Code branche	Ministère de l'Education nationale et de la Formation professionnelle EXAMEN DE FIN D'ETUDES SECONDAIRES TECHNIQUES Régime de la Formation de Technicien - Session 2012/2013	
MICEL		
Épreuve écrite	*Branche*	*Division / Section*
Durée épreuve **3 h**	**MICROÉLECTRONIQUE**	**TECAN**
Date épreuve 6.6.2013		

1. Externer Interrupt (16 Punkte)

Es soll ein interruptgesteuerter Frequenzzähler programmiert werden.

Dazu werden während einer Sekunde, beziehungsweise während einer Millisekunde mit Hilfe einer Interrupt-Routine die steigenden Flanken am Interrupt-Eingang **INT2** gezählt. Als globaler Zähler soll das Arbeitsregister **r20** verwendet werden.

Die Unterprogramme **W1s** und **W1ms** sind schon in der Zeischleifen-Bibliothek **SR_TIME_16M.asm** vorhanden und müssen nur eingebunden werden. Die Interrupts dürfen nur während der Zeitschleife erfolgen!

Das Umschalten der Torzeit Zeit während der gemessen wird, erfolgt mit einem Schalter an Port B (**PB0**):

S1 = 0 → Torzeit = 1 Sekunde
S1 = 1 → Torzeit = 1 Millisekunde

Die Zählerausgabe erfolgt an 8 LEDs (Port C) und soll im im Hauptprogramm erfolgen. Die Zeitschleifen befinden sich ebenfalls im Hauptprogramm.

Eine zusätzliche LED an PORTB (**PB1**) leuchtet, wenn der Schalter betätigt wurde.

a) Berechne die beiden Messbereiche des Zählers in Hertz. (2)

b) Notiere alle! benötigten Initialisierungen (Assemblercode mit Kommentaren). Nutze Maskierungen beziehungsweise die Befehle **cbi** und **sbi** damit nicht benötigte Bits in SF-Registern unverändert bleiben. (5)

c) Bei welchen SF-Registern dürfen die Befehle **sbi** und **cbi** eingesetzt werden? (1)

d) Schreibe das kommentierte Hauptprogramm und die kommentierte Interrupt-Service-Routine. (8)

2. Serielle Schnittstelle (16 Punkte)

Ein mit 16 MHz getakteter ATmega32 erhält über die serielle Schnittstelle (Datenformat: 8E1, Bitrate: 38400 bit/s) Kommandos aus zwei ASCII-Zeichen.

Eine Interrupt-Service-Routine, die durch das Eintreffen des ersten Zeichens des Kommandos aktiviert wird, wertet die Kommandos aus. Das Hauptprogramm hat keine Aufgaben und besteht aus einer Endlosschleife. Es wird keine Fehlerkontrolle durchgeführt.

Die Interrupt-Service-Routine, wird durch das erste erste Zeichen aufgerufen. Innerhalb der Routine wird mittels Polling auf das zweite Zeichen gewartet. Ist dieses eingetroffen, so werden folgende Aktionen ausgeführt:

Kommando:	Aufgaben
"C1"	LED an Port A 0 einschalten ASCII-Zeichen 'A' als Bestätigung zurücksenden
"C2"	LED an Port A 0 ausschalten ASCII-Zeichen 'A' als Bestätigung zurücksenden

Das Zurücksenden des ASCII-Zeichen **'A'** als Bestätigung erfolgt ebenfalls in der Interrupt-Service-Routine. Auch hier ist das Polling zu verwenden.

a) Zeichne den vollständigen zeitlichen Verlauf des Signals (SDU) des ersten Kommandos („C1“, 2 Zeichen!) an der Schnittstelle (EIA232!). (3)

b) Notiere alle! benötigten Initialisierungen (Assemblercode mit Kommentaren). (5)

c) Zeichne das ausführliche Flussdiagramm der Interrupt-Service-Routine. (8)

3. Digitale Meßtechnik (8 Punkte)

a) Erkläre die Begriffe “Wertdiskretisierung“ und „Zeitdiskretisierung“. (2+2)

b) Erkläre die Begriffe “Amplitudenunsicherheit" und "Aperturzeit“ anhand einer Skizze. (2)

c) Was ist der Aliasing-Effekt und wie lässt er sich verhindern? (2)

4. D/A-Wandler (5 Punkte)

Skizziere die Schaltung für ein R-2R-Netzwerk mit zwei Stellen (2^0 und 2^1) des Dualcodes. Erstelle eine Tabelle für alle vier Fälle und errechne die jeweiligen Ausgangsspannungen wenn $U_E = 3$ V beträgt. (5)

5. A/D-Wandler und Timer0 (15 Punkte)

Es sollen innerhalb einer Millisekunde zwei Spannungen am A/D-Wandler eingelesen werden. Das Wandeln der analogen Werte (Polling) erfolgt dazu in einer Interrupt-Routine des Timer0.

a) Der Timer0 soll mit Hilfe des Overflow-Interrupts so eingestellt werden, dass er nach 256 Zählschritten (keine Voreinstellung!) genau jede Millisekunde einen Interrupt auslöst. Um dies zu erreichen wird ein Quarz von 16,384 MHz verwendet. Berechne den zu verwendeten Vorteiler. Notiere deine Berechnungen. (1)

b) Notiere alle! Initialisierungen die für den Timer benötigt werden. (3)

c) Zum Einlesen der beiden Spannungen werden die beiden Eingänge **PA6** und **PA7 am Port A** verwendet. Die Werte werden per Polling mit 10 Bit gewandelt. Als Referenzspannung wird die Betriebsspannung des Controllers (+5 V) verwendet. Notiere die Assemblerzeilen, die zur Initialisierung des A/D-Wandlers benötigt werden. (1)

d) In der Timer-Interrupt-Routine werden die beiden 10-Bit A/D-Werte gewandelt, eingelesen und dann im SRAM in zwei getrennten Tabellen von je 256 Byte abgespeichert (indirekte Adressierung). In den beiden Tabellen mit den symbolischen Namen **ADCPA6** und **ADCPA7** wird immer zuerst das HighByte und dann erst das LowByte abgespeichert. Notiere alle Assembleranweisungen und Befehle die benötigt werden um die Tabellen nutzen zu können. (2)

e) Notiere alle Assemblerzeilen, die in der Interrupt-Service-Routine benötigt werden um die Wandlungen vorzunehmen und die Werte abzuspeichern. (6)

f) Nachdem jeweils 256 Byte abgespeichert wurden, laufen beide Tabellen über. Berschreibe eine Lösungsmöglichkeit, wie man dieses Problem lösen könnte (keine Assemblerzeilen!)? (2)