    R1, #8         ; 8 Zeichen einlesen
06C3 F100      L52RSTRW:ACALL L52READ ; Warten, bis ein Zeichen empfangen
06C5 F6        MOV    @R0, A      ; Zeichen im RAM ablegen
06C6 F108      ACALL  L52AUSG   ; Ausgabe des in A übergebenen Zeichens
06C8 F590      MOV    P1, A        ; Kontrollausgabe auf LEDs
06CA 08        INC    R0          ; Zeiger um 1 erhöhen
06CBD9F6      DJNZ   R1, L52RSTRW; Schleife 8mal durchlaufen
06CD 22        RET
06CE      L52WSTR: ;Ausgabe des 8 Byte Strings
06CE 7422      MOV    A, #""         ; Anfang des Ausgabestrings
06D0 F108      ACALL  L52AUSG   ; Ausgabe des in A übergebenen Zeichens
06D2 7908      MOV    R1, #8         ; 8 Byte übertragen
06D4 E6        L52WSTRW:MOV  A, @R0    ; Date lesen
06D5 F108      ACALL  L52AUSG   ; Ausgabe des in A übergebenen Zeichens
06D7 08        INC    R0          ; Zeiger um 1 erhöhen
06D8 D9FA      DJNZ   R1, L52WSTRW; 8 Byte ausgeben
06DA 7422      MOV    A, #""         ; Ende des Ausgabestrings
06DCF108      ACALL  L52AUSG   ; Ausgabe des in A übergebenen Zeichens
06DE 740D      MOV    A, #00DH      ; Carriage Return
06E0 F108      ACALL  L52AUSG   ; Ausgabe des in A übergebenen Zeichens
06E2 740A      MOV    A, #00AH      ; Linefeed
06E4 F108      ACALL  L52AUSG   ; Ausgabe des in A übergebenen Zeichens
06E6 22        RET
06E7      L52REEPR:;Lesen der ersten 8 Byte des EEPROMs
06E7 7820      MOV    R0, #020H     ; Beginn des RAM-Puffers
06E9 7400      MOV    A, #0        ; Adresse im EEPROM = 0
06EB F6        MOV    @R0, A      ; Adresse im RAM hinterlegen
06EC 7901      MOV    R1, #1        ; 1 Byte schreiben (Start-Adresse im EEPROM)
06EE 7A08      MOV    R2, #8        ; 8 Byte lesen
06F0 7BA0      MOV    R3, #ADR_EEPR; Slaveadresse des EEPROMs
06F2 1200D7    LCALL  I2C_WR      ; I2C-Baustein schreiben und lesen
06F5 22        RET
06F6      L52WEEPR: ;Schreiben der 8 Byte in das EEPROM
06F6 7820      MOV    R0, #20H     ; Zeiger auf Anfang im RAM
06F8 7909      MOV    R1, #9        ; Adresse im EEPROM und 8 Bytes schreiben
06FA 74A0      MOV    A, #ADR_EEPR; Slaveadresse des EEPROMs
06FC 1200BF    LCALL  I2C_WANZ   ; n-Byte über die I2C-Schnittstelle ausgeben
06FF 22        RET
0700      L52READ: ;Warten, bis ein Zeichen empfangen und es dann lesen
0700 3098FD    JNB    RI $         ; Warten, bis Zeichen empfangen
0703 C298      CLR    RI          ; Empfangsflag löschen
0705 E599      MOV    A, SBUF     ; Zeichen lesen
0707 22        RET
0708      L52AUSG: ;Ausgabe des in A übergebenen Zeichens auf die serielle Schnittstelle
0708 3099FD    JNB    TI $         ; Warten, bis der Sendepuffer leer ist
070B C299      CLR    TI          ; Sendeflag löschen
070D F599      MOV    SBUF, A     ; Zeichen in den Sendepuffer schreiben
070F 22        RET
0710      L52INIT: ;Initialisierung der seriellen Schnittstelle für Daten, Ein und Ausgabe
0710 759850    MOV    SCON,#50H   ; Mode 1, (8Bit Daten) Receive Enable
0713 758920    MOV    TMOD,#20H  ; Timer 0 keine Funktion
                                ; Timer 1 Mode 2 C/T =0, Gate =0
0716 758700    MOV    PCON, #00H ; SMOD =0, Teiler /2
0719 758DFD    MOV    TH1,#100H-3 ; Nachladewert 3 = 11,0592 MHz/12/16/2/
                                ; 9600 Baud
071C D28E      SETB   TR1        ; Timer starten (TCON)
071E D299      SETB   TI          ; Sendeflag setzen, damit die Datenausgabe
                                ; beginnen kann
0720 22        RET

```

sen bzw. Ausgeben von 8 Zeichen von bzw. zur seriellen Schnittstelle, das Lesen bzw. Schreiben der ersten 8 Bytes des EEPROMs, das Empfangen bzw. Senden eines Zeichens sowie die Initialisierung der seriellen Schnittstelle.

6.10.4.5 Multimaster-Zugriff

Abbildung 160 zeigt das Programm 53, welches permanent die ersten 8 Zeichen des EEPROMs ausliest. Durch die Multimasterfähigkeit der I²C-Treiber erlauben diese den quasi gleichzeitigen Zugriff auf ein EEPROM. Hardwaremäßig sind dazu 2 Mikrocontroller-Platinen erforderlich, von denen auf nur einer das EEPROM bestückt ist. Von der zweiten Platine werden die Masse, SDA- und SCL-Steuerleitungen mit den entsprechenden Pins des EEPROMs verbunden. Somit erhält man einen Bus, an dem 2 Mikrocontroller und 1 EEPROM angeschlossen sind. Abschließend kann an eine der beiden Schaltungen ein PC mit einem Terminalprogramm angeschlossen werden.

Nach dem Einstellen des Programmes 53 an den Drehschalter und Einschalten der Versorgungsspannungen versuchen beide Mikrocontroller auf das gemeinsame EEPROM zuzugreifen. Für jedes erfolgreiche und nicht erfolgreiche Auslesen des EEPROMs wird jeweils eine der Zählvariablen ZHL1 bzw. ZHL2 um 1 heruntergezählt. Sobald eine Variable bei 0 angelangt ist, werden bei erfolgreichem Zugriff die 8 Byte ausgegeben bzw. bei nicht erfolgreichem Zugriff ein „-“-Zeichen über die serielle Schnittstelle ausgegeben. Dadurch läßt sich die Ausgaberate um den Faktor 100 verlangsamen, wodurch ein langsamerer und übersichtlicher Bildschirm Aufbau gewährleistet ist.

Im Verhältnis zu den erfolgreichen Zugriffen ist die Anzahl der nicht erfolgreichen Zugriffe, die mit einem „-“-Zeichen quittiert werden, relativ hoch. Dieses liegt unter anderem daran, daß während der Belegung des Busses durch den anderen Master relativ viele kurze erfolglose Zugriffsversuche stattfinden. Bedingt dadurch, daß beide Prozessoren asynchron auf den Bus zugreifen, ist die Anzahl der Fehlversuche recht unterschiedlich.

Das in Abbildung 160 dargestellte Testprogramm benutzt die bereits in Abbildung 161 beschriebenen Unterprogramme von dem Testprogramm 52.


Im 21. Teil der Mikrocontroller-Grundlagen-Serie beschreiben wir universelle Routinen, die es ermöglichen, beliebige I²C-Bausteine vom PC aus über die Grundschaltung anzusteuern. 

Bild 160: Zeigt die zum Testprogramm 52 (Abbildung 159) gehörenden Unterprogramme