

Verfmengmachine Service Manual

Technische Informatica

Oktober 2008

Versiebeheer:

Aangepast door:

Datum aanpassing:
versie 1.0

adres

Haagse Hogeschool
Academie voor ICT & Media
Johanna Westerdijkplein 75
2521 EN 's-Gravenhage
tel: 070-4458400

onderwijs en dienstverlening

Opleiding Technische Informatica

Service manual verfmengmachine

Auteur
T.Mast

Auteursrechten
© Academie voor ICT & Media
versie 1.0

Inhoud

| | | |
|------|--|----|
| 1. | INLEIDING..... | 3 |
| 2. | HYDRAULICA..... | 5 |
| 2.1. | C1..... | 5 |
| 2.2. | C5..... | 6 |
| 2.3. | C11..... | 6 |
| 2.4. | Centrifugaalpompn | 6 |
| 3. | ELEKTRA | 9 |
| 3.1. | P1 t/m P6 en bodemvlotters..... | 9 |
| 3.2. | V1 t/m V7, P7, heater en agitator..... | 10 |
| 3.3. | Voeding en PLC connector | 11 |
| 4. | ELECTRONICA | 13 |
| 4.1. | Debiet pulsecounter | 13 |
| 4.2. | Heater simulator..... | 17 |
| 4.3. | Temperatuurregistratie simulator | 21 |
| 5. | LANDSCAPE HYDROULIEKDIAGRAM | 24 |

1. Inleiding

De verfmengmachine is ontworpen en gebouwd in opdracht van de Haagse Hogeschool ten behoeve van het PLC praktijkonderwijs.

De machine is een model, welke een fabrieksapparaat voorstelt die batches verf van verschillende kleuren kan produceren.

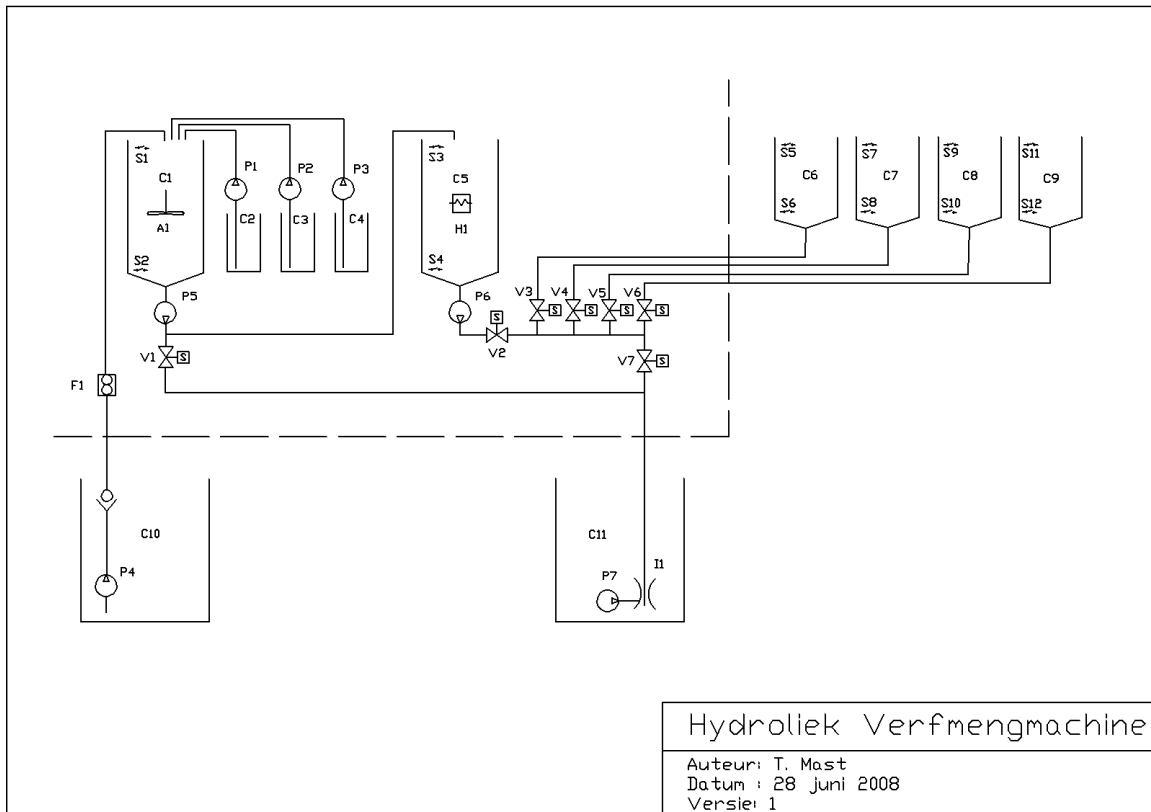
In het model wordt voedingskleurstof of inkt als kleurstof gebruikt, en water als oplosmiddel.

De belangrijkste onderdelen van de machine zijn de volgende:

- Een container waarin water en drie verschillende soorten kleurstof worden gemengd.
- Drie peristaltische kleerpompen met een constante flow.
- Een debietmeter waarmee de vulling van de eerste container kan worden gemeten.
- Een tweede container waarin een simulatie van een verhittingsproces kan plaatsvinden.
- Vier externe voorraadcontainers die in de verfvulfabriek kunnen worden geplaatst.
- Digitale in- en uitgangen om de machine te bedienen die werken op een 24 Volt spanning.
- Een frontpanel met schakelaars om de verschillende pompen en kleppen handmatig te bedienen.
- Vlotters die de verschillende vaten voor overvulling behoeden of een indicatie geven dat een vat bijna leeg is.
- Een injecteur welke onderdruk kan genereren om de verschillende vaten en leidingen leeg te zuigen.
- Een noodknop om het systeem handmatig uit te schakelen.

2. Hydraulica

Figuur 1 geeft een schematisch overzicht van de hydraulica van de verfmengmachine.



De componenten binnen de stippellijn bevinden zich op het frame van de machine. De overige componenten zijn extern geplaatst, maar maken onderdeel van de machine uit.

P1 t/m P3 zijn peristaltische pompen.

P4 t/m P7 zijn centrifugaal pompen. (standaard caravan pomp)

V1 t/m V7 zijn elektrische kleppen. (Bürkert solenoid valve, type 6027 small)

S1 t/m S12 zijn vlotterschakelaars.

F1 is een debietmeter. (Kobold DTK-12..G2)

A1 is een agitator. (roerwerk)

H1 is een fictief verwarmingselement.

2.1. C1

Vanuit de schoonwatertank container C10 wordt water gepompt naar C1. De leiding bevat een terugslagklep waardoor het water in de leiding blijft staan nadat de pomp klaar is met pompen. In de leiding is een debietmeter (F1) opgenomen waarmee de hoeveelheid water die de leiding passeert kan worden gemeten.

Op C1 zijn ook de drie kleurstofpompen (P1, P2 en P3) aangesloten. De peristaltische pompen hebben een constant debiet, mits de viscositeit van de vloeistof gelijk blijft.

Wanneer voedingskleurstoffen worden gebruikt, zal na verloop van tijd algengroei optreden. Een chloortablet in de schoonwatertank voorkomt dit. Teveel chloor in het water doet de kleurstof verbleken. Tevens veroorzaakt teveel chloor schuim in het water, waardoor de centrifugaalpompen lucht in het pomphuis kunnen krijgen.

In C1 zit een agitator (roerwerk) waarmee de vloeistoffen gemixed kunnen worden.

Normally closed vlotter S1 begrenst pompen P1, P2, P3 en P4. Als het vloeistofpeil zo hoog komt dat de vlotter onderbreekt, kunnen de pompen niet meer lopen. S2 begrenst niets en is slechts indicatief, alleen bedoeld om af te lezen.

Via klep V1 is het mogelijk de vloeistof uit C1 af te pompen.

2.2. C5

Via P5 kan de vulling naar C5 worden gepompt. Hier vindt het verwarmingsproces plaats.

S3 begrenst P5 om overvulling te voorkomen.

De heater H1 geeft na activeren een signaal af wanneer de juiste temperatuur is bereikt. S4 is indicatief.

Via klep V2 en één van de kleppen V3 t/m V6 is het mogelijk de vaten C6 t/m C9 te vullen.

De V3 t/m V6 zijn elektrisch begrensd zodat er slechts één tegelijkertijd open kan staan. Wanneer twee of meer kleppen worden aangestuurd zal geen enkele meer openen. Dit geldt ook wanneer men via de PLC een klep bekrachtigt terwijl er een andere handmatig wordt bekrachtigd.

Dezelfde kleppen worden begrensd door hun corresponderende vlotters. D.w.z.: S5 begrenst V3, S7 begrenst V4, S9 begrenst V5 en S11 begrenst V6. S6, S8, S10 en S12 zijn indicatief.

Wanneer een container overvuld is, is het daarom ook niet meer mogelijk de klep te openen om de container leeg te zuigen. Wil men een vat legen via de injecteur, dan zal men handmatig de juiste kleppen moeten openen, de afzuiging aanzetten, en de vlotter tegen de waterdruk in omlaag schuiven.

Tijdens het ontwerpen is overwogen niet de kleppen per stuk te begrenzen, maar P6. Dit houdt echter in dat er geen vaten meer gevuld kunnen worden zodra er één vat vol zit. De programmeur zou dan niet in staat zijn meerdere vaten tot hun maximum te vullen.

2.3. C11

De afvalcontainer C11 is uitgerust met een injecteur. De injecteur kan met behulp van een pomp een onderdruk in de leiding creëren. Wanneer een leiding die leeggezogen moet worden nog vol vloeistof staat, zal het afpompen direct starten. Wanneer er lucht in de leiding zit, duurt het langer omdat de luchtdruk in de slang eerst voldoende gedaald moet zijn.

2.4. Centrifugaalpompen

De pompen P4, P5, P6 en P7 zijn centrifugaalpompen. Een eigenschap van een centrifugaalpompe is dat de pompkamer met vloeistof vol moet zijn voordat hij kan pompen. Na het afpompen van een leiding waar een centrifugaalpompe in zit, ontstaat er een luchtbel in de pompkamer.

Wanneer een leiding met een centrifugaalpompe is leeggepompt, zal de pompe niet functioneren. Ook niet wanneer het vat voor de pompe geheel is gevuld. De operator/programmeur zal de

betreffende pomp moeten aanzetten en korte tijd (1 a 2 sec) de injecteur laten draaien, met de juiste kleppen geopend. Hiermee wordt de luchtbel in het pomphuis, afgepompt, waarna de centrifugaalpomp zal werken.

Het volgende voorbeeld toont de cyclus van schakelen voor een situatie waarin een volledig gevulde C1 wordt overgepompt naar C5, terwijl de leiding waar P5 in zit, hiervoor is afgepompt.

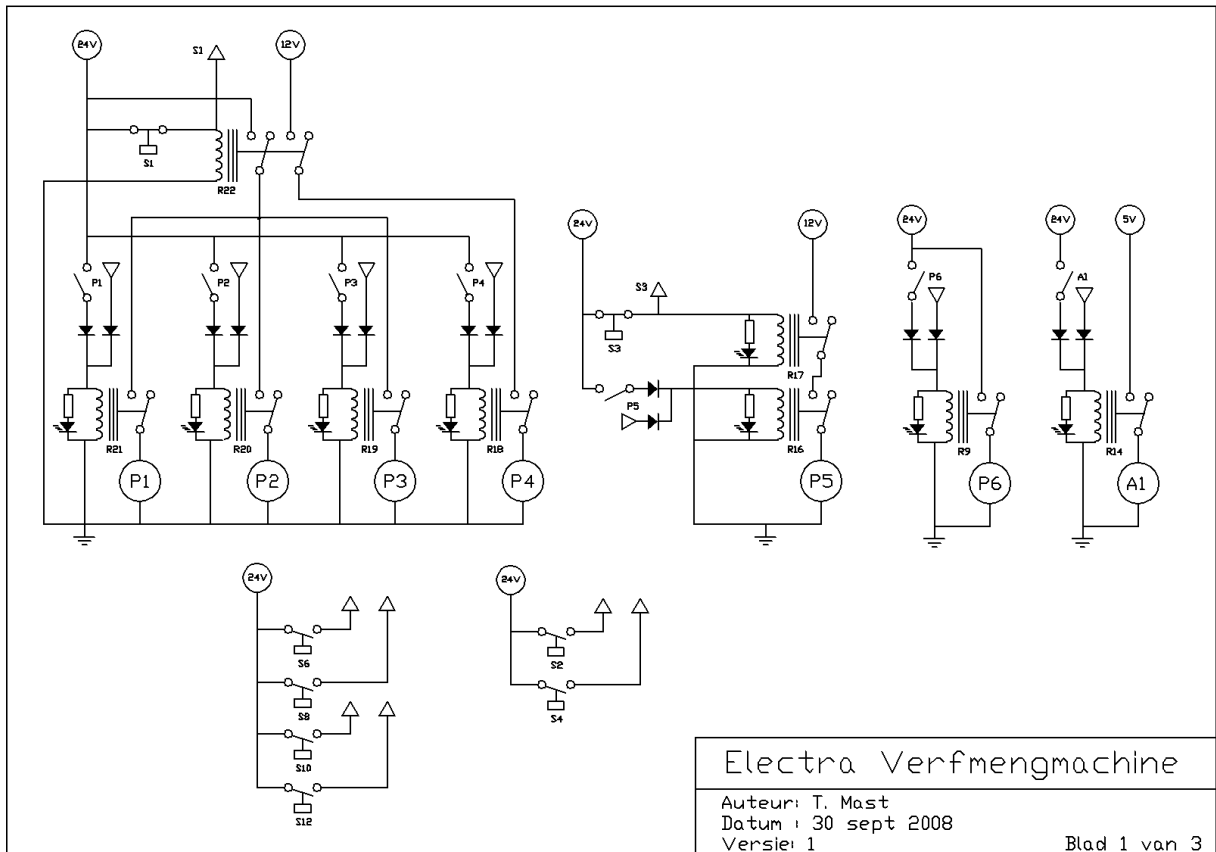
| Tijd in seconden | Actie |
|------------------|------------------|
| T=0 | P5 aan |
| T=1 | P7 aan V1 aan |
| T=3 | P7 uit V1 uit |

Hierna zal P5 de vloeistof naar C5 overpompen tot C1 leeg is, of door S3 wordt onderbroken.

3. Elektra

3.1. P1 t/m P6, A1 en bodemvlotters

Figuur 2 is het elektrische schema van pomp P1 t/m P6, A1 en de bodemvlotters.



Alle actuatoren worden via een relais aangestuurd. Hierdoor levert de PLC niet rechtstreeks bedrijfsstroom aan de actuatoren. Een geactiveerd relais met LED gebruikt 50 mA. Wanneer alle relais bekrachtigd zijn, loopt er vanaf de PLC een stroom van maximaal 1 Ampère. Dit is ruim onder het maximum van de PLC.

Alle actuatoren, op de agitator en heater na, zijn zowel door de PLC (het ingaande driehoekje) als manueel (de schakelaar) aan te sturen. De stuurspanningslijnen voor manueel- en PLC-bediening zijn middels diodes van elkaar gescheiden.

De relais zijn uitgerust met een LED op de magneetspoel. Hierdoor is aan de machine te zien welke relais zijn bekrachtigd. Tevens dient het LED als een ontvonkingsdiode.

Als gebruikelijk is het schema afgebeeld in een toestand waarin geen voedingsspanning op het circuit staat.

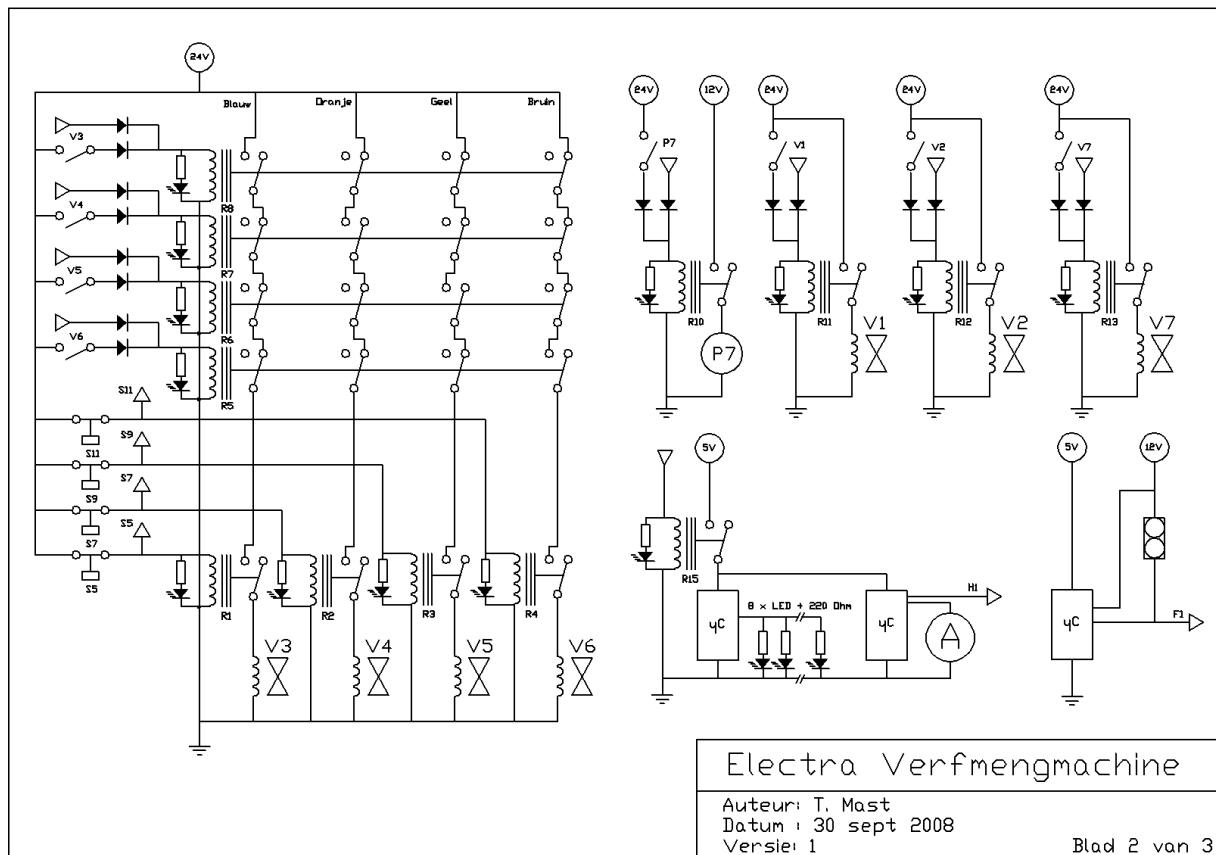
Pompen P1 t/m P4 kunnen middels de PLC of manueel worden geschakeld. De stroomtoevoer loopt via relais R22 welke afschakelt wanneer S1 opent. (vloeistofniveau te hoog)

Pompen P5 en P6 zijn op dezelfde manier begrensd.

De verschillende vlotters zijn op de 24V aangesloten om aan de PLC een signaal te kunnen geven.

3.2. V1 t/m V7, P7 en de heater

Figuur 2 is het elektrische schema van V1 t/m V7, P7 en de heater.



Kleppen V3 t/m V6 worden aangestuurd door een combinatie van relais R5 t/m R8. Iedere klepsturing loopt over de vier relais. Wanneer de relais in rust zijn, is de voedingsspanning voor een klep een gesloten circuit, op een relais na. Bekrachtigt men dit relais, dan is het circuit helemaal gesloten en wordt de klep bekrachtigd. Met het bekrachtigen van een relais, worden tevens de circuits voor de andere kleppen onderbroken. Het is dus niet mogelijk meer dan één klep tegelijkertijd te bekrachtigen.

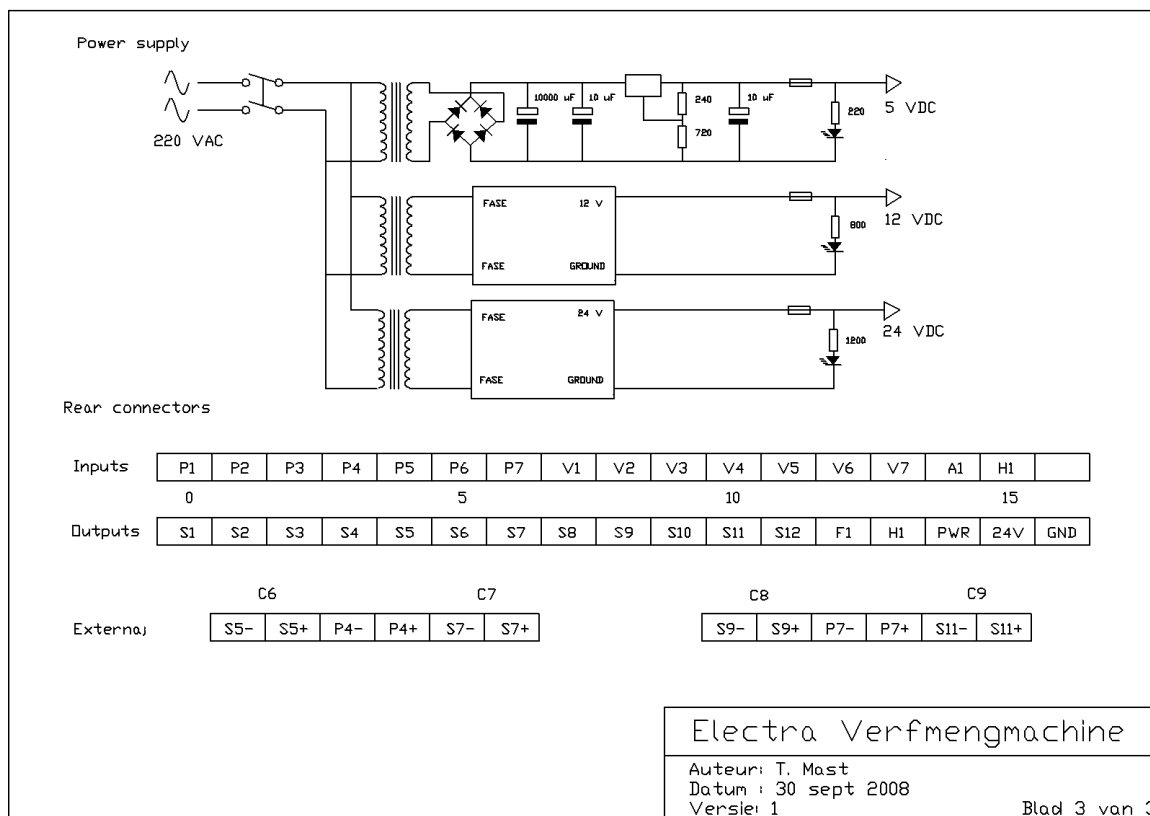
Relais R1 t/m R4 worden bekrachtigd door stromen die lopen via hun corresponderende vlotters. Zodra het vat maximaal gevuld is, schakelt de vlotter open en daarmee wordt het relais afgeschakeld.

Pomp P7, en kleppen V1, V2 en V7 zijn niet door vlotters begrensd.

Agitator A1 en heater H1 kunnen alleen via de PLC worden ingeschakeld.

3.3. Voeding en PLC connector

Figuur 4 is het elektrische schema van de voeding en de connector voor aansluiting op de PLC



De voeding levert drie gestabiliseerde spanningen. 5V voor de microcontrollers, 12V voor de centrifugaalpompen, en 24V voor de kleppen, peristaltische pompen en relais.

De 24V en 12V transformatoren kunnen 4 Ampère leveren. De secundaire spoel is aangesloten op een Velleman spanningsstabilisatiemodule K7203. Deze module kan een wisselspanning van 35V stabiliseren en omlaagregelen naar een spanning tussen de 30V en 3V. De module schakelt af bij oververhitting of wanneer er meer dan 3 Ampere vermogen wordt afgenomen. De uitgangen zijn afgezekerd met 4A traag zekeringen.

De 5V transformator kan 1 Ampère leveren. De spanning wordt gestabiliseerd door een LM317 stabilisator. De LM317 schakelt af bij 1.5 Ampere vermogen of oververhitting.

De inputs en outputs werken op een spanning van 24V voor hoog en 0V voor laag, de gebruikelijke standaard voor industriële elektrische besturing.

De polariteit van P4-/P4+ en P7-/P7+ doet niet ter zake, daar de pompen altijd dezelfde richting uit pompen.

De noodknop (niet in het schema opgenomen) onderbreekt de 24V voeding naar de machine. Het aanwezig zijn van de 24V voeding, kan worden afgelezen op de output PWR.

4. Electronica

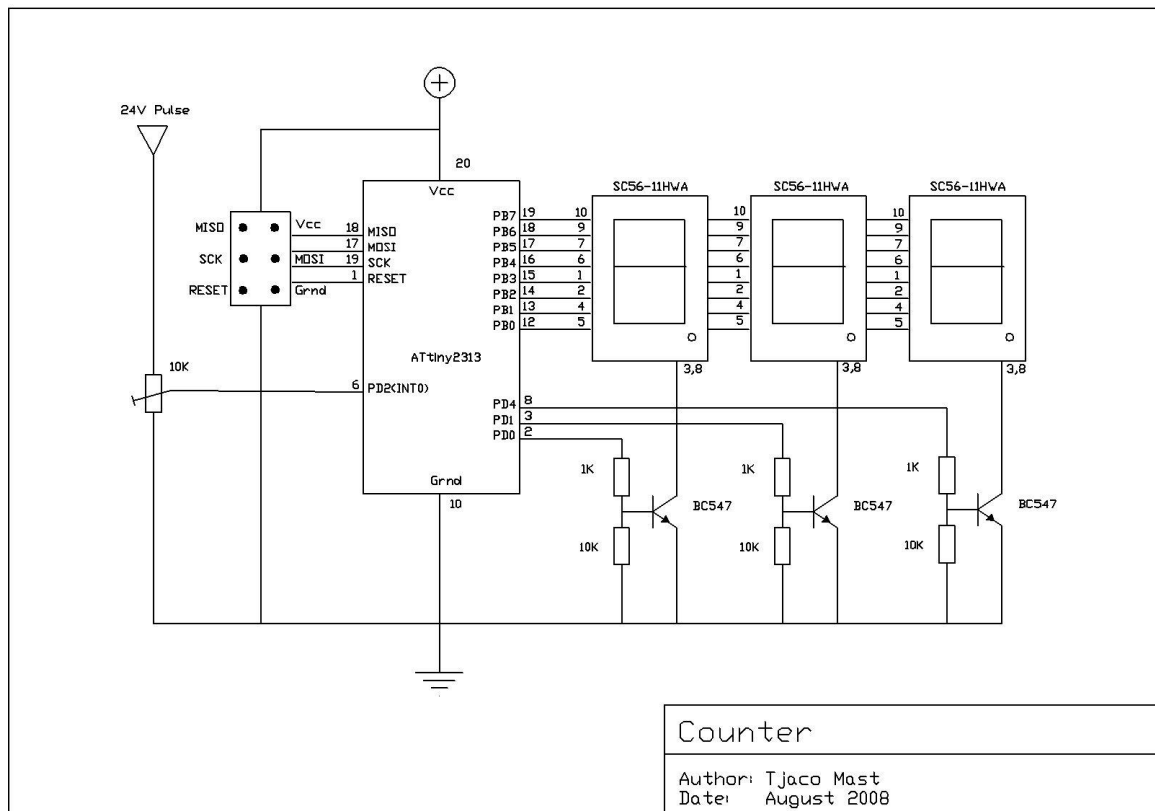
De machine is uitgerust met drie microcontrollerschakelingen voor het animeren van de mengmachine. Deze schakelingen zijn niet essentieel om de machine te laten functioneren.

Iedere schakeling is gebaseerd op een Atmel ATtiny2313. De software is geschreven in de gratis Atmel AVR IDE met de GNU AVR-GCC plugin als C compiler.

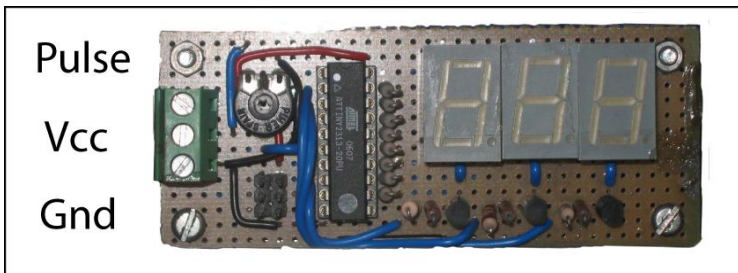
De programmeer-connector op de prints is compatible met de AVR In-System Programmer mkII

4.1. Debiet pulsecounter

De counter zit aangesloten op de debietmeter. Middels een spanningsdeler worden de pulsen van de debietmeter aan de counter toegevoerd. De counter deelt het aantal pulsen door twee en geeft deze weer op de displays. De counter staat wanneer de mengmachine aanstaat, continu onder spanning. Wanneer de counter vijftien seconden geen pulsen heeft ontvangen, gaat het display uit en wordt de teller op nul gezet.



4.1.1. Print



4.1.2. Code

```
/*
Counter

Tjaco Mast, may 2008

This software creates a digital up counter. At port B three 7-segment led displays
are connected in a paralel fashion. The common cathodes of the displays are
connected to pin 0, 1 and 4 of Port D. The system counts up on a rising edge on
INT0 (pin 6) When no new interrupts are recieved, the display shuts down after
TIMEOUT seconds.

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 3 of the License, or (at your option) any later
version. This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more
details. You should have received a copy of the GNU General Public License
along with this program; if not, see <http://www.gnu.org/licenses>.

*/

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/iotn2313.h>

// The seconds to shut down after last interrupt
#define TIMEOUT 10

int divide = 0;
int dummy;
char output = 0;
//to store the numbers of the 7 segment displays
uint8_t hundreds=0, tens=0, ones=0;
int total = 0;

int timeout = TIMEOUT;

/* signal handler for external interrupt int0 */
ISR(INT0_vect) {

    timeout = 0;

    // Count every second pulse
    if (divide){

        ones++;
        total++;
        divide = 0;
    }
}
```



```

    } else {

        divide = 1;
    }

    if (ones==10){

        tens++;
        ones=0;
    }

    if (tens==10){

        hundreds++;
        tens=0;
    }

    if (hundreds==10){

        hundreds=0;
    }
}

ISR(TIMER1_COMPA_vect,){

    if (timeout < TIMEOUT) timeout++;

}

uint8_t decimal_to_image(uint8_t decimal){

    uint8_t image;

    if (decimal==0) image = 0b01111110;
    if (decimal==1) image = 0b00010010;
    if (decimal==2) image = 0b10111100;
    if (decimal==3) image = 0b10110110;
    if (decimal==4) image = 0b11010010;
    if (decimal==5) image = 0b11100110;
    if (decimal==6) image = 0b11101110;
    if (decimal==7) image = 0b00110010;
    if (decimal==8) image = 0b11111110;
    if (decimal==9) image = 0b11110110;
    // Blank screen
    if (decimal==10) image = 0b00000000;

    return image;
}

void showimage(){

    // Do not touch the values
    int hun = hundreds;
    int ten = tens;
    int one = ones;

    if (total < 99) hun = 10;
    if (total < 9) ten = 10;
    if (total == 0) one = 10;

    // Set the display for some time
    for(dummy=0;dummy<100;dummy++){

        // Select display 0
        PORTD = 0b00000001;
        PORTB = decimal_to_image(hun);
    }

    for(dummy=0;dummy<100;dummy++){

        // Select display 1

```

```

        PORTD = 0b00010000;
        PORTB = decimal_to_image(ten);
    }

    for(dummy=0;dummy<100;dummy++){

        // Select display 2
        PORTD = 0b00000010;
        PORTB = decimal_to_image(one);
    }
}

int main(void){

    // All Port B pins as output
    DDRB = 255;
    // Port D pins 0, 1 and 4 as output
    DDRD = 0b00010011;

    /*
    Period between ticks:
    P = OCR * prescaler * (1/clock)
    OCR = P / (precaler * (1/clock))
    Clock in hz, P in secs
    */

    //P=1 sec if clock = 8 Mhz
    OCR1A = 7650;

    // Set prescaler 1024 and countmode to CTC
    TCCR1B |= (1<<WGM12) | (1<<CS12) | (1<<CS10);
    // Enable CTC interrupt
    TIMSK |= (1 << OCIE1A);

    // enable external int0
    GIMSK |= (1<<INT0);
    MCUCR |= (1<<ISC01) | (1<<ISC00);

    // Enable global interrupt
    sei();

    //loop forever
    while(1){

        showimage();
        if (timeout == TIMEOUT){

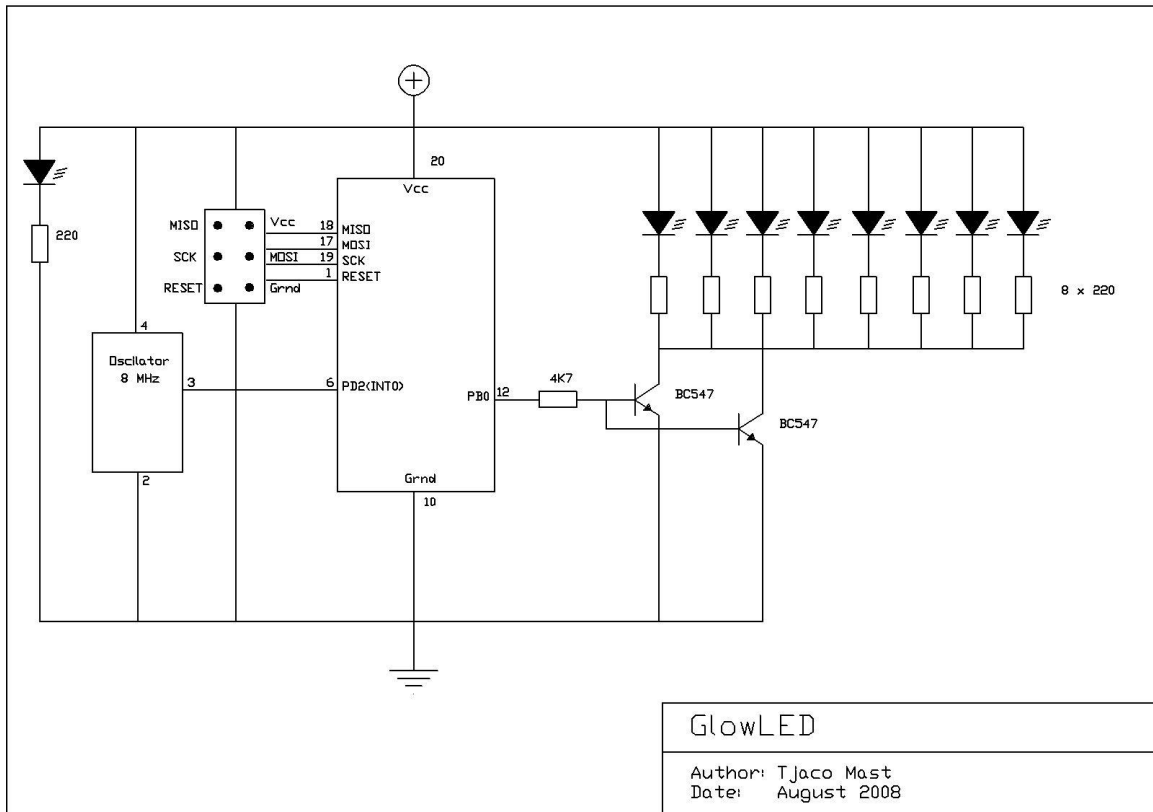
            ones = 0;
            tens = 0;
            hundreds = 0;
            total = 0;
        }
    }

    //useless but it fixes a warning
    return 0;
}

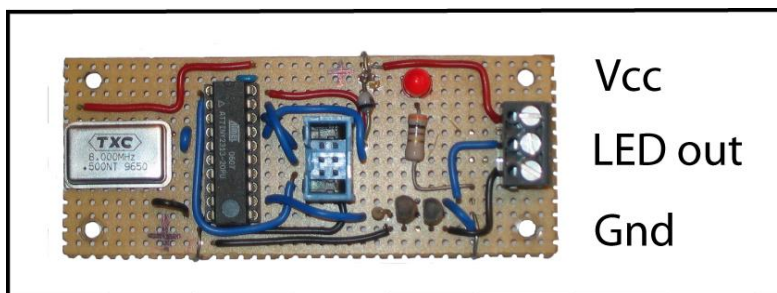
```

4.2. Heater simulator

Wanneer de schakling onder spanning wordt gezet, gloeien de LEDs op en af. Het dimmen van de leds gebeurt middels pulse width modulation.



4.2.1. Print



4.2.2. Code

```
/*  
GlowLED
```

```
Tjaco Mast, may 2008
```

```
This program continuously glows a LED up and down. The glowing is achieved  
by subdividing a period of 1/100th of a second by 100. By setting variable
```

'value' is determined how many of the subdivisions the led is on
 The range of values for 'value' are stored in array 'range'. The brightness
 of a glowing LED is not proportional to the percentage of time the LED is on
 during a period, so I had to figure out something of a curved range of numbers.
 This array is continuously being read from 0 to 24 and back again.

This program is free software; you can redistribute it and/or modify it under
 the terms of the GNU General Public License as published by the Free Software
 Foundation; either version 3 of the License, or (at your option) any later
 version. This program is distributed in the hope that it will be useful, but
 WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
 FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more
 details. You should have received a copy of the GNU General Public License
 along with this program; if not, see <<http://www.gnu.org/licenses>>.

```

*/

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/iotn2313.h>

char lamp = 0; //Lamp output
char value = 0; //How hard the lamp shines 0=off 100=full on
char tickcounter = 0; //This counter will be increased every 1/100000 sec.
int ms = 0; //This counter will be increased every 1/1000 sec.
char direction = 1; //1=glowing up, 0=glowing down
int pointer = 0; //Points to a element in 'range'

char range[25] = {1,1,1,2,2,2,3,3,3,6,8,10,12,16,20,27,34,44,58,75,80,85,90,100,100};

ISR(TIMER1_COMPA_vect){

    //Disable interrupts
    cli();

    tickcounter++;

    //100 Ticks is 1/1000th second
    if (tickcounter == 100){

        tickcounter = 0;
        ms++;
    }

    //Each 20 ms we increase or decrease the intensity of the LED
    if (ms==20){

        ms=0;
        if (direction) pointer++; else pointer--;
        if (pointer==24) direction = 0;
        if (pointer==0) direction = 1;
        value = range[pointer];
    }

    if (value > tickcounter) lamp = 255; else lamp = 0;
    PORTB = lamp;

    //Enable interrupts
    sei();
}

int main(void){

    //Port B to total output.
    DDRB = 255;

    //Period = 1/100000 sec if clock = 8 Mhz and OCR = 80.
    OCR1A = 80;
    //Set no prescaler and countmode to CTC.
    TCCR1B |= (1<<WGM12) | (1<<CS10);

```

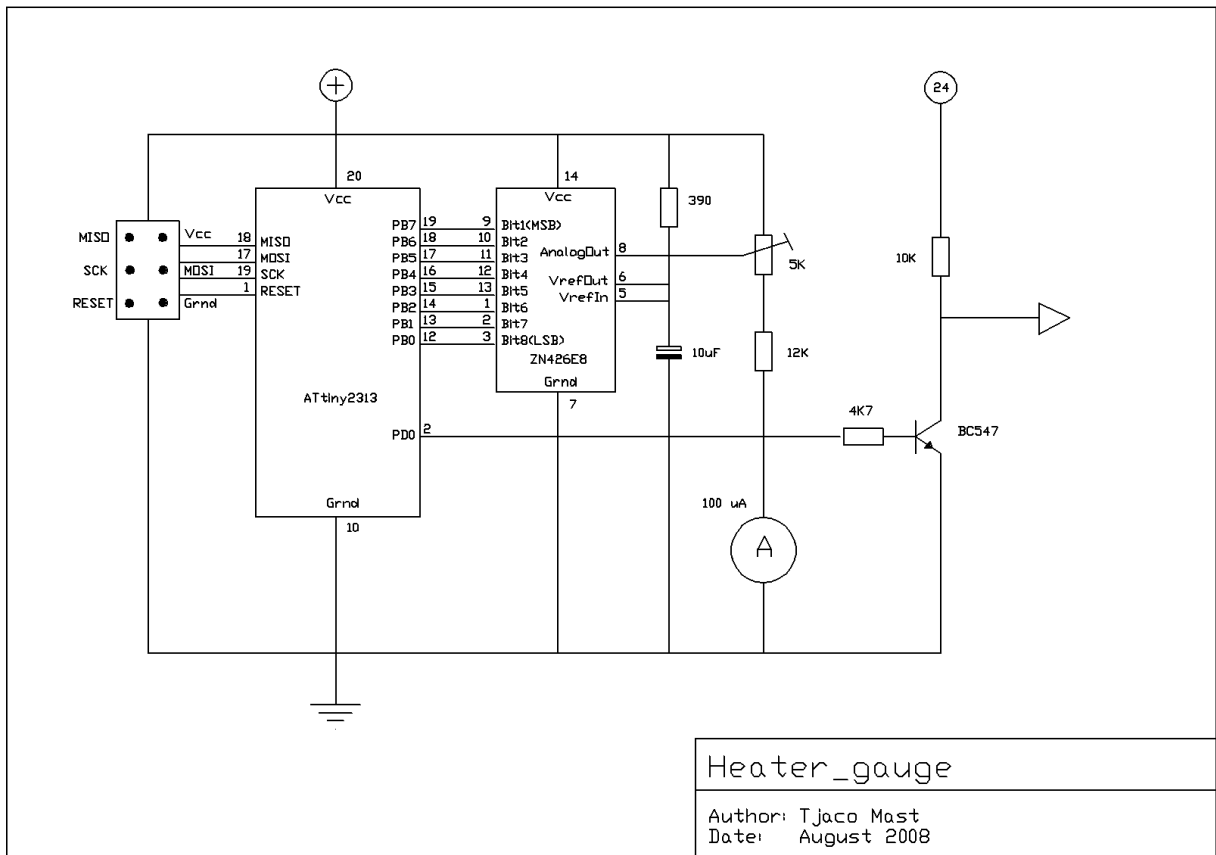
```
//Enable CTC interrupt.  
TIMSK |= (1 << OCIE1A);  
  
//Enable global interrupt.  
sei();  
  
//Loop forever.  
while(1){  
//Useless but it fixes a warning.  
return 0;  
}
```


4.3. Temperatuurregistratie simulator

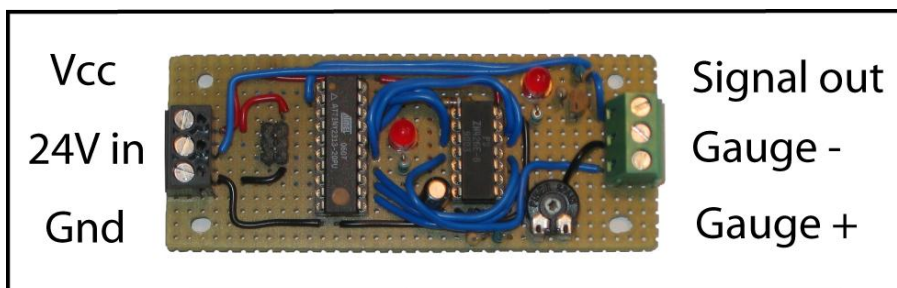
De heater_gauge simulator genereert een recht oplopende stroom , in 15 seconden van 0 tot 100 micro Ampère. De stroom komt uit een R2R netwerk, die digitaal door de microcontroller wordt aangestuurd. Hierna maakt deze een uitgang laag als indicatie dat de juist temperatuur is bereikt.

De output H1 (heater ready) is ook laag wanneer de heater niet is ingeschakeld. Het berekenen of de temperatuur is bereikt, gaat daarom met de volgende formule:

$$(\text{input H1} \wedge \overline{\text{output H1}})$$



4.3.1. Print



4.3.2. Code

```
/*
heater_gauge

Tjaco Mast, may 2008

heater_gauge simulates an temperaturemeter in a production process for
demonstrating purposes.

This program counts from 0 to 255 on port B in 15 seconds. Port B is connected
to a R2R network thus generating an increasing voltage. At the R2R network an
Ampere meter is connected. When the value 255 is reached, port B stays at this
value and pin PD0 is set to one, indicating that the right 'temperature' is
reached

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 3 of the License, or (at your option) any later
version. This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more
details. You should have received a copy of the GNU General Public License
along with this program; if not, see <http://www.gnu.org/licenses>.

*/

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/iotn2313.h>

char output = 0;

// signal handler for timer
ISR(TIMER1_COMPA_vect){

    if (output < 255){

        PORTB = output++;
    }
    if (output == 254) PORTD = 1;
}

int main(void){

    DDRB = 255;
    DDRD = 1;

    /*
    Period between ticks:
    P = OCR * prescaler * (1/clock)
    OCR = P / (precaler * (1/clock))
    Clock in hz, P in secs
    */

    //P=0.06 sec if clock = 8 Mhz
    OCR1A = 459;

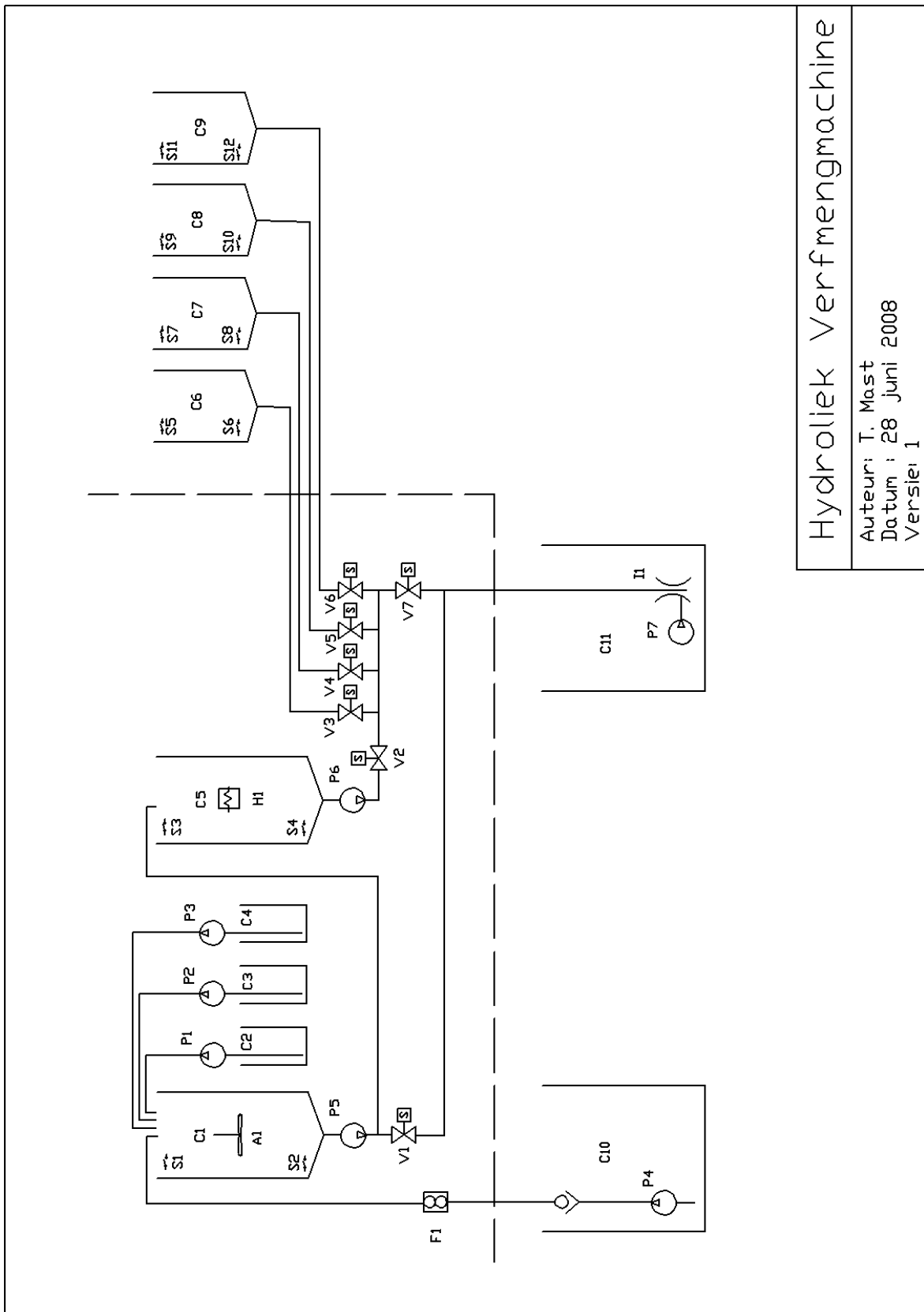
    // Set prescaler 1024 and countmode to CTC
    TCCR1B |= (1<<WGM12) | (1<<CS12) | (1<<CS10);
    // Enable CTC interrupt
    TIMSK |= (1 << OCIE1A);

    // Enable global interrupt
    sei();

    //Loop forever
    while(1){}
    //Useless but it fixes a warning
    return 0;
}
```


}

5. Landscape Hydrouliekdiagram



Hydrauliek Verfmgmaschine

Auteur: T. Mast
 Datum : 28 juni 2008
 Versie: 1