Yuye Andreas Meier, INB-2, 71610

Dokumentation – SNP

LED Design:

6 genutzte Zeilen: (Bei 0 leuchtet keine LED in jeweiliger Zeile usw.)

PORTC0 mit Sekunden-Einer (0-9)

PORTC1 mit Sekunden-Zehner (0-9)

PORTD4 mit Minuten-Einer (0-9)

PORTD5 mit Minuten-Zehner (0-9)

PORTB0 mit Stunden-Einer (0-9)

PORTB1 mit Stunden-Zehner (0-2)

Auswahlbegründung:

- Einfach darzustellen
- Leicht und schnell für Betrachter zu lesen
- Übersichtlich (andere Darstellungen können meiner Meinung nach schnell unübersichtlich werden wegen Platzmangel)

Funktionsweise:

Variablen

sekunde1	Sekunden-Einer
sekunde2	Sekunden-Zehner
minute1	Minuten-Einer
minute2	Minuten-Zehner
stunde1	Stunden-Einer
stunde2	Stunden-Zehner
counter	Schlafmoduszähler(30s)
ledcounter	Zähler für Zeitpunkt zum Ansteuern der Matrixzeilen

Funktionen

Inittimer0	Initialisiert den Timer0 mit einem Prescaler von 64 (8MHz/64= 125000Hz) aktiviert Overflow Interrupts und setzt Register auf 256-125 (es entstehen 1000OVF, da 1000x125=125000)
ISR(TIMER0_OVF_vect)	Timer0 OVF Interrupt Händler, bei Auftreten eines OVF, setzt Register wieder auf 256 – 125 Und erhöht den ledcounter (zählt von 0 bis 12)
Inittimer2	Initialisiert Timer2 mit einem Prescaler von 1024, dieser wird asynchron betrieben (RTC-Real Timer Counter, externer Uhrenquarz) und im CTC (Clear on Compare) Modus betrieben, d.h. wenn der OCR2 Wert von 31

	(32768Hz/1024 – 1=31) erreicht wird, wird der Compare Interrupt Händler aufgerufen und das Register zurückgesetzt
ISR(TIMER2_COMP_vect)	Compare Interrupt Händler von Timer2, verantwortlich zum Zählen der Uhr, d.h. er erhöht die Sekunden, Minuten und Stunden, sowie den counter für den Schlafmodus
ISR(INT0_vect)	Externer Interrupt Händler, dieser setzt den counter für den Schlafmodus immer auf 0 zurück und stellt den externen Interrupt wieder auf fallende Flanke (zeitweise lowlvl gesteuert, da sonst ext. Interrupt nicht im sleepmode funktioniert)
Aufruf(int zeit)	Aufruf wird von der Funktion zeitanzeige() aufgerufen um das Schieberegister zu füllen, dies geschieht über Clock Signale und jeweils das Einfügen einer 0 (leuchten) oder 1 (nicht leuchten), bei 1 leuchtet die Diode nicht da die LEDS sperren wenn von Kathoden Seite auf high gestellt ist.
Zeitanzeige()	Über ledcounter gesteuert, d.h. LEDs werden über den Timer0 gesteuert: Z.B. wenn ledcounter=0 dann wird Aufruf(sekunde1) aufgerufen und die Zeile angemacht, dann wenn ledcounter=2 ist wird sie wieder ausgemacht, Aufruf(sekunde2) getätigt und die neue Zeile angemacht usw. (An machen einer Zeile = 1 setzen, andersrum 0(Anodenseite)) Und nach Aufruf der Zeitanzeige jeweils immer Strobe an und aus am Ende um die Daten vom 8Bit Shift Register ins 8Bit Storage Register zu schieben.
Clear()	Schaltet vor sleepmode sicherheitshalber alle Zeilen aus
Int main()	Mainfunktion wird als allererstes vom Programm aufgerufen: Zunächst ausschalten des Watchdogs um Energie zu sparen(dieser kümmert sich um Errors und erzeugt Interrupt bei Fehlern, damit der Prozessor nicht abstürzt) Danach werden die Ausgänge der Ports festgelegt, danach der externe Interrupt auf PORTD2 (Eingang) auf 1gesetzt und über sein Register initialisiert mit zunächst fallender Flanke, danach werden die Timer initialisiert und die Startzeit gesetzt, danach wird der Schlafmodus (powersave) gesetzt und es werden globale Interrupt aktiviert (sei()), danach geht das Programm in die while Schleife, in welche sich das Register der Timer füllt, ab 30Sekunden wird ein Schlafmodus geschalten, sonst wird die Uhrzeit über zeitanzeige() ausgegeben

Yuye Andreas Meier, INB-2, 71610

Sleepmode (rund 10Mikroampere)

Sleepmode: power save, da im power down Modus der Timer2 nicht mehr an ist und in andren Sleepmodis, wie dem Idle Modus sind zu viele unnötige Clocks aktiviert, die im Schlafmodus Energie rauben.

Funktionsweise: Im Sleepmode powersave wacht die Uhr wegen des sekündigen Timer2 Interrupts für einen kurzen Moment auf, um die Uhrzeit zu bestimmen (kurzzeitige Erhöhung des Stromverbrauchs am Messgerät) und wechselt dann wieder in den sleepmode. In diesem kurzen Zeitraum ist es nur möglich einen Flanken gesteuerten, externen Interrupt auszulösen, deshalb verwendet man während des Sleepmodes den lowlvl Interrupt, damit es auch im Schlafmodus funktioniert und im aktiven Modus bei Betätigen des externen Interrupt wechselt man wieder auf die Rückflanke.