# Microcontroller programmeren

Inhoud

[Microcontroller programmeren 1](#_Toc191487409)

[1 Microcontrollers en programmeren 2](#_Toc191487410)

[1.1 voorbereiding 2](#_Toc191487411)

[1.2 practicum 5](#_Toc191487412)

[2 GPIO 6](#_Toc191487413)

[2.1 voorbereiding 6](#_Toc191487414)

[2.2 Practicum 6](#_Toc191487415)

[3 Interrupts 9](#_Toc191487416)

[3.1 Voorbereiding 9](#_Toc191487417)

[3.2 Practicum 10](#_Toc191487418)

## 1 Microcontrollers en programmeren

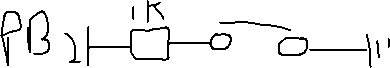
### voorbereiding

4. i/o staat voor input/output. Ermee word bedoet pins waar je data mee kan ontvangen of sturen. Peripherals zijn externe hardware waar data naar gestuurd kan worden of van kan worden ontvangen.
5. 1. ADCn: een chip die analoge naar digitale signalen omzet
   2. DACn: een chip die digitale naar analoge signalen omzet
   3. RTC: een klok
   4. TCAn: timer
6. 42
7. A screenshot of a computer program

   Description automatically generated
8. Input
9. PB3 / 7
10. De led heeft een 1k voorschakelweerstand en zit verbonden aan pb3. Als je de led naar ground verbind gaat hij aan



1. 0 want dan zit de led verbonden aan vcc
2. Pb2
3. Als je m indrukt gaat de i/o lijn naar gnd
4. Een restistor om te voorkomen dat de waarde gaat zweven



1. 0 want de i/o pin word verbonden aan gnd
2. Dat als je een knop indrukt dat hij eerst een aantal keer heel snel heen en weer gaat tussen aan en uit omdat het contact “stuiterd”
3. Met een condensator
4. A diagram of a machine

   Description automatically generated
   1. 7 segmenten display
   2. Oled display aansluiting
   3. Motor driver
   4. Knopjes
   5. Potmeter
   6. Sd kaart reader
   7. I2c connectie
   8. P1 connectie

|  |  |
| --- | --- |
| P1: 1 | Pb2 |
| 2 | Pf4 |
| 3 | Pf5 |
| 4 | Pa2 |
| 5 | Pa3 |
| 6 | Pa4 |
| 7 | Pa5 |
| 8 | Pa6 |
| 9 | Pa7 |
| 10 | Gnd |
| 11 | Pb0 |
| 12 | Pb1 |
| 13 | Pf2 |
| 14 | Pf3 |
| P2: 1 | vcc |
| 2 | Pd6 |
| 3 | Pd3 |
| 4 | Pd2 |
| 5 | Pd1 |
| 6 | Pd0 |
| 7 | Pd7 |
| 8 | Pd5 |
| 9 | Pd4 |
| 10 | Gnd |
| 11 | Pe3 |
| 12 | Pe2 |
| 13 | Pe1 |
| 14 | Pe0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Functie | Pin smu | Pin curiosity |
| Led1 | Indicatie led sd reader | P2.8 | Pd5 |
| Led2 | Indicatie led | P2.14 | Pe0 |
| Sw1 | Software defineerbare switch | P2.11 | Pe3 |
| Sw2 | Software defineerbare switch | P2.12 | Pe2 |
| Sw3 | Software defineerbare switch | P2.13 | Pe1 |
| pot | Potmeter te lezen met software | P2.9 | Pd4 |

1. Een korte pauze
2. \_delay\_ms(tijd in ms);

### practicum

2. Er moet “while”staan
3. Met dir an je aangeven of een pin input of output is  
   met out kan je een pin op hoog of laag zetten
4. Dir: initialisatie  
   out: loop
5. Dirset en outset
6. PORTE.DIR |= PIN0\_bm;  
   PORTE.OUT |= PIN0\_bm;
7. Outclr
8. Met een delay

## GPIO

### voorbereiding

1. 1. = maakt de variabele een bepaalde waarde
   2. ~ NOT operator. Draait een bit / bitreeks om.
   3. | OR operator. Als van 2 bitreeksen 1 van de 2 bits 1 is word het resultaat 1
   4. & AND operator. Alleen als van beide bitreeksen de bit 1 is word het resultaat 1
   5. ^ XOR operator. Alleen als 1 van de 2 bits uit een bitreeks 1 is word het resultaat 1
2. Dat zijn conditionele operatoren
3. 1111 1101
4. 1
5. 1100 1111
6. 1100 1101
7. 0000 0000
8. 1100 1101
9. 1
10. Or operator 1010 1010 | 0100 0000 🡪 1110 1010
11. XOR operator 1010 1010 ^ 1000 0000 🡪 0010 1010
12. Exact hetzelfde
13. |=
14. ^=
15. ^=

### Practicum

1. PORTB.DIR &= ~PIN2\_bm;
2. PORTB.IN & PIN2\_bm
3. 1 is indrukken 0 is los
4. PORTB.IN & PIN2\_bm
5. PORTB.IN is de hele bitreeks van de inputwaarden van de port b. PIN3\_bm is een bitmask van pin3 dus als de bit in de bitreeks 1 is, en de juiste bit (de enige hoge bit in de bit mask) hoog is, is de input 1
6. PORTB.DIR |= PIN3\_bm;   
     
   while (1) {  
   if(PORTB.IN & PIN2\_bm)  
    PORTB.OUTSET = PIN3\_bm;  
    } else {  
    PORTB.OUTCLR = PIN3\_bm;  
    }  
   }
7. Ja maar hij blijft aan
9. Zet een pull up resistor aan of uit per pin
10. PORTB.PIN2CTRL, PORT\_PULLUPEN\_bm;
11. PORTB.PIN2CTRL = PORT\_PULLUPEN\_bm;
12. Nu werkt het wel goed
13. Dan wacht je op de input van een extern apparaat inplaats van dat je een trigger (interrupt) krijgt
14. Omdat je niks anders kan toen terwijl je o peen input aan het wachten bent
15. Met een interrupt
16. Timer
17. Gedaan





24. A word elke druk groter










36. int main(void)

{

SYSTEM\_Initialize();

while (1) {

if(SW0\_GetValue()){

LED0\_SetHigh();

} else {

LED0\_SetLow();

}

}

}

2. if(SW0\_GetValue()) { LED0\_SetHigh(); } else { LED0\_SetLow(); };

## 3 Interrupts

### 3.1 Voorbereiding

1. constant bij de deur kijken
2. 1: je staat constant bij de deur dus je kan niks anders  
   2: Het kost enorm veel moeite
3. Je hoeft niet constant bij de deur te staan aangezien je een seintje krijgt als je pakketje er is. Hierdoor kan je iets anders / niks doen om uit te rusten.
4. Polling = geen deurbel, interrups = wel deurbel
5. Er is een externe pheripheral die een seintje stuurt als een bepaalde trigger afgaat
7. GPIO, timer, rtc, event system, adc, dac, uart, spi, i2c
8. PORTB.PIN2CTRL = PORT\_PULLUPEN\_bm;  
   PORTB.PIN2CTRL |= PORT\_ISC\_FALLING\_gc;
9. ISR(PORTB\_PORT\_vect) {

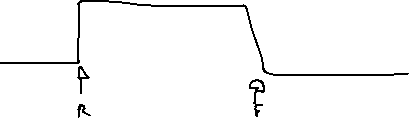
if (PORTB.INTFLAGS & (1 << PIN2\_bp)) {

PORTB.INTFLAGS = (1 << PIN2\_bp);  
 // HIER WORD CODE UITGEVOERD  
 PORTB.OUTTGL = (1 << PIN3\_bp);

}

}

1. Een nested interrupt is een interrupt die onderbroken kan worden door een hogere prioriteit interrupt. Dit is niet altijd maar soms wel wenselijk
2. Andere interrupts moeten wachten tot de isr klaar is dus als het sneller klaar is kunnen de volgende interrups uitgevoerd worden
3. Dat zijn geheugenadressen waar de microcontroller naar toe springt als interrupt getriggered word. Er zijn er 33
4. Rising is wanneer de trigger “ingedrukt”word falling is als hij “losgelaten” word



1. Rising edge trigger aanzetten, falling edge uitzetten
3. Idle, standby, power-down
4. Ja, anders stopt de microcontrooler met de code uitvoeren en crasht / reset hij
5. Idle
6. Pin change interrups,
7. 6

### 3.2 Practicum