ÉPREUVE ÉCRITE

Ministère de l'Education Nationale et
de la Formation Professionnelle

EXAMEN DE FIN D'ÉTUDES SECONDAIRES TECHNIQUES

Régime de la formation de technicien

Division: Électrotechnique

Section: Communication

BRANCHE: MICROÉLECTRONIQUE

SESSION: 2010 DATE: / /10 DURÉE: 3 h

1. Interrupts (11 Punkte)

a) Welchen Vorteil bringen Interrupts gegenüber dem Polling? (1)

b) Was kann passieren, wenn das Statusregister (**SREG**) in der Interrupt-Routine nicht gerettet wurde (Erklärung anhand eines konkreten Beispiels)? (2)

c) Was passiert, wenn ein spezifischer Interrupt auftritt, und die Interrupts global gesperrt sind? (2)

d) Der Zeitsignalsender DCF77 ist ein Langwellensender in Mainflingen, der die meisten funkgesteuerten Uhren im westlichen Europa mit der genauen Uhrzeit versorgt. Die Zeitinformationen werden als digitales Signal übertragen. Fertige Empfangsmodule liefern das sofort das digitale Signal, das mit Hilfe eines Controllers dekodiert werden kann.

Das DCF77-Signal liegt an **PD3** an. Eine Interruptroutine (ISR) dient der Dekodierung des Signals und soll bei einer Pegeländerungen aufgerufen werden. Durch Einlesen von **PD3** kann in der Routine festgestellt werden, ob eine steigende oder eine fallende Flanke auftrat.
Bevor die ganze Routine zur Dekodierung geschrieben wird, soll eine Minimalversion bei einer steigendenden Flanke eine LED an PD0 eingeschalten. Bei einer fallenden Flanke wird die LED wieder ausgeschaltet. Das Blinken der LED dient der Kontrolle des Empfangssignals.
Notiere die Initialisierungsbefehle in den zwei relevanten SF-Registern, die für externe Interrupts vorgesehen sind. Zeichne das Flussdiagramm zu dieser Minimalversion der Interruptroutine. (2+4)

2. Serielle Schnittstelle (14 Punkte)

Ein an den Controller ATmega32 angeschlossene 5 Meter lange Schnittstellenleitung soll mit einem sogenannten „Loopback" Test überprüft werden. Dazu wurden am Ende des Schnittstellenkabels die Leitungen TXD und RXD zum Testen miteinander verbunden. Es soll jetzt ein einziges Mal mit dem Controller interruptgesteuert der Buchstabe 'A' (0x41) gesendet und auch interruptgesteuert empfangen werden (1 Mbit/s, 7N1). Ist die Schnittstelle in Ordnung, so soll eine LED an PA0 aufleuchten. Dies passiert nur, wenn das

empfangene Zeichen und das gesendete Zeichen gleich sind, und wenn kein Fehlerflag gesetzt wurde. Das Hauptprogramm ist arbeitslos. Der Controller arbeitet mit 16 MHz.

a) Ist die Baudrate sinnvoll gewählt worden? Begründe. (1)

b) Notiere alle! benötigten Initialisierungen (Assemblercode und Direktiven mit Kommentaren). (7)

c) Schreibe die kommentierten Interrupt-Routinen. (6)

3. AD/DA-Wandler (30 Punkte)

a) Skizziere die Schaltung für einen AD-Wandler nach dem Wägeverfahren. (3)

b) Erkläre das Verfahren ausführlich anhand der Skizze aus a) und eines zu zeichnenden Flussdiagramms. (5)

c) Skizziere die Schaltung eines 3-stelligen D/A-Wandlers der an 3 Pin eines ATMega32 (PC0 bis PC2) angeschlossen wird. (3)

d) Welche acht Spannungen können am Ausgang des D/A Wandlers aus c) abgegriffen werden (VCC = 5V)? (2)

e) Am Port A des ATMega32 (16 MHz) sind 4 Temperatursensoren (Uref = AVCC) angeschlossen (PA0 bis PA3). Ungefähr einmal pro Sekunde (es steht das Unterprogramm W1s zur Verfügung) sollen die 4 Sensoren mittels Polling mit 8-Bit-Genauigkeit eingelesen werden. Das Polling soll in einem kleinen Unterprogramm erfolgen. Ein Schalter S0 an PB0 (interner Pull-UP) entscheidet über die Betriebsart. An PORTC sind 8 LEDs angeschlossen.

S0 = 0:
Alle Werte werden im SRAM abgespeichert (die Werte werden immer wieder überschrieben). Alle LEDs an PORTC sind aus.

S0 = 1:
Aus den vier Werten wird zusätzlich ein Mittelwert gebildet. Dieser wird an PortC ausgegeben.

Hinweis zur SRAM-Adressierung: Die direkte Adressierung vereinfacht das Programm.
Hinweis zur Mittelwertbildung: Die vier 8 Bit-Werte müssen wegen möglicher Überläufe in einem 16-Bit-Register addiert werden. Die Division durch vier erfolgt durch zweimaliges Verschieben des Doppelregisters um eine Stelle nach rechts (carry beachten!).

Notiere alle! benötigten Initialisierungen (Assemblercode und Direktiven mit Kommentaren). (4)
Schreibe das kommentierte Unterprogramm. (3)
Schreibe das kommentierte Hauptprogramm. (10)

4. Timer (5 Punkte)

Ein ATMega32 soll ein quarzgenaues Rechtecksignal mit einer Frequenz von 1000 Hz an PA0 ausgeben können (Quarz = 16 MHz, Vorteiler = 64). Die Impulsdauer soll dabei genau 0,2 ms (entspricht einer ISR-Frequenz von 5000 Hz!) betragen. Da Impuls- und Pausendauer nicht gleich gross, sind muss der voreingestellte Wert in der Interruptroutine bei jedem neuen Aufruf der Routine verändert werden.

Der momentane Zustand von PA0 gibt dabei vor, welcher Wert für die Voreinstellung verwendet werden soll.

Das Hauptprogramm lässt eine LED an PB0 im Sekundentakt blinken (entsprechende Zeitschleife steht in SR_TIME_16M.asm zur Verfügung).

a) Skizziere das Rechtecksignal (inkl. Achsen!). (1)

b) Berechne die beiden Werte für die Voreinstellung (Startwerte des Timers). (2)

c) „Da Impuls und Pausendauer nicht gleich gross sind, muss der voreingestellte Wert in der Interruptroutine bei jedem neuen Aufruf der Routine verändert werden."
Beschreibe kurz eine Lösung (kein Programm) wie man dies praktisch bewerkstelligen könnte. (2)