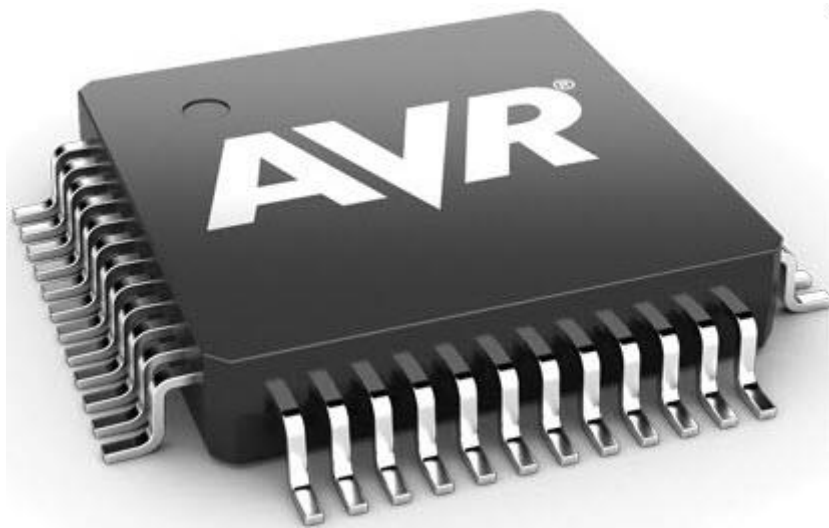


Microcontrollers Avans Ti



Aaron Israëls en Maarten Nieuwenhuize

https://github.com/Maartenwn/Microcontrollers-Aaron_Isra-Is_en_Maarten-Nieuwenhuize

A1

The good: Interrupts kunnen events afhandelen zonder dat de CPU veel kracht hiervoor nodig heeft. Een interrupt blocked de CPU niet.

The bad: de code zal mogelijk al zijn tijd besteden aan het uitvoeren van de interrupts, en zal dan weinig tijd over hebben voor zijn standaard routine, waardoor hij zal "starven". bijvoorbeeld als 2 interrupts elkaar constant triggeren zal de main loop geen CPU tijd meer hebben.

The Ugly: wanneer een variabel verandert wordt door interrupts kan je er niet langer vanuit gaan dat deze constant zullen blijven; je weet niet wanneer de interrupts uitgevoerd zullen worden. het is altijd maar de vraag of je code van A naar B kan komen zonder onderbroken te worden.

B1

Voor de main loop worden de interrupts aangezet. En de globale interruptflag aangezet.

De main loop toggled port 7 dit gebeurt door een xor operatie. Als links bit 7 aanstaat en rechts bit 7 aanstaat resulteert dit in dat bit 7 uit gaat. Als links bit 7 uitstaat en rechts bit 7 aanstaat resulteert dit in dat bit 7 aangaat.

1

Als je op knop D0 klikt wordt deze code uitgevoerd:

```
PORTD |= (1<<5);
```

Hier wordt PORTD vergeleken met 0x10. Doordat bij 1<<5 altijd de 5de bit aanstaat wordt or= operatie op bit 5 altijd 1.

Als je op knop D1 klikt wordt deze code uitgevoerd:

```
PORTD &= ~(1<<5);
```

1 << 5 zorgt dat de 5de bit aangaat. de ~ operator flipt alle bits over dit betekent dat het getal 0b11101111. Omdat de &= operator gebruikt wordt en bit 5 altijd 0 is, resulteert dit dat de 5 bit na de & operatie altijd 0 is.

B2

Gedaan Looplicht met interrupts <https://youtu.be/MbOt8OOggTo>

B3

Gedaan -> 7seq ophogen met interrupts <https://youtu.be/PFpU6lJAsgA>

B4

Gedaan -> 7seq met patroon <https://youtu.be/jSylPbVKNgg>