Zeno Scheltens

[z.f.scheltens@st.hanze.nl](mailto:z.f.scheltens@st.hanze.nl)

ELVE1 groep 3

Kit Yi Feng

[k.y.feng@st.hanze.nl](mailto:k.y.feng@st.hanze.nl)

ELVM1 groep 1

Ontwerp en evaluatie document programmeren

Communicatie

Serieuze communicatie!

Inhoud

[Analyse 3](#_Toc498286840)

[Ontwerp 4](#_Toc498286841)

[Flowcharts 4](#_Toc498286842)

[Main() 4](#_Toc498286843)

[Functies 4](#_Toc498286844)

[Implementatie 5](#_Toc498286845)

[Evaluatie 6](#_Toc498286846)

# Analyse

Voor het project de zonnekracht meter moet er een verbinding opgezet worden tussen de microcontroller en een pc om realtime de zonnekracht te kunnen uitlezen. De microcontroller is echter niet uitgerust met functionaliteit om een directe verbinding op te kunnen zetten met een pc.

Voor deze opdracht zetten we een verbinding op tussen de microcontroller en de pc via een Arduino UNO. De Arduino UNO beschikt over de ATmega16U2 microcontroller. Deze microcontroller kan seriële signalen omzetten naar een signaal over USB.

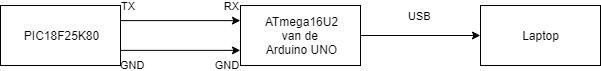
Voor het tot stand brengen van communicatie tussen de microcontroller en de pc moet de microcontroller ingesteld worden en moet op de pc het signaal ook op de juiste snelheid worden uitgelezen.

Formele eisen:

* Het programma kan communiceren met de laptop

# Ontwerp

Om de communicatie tussen de microcontroller tot stand te brengen word er een Arduino UNO tussen gezet zoals uitgebeeld in Figuur 1.



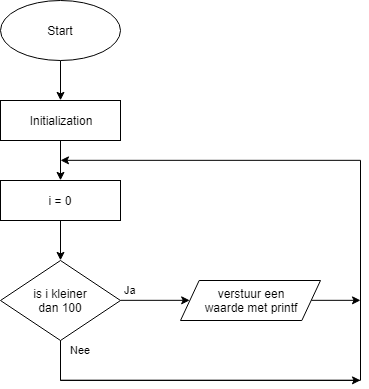
Figuur 1: Blokschema seriële communicatie met de PIC18F25K80

Verder moeten er verschillende dingen worden ingesteld via registers om de eigenschappen van de communicatie vast te leggen. Dit moet met het TXSTA, RCSTA, BAUDCON en SPBRG register.

Daarna word met een stukje code een signaal over de verbinding verstuurd dat kan worden ontvangen met de seriële monitor.

De printf functie regelt het formateren van de tekst en roept de functie putch() aan om dan daadwerkelijk elke byte te versturen. Standaard is deze functie leeg en moet dan nog worden geprogrammeerd om bij het project te passen waar je de functie voor wilt gebruiken.

## Flowcharts



# Implementatie

1. /\*
2. \*  Embedded Programming 2018
3. \*
4. \*  Opdracht: Serieuze Communicatie
5. \*
6. \*  Code by
7. \*      - Zeno Scheltens
8. \*      - Kit Yi Feng
9. \*
10. \*/
12. #include <stdio.h>
13. #include <xc.h>
14. #include "fuses.h"
16. #define \_XTAL\_FREQ 8000000  // X-tal = 8 MHz
18. // specific putch() function for this project, needed for printf()
19. **void** putch (**char** c) {
20. **while** (TXSTA1bits.TRMT == 0)    {
21. ;
22. }
23. TXREG1 = c;
25. }
27. **void** main() {
29. // Ini fase
31. TRISCbits.TRISC6 = 0;   // Tx1 output
32. PIE3bits.RC2IE = 0;     // disable Rx interrupt USART2
33. PIE3bits.TX2IE = 0;     // disable Tx interrupt USART2
34. PIE1bits.RC1IE = 0;     // disable Rx interrupt USART1
35. PIE1bits.TX1IE = 0;     // disable Tx interrupt USART1
36. TXSTA1 = 0xA0;          //
37. RCSTA1 = 0x80;          //
38. BAUDCON1 = 0xC0;        //
39. SPBRG1 = 12;            //
41. // port settings
43. TRISCbits.TRISC5 = 0;           // portC output
44. LATCbits.LATC5 = 1;             // LED1 groen aan
46. // variables
48. **int** i;
50. // main fase
52. **while** (1)   {
53. **for** (i = 0;i < 100;i++)  {
54. printf("test = %d\n\r", i);
55. }
56. }
57. }

# Evaluatie