

Hoofdstuk 5 – De microcontroller programmeren

Thuisopdracht 1

- a) Inputs voor de microcontroller zijn: schakelaars, P1 poort
- b) Outputs voor de microcontroller zijn: OLED display, 7-segmentsdisplay (via decoder), LED
- c) Alle pinnen met blauw of groen rechtsonder in figuur 2.1 van de pinout (datasheet):

 Digital Function Only

 Analog Function

- d) Digital function only, betekent dat deze pin alleen digitale signalen kan verwerken. De pin leest en schrijft dus alleen nullen en enen.
- e) Deze pinnen kunnen naast digitale signalen ook analoge signalen inlezen en uitsturen.

Thuisopdracht 2

- a) $10 \times 16 + 5 = 165$
- b) $3 \times 16 + 15 = 63$
- c) $1 \times 16^2 + 3 \times 16 + 12 = 316$
- d) $1100\ 1010 = 12$ en $10 = 0xCA$
- e) $10\ 1110 = 2$ en $14 = 0x2E$
- f) $1\ 1011 = 1$ en $11 = 0x1B$
- g) $0100\ 0000 = 1000000$
- h) $0111\ 1011 = 1111011$
- i) $1110\ 0010\ 1000$

Thuisopdracht 3

- a) PE3
- b) PORTE PIN3
- c) Ja
- d) Er hoeft niets ingesteld te worden, want de pinnen zijn standaard input (dus staat al goed). De pull-up is extern aanwezig en hoeft dus niet softwarematig aangezet te worden
- e) Haal uit het schema wanneer de schakelaar hoog of laag is.

SW1	Waarde
Ingedrukt	0
Niet ingedrukt	1

Thuisopdracht 4

- a) PEO
- b) PORTE PIN0
- c) `PORTE.DIR = PORTE.DIR | PIN0_bm;`
- d)

```
#include <avr/io.h>

int main()
{
    PORTE.DIR = PORTE.DIR | PIN0_bm;

    while(1)
    {
    }
}
```
- e) LED gaat aan als pin hoog ('1') is.

Labopdracht 1

- a) ..
- b) SW2 zit aan pin PE2. Deze schakelaar heeft een externe pull-up, dus er hoeft niets ingesteld te worden om de pin goed te gebruiken.
- c) LED2 zit op PE0:
`PORTE.DIR = PORTE.DIR | PIN0_bm;`
- d) Zelfde programma als in de reader, alleen zijn nu de regels onder *if* en *else* omgewisseld. Bovendien is er een andere inputpin (achter *if*):

```
#include <avr/io.h>

int main()
{
    PORTE.DIR = PORTE.DIR | PIN0_bm;

    while(1)
    {
        if (PORTE.IN & PIN2_bm)
        {
            PORTE.OUTSET = PIN0_bm;
        }
        else
        {
            PORTE.OUTCLR = PIN0_bm;
        }
    }
}
```

Labopdracht 2

- a) $Z = \bar{A}\bar{B}\bar{C} + AB\bar{C} + ABC$
- b) $Z = B\bar{C}(\bar{A} + A) + AB(\bar{C} + C) = B\bar{C} + AB = B(A + \bar{C})$
- c) SW1 = PE3, SW2 = PE2, SW3 = PE1
Vergeet niet dat de switches geïnverteerd zijn (dus 0 wanneer ingedrukt en 1 wanneer niet ingedrukt). Voor de formule betekent dit dat je alle variabelen ook moet inverteren, dus overal waar B staat, programmeer je !B, etc.

```
#include <avr/io.h>

int main()
{
    PORTE.DIR = PORTE.DIR | PIN0_bm;

    while(1)
    {
        if (!(PORTE.IN & PIN2_bm) && (!(PORTE.IN & PIN3_bm) || ((PORTE.IN & PIN1_bm))))
        {
            PORTE.OUTSET = PIN0_bm;
        }
        else
        {
            PORTE.OUTCLR = PIN0_bm;
        }
    }
}
```

Labopdracht 3

```
#include <avr/io.h>

int main()
{
    PORTE.DIR = PORTE.DIR | PIN0_bm;

    while(1)
    {
        PORTE.OUTSET = PIN0_bm;
        PORTE.OUTCLR = PIN0_bm;
    }
}
```

Klok draait default op 4 MHz.

Labopdracht 4

a) Tabel:

Pin decoder	Verbonden pin microcontroller
D0	PA4
D1	PA5
D2	PA6
D3	PA7
LE	PF2
LT	Op VCC aangesloten (= '1')
BL	PF3

Even terug naar vorige week: om cijfers af te beelden moest LE laag zijn en BL en LT hoog.

b) We kunnen de vier pinnen van poort A tegelijk instellen:

Bit 4,5,6 en 7 moeten hoog worden (we beginnen rechts vanaf 0 te tellen): "11110000".

Dit kunnen we per 4 bits omzetten naar hexadecimaal: F0

Hetzelfde geldt voor de twee pinnen van poort F: PF2 en PF3

"00001100" wordt hexadecimaal: 0C

c)

```
#include <avr/io.h>
```

```
int main()
{
    PORTA.DIR = PORTA.DIR | 0xF0;    //alle vier de pinnen in PORTA
    PORTF.DIR = PORTF.DIR | 0x0C;    //alle twee de pinnen in PORTF

    while(1)
    {
        PORTF.OUTCLR = PIN2_bm;      //LE = 0
        PORTF.OUTSET = PIN3_bm;      //BL = 1

        PORTA.OUTSET = PIN4_bm;      //D0 = 1
        PORTA.OUTSET = PIN5_bm;      //D1 = 1
        PORTA.OUTCLR = PIN6_bm;      //D2 = 0
        PORTA.OUTCLR = PIN7_bm;      //D3 = 0
    }
}
```

```
#include <avr/io.h>
```

```
int main()
{
    PORTA.DIR = PORTA.DIR | 0xF0;    //alle vier de pinnen in PORTA
    PORTF.DIR = PORTF.DIR | 0x0C;    //alle twee de pinnen in PORTF

    while(1)
    {
        PORTF.OUTCLR = PIN2_bm;      //LE = 0
        PORTF.OUTSET = PIN3_bm;      //BL = 1

        PORTA.OUTSET = PIN4_bm;      //D0 = 1
        PORTA.OUTSET = PIN5_bm;      //D1 = 1
        PORTA.OUTCLR = PIN6_bm;      //D2 = 0
        PORTA.OUTCLR = PIN7_bm;      //D3 = 0
    }
}
```

d) SW1 = PE3, SW2 = PE2, SW3 = PE1

Vergeet niet dat de switches geïnverteerd zijn (dus 0 wanneer ingedrukt en 1 wanneer niet ingedrukt).

Optie 1: gebruik een variabele om te tellen hoeveel schakelaars er ingedrukt zijn en switch-case om elk cijfer op het display te laten zien:

```
#include <avr/io.h>

int main()
{
    PORTA.DIR = PORTA.DIR | 0xF0;    //alle vier de pinnen in PORTA
    PORTF.DIR = PORTF.DIR | 0x0C;    //alle twee de pinnen in PORTF

    PORTF.OUTCLR = PIN2_bm; //LE = 0
    PORTF.OUTSET = PIN3_bm; //BL = 1

    while(1)
    {
        int schakelaar = 0;

        if(!(PORTE.IN & PIN1_bm))
        {
            schakelaar++;
        }
        if(!(PORTE.IN & PIN2_bm))
        {
            schakelaar++;
        }
        if(!(PORTE.IN & PIN3_bm))
        {
            schakelaar++;
        }

        switch(schakelaar)
        {
            case 0:
                PORTA.OUTCLR = PIN4_bm;    //D0 = 0
                PORTA.OUTCLR = PIN5_bm;    //D1 = 0
                PORTA.OUTCLR = PIN6_bm;    //D2 = 0
                PORTA.OUTCLR = PIN7_bm;    //D3 = 0
                break;
            case 1:
                PORTA.OUTSET = PIN4_bm;    //D0 = 1
                PORTA.OUTCLR = PIN5_bm;    //D1 = 0
                PORTA.OUTCLR = PIN6_bm;    //D2 = 0
                PORTA.OUTCLR = PIN7_bm;    //D3 = 0
                break;
            case 2:
                PORTA.OUTCLR = PIN4_bm;    //D0 = 0
                PORTA.OUTSET = PIN5_bm;    //D1 = 1
                PORTA.OUTCLR = PIN6_bm;    //D2 = 0
                PORTA.OUTCLR = PIN7_bm;    //D3 = 0
                break;
            case 3:
                PORTA.OUTSET = PIN4_bm;    //D0 = 1
                PORTA.OUTSET = PIN5_bm;    //D1 = 1
                PORTA.OUTCLR = PIN6_bm;    //D2 = 0
                PORTA.OUTCLR = PIN7_bm;    //D3 = 0
                break;
        }
    }
}
```

```

    }
}
}

```

Alternatief (minder netjes): check met een if en logische operatoren hoeveel knoppen er tegelijkertijd worden ingedrukt en

```

#include <avr/io.h>

int main()
{
    PORTA.DIR = PORTA.DIR | 0xF0;    //alle vier de pinnen in PORTA
    PORTF.DIR = PORTF.DIR | 0x0C;    //alle twee de pinnen in PORTF

    PORTF.OUTCLR = PIN2_bm;    //LE = 0
    PORTF.OUTSET = PIN3_bm;    //BL = 1

    while(1)
    {
        // alle drukknoppen NIET ingedrukt = cijfer 0
        if((PORTE.IN & PIN1_bm) && (PORTE.IN & PIN2_bm) && (PORTE.IN & PIN3_bm))
        {
            PORTA.OUTCLR = PIN4_bm;    //D0 = 0
            PORTA.OUTCLR = PIN5_bm;    //D1 = 0
            PORTA.OUTCLR = PIN6_bm;    //D2 = 0
            PORTA.OUTCLR = PIN7_bm;    //D3 = 0
        }
        else
        {
            // alle drukknoppen WEL ingedrukt = cijfer 3
            if(!(PORTE.IN & PIN1_bm) && !(PORTE.IN & PIN2_bm) && !(PORTE.IN &
PIN3_bm))
            {
                PORTA.OUTSET = PIN4_bm;    //D0 = 1
                PORTA.OUTSET = PIN5_bm;    //D1 = 1
                PORTA.OUTCLR = PIN6_bm;    //D2 = 0
                PORTA.OUTCLR = PIN7_bm;    //D3 = 0
            }
            else
            {
                // twee drukknoppen ingedrukt = cijfer 2
                if(((!(PORTE.IN & PIN1_bm) && !(PORTE.IN & PIN2_bm)) ||
(!(PORTE.IN & PIN3_bm)&&!(PORTE.IN & PIN1_bm)) ||
(!(PORTE.IN & PIN3_bm)&&!(PORTE.IN & PIN2_bm)) )
                {
                    PORTA.OUTCLR = PIN4_bm;    //D0 = 0
                    PORTA.OUTSET = PIN5_bm;    //D1 = 1
                    PORTA.OUTCLR = PIN6_bm;    //D2 = 0
                    PORTA.OUTCLR = PIN7_bm;    //D3 = 0
                }
            }
            else
            {
                // één drukknop ingedrukt = cijfer 1
                PORTA.OUTSET = PIN4_bm;    //D0 = 1
                PORTA.OUTCLR = PIN5_bm;    //D1 = 0
                PORTA.OUTCLR = PIN6_bm;    //D2 = 0
                PORTA.OUTCLR = PIN7_bm;    //D3 = 0
            }
        }
    }
}
}
}

```