



PROFE 2

Lycée Technique des Arts et Métiers Projet de fin d' études 2014/2015

Rédacteur: ACOSTA Alejandro
Classe: T3EE
Tuteur: FELTES Jean Claude
Projet: Messstation im Musée Nationale des Mines

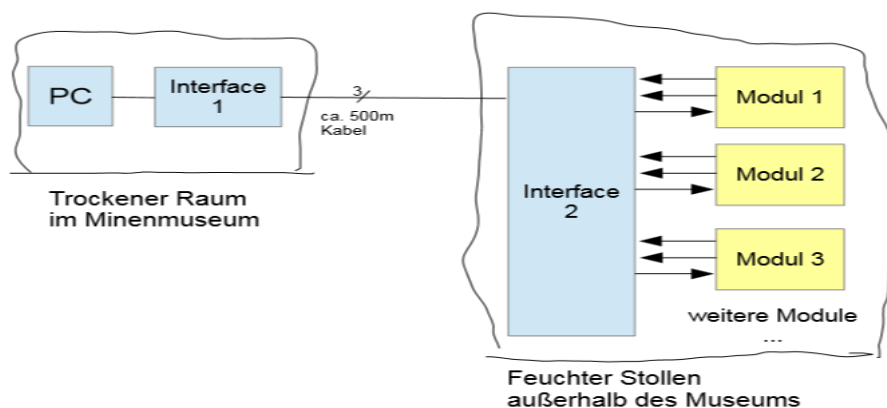
Inhaltsverzeichnis

Einleitung.....	3
Projekt	3
Spezifikationen für Module	4
Windgeschwindigkeits- und Windrichtungsmessung.....	5
Einleitung.....	5
Windgeschwindigkeitsmessung.....	5
Timer0 Als Counter.....	7
Test des Timer0 Als Counter.....	7
Windrichtungserfassung.....	8
DS18B20.....	8
Beschaltung des DS18B20.....	9
Erste Versuche mit DS18B20.....	9
Schaltung.....	11
Mikrocontroller Atmega 8.....	11
Windgeschwindigkeitsmesser.....	11
Windrichtungsmesser.....	11
Programmierung.....	12
Flussdiagramme.....	12
Serielle Kommunikation und die Befehlsverarbeitung.....	13
Schwierigkeiten bei der Programmierung.....	14
Serieller Interrupt	14
Befehlskette.....	14
Unterprogramm Bereitschaft.....	15
Unterprogramm Adressabfrage	15
Unterprogramm Befehlsabfrage.....	15
Datensenden.....	15
Reset.....	15
Eigenständigkeitserklärung.....	16
Anhang.....	17
Schaltungen.....	17
Windrichtungsmesser.....	17
Atmega 8.....	18
Layout.....	19
Modul.....	20
Bascom Programm.....	21
Bascom Programm Online.....	25
Quellen.....	26

Einleitung

Projekt

In diesem Projekt geht es darum eine Messstation im Musée Nationale des Mines umzubauen. Die Bereits bestehende Messstation misst den CO₂ wert, nun soll die Messstation erweitert werden. Die Station soll den Sauerstoffwert, die Windgeschwindigkeit sowie Windrichtung und Seismischen Aktivitäten erfassen. Die Messstation ist über ein Kabel mit einer Empfangsstation (Masterstation) verbunden. Die Masterstation schickt Kommandos zur Messstation, empfängt die Daten und speichert sie in einer LOG-Datei.



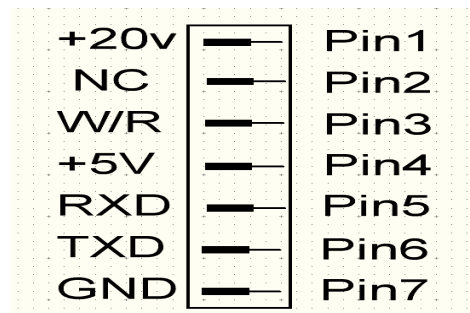
Für das Projekt 2014/2015 sollen die gelb markierten Module einer neuen Messstation entwickelt und gebaut werden. Die blau markierten Teile bestehen oder werden vom Betreuer realisiert. Außerdem soll eine Temperaturregelung eingebaut werden, damit die Innentemperatur der Mess-Station auf ca. 20°C geregelt wird, zur Korrosionsvermeidung. Alle Module werden an den Bus des Interfaces 2 angeschlossen. Die Steuerung erfolgt über TxD und RxD der seriellen Schnittstelle und W/R zur Kontrolle der Datenrichtung.

Spezifikationen für Module

- 1) Zur Verfügung steht eine Betriebsspannung von +5V und +20V
- 2) Daten und Kommandos werden über die serielle Schnittstelle ausgetauscht. Die Baudrate soll mit einem Jumper auf 2400 oder 9600 einstellbar sein.
- 3) Jedes Modul soll seine eigene variable Adresse haben. Diese soll über einen Dip Schalter einstellbar sein.
- 4) Module sollen auf Befehl/Kommando reagieren
 - Befehl: - ! bringt die Module in Bereitschaft
 - Adresse Moduladresse wählbar im Bereich der lesbaren ASCII-Zeichen
 - Kommando "s"=sende Daten "i"=sende Modulinfo "r"=Reset

Beispiele : !Ds / !Di / !Dr befehle für Modul mit Adresse D

- 5) Für Debuggingzwecke benutzen wir einen zweiten Jumper dieser gibt den Befehl bei jedem Durchlauf die Daten zu senden.
 - JP2 gesetzt: kontinuierlich senden
 - JP2 offen: auf Kommando senden
- 6) Die Daten werden in Textform gesendet, durch Tabulatorzeichen (0x09) getrennt. Am Ende wird "EOD" (= End of Data) und ein Zeilenvorschub gesendet (0x0D 0x0A).
- 7) Alle Module haben einen Alive-Ticker, der jede Sekunde inkrementiert wird. So kann man sehen ob ein Modul aktiv oder abgestürzt ist.
- 8) Die Aktivität wird von einem Watchdog überwacht
- 9) Zur Kontrolle der Übertragung hat jedes Modul einen Steuerausgang WR (Write / Receive), den es auf H setzt solange es Daten sendet.
- 10) Module sollen alle die Gleiche Pinbelegung haben. Damit diese leicht ausgewechselt werden können. Wir haben uns auf diese Pinbelegung geeinigt:



Windgeschwindigkeits- und Windrichtungsmessung

Einleitung

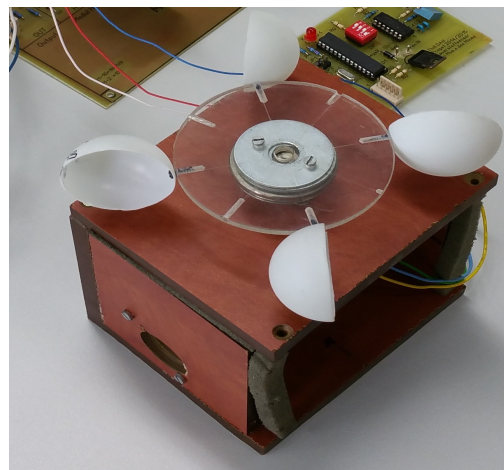
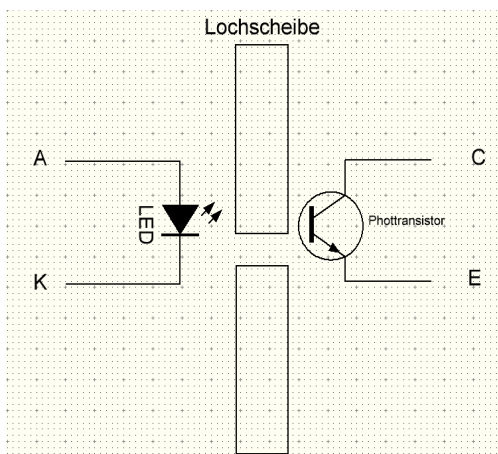
Der Windgeschwindigkeitsmesser soll die Windstärke in der Mine messen und die Daten zur Master Station senden. Diese Messung erfolgt mit Hilfe eines Windrades, eine und eine Lochscheibe. Dieser Aufbau war bereits vorhanden.

Der Windrichtungssensor soll die Windrichtung erfassen. Da diese in der Mine nur von zwei Richtungen kommt, Luxemburg oder Frankreich, habe ich mich entschlossen den Wind mit zwei Temperatursensoren und ein Leistungstransistor zu erfassen. Die Gemessenen Daten werden danach nach abfrage zur Master Station gesendet.

Windgeschwindigkeitsmessung

Um die Windgeschwindigkeit zu messen benutze ich einen vorhandenes Windrand das in einem Gehäuse mit einer 8 Löcher Lochscheibe und einer Gabellichtschanke verbunden war.

Durch die Drehung des Windrades dreht die Lochscheibe diese dreht in der Gabellichtschanke dadurch entsteht eine Pulsierende Gleichspannung. Diese wurde im Mikrokontroller verarbeitet und in eine Geschwindigkeit umgewandelt.

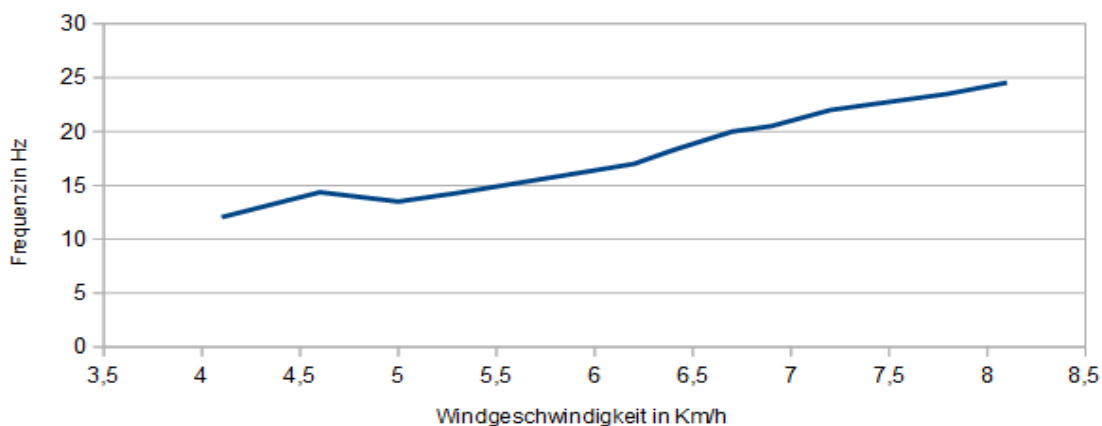


Um die Pulsierende Gleichspannung in eine Geschwindigkeit umzuwandeln benutzte ich die Funktion Timer0 als counter. Diese zählte die Impulse während eine bestimmten Zeit man erhält eine Frequenz. Um eine höhere Genauigkeit zu erhalten entschloss ich mir jedes mal die Impulse während 2 Sekunden zu zählen um nicht den Mikrocontroller lange aufzuhalten. Die gezählten Impulse wurden dann durch die Zeit geteilt. Die erhaltene Frequenz wurde mit einer Konstante Multipliziert um die Windgeschwindigkeit zu erhalten da die Drehung des Windrads Proportional zur Windgeschwindigkeit ist.

Um die Konstante herauszufinden wurden ein Paar Tests durchgeführt. Dazu habe ich das Windrad mit einer konstante Windgeschwindigkeit drehen lassen mit Hilfe eines Windkanals und mit einem Professionellen Windmesser dann die Windgeschwindigkeit gemessen. Danach habe ich die Frequenz der das Windrad liefert gemessen. Durch Teilung der Geschwindigkeit mit der Frequenz erhält man die Konstante. Dies habe ich mit verschiedenen Windgeschwindigkeiten durchgeführt und bekam folgende Ergebnisse.

Windgeschwindigkeit	Frequenz	Konstante	Durchschnitt
4,1	12,05	0,340	0,38
4,6	14,37	0,320	
5	13,5	0,370	
5,3	14,3	0,371	
6,2	17	0,365	
6,4	18,28	0,350	
6,7	20	0,335	
6,9	20,5	0,337	
7,2	22	0,327	
7,8	23,5	0,332	
8,1	24,54	0,330	

Verhalten Windgeschwindigkeit und Frequenz



In der obigen kurve sieht man dass Windgeschwindigkeit und die Frequenz das wir von der Gabellichtschränke erhalten ungefähr Proportional sind.

Timer0 Als Counter

Timer0 ist ein 8-Bit Timer welche eine Auflösung von 256 (2^8) hat. Ich habe mich entschieden den Timer0 zu benutzen da seine Auflösung von 8Bit völlig reicht. Denn wenn der Counter 256 in 2 Sekunden erreichen würde hätten wir eine Frequenz von 125,3 Hz das wäre eine Windstärke von $125,3\text{HZ} \cdot 0,38\text{Km/h} \cdot \text{Hz} = 47\text{Km/h}$ und ich denke dies ist in der Mine unmöglich.

Test des Timer0 Als Counter

```

$crystal = 8000000                                'Externer Quarz mit 8Mhz
$regfile = "m16def.dat"                            'Verwendung eines ATMEGA16Controllers

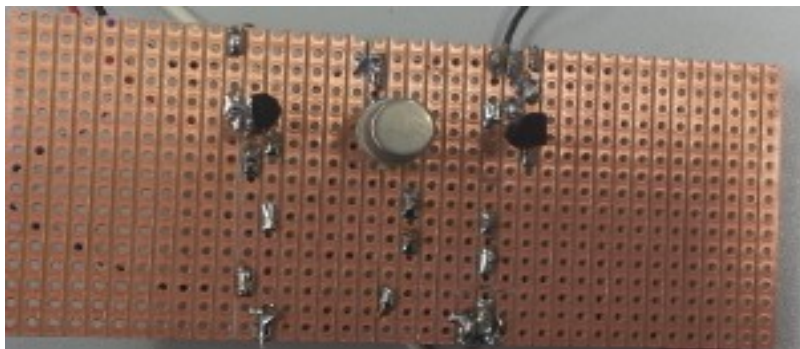
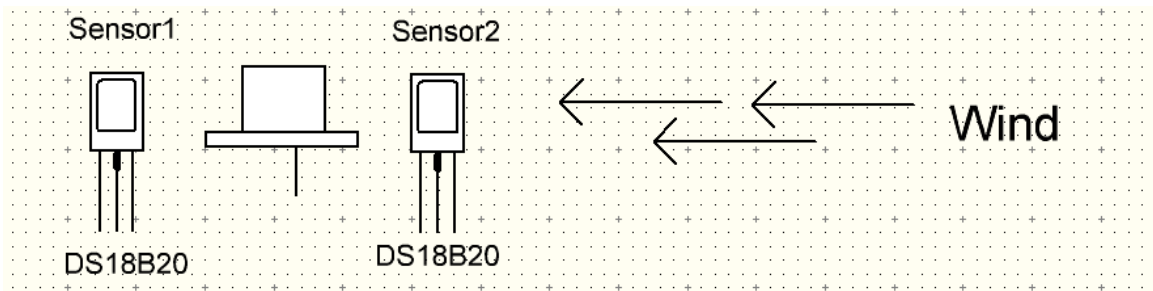
$hwstack = 100
$swstack = 100
$framesize = 100
'Initialiesierungen
Config Portb.0 = Input
Config Portc = Output
Config Lcdpin = Pin , Db4 = Portc.3 , Db5 = Portc.2 , Db6 = Portc.1 , Db7 = Portc.0 , E = Portc.4 ,
Rs = Portc.5
Config Lcd = 16 * 2
Cls

    Config Timer0 = Counter , Edge = Rising          'Timer0 als Counter Zählt die Impulse
    bei Positiver Flanke
Do
    Tcnt0 = 0
    Wait 3
    Print Tcnt0
    Cls
    Lcd Tcnt0
    Waitms 20
Loop
  
```

Einfaches Programm der die Impulse am PortB.0 während 3 Sekunden zählt und diese dann an der LCD anzeigt.

Windrichtungserfassung

Da in der Mine wo die unsere Messstation stehen wird der Wind nur von zwei Seiten bläst: Luxemburg oder Frankreich, habe ich mir überlegt die Windrichtung mit Hilfe von zwei Temperatur Sensoren zu erfassen. Zwischen den zwei Sensoren ist ein Leistungstransistor dieser dient dazu die Umgebung der Sensoren zu erhitzen. Damit diese bei der kleinsten Brise schon ein Temperatur unterschied merken. Außerdem werden Sensoren und Transistor in eine Art Tunnel stecken damit die Hitze nicht verloren wird.



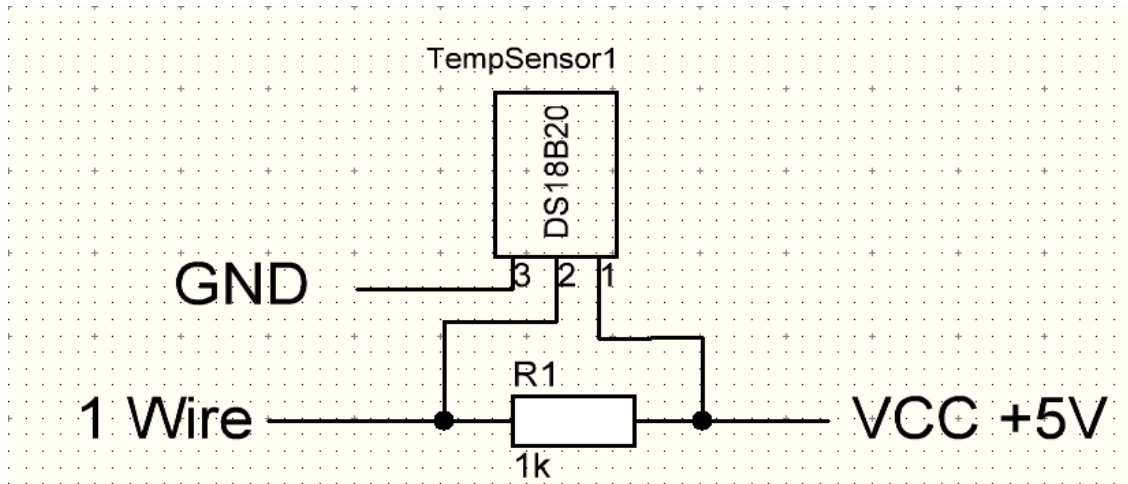
Im Obigen Bild sieht man dass wenn der Wind bläst dieser gegen den Sensor 2 prallt. Deswegen wird die Temperatur am Sensor2 kälter als die am Sensor1 sein. Der Transistor dient auch als Schutz damit nur ein Sensor vom Wind getroffen wird.

DS18B20

Der digitale Temperatursensor DS18B20 von Dallas Semiconductors eignet sich für einfache Temperaturmessungen mit einer 9 bis 12 Bits Genauigkeit also von bis zu 0,0625 Grad im Bereich $-55\text{ °C} \dots +125\text{ °C}$. Der Sensor ermöglicht ein direktes digitales Auslesen der gemessenen Temperatur, ohne dass A/D-Wandler benötigt werden und ist von Herstellerseite schon kalibriert. Er benutzt das 1-wire (one wire) Protokoll d.h. die Ansteuerung erfolgt über ein Eindraht-Interface (One Wire). Es können nahezu beliebig viele Sensoren auf einer einzelnen 1-wire Leitung angeschlossen werden, da jeder Sensor mit 64-Bit-ID adressiert wird. In diesem Fall wird zuerst die ID des Sensors per Software bestimmt und anschließend die Temperaturmessungen ausgeführt.

Datenblatt : <http://home.arcor.de/rachid.nouna/Program/datasheet/DS18B20.pdf>

Beschaltung des DS18B20



Erste Versuche mit DS18B20

Temperatur Messung mit 1 Sensor.

```

$crystal = 16000000
$regfile = "m16def.dat"
$hwstack = 100
$framesize = 100
$swstack = 100
$baud = 9600
'DS18B20
Config lwire = Portd.7
Config Lcdpin = Pin , Db4 = Portb.3 , Db5 = Portb.2 , Db6 = Portb.1 , Db7 = Portb.0 , E = Portb.4 ,
Rs = Portb.5

Dim Dsread(9) As Byte
Dim Itemperature As Integer At Dsread Overlay
Dim Temperature As Single
Do
  'sensor 1
  Gosub Measuretemperature
  Print Temperature
  Wait 1
  Cls
  Locate 1 , 2
  Lcd "Temperature"
  Locate 2 , 2
  Lcd Temperature
Loop
'-----
Measuretemperature:
  'Messen
  lwreset
  lwwrite &HCC
  lwwrite &H44
  Waitus 200
  'auslesen
  lwreset
  lwwrite &HCC
  lwwrite &HBE
  Dsread(1) = lwread(9)
  Temperature = Itemperature * 0.0625
Return
  
```

Temperatur Messung mit 2 Sensoren.

```
$crystal = 8000000
$regfile = "m16def.dat"
$hwstack = 100
$framesize = 100
$swstack = 100
$baud = 9600
'DS18B20
Config lwire = Portd.7
Config lwire = Portd.6
Config Lcdpin = Pin , Db4 = Portc.3 , Db5 = Portc.2 , Db6 = Portc.1 , Db7 = Portc.0 , E = Portc.4 ,
Rs = Portc.5
Dim Thpinr As Byte
Dim Dsread(9) As Byte
Dim Itemperature As Integer
Dim Temperature As Single
Dim S As String * 5
'-----
Do
  Cls
  'sensor 1
  Thpinr = 7
  Gosub Measuretemperature
  Print Temperature;
  Print Chr(9);
  Locate 1 , 1
  Lcd "Temp1:"
  Locate 1 , 7
  Lcd Temperature
  'sensor 2
  Thpinr = 6
  Gosub Measuretemperature
  Print Temperature;
  Print
  Locate 2 , 1
  Lcd "Temp2:"
  Locate 2 , 7
  Lcd Temperature
  Wait 1
Loop
'-----
Measuretemperature:
'Messen
lwreset Pind , Thpinr
lwwrite &HCC , 1 , Pind , Thpinr
lwwrite &H44 , 1 , Pind , Thpinr
Waitus 200
lwreset Pind , Thpinr
lwwrite &HCC , 1 , Pind , Thpinr
lwwrite &HBE , 1 , Pind , Thpinr

Dsread(1) = lwread(9 , Pind , Thpinr)

Itemperature = Makeint(dsread(1) , Dsread(2))

Temperature = Itemperature * 0.0625

Return
```

Schaltung

Meine Schaltung besteht aus 3 Teile:

- Mikrocontroller Atmega 8
 - Baudrate
 - Dauerhaftsenden
 - Adresse
 - Programmiergerät
 - Eingänge für Messsensoren
- Windgeschwindigkeitsmesser
- Windrichtungsmesser

Mikrocontroller Atmega 8

Beschaltung

- Reset an PC6 mit Pullupwiderstand.
- Betriebsspannung an VCC 5V geglättet durch 2 Kondensatoren und Masse an GND. Externer Oszillator 8Mhz an XTAL1 und XTAL2 mit 2 parallel geschaltete 22pf Kondensatoren.
- Die Baudrate wird mit 1 Jumper als Pullup an PD2 bestimmt.
- Dauerhaftsenden ebenfalls mit 1 Jumper als Pullup an PD3.
- Die Adresse wird mit einem 4Pin Dip Schalter bestimmt als Pullup an PortC bestimmt.
- Der Programmiergerät wird mit dem Reset, SCK, MISO und MOSI verbunden.
- Die Eingänge für die Messungen sind PD6 für den ersten Temperatur Sensor und PD7 für den zweiten. Der Windrad wird am T0/ PD4 angeschlossen.

Windgeschwindigkeitsmesser

Der Windgeschwindigkeitsmesser besteht nur aus der Gabellichtschranke, dem Windrad und der Lochscheibe. Die Gabellichtschranke wird mit 5V Betriebsspannung betrieben. Die genauere Schaltung weiß ich leider nicht da ich mich nicht richtig darum gekümmert habe. Ich habe nur die Funktionsweise analysiert.

Windrichtungsmesser

Der Windrichtungsmesser besteht aus zwei Temperatursensoren (DS18B20) Diese werden mit 5V Betriebsspannung betrieben. Die Datenleitung wird mit einem 4.1kOhm Pullup Widerstand am Mikrocontroller geschaltet. Der Transistor ist ein 2N22A dessen Strom auf 200mA mit Hilfe eines 1,1 kOhm Widerstand an der Basis begrenzt wird. Der Transistor heizt sich auf bis zu 75 °C dies habe ich mit Hilfe einer Wärmebildkamera gemessen die mir während meines Praktikums bei Cargolux zur Verfügung gestellt wurde.

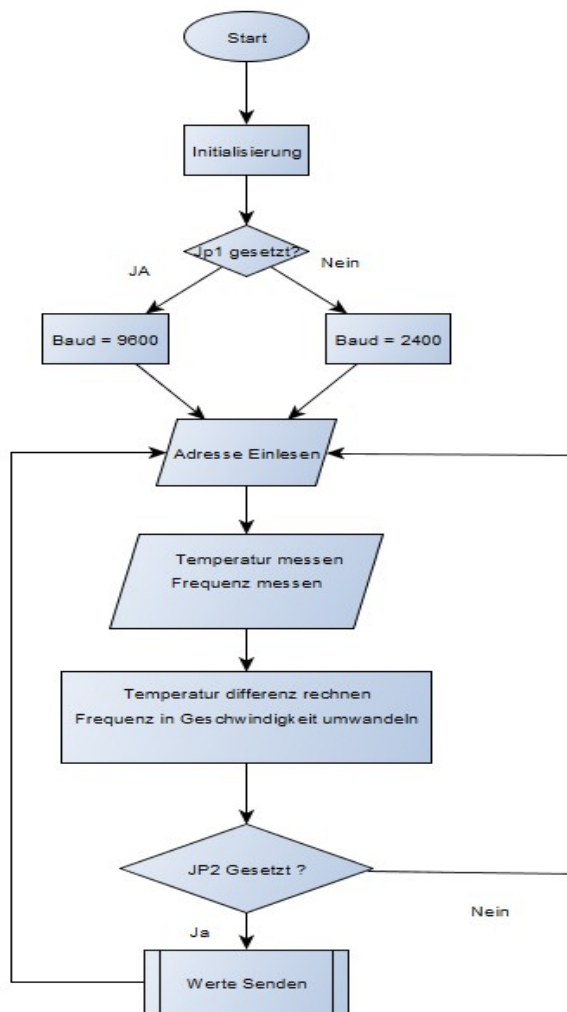
Programmierung

Die Programmierung Besteht aus zwei Teilen:

- Erstes die Werte der Sensoren einlesen und verarbeiten.
- Zweitens die Serielle Kommunikation und die Befehlsverarbeitung.

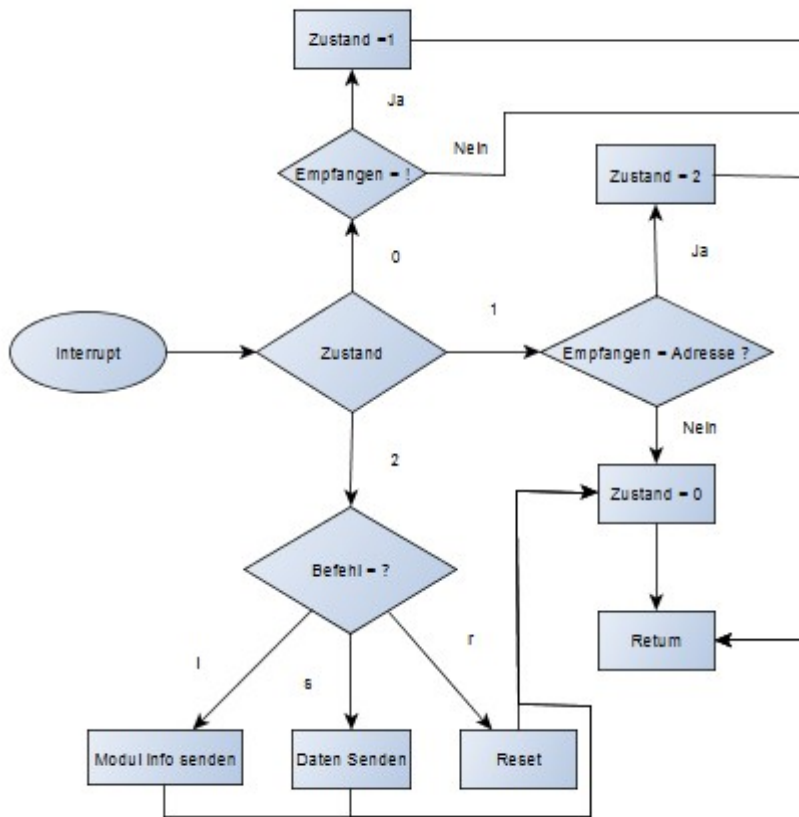
Flussdiagramme

Werte der Sensoren einlesen und verarbeiten



Als Erstes wird geschaut ob der Jumper1 der für die Baud zuständig ist gesetzt ist. Falls ja wird mit einer Baud von 9600 gearbeitet andersrum mit einer Baud von 2400. Dies wird nur beim einschalten des Mikrocontrollers oder beim Reset kontrolliert. Danach wird die Adresse eingelesen so wie die Werte der Sensoren. Diese werden dann umgewandelt. Sollte der Jumper zwei gesetzt sein werden dann die Daten gesendet und dieser Prozess wird wiederholt. Sollte Jumper zwei nicht gesetzt sein wird dann der Prozess nur wiederholt.

Serielle Kommunikation und die Befehlsverarbeitung



Wir arbeiten mit dem seriellen Interrupt. Dieser unterbricht den Programm beim erhalten eines Zeichen. Führt ein Unterprogramm aus und kehrt dann zum Programm zurück. Im Unterprogramm wird die Befehlskette bearbeitet.

Schwierigkeiten bei der Programmierung

Serieller Interrupt

Unterbricht den Programm führt ein Unterprogramm aus und kehrt zu Programm zurück
 Wenn ein Zeichen durch die serielle Kommunikation empfangen wird.
 Damit der Mikrocontroller auf Interrupts reagiert muss man 3 schritte folgen:

```

-----
'Serialinterrupt
-----
On Urxc Serialinterrupt           'Seriellerinterrupt Aktiviert
Enable Urxc                       'Interrupt ist aktiv
Enable Interrupts                 'allgemeine Interrupts aktivieren
  
```

Empfängt dieser ein Zeichen führt dieser den Unterprogramm aus.
 Hier Schrittabfrage was die Befehlskette ist verarbeitet.

Serialinterrupt:

```

  Disable Urxc                     'Interupt wird Abgeschaltet
  Gosub Schrittabfrage             'Abrufen des unterprogramms
  Return
  
```

Damit es keine Störungen gibt durch kommenden Zeichen wird der Interrupt deaktiviert.

```
Disable Urxc                       'Interupt wird Abgeschaltet
```

Und mit `Enable Urxc` wieder aktiviert.

Um zu Programm zurückzukehren wird den Befehl `Return` genutzt.

Befehlskette

Erhält man ein Zeichen durch den Seriellen Interrupt, wird dieser in einer Variabel
 gespeichert da den Seriellen Interrupt nur ein Zeichen speichert. Im Programm ist dies
 Variable „Empfangen“. Die Variable „Schritt“ sagt uns in welchen schritt wir uns befinden.

Schrittabfrage:

```

Empfangen = Udr                   'UDR-Register wird als Variabel
abgespeichert

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
Return
  
```

Unterprogramm Bereitschaft

Wird der Zeichen "!" empfangen wird der Schritt auf 1 gesetzt.

Dies bedeutet dass die Module abgefragt werden.

Wird ein anderes Zeichen empfangen wird werden die Variablen wieder auf null gesetzt.

Unterprogramm Adressabfrage

Empfängt man als nächstes du zum Modul dazugehörigen Adresse wird der Schritt auf 2 gesetzt. Mein Modul war das Modul mit Adresse D. Wird ein anderes Zeichen empfangen wird werden die Variablen wieder auf null gesetzt.

Unterprogramm Befehlsabfrage

Hier wird der Befehl verarbeitet dieser kann „i“ für Modul info, „s“ für sende Daten oder „r“ für Reset

Datensenden

Die Daten werden mit dem Print Befehl gesendet. Die Daten werden mit " ; " getrennt um diese in eine reihe senden zu können und mit Chr(9) getrennt werden. (Chr(9) = Tab)
Außerdem war eine der Anforderungen dass man sehen soll wenn was gesendet wird.
Dies wird mit einer LED ausgeführt die kurz vor dem senden eingeschaltet wird und kurz nachdem senden ausgeschaltet.(W/R)

```

Portb.2 = 1                                     'Aktivieren des Senden PINNs (W/R)
Waitms 30                                       'Reaktionzeit Abwarten
Print Ticks ; Chr(9) ; Temperature1 ; Chr(9) ; Temperature2 ; Chr(9) ; Diff ; Chr(9)
; V ; Chr(9) ; "EOD"
Waitms 30                                       'Reaktionzeit Abwarten
Portb.2 = 0                                       'Deaktivieren des Senden PINNs
Schritt = 0                                       'Schritt auf null zurückgesetzt
Enable Urxc                                       'Interrups werden wieder eingeschaltet

Return

```

Reset

Um Per Befehl reseten zu können. Benutze ich die Watchdog funktion.

Dieser ist ein Timer der während einer bestimmten Zeit läuft dieser muss jedes mal bevor die Zeit abläuft resetet werden sonst resete dieser den Mikrocontroller. Dieser wird auch benutzt um den Betrieb zu überwachen.

Im Befehlsabfrage wird falls der Zeichen "r" empfangen wird ein Watchdog konfiguriert der nach 1 Sekunde ablaufen soll. Mit ein Wait Befehl wird dann dort das Programm blockiert dies führt dazu dass der Watchdog den Mikrocontroller resetet.

```

'Reset

Elseif Empfangen = 114 Then
    Config Watchdog = 512
    Start Watchdog                                       'Abfrage des Befehls Daten Zurücksetzen

"r"

Wait 1
Enable Urxc                                       '1S warten um watchdog auszulösen

```

Eigenständigkeitserklärung

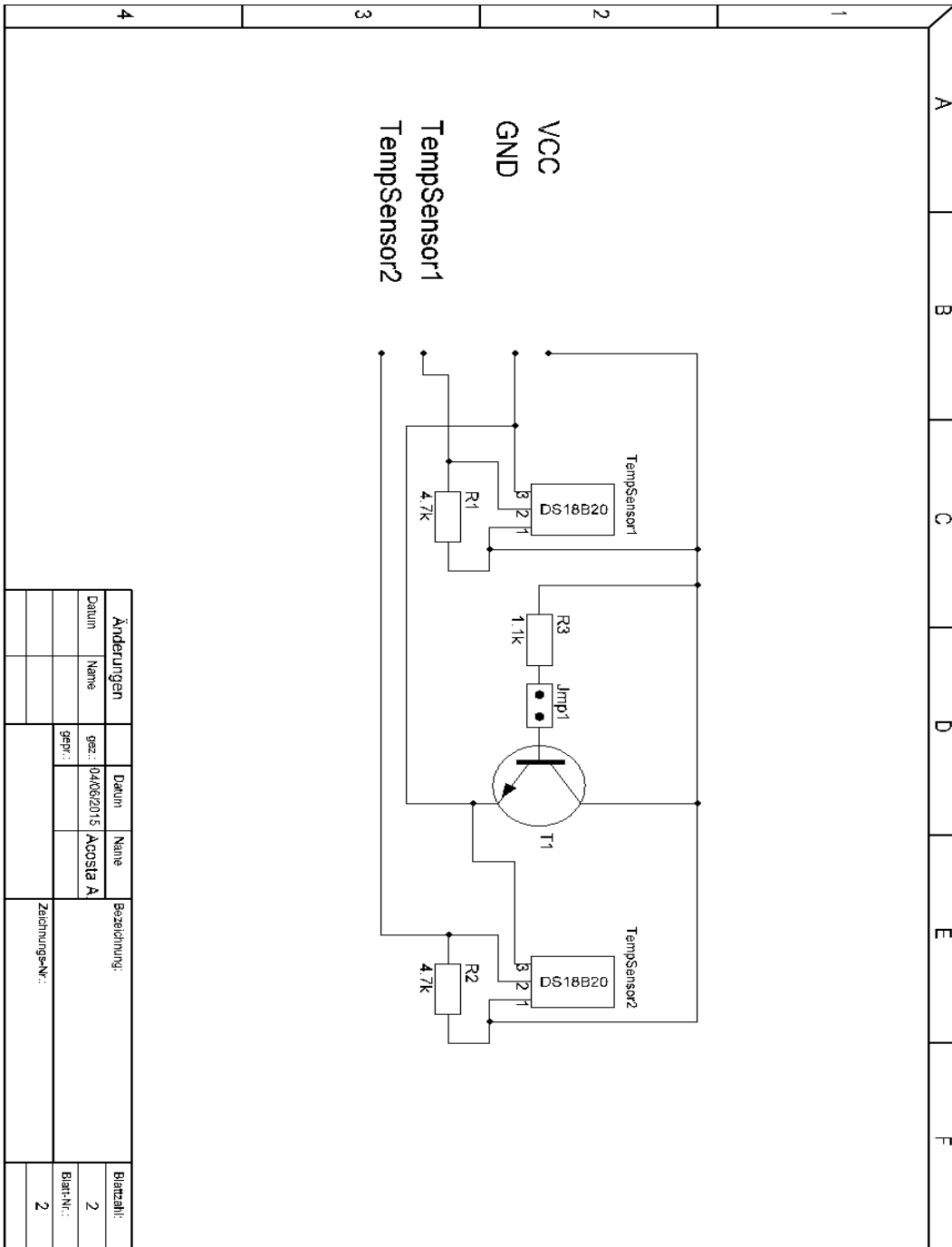
Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe.
Alle Passagen, die ich aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich mit Angabe der Quelle kenntlich gemacht.

Acosta Alejandro
Mersch 04/06/2015

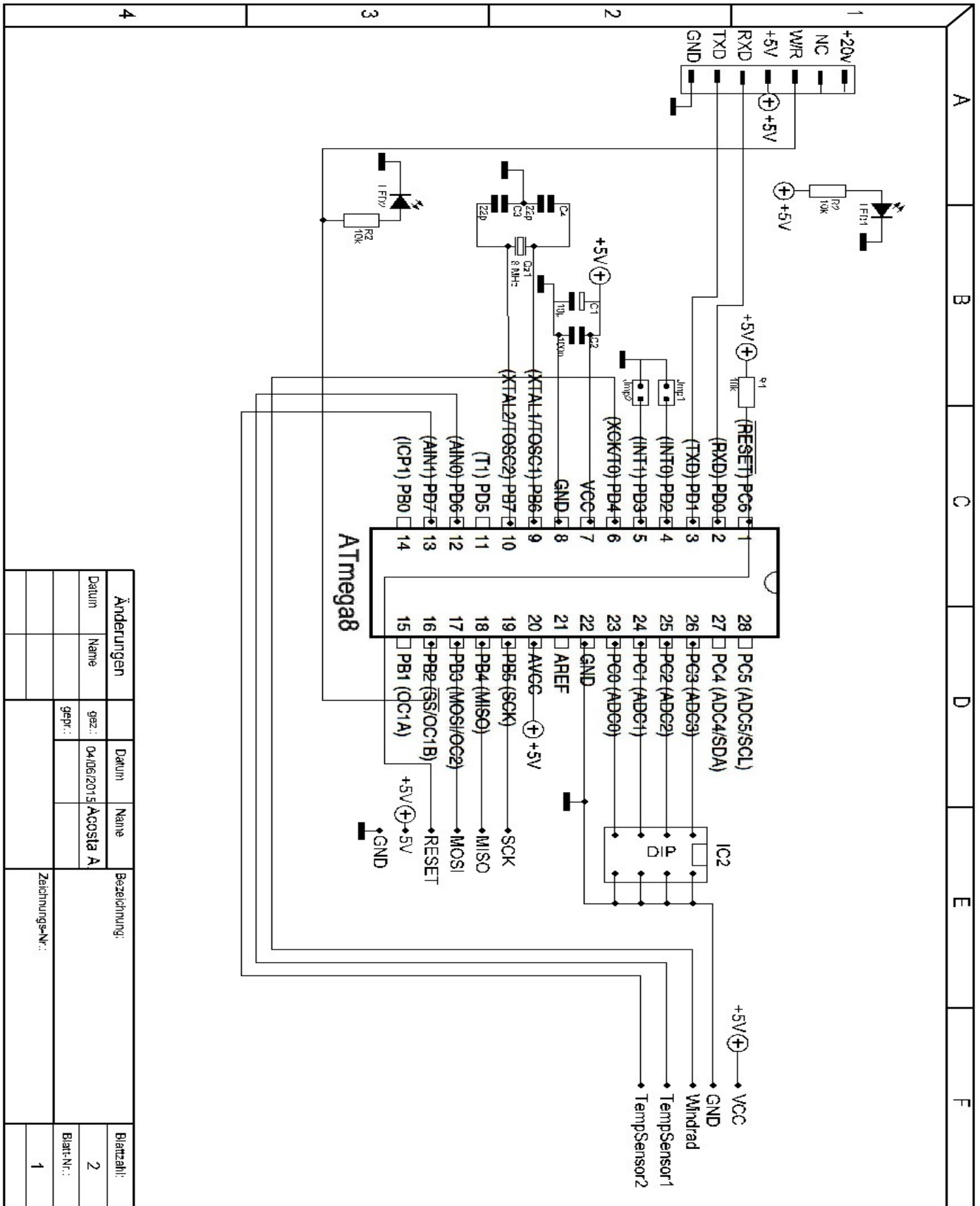
Anhang

Schaltungen

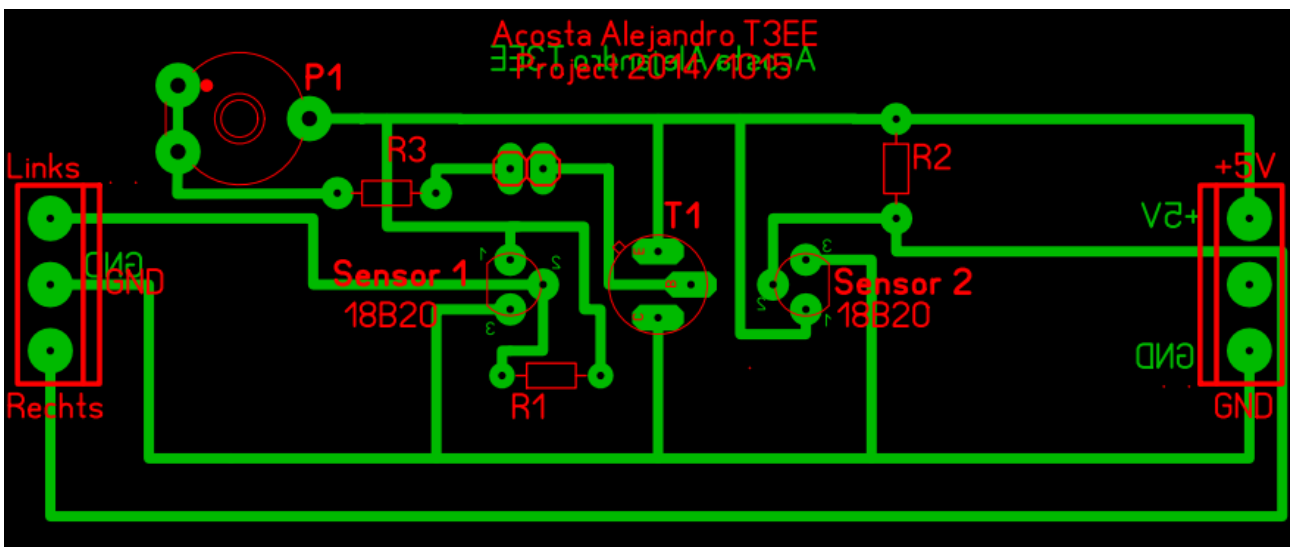
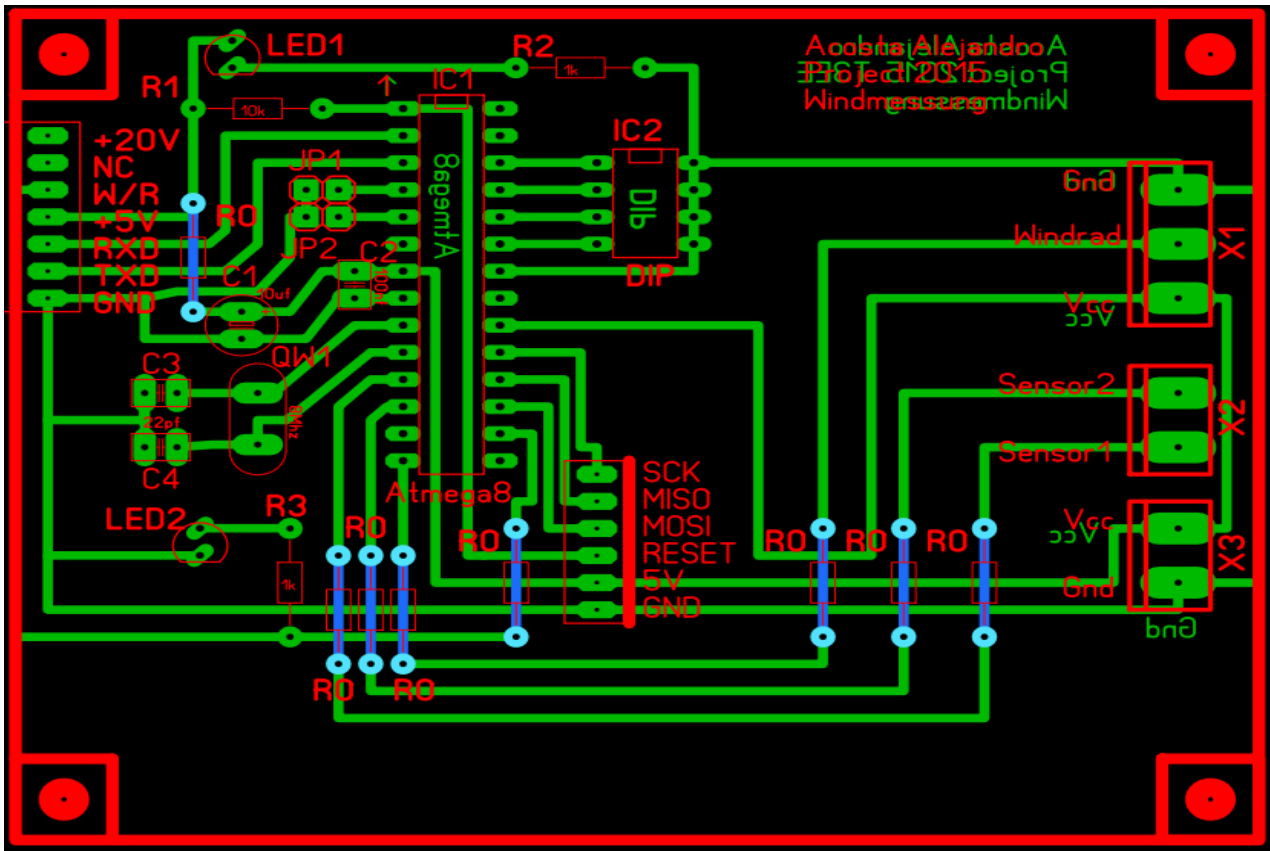
Windrichtungsmesser



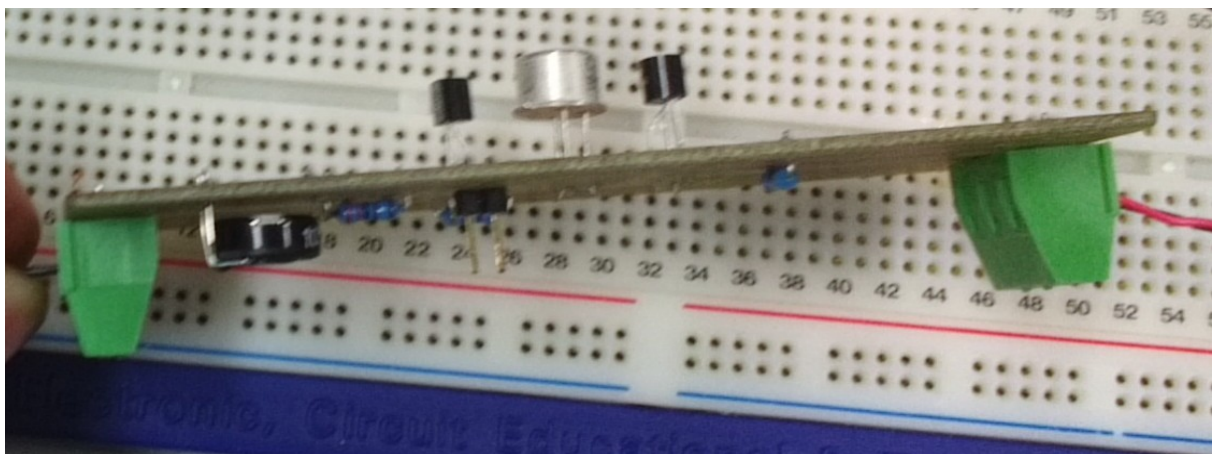
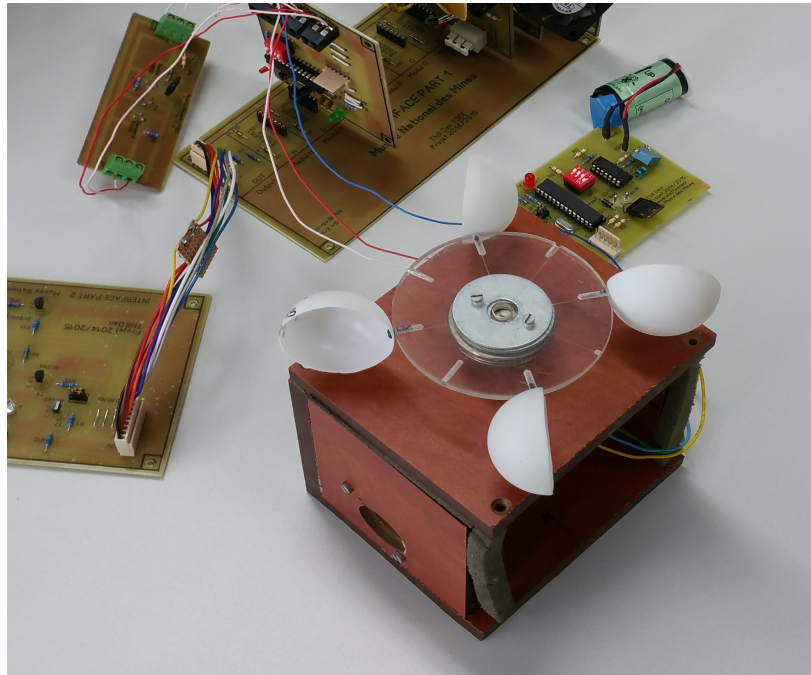
Atmega 8



Layout



Modul



Bascom Programm

```

*****
'Autor:      Acosta Alejandro
'Programm:   Messung von Windgeschwindigkeit und Windrichtung
'Schule:     LTAM
'Klasse:     T3EE
*****

'*****
'
'              INITIALISIERUNG
'*****

'Hardware
'-----
$crystal = 8000000           'Externer Quarz 8Mhz
$regfile = "m8def.dat"      'Atmega 8 Controller
$hwstack = 100
$framesize = 100
$swwstack = 100
'-----

'Inputs/Outputs
'-----
'DS18B20 1Wire Temperatur Sensor
Config lwire = Portd.5       'Input Sensor 1
Config lwire = Portd.6       'Input Sensor 2
'.....
Config Pind.4 = Input        'Input Windrad   PORTD.4
Config Pind.2 = Input        'JP Baudrate
Config Pind.3 = Input        'JP Dauerhaft senden
Config Portc = Input         'Adresseingang

Config Portb.2 = Output      'W/R Meldung
'-----
'Timer0 als Counter/Zähler
'-----
Config Timer0 = Counter , Edge = Rising      'Zählt die Impulse bei Positiver Flanke
'-----

'Pullups
'-----
Portc = 15                       'Pullup der Adresse
Portd.2 = 1                       'Pullup JP1
Portd.3 = 1                       'Pullup JP2
'-----

'Variablen
'-----
Dim Thpinnr As Byte
Dim Dsread(9) As Byte
Dim Itemperature As Integer
Dim Temperature As Single
Dim Temperature1 As Single
Dim Temperature2 As Single
Dim Tdiff As Single
Dim S As String * 5

Dim F As Integer
Dim Impulse As Integer
Dim Geschwindigkeit As Single
Dim V As String * 10

Dim Schritt As Byte           'Abfragevariable um nächsten Schritt
einzulesen
Dim Adresse As Byte          'Angeschlossene adresse binär mit ASCII
chart
Dim Empfangen As Byte
Dim X As Integer
Dim Ticks As Integer
'-----

'Baudrate
'-----
If Pind.2 = 0 Then           'Jumper1 gesetzt
Baud = 2400

```

```

Else                                                    'Jumper1 nicht gesetzt
Baud = 9600
End If
-----
'Watchdog
-----
'Config Watchdog = 2048                                '"Totmann" Timer
'Start Watchdog                                        '"Totmann" Überwachung starten
-----
'Serialinterrupt
-----
On Urxc Serialinterrupt                                'Seriellerinterrupt Aktiviert
Enable Urxc                                           'Interrup ist aktiv
Enable Interrupts                                     'allgemeine Interrupts aktivieren
'*****
'                               Hauptprogramm
'*****

Do
-----
'Adresse
'Diese wird als als ASCII Zeichen eingelesen
-----
    Adresse = Pinc                                    'Adresse Einlesen
    Toggle Adresse                                  'Adresse Invertieren um Positive Logig
zu erhalten
    Adresse = Adresse And &B00001111                'Maskieren um nur erste 4bit zu erhalten
    Adresse = 64 + Adresse
-----
'Windgeschwindigkeitmessung
-----

    Tcnt0 = 0                                        'Der Zähler wir auf 0 Gesetzt
    Wait 2                                           'Wartet 2s und Zählt die Impulse der
Gabellichtschranke

    Impulse = Tcnt0                                  'TCNT0 Wird als Impulse definiert
    F = Impulse / 2                                  'Impulse werden durch Zeit geteilt=
Frequenz

    Geschwindigkeit = F * 0.38                       'Geschwindigkeit ist die Frequenz mal
eine Konstante siehe Bericht

'Reset Watchdog                                       'Watchdog wird Resetet

-----
'Windrichtungserfassung Wind kommt von der Seite des kaltesten Sensors
-----
'                               sensor 1
'.....
Thpinnr = 5                                           'Sensor an PortD.5
Gosub Measuretemperature                             'Zum unterprogramm gehen

Temperature1 = Temperature                           'Die Temperatur wird als Temperatur1
definiert damit man die von nächsten sensor messen kann

'                               sensor 2
'.....
Thpinnr = 6                                           'Sensor an PortD.6
Gosub Measuretemperature                             'Zum unterprogramm gehen

Temperature2 = Temperature                           'Die Temperatur wird als Temperatur2
definiert

'                               Temperatur Unterschied
'.....

Tdiff = Temperature1 - Temperature2                 'Die differenz zeigt wie stark der
unterschied ist

'Reset Watchdog
-----
'Ticks um zu sehen wie oft der Programm durchlaufen wurde
-----

```

```

X = X + 2                                     'Der Livecounter wird gesetzt
Ticks = X                                    'Ticks
-----
'Dauerhaftsenden Nur zum Testen
-----
If Pind.3 = 0 Then                          'wenn JP2 gesetzt dann dauerhaftsenden
  Portb.2 = 1                                'Aktivieren des Senden PINNs (W/R)
  Waitms 30                                 'Reaktionzeit Abwarten
  Print Ticks ; Chr(9) ; Temperature1 ; Chr(9) ; Temperature2 ; Chr(9) ; Diff ; Chr(9) ; V ;
Chr(9) ; "EOD"                               'Daten Senden
  Waitms 30                                 'Reaktionzeit Abwarten
  Portb.2 = 0                                'Deaktivieren des Senden PINNs (W/R)

End If

Loop
-----
'
'                                             Adresseverarbeitung
'
-----

Serialinterrupt:

  Disable Urxc                               'Interupt wird Abgeschaltet

  Gosub Schrittabfrage                       'Abrufen des unterprogramms

  Return

-----
'
'                                             Unterprogramme
'
-----

Measuretemperature:

  'Messen
  lwreset Pind , Thpinnr
  lwwrite &HCC , 1 , Pind , Thpinnr         'alle herhören
  lwwrite &H44 , 1 , Pind , Thpinnr        'Messen

  Waitus 200

  'Auslesen
  lwreset Pind , Thpinnr
  lwwrite &HCC , 1 , Pind , Thpinnr
  lwwrite &HBE , 1 , Pind , Thpinnr        'RAM auslesen

  Dsread(1) = lread(9 , Pind , Thpinnr)

  Itemperature = Makeint(dsread(1) , Dsread(2))

  Temperature = Itemperature * 0.0625

Return

-----
'
'                                             Schrittabfrage
'
-----

Schrittabfrage:

Empfangen = Udr                               'UDR-Register wird als Variabel
abgespeichert

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

```

Return

```

-----
'
'                                     Bereitschaftabfrage
'
-----
  
```

Bereitschaft:

```

If Empfangen = 33 Then                                     'Abfrage des Bereitschaftssignal "!"
  Schritt = 1                                                 'Schrittvariabel wird auf 1 gesetzt
  Enable Urxc                                               'Interrups werden wieder eingeschaltet

Else
  Schritt = 0                                                 'Schritt auf null zurückgesetzt
  Empfangen = 0                                              'Aus UDR gelesener Wert auf Null gesetzt
  Enable Urxc                                               'Interrups werden wieder eingeschaltet
  
```

End If

Return

```

-----
'
'                                     Addressabfrage
'
-----
  
```

Addressabfrage:

```

If Adresse = Empfangen Then                               'Abfrage der Adresse
  Schritt = 2                                                 'Schrittvariabel wird auf 2 gesetzt
  Enable Urxc                                               'Interrups werden wieder eingeschaltet

  Else
    Empfangen = 0                                             'Aus UDR gelesener Wert auf Null gesetzt
    Schritt = 0                                              'Schritt auf null zurückgesetzt
    Enable Urxc                                             'Interrups werden wieder eingeschaltet
  
```

End If

Return

```

-----
'
'                                     Befehlsabfrage
'
-----
  
```

Befehlsabfrage:

```

'Systeminformationen
  If Empfangen = 105 Then                                   'Abfrage des Befehls Systeminformationen
  "i"
  Gosub Systeminformationen

'Daten senden
  Elseif Empfangen = 115 Then                               'Abfrage des Befehls Daten senden "s"
  Gosub Daten_senden

'Reset
  Elseif Empfangen = 114 Then
    Config Watchdog = 512
    Start Watchdog
  'Abfrage des Befehls Daten Zurücksetzen
  "r"
  Wait 1
  Enable Urxc                                               'IS warten um watchdog auszulösen
  
```



```

Else
Empfangen = 0
Schritt = 0
Enable Urxc

End If
  
```

Return

```

'***** Systeminformationen
*****
  
```

Systeminformationen:

```

Portb.2 = 1           'Aktivieren des Senden PINNs (W/R)
Waitms 30            'Reaktionzeit Abwarten
Print "#Modul Wetterstation"
Print "#Acosta Alejandro"
Print "#LTAM T3EE 2014/2015"
Print "#Adresse: " ; Chr(adresse)
Print "#"
Print "#Befehle: "
Print "#!" ; Chr(adresse) ; "s Daten Senden "
Print "#!" ; Chr(adresse) ; "i Info"
Print "#!" ; Chr(adresse) ; "r Reset"
Print "#"
Print "#Version: " ; Version()
Print "#"
Print "# ; "Ticks" ; Chr(9) ; "Sensor1" ; Chr(9) ; "Sensor2" ; Chr(9) ; "Temp
Differenz" ; Chr(9) ; "Windgeschwindigkeit" ; Chr(9) ; " EOD"
Print "#"
Waitms 30            'Reaktionzeit Abwarten
Portb.2 = 0          'Deaktivieren des Senden PINNs
Schritt = 0          'Schritt auf null zurückgesetzt
Enable Urxc         'Interrups werden wieder eingeschaltet

Return
  
```

```

'***** Daten senden
*****
  
```

Daten_senden:

```

Portb.2 = 1           'Aktivieren des Senden PINNs (W/R)
Waitms 30            'Reaktionzeit Abwarten
Print Ticks ; Chr(9) ; Temperature1 ; Chr(9) ; Temperature2 ; Chr(9) ; Diff ; Chr(9)
; V ; Chr(9) ; "EOD"
Waitms 30            'Reaktionzeit Abwarten
Portb.2 = 0          'Deaktivieren des Senden PINNs
Schritt = 0          'Schritt auf null zurückgesetzt
Enable Urxc         'Interrups werden wieder eingeschaltet

Return
  
```

Bascom Programm Online

Der Bascom Programm ist hier Elektronisch zu finden:

<https://onedrive.live.com/redir?resid=7EFD775F35CADBE1!106&authkey=!AMHxHRJvgqStXVQ&ithint=file%2cdocx>

Quellen

<http://staff.ltam.lu/feljc/home.html>

<http://www.weigu.lu/>

<http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

<http://avrhelp.mcselec.com/>

http://staff.ltam.lu/feljc/school/projets13/CO2_MMR/CO2_Tec_Doc/Temperaturmessung.pdf

http://staff.ltam.lu/feljc/electronics/bascom/BASCOM_Tutorial_2.pdf

http://www.produktinfo.conrad.com/datenblaetter/175000-199999/179999-da-01-en-GABELLICHTSCHRANKE_THT_SX_EE1041_5MM.pdf