Interrupts

Julien Stumpf

Gliederung

1. Was sind Interrupts?
2. Wie funktioniert das?
3. Wozu Interrupts?
4. Tipps

5. Interrupts beim Atmega32

6. Fragen
7. Quellen

Was sind Interrupts?

- Von engl. „to interrupt“ → unterbrechen

- Unterbrechen Programmablauf
- Führen zeitkritische Operation aus
- Lassen Programm weiterlaufen

Wie funktioniert das?

Programmablauf:

…
#1012
#1013
#1014

Interrupt Service Routine (ISR)

#1015
#1016
#1017
….

(IRQ = Interrupt Request)

Wozu Interrupts?

Beispiele für Auslösungen:

- Änderung des Eingangspegels an PIN
- Timer abgelaufen

- Messung am AD-Wandler abgeschlossen
- Serielle Übertragung abgeschlossen
- …
- …

Tipps

Interruptroutine kurz halten

- Andernfalls können Interrupts „verschluckt“ werden
- Keine Schleifen, Berechnungen, Ausgaben…
- Steuervariablen/Flags verwenden (Stichwort: `volatile`)

Atmega32

Atmega32

Interrupts werden nur ausgeführt wenn:

- „`#include <avr/interrupt.h>`“
- Interrupts global aktiviert wurden „`sei();`“
- Bestimmtes Maskenbit gesetzt wurde
- Der Interrupt auftritt

Atmega32

Interrupt Vectors in ATmega32

Table 18. Reset and Interrupt Vectors

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$002	INT0	External Interrupt Request 0
3	$004	INT1	External Interrupt Request 1
4	$006	INT2	External Interrupt Request 2
5	$008	TIMER2 COMP	Timer/Counter2 Compare Match
6	$00A	TIMER2 OVF	Timer/Counter2 Overflow
7	$00C	TIMER1 CAPT	Timer/Counter1 Capture Event
8	$00E	TIMER1 COMPA	Timer/Counter1 Compare Match A
9	$010	TIMER1 COMPB	Timer/Counter1 Compare Match B
10	$012	TIMER1 OVF	Timer/Counter1 Overflow
11	$014	TIMER0 COMP	Timer/Counter0 Compare Match
12	$016	TIMER0 OVF	Timer/Counter0 Overflow
13	$018	SPI, STC	Serial Transfer Complete

Datenblatt
Seite 44

Atmega32

```
#include <avr/io.h>
#include <avr/interrupt.h>       //Stellt Interrupt-Funktionen bereit
#include <avr/iom32.h>           //Erleichtert Benennung der Interrupts
//...
volatile int Interrupt_erkannt;
//...
ISR(INT2_vect)  //Interrupt für INT2 ("_vect" ranhengen)
{
      cli();     //deaktiviert Interrupts
      Interrupt_erkannt = 1;
}
//...
int main(void)
{
      Interrupt_erkannt = 0;
      GICR |= (1<<INT2);    //Aktiviert INT2 als Interrupt PIN
      MCUCSR |= (1<<ISC2);  //Interrupt INT2 bei steigender Flanke
      sei();     //Interrupts global aktivieren
      while(1)
      {
            //TODO...
            if(Interrupt_erkannt = 1)
            {
                  //Interrupt hat funktioniert
                  //DOOTHERSTUFF
            }
      //...
      }
}
```

Fragen?

Quellen

- Datenblatt des Atmega32
- mikrocontroller.net
- rn-wissen.de
- Wikipedia.de

Ende