

PROFE2

Projet de fin d'études 2014/2015 Semester 1+2

Projet: Messstation im Musée Nationale des Mines
Modul: Sauerstoffsensor

Thill Dan
T3EE
LTAM

Inhaltsverzeichnis

1.....	Einleitung
1.....	Was ist dieses Projekt?
2.....	Anforderungen an die Messstation!
2.....	Anforderungen an die Messmodule!
3.....	Besichtigung des Musée des Mines
5.....	Der Sauerstoffsensor
5.....	Einleitung
5.....	Bauteile Auswahl und Tests

5.....	Sauerstoffsensor
7.....	Operationsverstärker
7.....	Micro Controller (uC)
9.....	Die Schaltung
9.....	Aufbau
9.....	Der Sauerstoffsensor und der Verstärker
10.....	Der uController und die Schnittstellen
11.....	Der Transistor und Ventilator
12.....	Der Schaltplan
12.....	Materialliste
14.....	Layout
15.....	Programmierung
15.....	Das Flussdiagramm
16.....	Die Analogwertverarbeitung
17.....	Der Serielle Interrupt und die State Maschine und Daten senden
22.....	Die Puls Weiten Modulation (PWM)
22.....	Der Watchdog
23.....	Das Interface
23.....	Einleitung
23.....	Die Modulkopplung
23.....	Der Seriell zu Tristate Wandler (Modulseite)
23.....	Der Seriell zu Tristate Wandler (Computerseite)
24.....	Der Computeradapter
24.....	Schaltungen
24.....	Die Modulkopplung
25.....	Der Seriell zu Tristate Wandler (Modulseite)
29.....	Der Seriell zu Tristate Wandler (Computerseite)
29.....	Der Computeradapter
31.....	Foto-Blockaufbau der gesamten Messstation

Einleitung

?Was ist dieses Projekt

Bei dem Projekt Messstation Musée Nationale des Mines geht es darum eine Messstation, zum Erfassen von verschiedenen Daten im Musée des Mines in Rumelange nahe der

französischen Grenze zu entwickeln und aufzubauen. Die Station soll die Möglichkeit bieten den Sauerstoffwert, die Windstärke sowie Richtung als auch die Seismischen Aktivitäten aufzeichnen und speichern

!Anforderungen an die Messstation

.Die Messstation soll verschiedene Anforderungen erfüllen und Aufgaben erledigen können
Die Anlage soll verschiedene Messwerte aufnehmen können, unser Team entwickelt .1
deshalb mehrere Module

Windrichtung und Geschwindigkeitssensor um bestimmen zu können ob der Wind -
,aus Luxembourg oder aus Frankreich kommt

,Einen Seismograph um jegliche Bodenbewegung aufzeichnen zu können -

,Ein Sauerstoffsensor um eventuelle Schwankungen messen zu können -

Einer Temperaturmessung und Regelung um die Feuchtigkeit zu minimieren -

.Der hohen Luftfeuchtigkeit widerstehen zu können .2

Die Messstation soll außerdem die Daten an einen Computer senden können und .3
(speichern. (Später soll anstatt eines Computer ein Raspberry PI eingesetzt werden

!Anforderungen an die Messmodule

.Die Messelemente sollen auch verschiedene Standardanforderungen erfüllen

:Das Modul soll

,Mit einer Spannung von 20V und 5V auskommen .1

,Mit einer variablen Adresse ausgestattet sein .2

Die Möglichkeit habe über ein Jumper die Baudrate von 9600 auf 2400 setzen zu .3
(können (erst nach rücksetzen des Moduls

,Außerdem zu Debugging zwecken .4

einen Jumper zum dauerhaften senden besitzen (a

.eine LED besitzen die den Betrieb anzeigt und anzeigt wenn etwas gesendet wird (b

Die Module sollen einen Stecker mit Standardpinbelegung besitzen um einfach
,ausgewechselt und erweitert zu werden. Wir haben uns auf folgendes geeinigt

.5

20V+	Pin 1
(Not Connected (NC	Pin 2
(Write/not Receive (W/R	Pin 3
5V+	Pin 4
RXD	Pin 5
TXD	Pin 6
GND	Pin 7

.Die Module sollen auf verschiedene Standardbefehle reagieren .6

Um dies zu tun muss eine vorab besprochene Satzzeichenkette eingehalten werden. Diese Besteht aus drei Zeichen, das erste ist ein „!“ welches die Module in Bereitschaft bringen soll das zweite Zeichen ist die Moduladresse (Wählbar von „A“- „,O“) und das dritte Zeichen ist der Befehl. Die Befehle für jedes Element sind

Komplette Kette für Modul A	Aufgabe	Befehl
"As!,,	Senden der gemessenen Daten	"s,,
"Ai!,,	Senden der Modulinfo	"i,,
"Ar!,,	Rücksetzen den Moduls	"r,,

Teile des Befehls

s	A	!
(Befehl (Daten senden	(Moduladresse (Modul A	Bereitschaft

[Besichtigung des Musée des Mines](#)

In den Karneval Ferien bot uns Herr Feltes die Möglichkeit die alte Messstation im Musée des Mines zu besichtigen. Dort konnten wir auch sofort planen wir die neue Station aufhängen und befestigen. Wir trafen uns gegen 10 Uhr beim Schuppen der Lock und fuhren von dort aus mit einem Auto in die Mine hinein zuerst hielten wir bei der Masterstation die zum Aktuellen Zeitpunkt aus einem alten Computer und einem Interface besteht. Danach ging es zu Fuß weiter. Wir gingen durch eine Stahltür in den abgesperrten Bereich. Beim Stahlgitter angekommen, welches die Grenze zwischen Luxembourg und Frankreich markiert, erreichten wir unser Ziel an der Wand Links der Gittertür befand sich die Messstation mit den Sensoren. Nachdem wir diese etwas genauer untersucht hatten suchten wir nach einer Stelle an der wir unsere neue Messstation befestigen konnten. Wir fanden einen Stahlträger an den wir die Messstation so gut befestigen können, so dass Seismograf optimal festgehalten wird um beste Resultate zu erzielen

Der Sauerstoffsensor

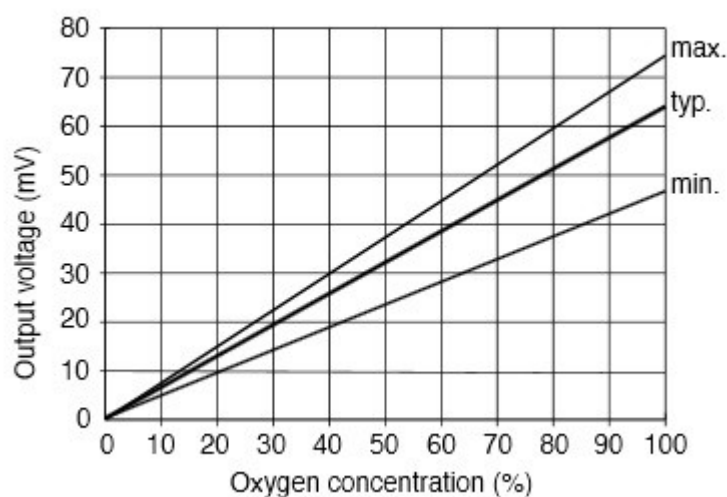
Einleitung

Der Sauerstoffsensor auch noch Sauerstoffmesselement oder Sauerstoffmessmodul genannt soll den aktuellen Sauerstoffgehalt im Musée des Mines messen. Um dies zu tun musste ich mir zuvor überlegen wie ich diesen korrekt messen kann. Da wir einen Yaesu KE-25 Sauerstoffsensor in der Schule hatten beschloss ich diesen zu nutzen

Bauteile Auswahl und Tests

Sauerstoffsensor

Da wir wie schon erwähnt den Sauerstoffsensor des Typs Yuasa KE-25 in der Schule hatten beschloss ich diesen zu nehmen. Dieser gibt abhängig vom Sauerstoffgehalt eine Spannung ab



Da es sich bei dieser Spannung nur um einige Milli-Volt handelt muss diese verstärkt werden um ein gut interpretierbares Signal zu erhalten

Der Sauerstoffsensor hat verschiedene Vor- und Nachteile

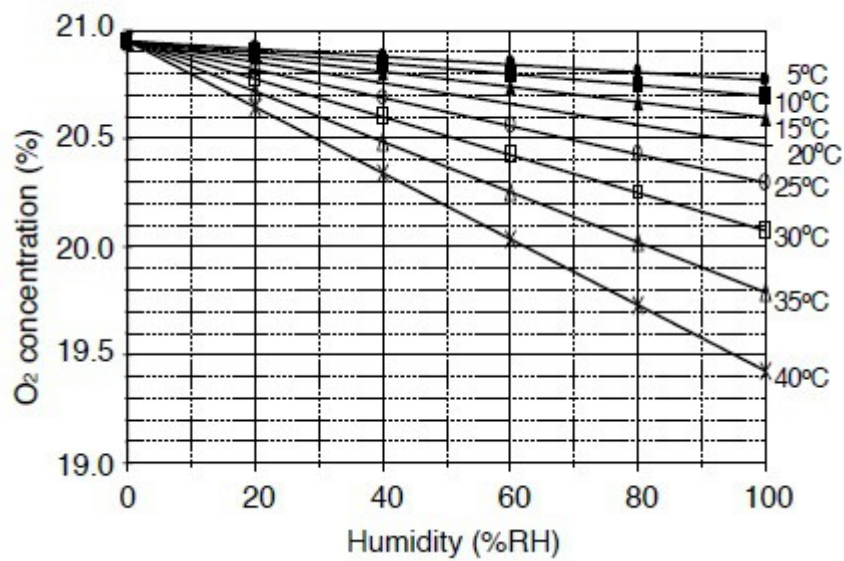
Nachteile	Vorteile
Das Signal hat nur einige mV	Das Signal steigt konstant bei steigendem Sauerstoffgehalt
Das Signal des Sensors steigt nur sehr langsam 30 Sekunden Reaktionszeit	Der Sensor hat nur eine kleine Ungenauigkeit

Die Trägheit des Sensors ist zwar groß aber tolerierbar und um das Signal zu verstärken kann man ein OPV nutzen

Die erste Arbeitsstunde testete ich den Sensor und bei der sich heraus dass dieser defekt war bei +-21% O₂ welches sich in der Luft befand erzeugte der Sensor nur eine Spannung von 4mV also eine Wert außerhalb der vom Hersteller angegebenen Toleranz, so musste ein neuer Sensor bestellt werden



.Der Sensor hat außerdem noch den Vorteil dieser ist nicht sehr feuchtigkeitsempfindlich



.Bei normalen Arbeitsbedingungen ergibt der Sensor einen maximalen Fehler von 0,5%
Ganzes Datenblatt unter:

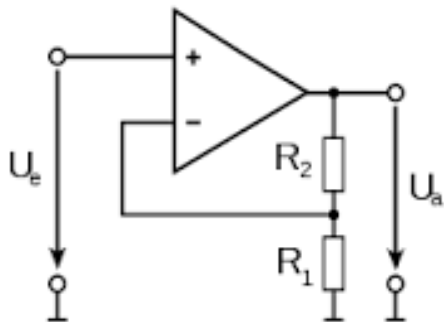
Der Operationsverstärker wird genutzt um das sehr schwache Signal des Sauerstoffsensors in ein gut lesbares Signal umzuwandeln

Um dies zu realisieren beschloss ich das Signal auf 2,1 Volt zu verstärken so kann das Signal besser verstanden werden. 21% => 2,1V Um dies zu erreichen beschloss ich die nicht

invertierende OPV Schaltung zu nutzen. Hierfür galt

Sensorspannung = 7mV Gewünschte Spannung = 2,1V also galt die Rechnung $V = 2,1V / 0,007V = 300$

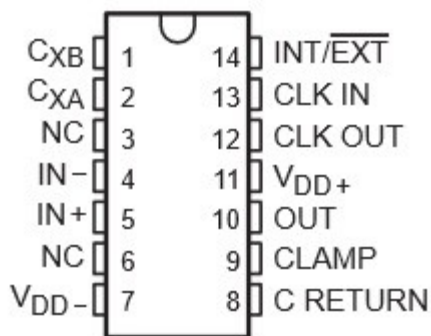
Zur Berechnung der Widerstände nutzte ich die Formel $V = 1 + R1/R2$ so ergaben sich die Widerstände von $R1 = 1 \text{ k}\Omega$ $R2 = 175 \text{ k}\Omega$ und zur größerer Flexibilität nutze ich nun 3 Widerstände $R1 = 1 \text{ k}\Omega$ und $R2 = 100 \text{ k}\Omega$ und $R3 = 0-100 \text{ k}\Omega$ in Reihe



Bei den ersten Tests stellte sich heraus dass ein normaler OPV die Offsetspannung zu hoch ist, diese lagen bei 100 mV also war schon dort ein Fehler von 1% des Sauerstoffgehaltes zu finden

So suchte ich bei Texas Instruments nach einen OPV mit sehr niedriger Offsetspannung.

Nach einigem Suchen fand ich den TLC2652ACN OPV welcher eine Offsetspannung von nur 1uV besitzt



Wie man hier sieht besitzt dieser OPV viele spezielle Pins am Cxa und Cxb muss ein jeweils ein Kondensator gegen Masse angeschlossen werden da der OPV diese braucht um das Signal zu erzeugen, da dies kein einfacher Verstärker sondern ein „Precision Chopper“ OPV ist,

welcher das Signal wechselrichtet dieses verstärkt und dann glättet

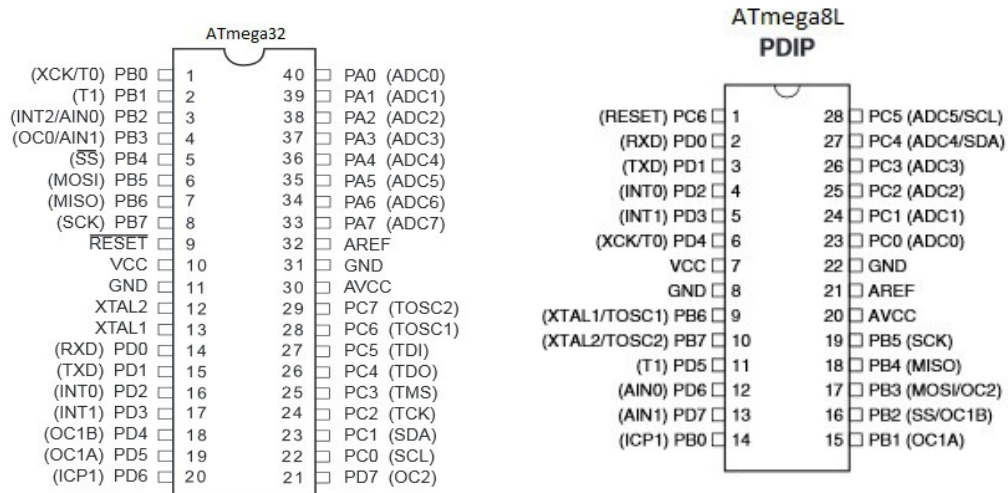
Komplettes Datenblatt:

(Micro Controller (uC

Beim Micro Controller hatten wir verschiedene Modelle zur Auswahl. Da hier nur ein aktueller Wert ausgegeben muss und dafür nicht viel Speicher gebraucht entschied ich mich aus Platzersparnis für den ATmega8L. Diesen hatten wir auf Lager und eignet sich für alle meine Aufgaben und Bedürfnisse

Zum Testen der ersten Programme nutzte ich hingegen den ATmega32 da dieser auf unserem .Test Bord montiert ist

Da ich den ADC des uC brauche um das analoge Signal des OPV einzulesen durfte die Spannung nicht über die Referenz Spannung des uC hinausgeht. Die Referenz Spannung des ATmega8 und Atmega32 sind beide 2,56V. Um diese Spannung zu übersteigen, müsste der .Sauerstoffgehalt auf über 25% steigen, was nicht möglich ist, so ergibt dies kein Problem



:Datenblatt des ATmega8L

:Datenblatt des ATmega32

Die SchaltungAufbau

Die Schaltung besteht aus 3 Teilen
Dem Sauerstoffsensor mit dem Verstärker -
Dem uController mit den Schnittstellen -
Dem Transistor mit dem Ventilator -

Der Sauerstoffsensor und der Verstärker

Wie schon erwähnt gibt der Sauerstoffsensor eine Spannung von 7-12mV ab und um diese zu
.verstärken nutze ich einen nicht invertierenden Verstärker

.Beim Sauerstoffsensor handelt es sich den Yuasa KE-25 den wir auf Lager hatten
Der Sensor wird am + Eingang des OPV angeschlossen und am Ausgang sollen 2,1V
rauskommen. Die Widerstände die also für die Verstärkung gebraucht werden rechnen sich
.wie folgt

$$V=1+R2/R1$$

V rechnet sich wie folgt

$$V= U_{out}/U_{in} =2,1/0,012 =175$$

R1 = 1K gewählt

$$R2=R1/(V-1)=1K/174$$

$$R2= 174 \text{ kOHM}$$

Gewählt R2 =100 kOHM + R3 0-100 kOHM

Durch die hohe Offsetspannung der normalen Verstärker nutze ich den TLC2652ACN

Da der OPV kein normaler Verstärker ist braucht dieser noch 2 10uF Kondensatoren am Pin
.Cxa und Cxb gegen Masse

Der uController und die Schnittstellen

Beim Definitiven uController handelt es sich um den ATmega8L. Gewählt wegen seiner .kleinen Größe und da wir diesen auch auf Lager hatten

Die Beschaltung des uControllers besteht aus kleinen Teilschaltungen

Der erste Teil ist die Betriebsbeschaltung des uControllers besteht aus dem Pullupwiderstand an Reset Pin, der 5V am VCC und am AVCC, sowie Masse an den zwei GND Pins. An den Pins XTAL1 und XTAL2 wurde jeweils 1 Pin eines Quarzoszillator (8MHz bei dem definitiven Sensor, 16MHz beim Prototyp) angeschlossen und auch jeweils ein 22 pF .Kondensator gegen Masse angeschlossen

Der zweite Teil besteht aus den 4 Pins für die Adresse die an einem Dip-Schalter

.angeschlossen sind und 2 Jumper einer für die Baudrate der andere zum dauerhaften Senden

Der dritte Teil ist die serielle Schnittstelle mit dem RXD, TXD und einem zusätzlichen

.Tristate Pin

Der vierte Teil ist der Programmieradapter mit dem MOSI, dem MISO, dem SCK und dem .RESET Pin

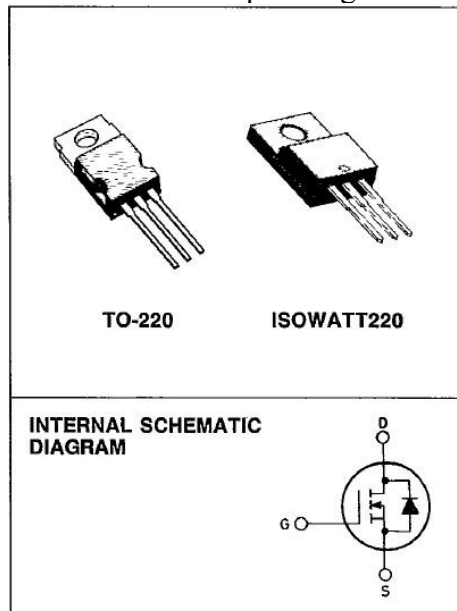
Der letzte Teil besteht aus dem OC1A Pin für das PWM Signal für den Ventilator (Details im .(Teil Programmierung, PWM

Aufgabe	Bezeichnung	Pin
Rücksetzpin des uController	RESET	1
Empfangspin der seriellen Schnittstelle	RXD	2
Sendepin der seriellen Schnittstelle	TXD	3
JP1 Baudrate Einstellungspin	PD2	4
JP2 Dauerhaftsendepin	PD3	5
Tristate Ausgangspin	PD4	6
Betriebsspannung Pin	VCC	7
Masse Pin	GND	8
Quarz Pin 1	XTAL1	9
Quarz Pin 2	XTAL2	10
PWM Pin für Ventilator	OC1A	15
Programmieradapter	MOSI	17
Programmieradapter	MISO	18
Programmieradapter	SCK	19
Betriebsspannung ADC Wandler	AVCC	20
Masse ADC Wandler	GND	22
Adresseingang	PC0	23
Adresseingang	PC1	24
Adresseingang	PC2	25
Adresseingang	PC3	26
Analogeingang Sauerstoffgehalt	ADC5	28

Der Transistor und Ventilator

Der Ventilator, welcher mir zur Verfügung stand, war ein 12V Ventilator. Da jedoch nur 5V sowie 20V zur Verfügung stehen, beschloss ich, aus Energieeffizienzgründen keinen .Festspannungsregler zu nutzen sondern einen PWM gesteuerten Transistor

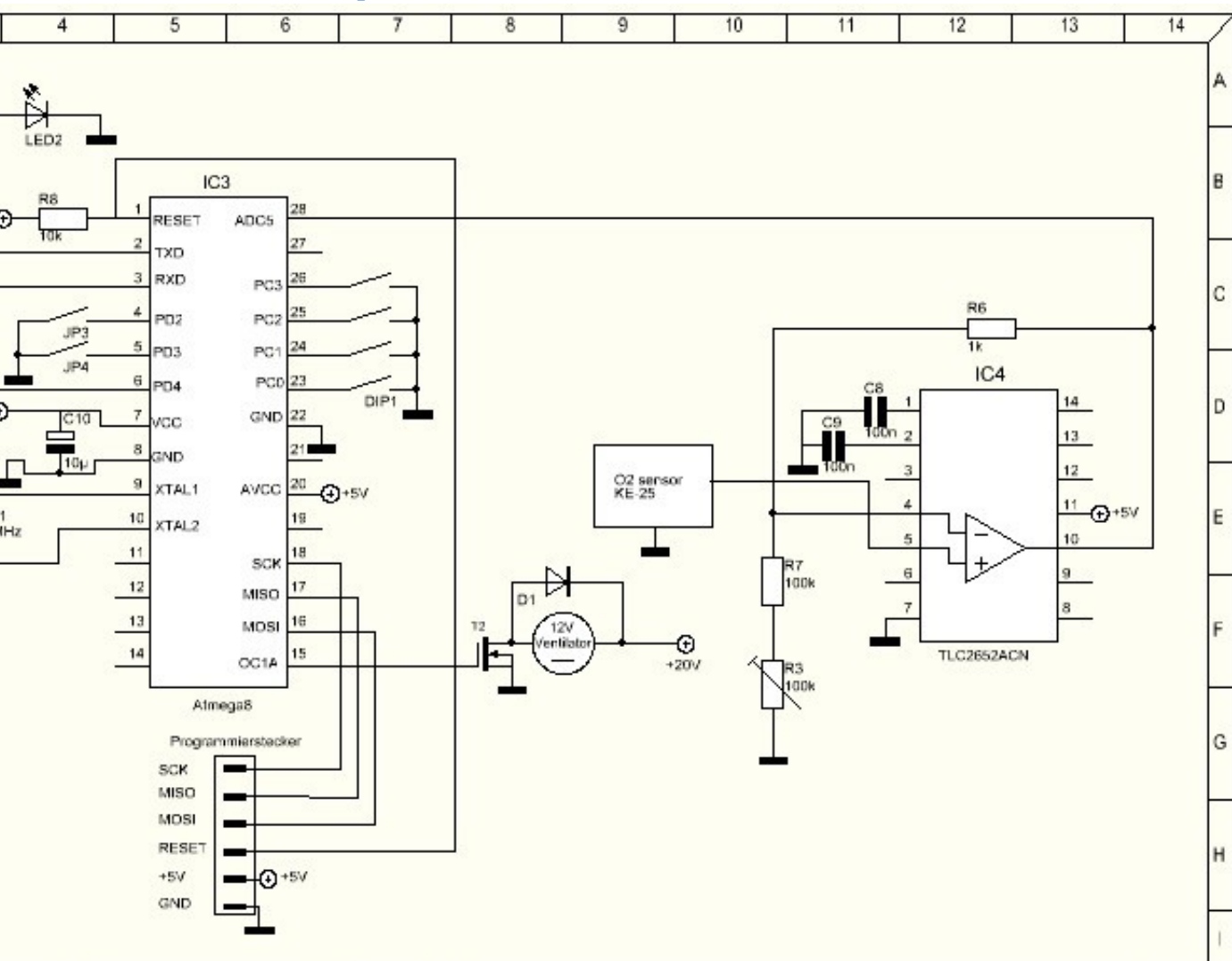
Da wir nicht genau wussten, welchen Strom der definitive Ventilator aufnimmt, beschlossen wir einen relativ großen Transistor einzusetzen. Da der uController außerdem eine Spannung von 5V ausgibt, beschloss ich einen MOSFET Transistor einzusetzen. Als einzige Wahl für diese Anforderungen blieb der IRF530 übrig. Dieser war der einzige N-Kanal MOSFET, den .wir für 5V Gate Spannung hatten



TYPE	V_{DSS}	$R_{DS(on)}$	I_D ■
IRF530	100 V	0.16 Ω	14 A
IRF530FI	100 V	0.16 Ω	9 A
IRF531	80 V	0.16 Ω	14 A
IRF531FI	80 V	0.16 Ω	9 A
IRF532	100 V	0.23 Ω	12 A
IRF532FI	100 V	0.23 Ω	8 A
IRF533	80 V	0.23 Ω	12 A
IRF533FI	80 V	0.23 Ω	8 A

MOSFET Datenblatt:

Der Schaltplan

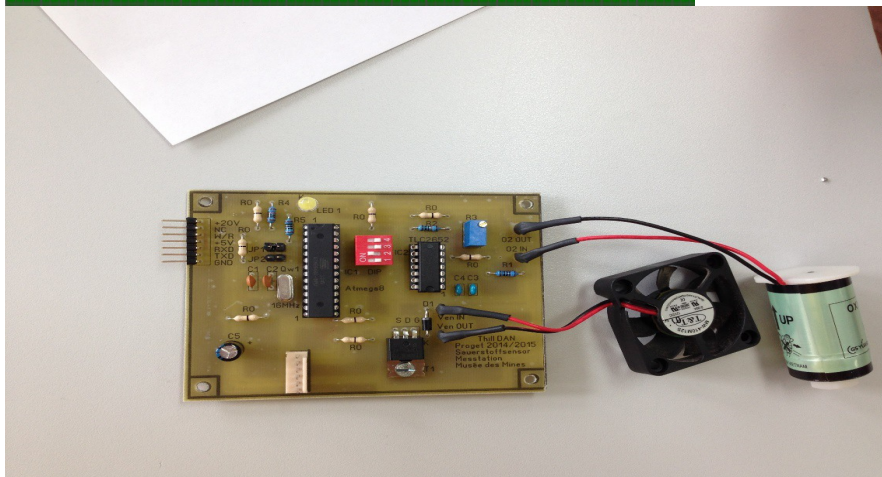
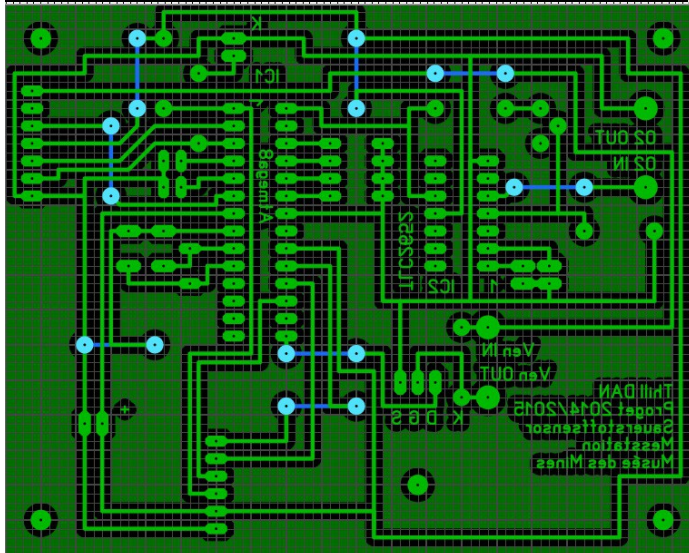
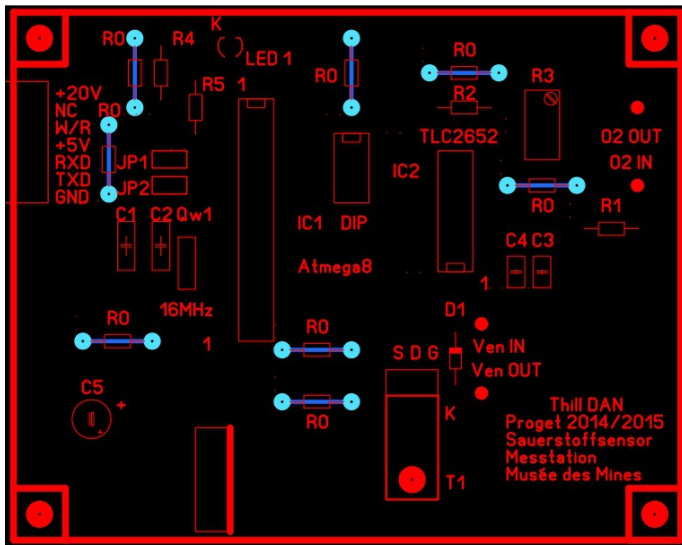


	Datum	Name	Bezeichnung	Blatt
gez.:	19.05.2015	Thill Dan	Sauerstoffsensormodul Messstation	1
gepr.:			Musée des mines 2014/2015	von
Norm.:			Zeichnungs-Nr.: 20150519001	3

Materialliste

Größe	Anzahl	Wert	Bauteil	Bauteil Name
mm 10	8	Ohm 0	Brücke	R0
mm 10	1	1k Ohm	Widerstand	R1
mm 10	1	k Ohm 100	Widerstand	R2
mm 7,5	1	k Ohm 0-100	Trimmer	R3
mm 10	1	k Ohm 15	Widerstand	R4
mm 10	1	Ohm 220	Widerstand	R5
mm 2,5	2		Jumper	JP1/JP2
mm 5	1	MHz 16	Quartz Oszillator	QW1
mm 2,5	2	22nF	Keramikkondensator	C1/C2
mm 2,5	2	uF 10	Keramikkondensator	C3/C4
mm 2,5	1	uF 100	Elektrolytkondensator	C5

DIP28	1	ATMEGA8	Micro Controller	IC1
DIP14	1	TLC2652ACN	Operationsverstärker	IC2
DIP8	1	4X	Dipschalter	DIP
mm 7,5	1	IRF530	MosFET-Transistor	T1
mm 10	1	1N4007	Diode	D1
mm 2,5	1	Rot	Leuchtdiode	LED1
	4		Printnagel	
100mm x 80mm	1		Platine	
DIP28	1		IC-Sockel	
DIP14	1		IC-Sockel	
DIP8	1		IC-Sockel	
50mm / 50mm	1	12V	Ventilator	VEN
R15mm	1	Ke-25	Sauerstoffsensor	O2
15mm/3mm	1	Pinn 6	Programmiersteckleiste	
mm 17,5	1	Pinn 7	Messerstecker 90grad	



Programmierung

Mein Programm für den uController besteht aus 3 Teilen

Der Analogwertverarbeitung -

Die Befehlsverarbeitung mit der seriellen Kommunikation -

Die PWM Ansteuerung des Ventilators -

:Hier noch ein paar spezielle Teile

Die Analogwertverarbeitung mithilfe des ADC -

Der serielle Interrupt -

Seriellm senden von Daten -

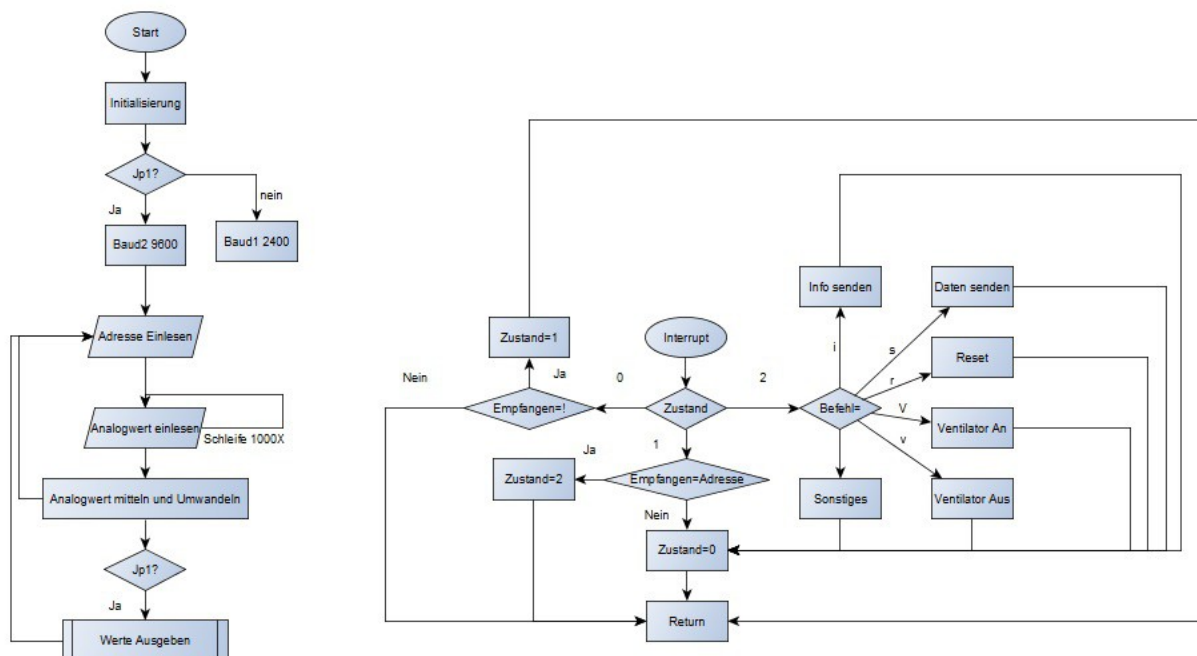
Die „State Maschine“ für die Befehlsverarbeitung -

Die PWM Konfiguration -

Der Watchdog -

-

Das Flussdiagramm



Die Analogwertverarbeitung

Um einen Analogwert an einem uController einzulesen braucht man die ADC Funktion und einen dazugehörigen Eingang. Diese sind als ADC gekennzeichnet. Beim ATmega8 ist dies .u.a. der PC5 Pin dies ist der ADC5 Pin

:Der Analogdigitalwandler oder auch ADC genannt wird wie folgt konfiguriert

Config Adc = Single , Prescaler = Auto , Reference = Internal

Der ADC wird als unendliche Variable definiert die Größe wird automatisch angepasst und .die Referenzspannung wird auf die interne 2,56VG gesetzt

Start Adc 'Start der Analogwert Verarbeitung

.Mit diesem Befehl wird die Analogwertverarbeitung

Adc5 = Getadc(5) 'Istwert wird eingelesen

Mit dem Getadc(Eingang) kann ich den aktuellen Analogwert in eine Variabel einschreiben .und später verwenden zum weiterrechnen

Umwandelfaktor = $1024 / 25.6$ '1024= Maximaler Wert des AD

Wandlers; $25,6 \Rightarrow 2,56$ Interne Referenzspannung $\Rightarrow 25,6\%$ O2

Da es sich beim ADC um eine 10 Bit Variable handelt muss der eingelesene Wert wieder in .einen verständlichen Wert umgewandelt werden dies funktioniert mit dieser Funktion

O2_mittelwert = Analogwert / 1000 'der Wert wird von den 1000 Werten zurückgerechnet auf einen Mittelwert

O2_mittelwert = O2_mittelwert / Umwandelfaktor 'der Eingelesene Wert wird auf einen verständlichen Wert zurückgerechnet

O2_lesbar = Str(o2_mittelwert) 'Der erhaltene Wert wird umgewandelt

O2_lesbar = Fusing(o2_mittelwert, "#.##") 'um den Wert auf zwei Stellen hinter dem Komma zu begrenzen

Da ich mit einem Mittelwert aus 1000 Werten arbeite sieht die Umrechnung wie vorherige .Zeilen aus

Der Serielle Interrupt und die State Maschine und Daten senden

.Diese Funktionen dienen zur Befehlsverarbeitung und somit dem Senden der Daten
 :Um auf jedes von der Masterstation gesendete Zeichen zu empfangen muss man entweder
 .A auf ein Zeichen warten was für diese Art von Programm sich nicht eignet -
 B mit einem Seriellen Interrupt. Dieser unterbricht bei jedem empfangenem Zeichen die -
 Arbeit und kehrt zu der Stelle zurück nach dem er die gewünschte Funktion erledigt hat. Um
 auf die richtige Reihenfolge von Zeichen zu reagieren wird eine „State Maschine“ gebraucht.
 .Wenn die richtige Reihenfolge eingehalten wird werden schlussendlich die Daten gesendet

Der Serielle Interrupt

Dieser unterbricht ein Programm und ruft ein Unterprogramm auf wenn ein Zeichen über
 .serielle Kommunikation empfangen wird

```
On Urxc Serialinterrupt          'Serieller Interrupt eingeschaltet
Enable Urxc
Enable Interrupts
```

Die erste Zeile aktiviert den Seriellen Interrupt

.Die zweite- sowie die dritte Zeile stellen diesen scharf

```
:Serialinterrupt
```

```
Disable Urxc                    'Interrupt wird Abgeschaltet
```

```
Empfangen = Udr                 'UDR-Register wird als Variabel abgespeichert
```

Hier wird das Serielle Interrupt Unterprogramm abgerufen, der Serielle Interrupt abgeschaltet
 um spätere Störungen zu vermeiden und das empfangene Zeichen in eine Variabel

.abgespeichert um dieses weiterverwenden zu können

```
Enable Urxc                      'Interrupts werden wieder eingeschaltet
```

Um Die Interrupts wieder einzuschalten wird dieser Befehl verwendet, um wieder zum

.Hauptprogramm zurück zu kehren der folgende Befehl zeigt dies

```
Return
```

Die „State Maschine“ wird verwendet um z.B. eine Folge von Zeichen richtig zu
 .interpretieren

.Hier in diesem Fall wird der empfangene Befehl untersucht und verarbeitet

"Die Station sendet einen Befehl unter folgendem Format „!Bs

.Beim „!“ sollen sich alle Modul bereitmachen

.Der „B“ ist die Adresse meines Moduls

.Der „s“ steht für Daten senden

Da der die Serielle Interrupt Variable nur ein Zeichen speichern kann nutze ich zusätzlich

.eine Schrittvariabel um zu zpeichern welches Zeichen übereinstimmt

```
If Schritt = 0 Then                'Abfrage der Schrittvariabel Schritt 0
Gosub Bereitschaft                'Unterprogramm Bereitschaft Abrufen
Elseif Schritt = 1 Then           'Abfrage der Schrittvariabel Schritt 1
Gosub Addressabfrage
Elseif Schritt = 2 Then           'Abfrage der Schrittvariabel Schritt 2
:Gosub Befehlsabfrage
End If
```

.Hier sieht man die Abfrage der Schrittvariabel

```
"!" If Empfangen = 33 Then        'Abfrage des Bereitschaftssignal
Schritt = 1                        'Schrittvariabel wird auf 1 gesetzt
Enable Urxc                        'Interrups werden wieder eingeschaltet
Mit dieser Abfrage wird geprüft ob das Zeichen ein „!“ war wenn ja wird der Schritt auf 1
.gesetzt
If Adresse = Empfangen Then       'Abfrage der Adresse
Schritt = 2                        'Schrittvariabel wird auf 2 gesetzt
Enable Urxc                        'Interrups werden wieder eingeschaltet
```

Beim zweiten Durchlauf wird die Adresse geprüft und der Schritt auf 2 gesetzt die Adresse ist
 .hier Variabel da diese im laufenden Betrieb geändert werden kann

.Da es beim dritten Durchlauf mehrere Möglichkeiten gibt ist diese Abfrage etwas größer

```
:Befehlsabfrage
.....Systeminformationen.....'
If Empfangen = 105 Then            'Abfrage des Befehls Systeminformationen
""i
Gosub Systeminformationen
End If
.....Daten senden.....'
"If Empfangen = 115 Then           'Abfrage des Befehls Daten senden "s
Gosub Datensenden
End If
.....Reset.....'
If Empfangen = 114 Then            'Abfrage des Befehls Daten Zurücksetzen
""r
```

```

Wait 1
Schritt = 0          '1S warten um watchdog auszulösen
End If
.....Ventilator_EIN.....'
'If Empfangen = 86 Then          'Abfrage des Befehls Ventilator_EIN "V
Pwm1a = 153          'PWM max 153 für 12V 80=6V
Schritt = 0          'Schritt auf Null zurückgesetzt
Enable Urxc          'Interrupts werden wieder eingeschaltet
End If
.....Ventilator_AUS.....'
'If Empfangen = 118 Then          'Abfrage des Befehls Ventilator_AUS "v
Pwm1a = 0            'PWM für 0V
Schritt = 0          'Schritt auf Null zurückgesetzt
Enable Urxc          'Interrupts werden wieder eingeschaltet
Else
Schritt = 0          'Schritt auf Null zurückgesetzt
Enable Urxc          'Interrupts werden wieder eingeschaltet
End If

```

- .Wenn das dritte Zeichen „i“ ist sollen die Systeminformationen gesendet werden
- .Wenn das dritte Zeichen „s“ ist sollen die gemessenen Daten gesendet werden
- .Wenn das dritte Zeichen „r“ ist soll das Modul rückgesetzt werden
- Wenn das dritte Zeichen „V“ ist soll der Ventilator eingeschaltet werden
- .Wenn das dritte Zeichen „v“ ist soll der Ventilator ausgeschaltet werden
- .Unabhängig vom Zeichen wird der Schritt wieder auf null gesetzt

Serielle Datenübertragung

- .Für das Senden von Daten wird an sich nur ein Befehl gebraucht nämlich der „print“ Befehl
- Um mehrere Daten in einer Reihe zu senden müssen diese mit „;“ getrennt sein und um
- .Wörter zu senden, müssen diese in Gänsefüßchen stehen

```

Waitms 30          'Reaktionszeit Abwarten
Print Ticks ; Chr(9) ; O2_lesbar ; Chr(9) ; "EOD"      'Daten Senden
Waitms 30          'Reaktionszeit Abwarten

```

Da wir hier mit einem Tristane Signal arbeiten müssen wird zusätzlich 30ms vor dem senden ein Pin eingeschaltet und 30ms danach ausgeschaltet dies dient zum Schutz vom beidseitigen .senden

Das Chr(9) steht für das ASCII Zeichen der Tab um sicher zu gehen dass dieses Zeichen .korrekt angezeigt wird

Bei der Verbindung der seriellen Datenübertragung zum Computer muss darauf geachtet werden dass der Computer mit 12V und der uController mit 5V. Außerdem sind die Logische 1 beim Computer 12V und beim uController GND, sowie die Logische 0 beim uController .5V und beim Computer GND

(Die Puls Weiten Modulation (PWM

Da ich nur 20V für meinen Ventilator zur Verfügung hatte beschloss ich ein PWM Signal zu nutzen

.Um dies zu konfigurieren muss man den Timer1 konfigurieren

Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Down , Prescale = 256

'PWM für Ventilator

Der Timer1 wird als PWM konfiguriert. Das PWM Signal wird als 8 Bit variabel definiert.

Bei höherem Zählerwert als das eingestellte Verhältnis wird das Signal auf null gesetzt. Die

.Prescale definiert die Pulsfrequenz

Pwm1a = 153

'PWM max 153 für 12V 80=6V

.Mit diesem Befehl wird das Puls Pausen Verhältnis definiert

.Für 12V Mittelwert muss das Verhältnis $(255/20) \times 12$ sein dies ergibt einen Wert von 153

.Zum ausschalten wird die Variabel auf Null gesetzt

Der Watchdog

Dieser Befehl dient zur Überprüfung des Betriebs. Dies ist ein Timer welcher in regelmäßigen Abständen rückgesetzt werden muss. Wenn dies nicht der Fall ist wird der

.uContoller rückgesetzt

.Der Watchdog dient also als Art Totmannschalter

Config Watchdog = 1024

""Totmann" Timer

Start Watchdog

""Totmann" Überwachung starten

Der Timer wird hier auf 1 Sekunde gestellt und auch sofort eingeschaltet wenn der Timer

.nicht innerhalb 1s rückgesetzt wird löst dieser aus

Reset Watchdog

""Totmann" bestätigen

Wenn dieser Befehl kommt wird der Totmannschalter betätigt und das Programm läuft

.weiter

Das Interface

Einleitung

Das Interface ist das Teil welches eine Verbindung von den Modulen zur Masterstation herstellt

Das Interface erfüllt verschiedene Funktionen es soll die Module mit der Masterstation verbinden ohne dass sich die Module gegenseitig stören und es soll das Signal auf ein Tristatesignal umwandeln und wieder zurück, da nur ein 3 adriges Kabel und kein 4 adriges (+20V, Data und GND stehen zur Verfügung). Außerdem soll das Interface einen

.Computeradapter besitzen

So teilt sich das Interface in 4 Teile auf

Die Modulkopplung .1

(Der Seriell zu Tristate Wandler (Moduleseite) .2

(Der Seriell zu Tristate Wandler (Computerseite) .3

Der Computeradapter .4

Die Modulkopplung

Die Modulkopplung ist der Teil des Interface welches dafür zuständig ist dass die Module alle Daten empfangen und senden können ohne sich gegenseitig zu zerstören und das W/R .Signal auch nicht kurzgeschlossen wird

(Der Seriell zu Tristate Wandler (Moduleseite)

Der Wandler setzt an sich das TXD in ein Tristatesignal um wenn etwas gesendet werden muss wird das W/R Signal des uControllers aktiviert und das Signal wird freigeschaltet. Wenn nicht gesendet wird soll das W/R Signal null sein um die Schaltung in Tristate zu setzen und Daten so einfach empfangen werden können. Für den RXD Pin wird das Signal nur auf 5V begrenzt. Zusätzlich ist in diesem Teil noch ein Schaltnetzteil welches die .Spannung auf 5V runtergesetzt

(Der Seriell zu Tristate Wandler (Computerseite)

Dieser Teil ist identisch mit dem Wandler an der Moduleseite der einzige Unterschied liegt .darin dass dieser Teil kein Schaltnetzteil besitzt

.Der Computeradapter soll den Spannungs- und Pegelunterschied ausgleichen
Das heißt, dass die Spannung von 20v auf 12V runtergesetzt wird und dass alle Signale
.Invertiert werden müssen

Schaltungen

:Die Teilschaltungen sind
Die Modulkopplung .1

(Der Seriell zu Tristate Wandler (Moduleseite .2

(Der Seriell zu Tristate Wandler (Computerseite .3

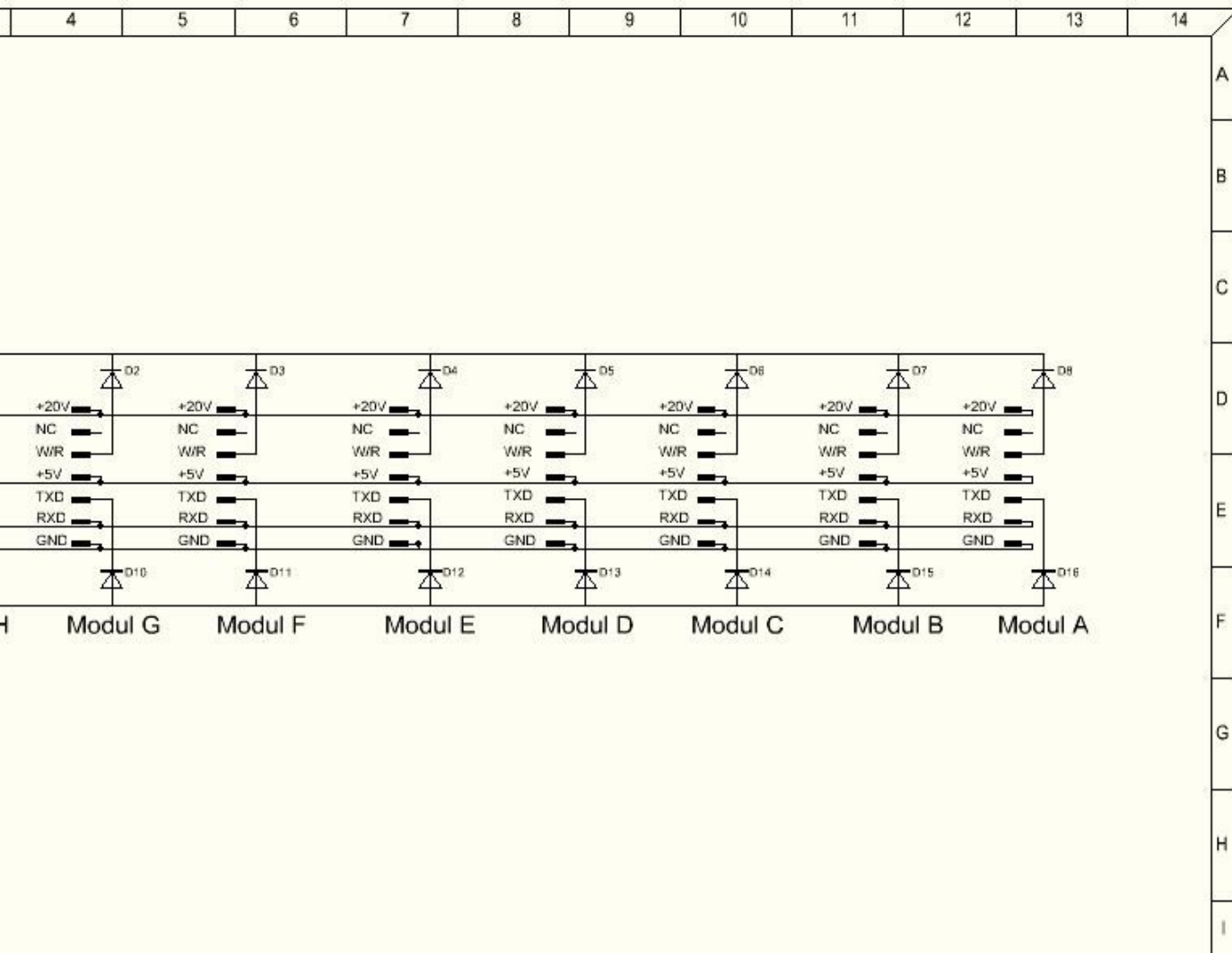
Der Computeradapter .4

Die Modulkopplung

Bei der Modulkopplung werden an sich alle Pins sofort verbunden nur der RXD und der W/R werden über Dioden Technik verbunden die RXD werden als AND verbunden da der Ruhezustand immer +5V ist. Der W/R wird als OR verbunden da jedes Modul dies auf +5V .setzt wenn es sendet

Materialste

Größe	Anzahl	Wert	Bauteil	Bauteil Name
10mm	16	1n4102	Dioden	D1-D16
10mm	2	kOhm 10	Widerstand	R1-R2
20mm	8	Pin 7	Messersteckerbuchsen	
30mm	1	Pin 10	Regenbogenkabelstecker	
200mm x 100mm			Platine	



	Datum	Name	Bezeichnung	Blatt
gez.:	19.05.2015	Thill Dan	Interface Teil1 Messstation Musée des mines 2014/2015 Modulkopplung	1
gepr.:				von
Norm:			Zeichnungs-Nr.: 20150519002	2

(Der Seriell zu Tristate Wandler (Modulseite

:Wenn Tristate = H

T4, T5, T6, sind leitend: T2 und T3 sind ohne Basisstrom, also sperrend, der Ausgang ist .hochohmig

:Wenn Tristate = L

.T4, T5, T6, sperren, der Totem pole Ausgang ist aktiviert

:TxD = 0

T1 sperrt, T3 erhält Basisstrom über RC1, T3 zieht den Ausgang auf 0

TxD = 1

(T1 ist leitend, T2 erhält Basisstrom und ist leitend, T2 zieht den Ausgang auf H (+20V

:Die LED verhindert ein Problem das bei TxD=1 entstehen würde

Der Basisstrom von T3 macht einen Spannungsabfall an RC1 von ca. 2V, so dass T2 nicht .sperren würde. Da an der LED ca. 2V abfallen ist das Problem gelöst

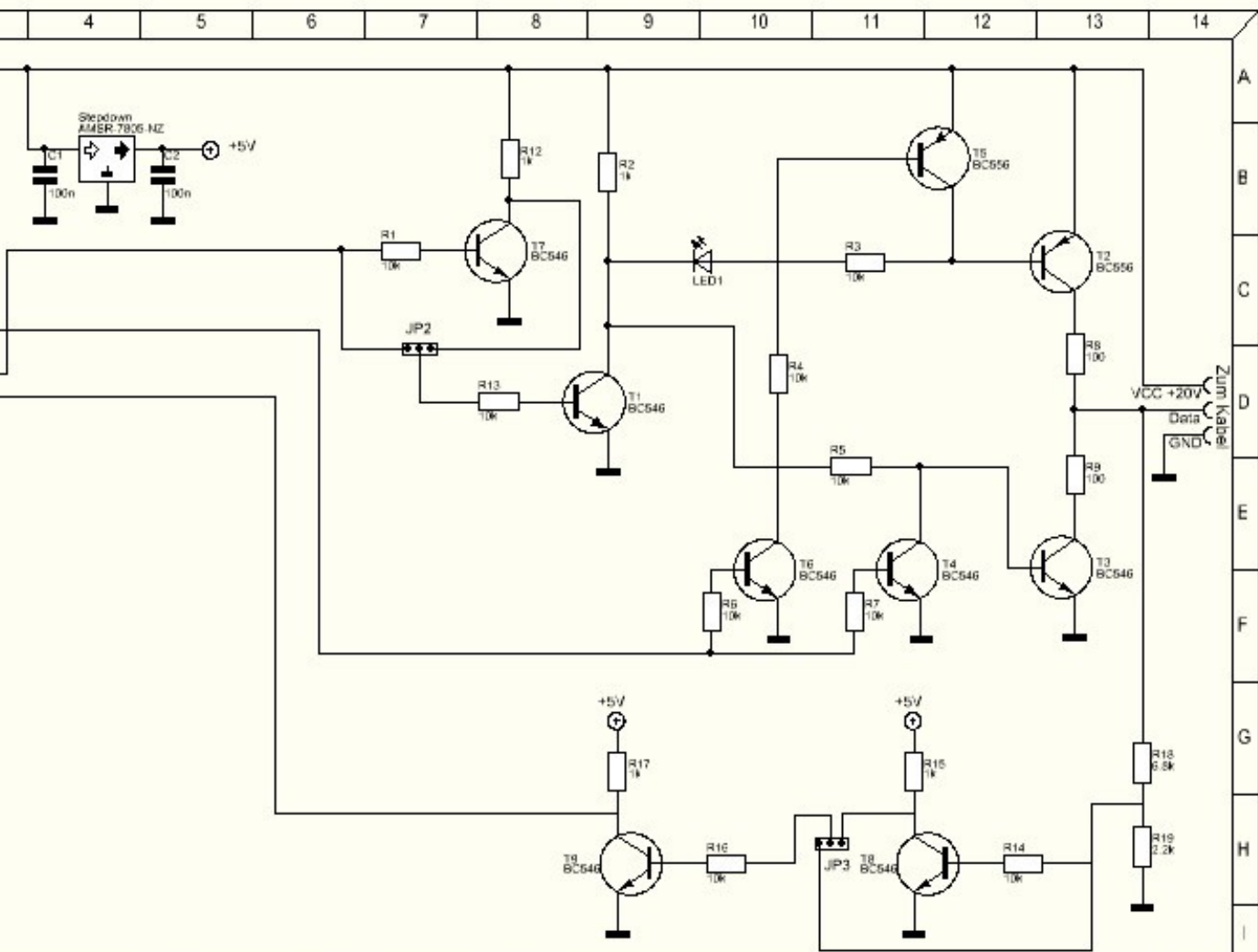
T8 und T9 dienen zur Begrenzung der Spannung am RXD Pin der uController auf 5V

T7 und T8 dienen auch als Inverter für das TXD, RXD Signal falls erwünscht

Da die uController nur W/R und nicht Tristate arbeiten ist im Kabel dieses Signal außerdem auch noch ein fixer Inverter eingebaut

Materialliste

Größe	Anzahl	Wert	Bauteil	Bauteil Name
	2	PNP	Transistor	T2+T5
	2	NPN	Transistor	T1+T3+T4+T6-T9
10mm	9	10 kOhm	Basiswiderstände	R1+R3+R4+R5+R6+R7+R13+R14+R15
10mm	4	1 kOhm	Kollektorwiderstände	R2+R12+R15+R17
10mm	2	100 Ohm	Widerstände	R8+R9
10mm	1	6,8 kOhm	Widerstand	R18
10mm	1	2,2 kOhm	Widerstand	R19
30mm	1	Pin 10	Regenbogenkabelstecker	
15mm	1	Pin 3	Printklemmen	
160mm x 100mm			Platine	
15mm x 5mm	1	ASMR- 7805- NZ	Stepdown Converter	
2,5mm	2	1uF	Kondensator	C1+C2



	Datum	Name	Bezeichnung	Blatt
gez.:	19.05.2015	Thill Dan	Interface Teil2 Messtation	3
gepr.:			Musée des mines 2014/2015	
Norm:			Zeichnungs-Nr.:	von
			20150519003	3

[\(Der Seriell zu Tristate Wandler \(Computerseite](#)

Siehe:

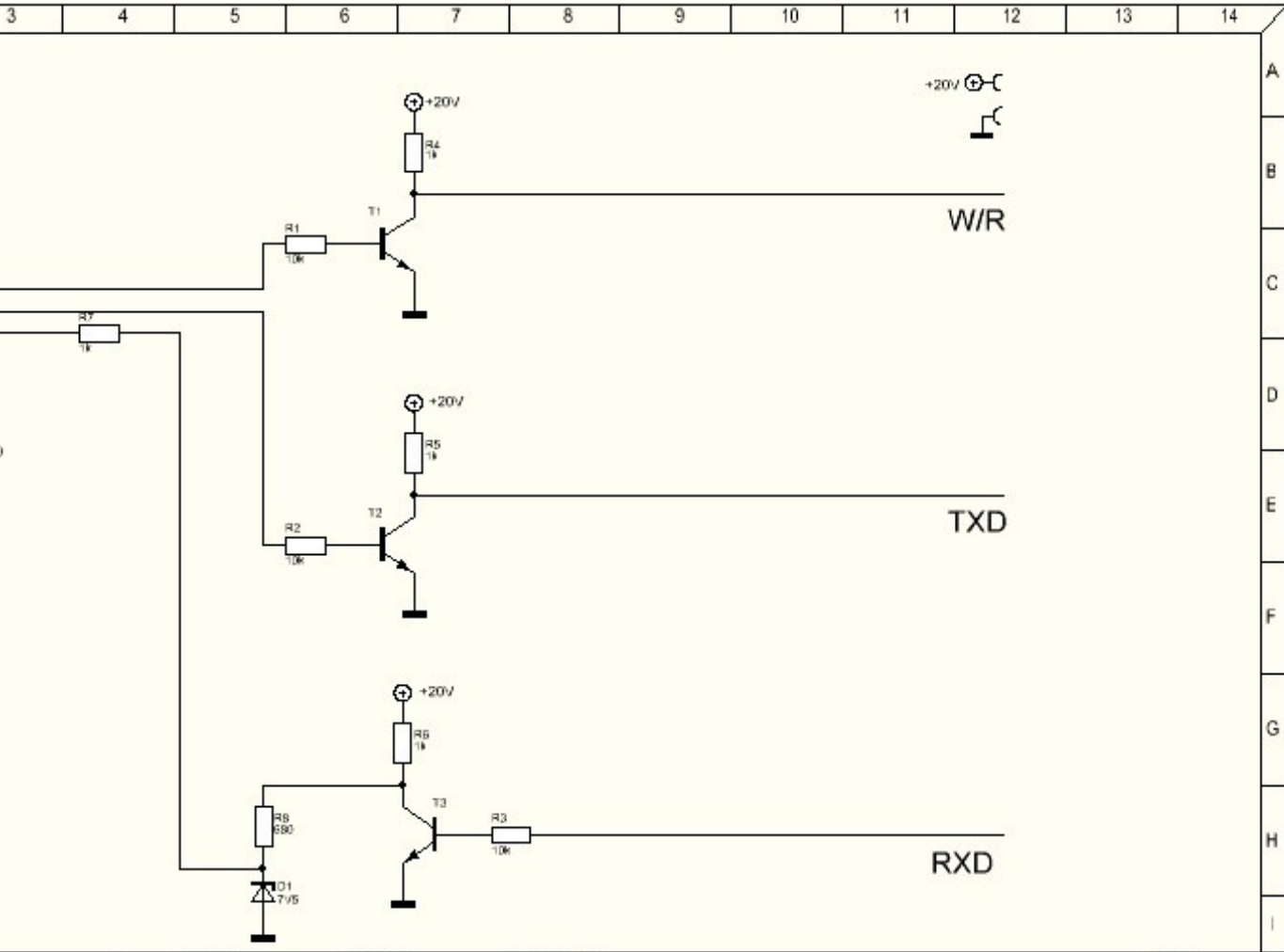
[Der Computeradapter](#)

.Der Adapter besteht an sich nur aus drei Transistoren als Schalter und einer Zenerdiode

Materialliste

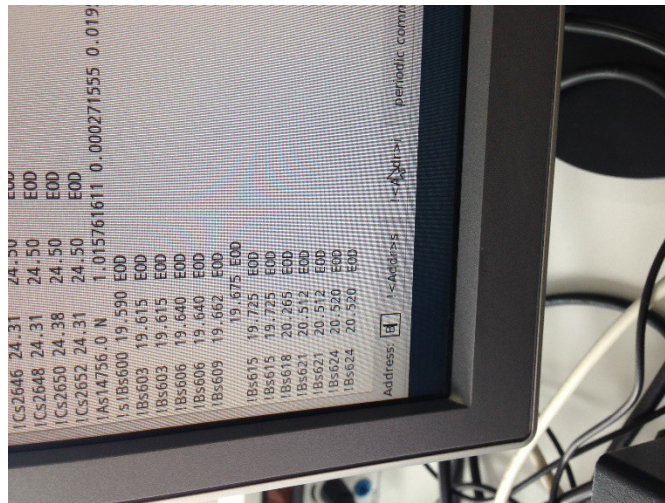
Anzahl	Wert	Bauteil	Bauteil Name
1	7.5V	Zenerdiode	ZD1
3	kOhm 10	Widerstand	R1-R3
3	kOhm 1	Widerstand	R4-R6

1	Pin 4	Messersteckerbuchse	
1	Pin 4	Messerstecker	
1	Pin 2	Printklemme	
1	kOhm 1	Widerstand	R7
1	Ohm 680	Widerstand	R8
		Lochrasterplatine	

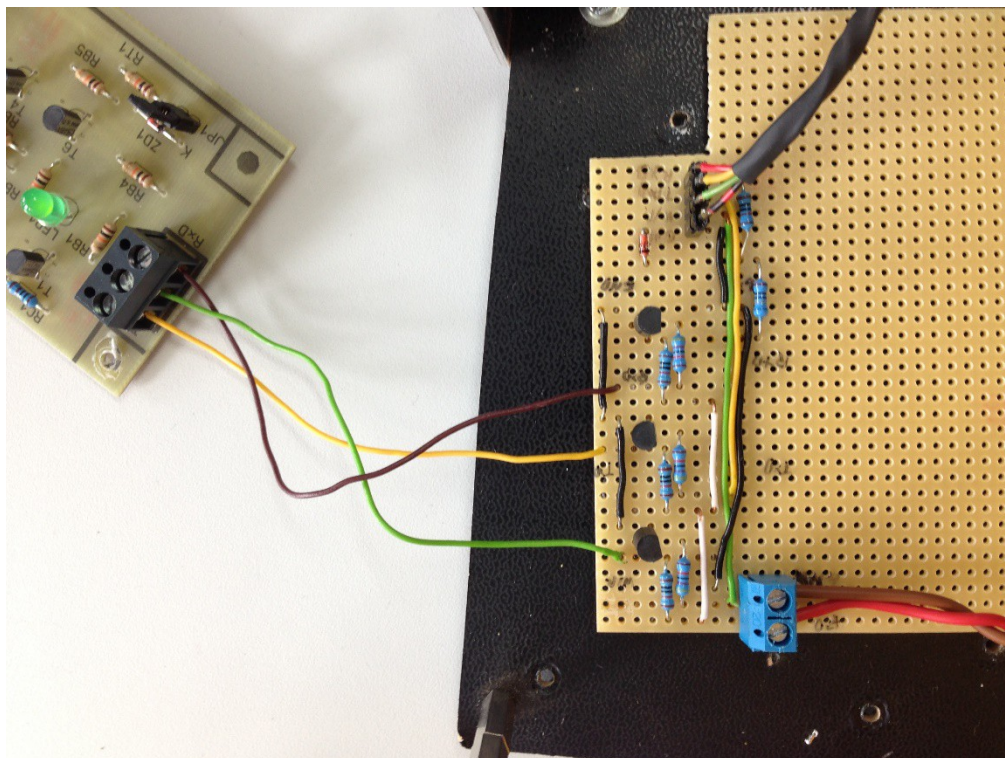


	Datum	Name	Bezeichnung	Blatt
gez.:		Thill Dan	Interface Teil3 Messtation	4
gepr.:			Musée des mines 2014/2015	Computeradapter
Norm:			Zeichnungs-Nr.:	von
			20150525001	4

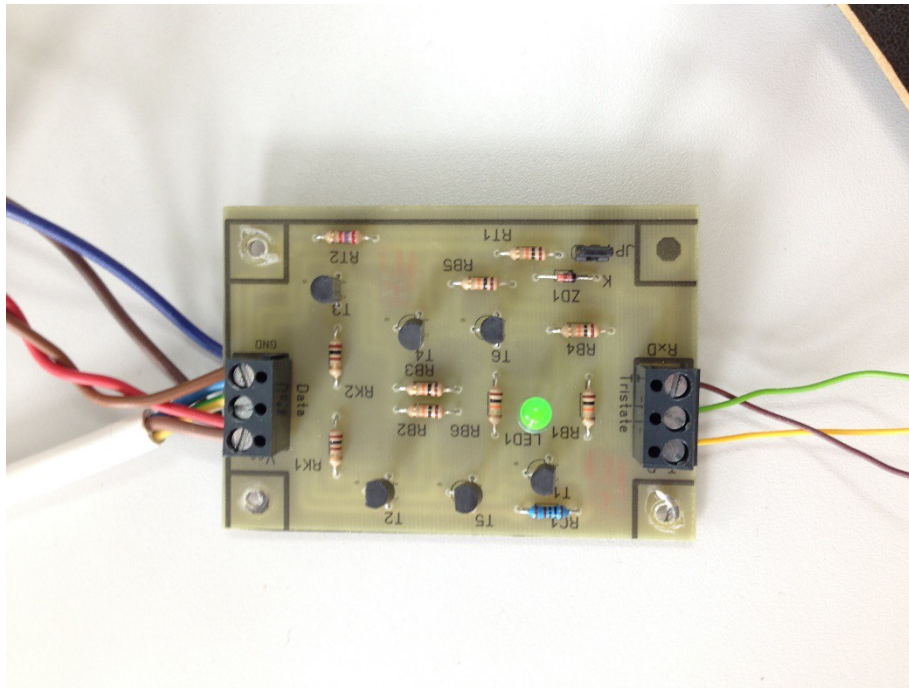
Foto-Blockaufbau der gesamten Messstation



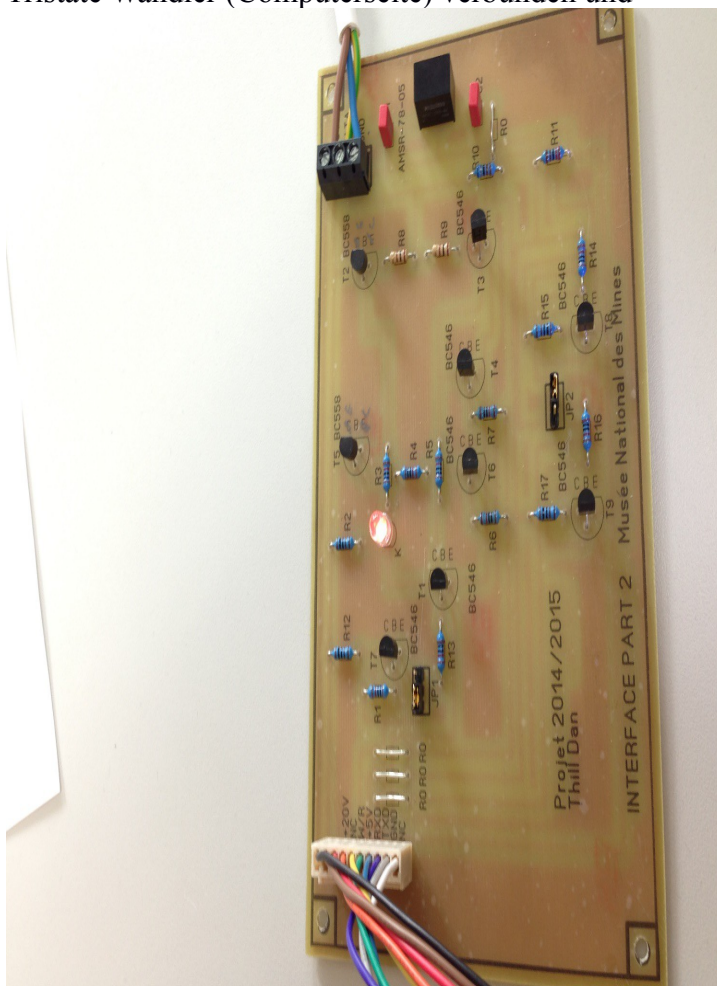
Der Computer ist mit dem



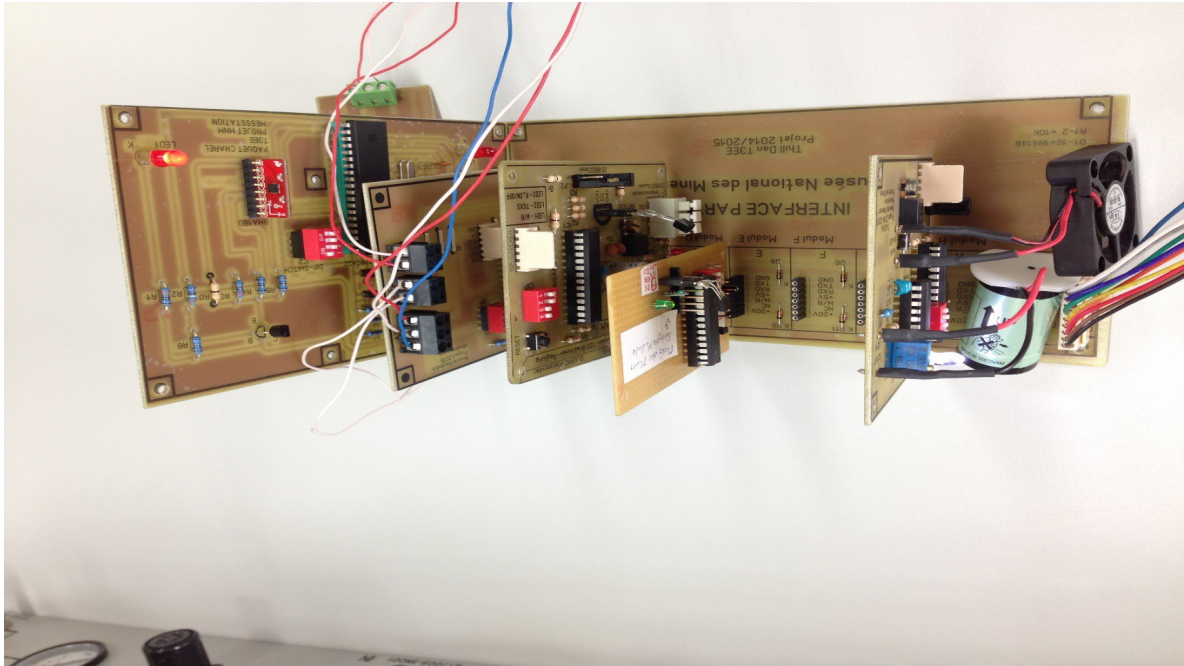
Computeradapter verbunden dieser ist mit dem



Tristate Wandler (Computerseite) verbunden und



Über das Kabel mit dem Tristate Wandler (Modulseite) zum Schluss geht es



zum Modulkopplungsteil mit den Modulen