

Project SMU

Slimme Meter Uitlezer

Docenten:

- Jan Kampen: j.kampen@windesheim.nl
- Anke Kuijk: a.kuijk@windesheim.nl
- Richard Rosing: r.rosing@windesheim.nl
- Henk Bouwman: h.bouwman@windesheim.nl



Wat gaan we doen?

Week 1: Kennismaken met de Slimme Meter en de SMU

Week 2: Het meten van spanning en stroom

Week 3: Digitale signalen en schakelaars

Week 4: Booleaanse algebra en het 7-segmentsdisplay

Week 5: De microcontroller programmeren

Week 6: Seriële communicatie, OLED display, SD-kaart

Week 7: Alles afmaken

Week 8: Toetsing



Seriële communicatie

- Data telegram wordt serieel verstuurd vanuit de P1 poort

|/Ene5\XS210 ESMR 5.0

```

1-3:0.2.8(50)
0-0:1.0.0(210905102806S)
0-0:96.1.1(4530303437303030303437363638383138)
1-0:1.8.1(004028.304*kWh)
1-0:1.8.2(003737.450*kWh)
1-0:2.8.1(000000.006*kWh)
1-0:2.8.2(000000.000*kWh)
0-0:96.14.0(0001)
1-0:1.7.0(00.000*kW)
1-.7.0(00.000*kW)
0-0:96.7.21(00333)
0-0:96.7.9(00010)
1-0:99.97.0(6)(0-0:96.7.19)(210905071327S)(0000001502*s)(210905064158S)(0000599056*s)(210829075513S)(0001197412*s)(210815032123S)(0000834818*s)
(200206100335W)(0000001622*s)(181012010000S)(0000000335*s)
1-0:32.32.0(00011)
1-.32.36.0(00000)
0-0:96.13.0()
1-0:32.7.0(233.0*V)
1-0:31.7.0(000*A)
1-0:21.7.0(00.000*kW)
1-0:22.7.0(00.000*kW)
0-1:24.1.0(003)
0-1:96.1.0(4730303538353330303337303333373139)
0-1:24.2.1(632525252525S)(00000.000*m3)
!FCB2

```



Seriële communicatie

- Wat is van belang?
 - Snelheid (baudrate)
 - Pariteit
 - Aantal databits
 - Aantal stopbits
- Bijvoorbeeld **9600,N,8,1**



Seriële communicatie

- Vanuit de microcontroller registers goedzetten
- Flinke uitzoekklus....
- Code krijg je cadeau: zie dictaat

```
void USART3_init(void)
{
    // AVR128DB48
    PORTB.DIRSET = PIN0_bm;           /* set pin 0 of PORT B (TXd) as output*/
    PORTB.DIRCLR = PIN1_bm;          /* set pin 1 of PORT B (RXd) as input*/

    USART3.BAUD = (uint16_t)(USART3_BAUD_RATE(9600)); /* set the baud rate*/

    USART3.CTRLA = USART_CHSIZE0_bm | USART_CHSIZE1_bm; /* set the data format to 8-bit*/

    USART3.CTRLB |= USART_TXEN_bm;    /* enable transmitter*/
}
```

```
void USART3_sendChar(char c)
{
    while(!(USART3.STATUS & USART_DREIF_bm))
    {
        ;
    }

    USART3.TXDATAL = c;
}
```

```
void USART3_sendString(char *str)
{
    for(size_t i = 0; i < strlen(str); i++)
    {
        USART3_sendChar(str[i]);
    }
}
```



Seriële communicatie

- Alleen functies gebruiken

```
int main(void)
{
    USART3_init();

    while (1)
    {
        USART3_sendString("Hello World! \r\n");
    }
}
```



Seriële communicatie

- Variabele printen: tussenstap maken
- Printen in char array (basis programmeren les 5)
- Let op grootte array!!!!
- Char array serieel versturen

```
uint8_t counter = 10;  
char buffer[20];
```

```
sprintf(buffer, "%d", counter);  
USART3_sendString(buffer);
```



OLED display



- Hiervoor gebruik je aangereikte functies vanuit template project (zie BrightSpace)
- Ook grafische mogelijkheden

Functie	Toelichting
<code>GLCD_Setup();</code>	Voordat je het display kunt gebruiken moeten we het initialiseren. Ook worden in deze functie de I2C communicatiepinnen geïnitieerd.
<code>GLCD_SetFont(Font5x8, 5, 8, GLCD_Overwrite);</code>	Met deze functie zet je het lettertype van het display goed. Op dit moment is er alleen ondersteuning voor een 5x8 font.
<code>GLCD_GotoXY(20, 10);</code>	Met deze functie kun je de positie bepalen waar de tekst wordt geplaatst.
<code>sprintf(buffer, "%3d", i);</code>	Met deze functie kan je een variabele omzetten naar tekst.
<code>GLCD_PrintString(buffer);</code>	Met deze functie kun je een tekst weergeven. De tekst wordt weergegeven op de positie die je hebt ingesteld met de <code>GLCD_GotoXY</code> functie. Als je een variabele wilt weergeven moet je deze omzetten naar een tekst.
<code>GLCD_Render();</code>	Wellicht de meest belangrijke functie: de <code>GLCD_Render</code> functie zorgt er voor dat de data die klaargezet is naar het display wordt verstuurd.



SD card



- Ook hiervoor gebruik je aangereikte functies vanuit template project (zie BrightSpace)

<u>f_mount()</u>	Zorgt er voor dat de SD kaart geschikt wordt gemaakt voor gebruik. Het systeem gaat uit van een geformatteerde SD kaart. Zorg er voor dat je deze voor gebruik FAT32 formatteert. Deze functie moet je dus altijd als eerste aanroepen. Hier worden ook de SPI pennen juist geïnitieerd.
<u>f_open()</u>	Opent het bestand op de SD kaart. Geef de naam van het bestand mee. De mode bepaalt wat er moet gebeuren als het bestand al op de SD kaart staat. Zie voor de verschillende modi de instructie bij de functie op http://elmschan.org/fsw/ff/doc/open.html .
<u>f_write()</u>	Schrijft data in het bestand op de SD kaart. Je geeft de data mee, maar ook het aantal karakters dat geschreven moeten worden.
<u>f_close()</u>	Als je klaar bent met het schrijven van het bestand sluit je deze met <u>f_close</u> .



Aan de slag!

- Ga naar leren.windesheim.nl
(zoek de cursus EDPD.22, project SMU)
- Voer de opdrachten van week 6 uit.
- Week 7 is uitloopweek

