

Analyse Heizkörperthermostat

Honeywell HR20

Version: 0.6

Datum: 19.03.2008

Historie:

Version	Datum	Bemerkung
0.1	19.12.07	Erstellung
0.2	29.01.08	erweitert um Serielle Schnittstelle
0.3	31.01.08	erweitert um LCD Konfiguration
0.4	1.02.08	LCD Konfiguration für Simulator im SVN
0.5	8.02.08	PWM-Signal = $1\text{MHz}/16 = 15.625\text{kHz}$
0.6	24.02.08	Fehler im Schaltplan Pin 21,52 und 64 an 3V3.

Disclaimer:

Alle Informationen in diesem Dokument wurden durch Analyse eines Heizkörper Thermostaten HR20 mit der Software-Version 2.04 gewonnen.

Der Autor übernimmt keinerlei Gewähr für die Aktualität, Korrektheit, Vollständigkeit oder Qualität der bereitgestellten Informationen.

Auch kann keine Gewähr für die Richtigkeit der hier gemachten Angaben gemacht werden. Haftung oder Gewährleistung sind grundsätzlich ausgeschlossen.

Index

1	Einführung	5
2	Ausgangssituation.....	6
3	Schaltplan	7
3.1	Motorsteuerung	9
3.2	Motorüberwachung	10
3.3	Benutzereingaben	10
3.4	Thermostat auf Heizung montiert.....	10
3.5	Temperaturmessung	11
3.6	LC Display.....	11
3.7	Steckerbelegung externer Anschluss.....	12
4	Messungen an Originalsteuerung	13
4.1	Ansteuerung des Motors	13
4.2	Inkrementalgeber am Einstellrad.....	14
4.3	Thermofühler.....	15
4.4	LC-Display.....	16
5	Serielle Schnittstelle.....	20
6	Auslesen über JTAG.....	22
7	Referenzen	24

1 Einführung

Das Heizkörperthermostat HR20 von Honeywell ist ein elektronisches Thermostat, welches Heizkörperventile ansteuert um die Raumtemperatur zu regeln. Der besondere Vorteil dabei ist, dass sich die Solltemperatur nicht nur einstellbar ist, sondern zu vorgegeben Zeiten per Schaltuhr geändert werden kann. So ist es möglich die Solltemperatur an Arbeitstagen von 7:00 bis 9:00 und von 17:00 bis 23:00 auf 20°C und in der verbleibenden Zeit auf 18°C zu stellen um Energie zu sparen.

Der Nachteil ist, dass die Einstellung der Schaltzeiten auf zwei Intervalle pro Wochentag beschränkt ist und ein Umprogrammieren sehr aufwändig ist. Insbesondere, wenn man in einer Wohnung mehrere Heizkörper mit den Thermostaten ausstattet.

Ist zum Beispiel am folgenden Tag die betriebliche Weihnachtsfeier und man kommt erst gegen 22:00 Heim, oder ist dort ein Feiertag und man ist die ganze Zeit zu Hause, muss man an jeden einzelnen Thermostat um den Sollwert von Hand vorgeben, bzw. die Schaltzeit umprogrammieren.

Wünschenswert wäre es, wenn eine zentrale Steuerung die Thermostate programmieren könnte. Auch wäre es von Vorteil, wenn die Ist-Temperatur an anderer Stelle im Raum gemessen würde um den Messfehler in der Nähe der Heizung auszugleichen.

Ziel ist es das Thermostat um eine solche Kommunikationsschnittstelle zu erweitern. Die Kommunikation sollte hierbei frei wählbar sein, sie kann kabelgebunden (RS232, SPI, CAN) oder über Funk erfolgen.

Technische Daten des HR20:

- Regelverhalten: PI-Algorithmus
- Ansprechempfindlichkeit: 0,1°C
- Regelbereich: 8-28°C bzw. ON/OFF
- Fenster-Offen-Erkennung:
Bei Temperatur-Änderung von -1,5°C/3Min. schließt das Ventil und öffnet wieder bei Temperaturanstieg oder nach 45 Minuten
- Ventilsfreispülung gegen Kalkablagerungen 1x wöchentlich

2 Ausgangssituation

Das Herzstück des Honeywell HR20 ist ein ATmega169 Controller der Firma Atmel. Dieser misst Uhrzeit und Temperatur und steuert über einen Motor das Ventil des Heizkörpers an. Über einen von Außen zugänglichen Stecker besteht die Möglichkeit den Controller neu zu programmieren.

Zusätzlich liegen an dem Stecker die Batteriespannung, sowie die seriellen Kommunikationssignale RxD und TxD des Controllers an. Das werkseitig implementierte Protokoll ist unbekannt.

Die angestrebte Vorgehensweise ist die folgende:

1. Erstellung des Schaltplanes
2. Messung der diversen Signale der Original Software
3. Programmieren der gleichen Funktionalität
4. Erweitern des Programms um zusätzliche Funktionen
 - a. Steuerung über externe Schnittstelle
 - b. Programmierung über externe Schnittstelle
 - c. Speichern der Messwerte (Temperaturlogger)
 - d. Erweiterung des Regelbereichs
 - e. Freie Wählbarkeit des Zeitpunktes der Ventilsfreispülung
5. Erweitern des Thermostaten um zusätzliche Schnittstellen
 - a. RS232 Schnittstelle
 - b. CAN Schnittstelle
 - c. RFM12 Module (ISM Band 433MHz)

Durch die Möglichkeit den Controller neu zu Programmieren, ohne das Thermostat zu öffnen, können diese einfach mit den neuen Funktionen nachgerüstet werden.

3 Schaltplan

Um den Schaltplan zu erhalten wurde das Thermostat zerlegt und die Platine von beiden Seiten fotografiert. Abbildung 1 zeigt Vorder- und Rückseite der Platine:

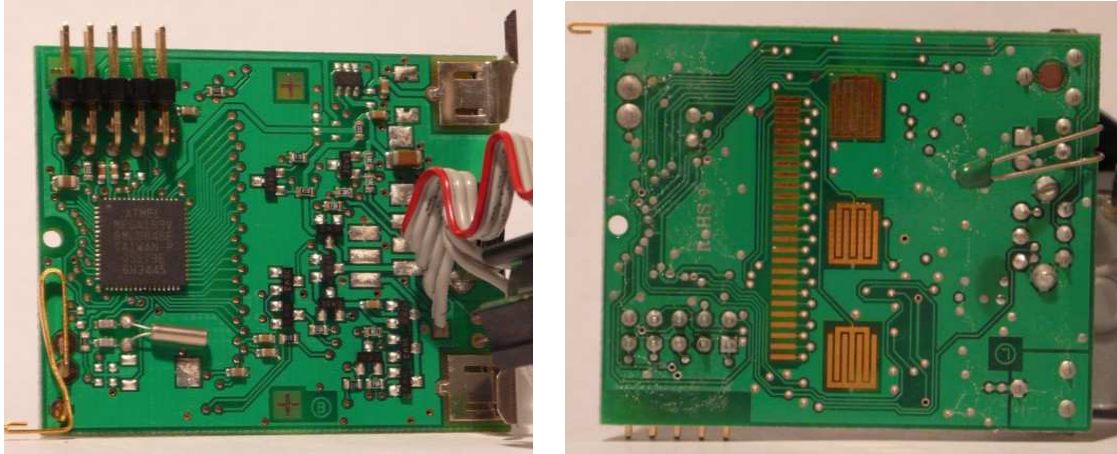


Abbildung 1: Vorder- und Rückseite der Platine

Zusätzlich besteht der Thermostat aus einem Motor, welche über ein Getriebe die Spindel antreibt, die das Ventil des Heizkörpers betätigt. Um die Funktion des Motors zu überwachen befinden sich an der ersten Getriebestufe zwei kleine silberne Punkte. Eine Reflexlichtschranke erkennt diese Punkte, wenn sie sich einmal pro Umdrehung an Ihr vorbei bewegen.

Ein LC-Display welches ohne zusätzlichen Displaycontroller direkt an den ATmega169 angeschlossen ist zeigt die Sollwerte und Schaltzeiten an und dient ferner der Programmierung, sowie Einstellung der Solltemperatur und Uhrzeit.

Die Benutzereingaben erfolgen über drei Tasten, AUTO/MANU, °C und PROG, sowie über ein großes Stellrad vor Kopf des Gerätes, welches einen Impulsgeber antreibt.

Die einzelnen Bauteile auf der Platine wurden benannt und anhand der Bilder die Leitungsführung nachvollzogen. Abbildung 2 zeigt die Lage der Bauteile auf der Platine.

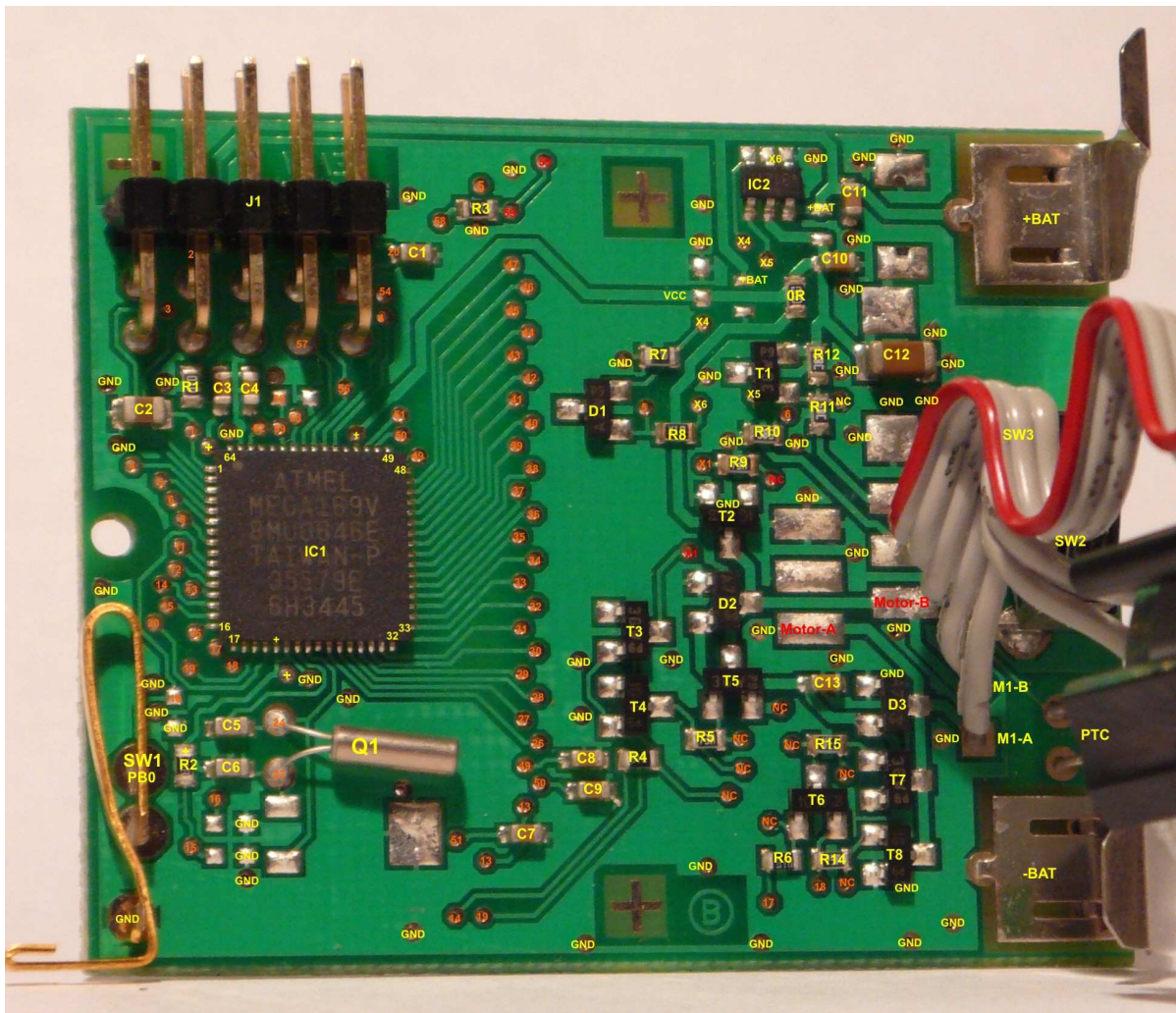


Abbildung 2: Lage der Bauteile auf der Platine

An den Stellen, an denen die Leiterbahnen durch andere Bauteile verdeckt sind wurden die Verbindungen messtechnisch ermittelt.

Die Bauteile konnten größtenteils anhand des aufgedruckten SMD Codes ermittelt werden. Die Werte der Kondensatoren wurden nicht ermittelt, für die Funktionsweise der Schaltung sind sie auch nicht von Bedeutung.

Bezeichnung des IC2 mit dem SMD Code „312h“ konnte nicht ermittelt werden, anhand der Beschaltung scheint es aber die Spannung zu regeln, sowie die den Sender der Reflexlichtschranke mit Spannung zu versorgen.

Alle andern Bauteile sind handelsübliche Bauelemente.

Durch das Übertragen der Verbindungen und entsprechende Anordnung der Bauteile ergab sich der Schaltplan des Thermostaten, wie er in Abbildung 3 zu erkennen ist.

3. Schaltplan

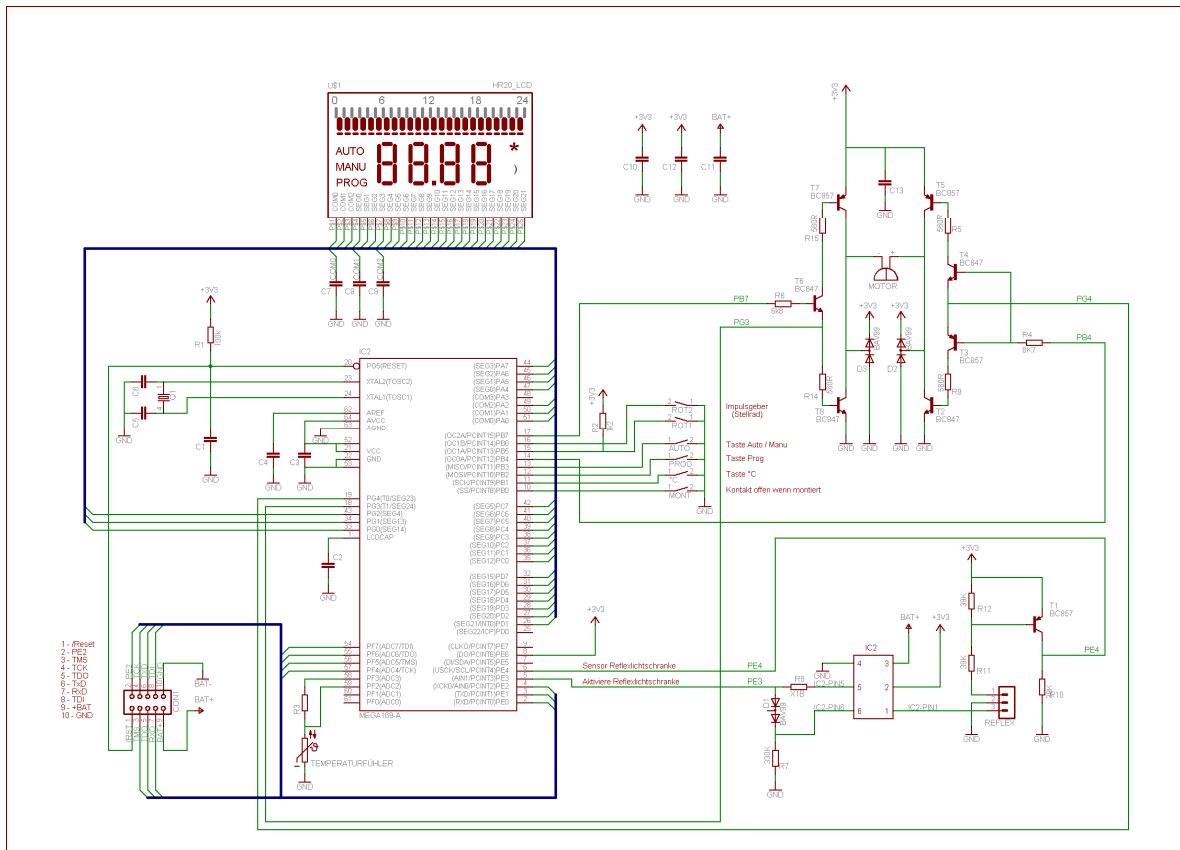


Abbildung 3: Schaltplan des HR20

Die Folgenden Funktionen übernehmen die einzelnen Aus- bzw. Eingänge des Mikrocontrollers:

3.1 Motorsteuerung

Der Motor wird über eine H-Brücke angesteuert, so ist es möglich die Laufrichtung umzuschalten. Die Brücke wird angesteuert über die Ausgänge PB4, PB7, PG3 und PG4. Tabelle 1 zeigt die für die Drehrichtungen des Motors die Zustände der Ausgänge:

Richtung	PB4	PB7	PG3	PG4
Stop	LOW	LOW	LOW	LOW
Ventil zu	LOW	HIGH	LOW	HIGH
Ventil auf	HIGH	LOW	HIGH	LOW

Tabelle 1: Motorsteuerung

3.2 Motorüberwachung

Um zu Überwachen, ob der Motor sich dreht, ist eine Reflexlichtschranke angeschlossen. Sie wird mit einem HIGH an dem Ausgang PE3 aktiviert. Der Status wird an den Eingang PE4 geleitet.

Anschluss	Richtung	Anschluss A
PE3	Out	Aktiviere Reflexlichtschranke
PE4	In	Signal Reflexlichtschranke

Tabelle 2: Pinbelegung Reflexlichtschranke

3.3 Benutzereingaben

Benutzereingaben erfolgen über drei Tasten, sowie ein großes Drehrad, welches einen Inkrementalgeber steuert, sie sind alle gegen GND und an den Controller wie folgt angeschlossen:

Anschluss	Richtung	Funktion
PB1	In	Taste °C
PB2	In	Taste PROG
PB3	In	Taste AUTO / MANU
PB5	In	Inkrementalgeber Anschluss A
PB6	In	Inkrementalgeber Anschluss B

Tabelle 3: Pinbelegung Eingabetasten und Implusegeber

3.4 Thermostat auf Heizung montiert

Ein Kontakt im Inneren des HR20 erkennt, ob der Thermostat an dem Heizungsventil montiert ist. Ist das Thermostat nicht montiert, schließt der Kontakt die Verbindung zu GND.

Anschluss	Richtung	Funktion
PB0	In	Kontakt Thermostat nicht montiert

Tabelle 4: Pinbelegung Kontakt Thermostat montiert

3.5 Temperaturmessung

Die Temperaturmessung erfolgt über einen Messwiderstand (NTC). Um den Stromverbrauch zu minimieren ist der Spannungsteiler aus R3 und dem NTC nicht an die Versorgungsspannung, sondern an einen Ausgang des Controllers angeschlossen.

Anschluss	Richtung	Funktion
PF3	Out	Spannungsteiler aktivieren
PF2	In	Messwert Temperatur Analog

Tabelle 5: Pinbelegung Temperaturfühler

3.6 LC Display

Der ATmega169 verfügt über einen Internen LCD Controller. Das LCD ist an die folgenden Ausgänge des Prozessors angeschlossen:

Anschluss	Richtung	Funktion
PA0-2	Out	Com0, Com1 und Com2
PA4-7	Out	Seg0 - Seg3
PG2	Out	Seg4
PC7 - PC0	Out	Seg5 - Seg12
PG1 - PG0	Out	Seg13 - Seg14
PD7 - PD1	Out	Seg15 - Seg21

Tabelle 6: Pinbelegung LC-Display

Das Display verfügt über 4 Sieben Segment Ziffern, 24 Quadraten nebeneinander, welche die Uhrzeit von 0:00-24:00 symbolisieren und 24 Strichen, jeweils unter den Quadraten. Zusätzlich befinden sich drei Symbole auf der rechten Seite: Sonne, Mond, Stern und drei Worte auf der linken Seite: ATO, MANU du PROG.

3.7 Steckerbelegung externer Anschluss

An dem externen Steckerkontakt liegen JTAG, RxD, TxD sowie die Batteriespannung an.

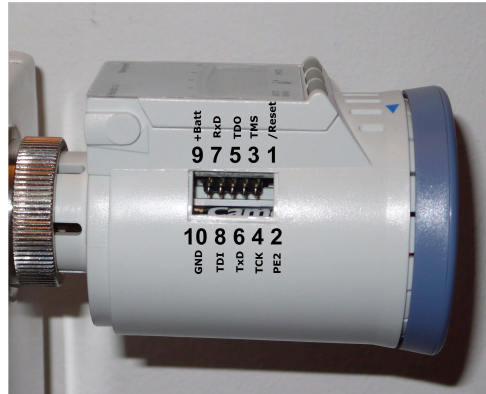


Abbildung 4: Pin-Nummerierung externer Anschluss

Gemäß der Kennzeichnung des Pin 1 auf der Platine und der Standardnummerierung für solche Stecker wurde in diesem Dokument die Nummerierung gewählt. Bei zugeschraubtem Thermostat ist Pin 1 der Pin, welcher am nächsten zu der Taste AUTO/MANU ist. Der Pin darunter ist Pin2. Die ungeraden Pins sind dem Display, die geraden der Platine zugewandt. Abbildung 4 zeigt die Lage der einzelnen Pins.

Pin	Pin Atmel 169	Funktion
1	/Reset	Reset
2	PE2	
3	TMS	JTAG
4	TCK	JTAG
5	TDO	JTAG
6	TxD	Serielle Ausgabe
7	RxD	Serielle Eingabe
8	TDI	JTAG
9	+ Batt	Batterie
10	GND	

Tabelle 7: Pinbelegung externer Anschluss

4 Messungen an Originalsteuerung

Um die Funktionalitäten der Originalsteuerung, insbesondere der Motorsteuerung, des Impulsgebers und der LCD Ansteuerung nachzuvollziehen wurden die Ausgangssignale des Controllers mit einem Logikanalysator und einem Oszilloskop gemessen.

4.1 Ansteuerung des Motors

Wie schon in Kapitel 3.1 vermutet werden mit den vier Ausgängen des Controllers gemäß der Tabelle die Drehrichtung des Motors vorgegeben.

Jedoch wird mit dem Pin PB4 (OC0A) ein PWM Signal mit einer Frequenz von 15,625 kHz ($1 \text{ MHz} / 64$) erzeugt, mit dem die dem Motor zugeführte Leistung eingestellt wird. Abbildung 5 zeigt die PWM Signale für das Schließen des Ventils, Abbildung 6 für das Öffnen.

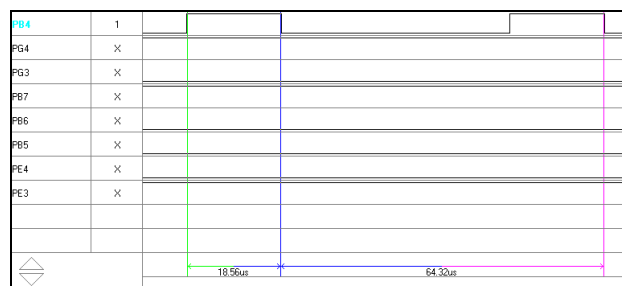


Abbildung 5: Signale Ventil schließen

Bei Öffnen des Ventils, muss PB4 HIGH sein es wird aber mit einem Puls/Pause Verhältnis von ca. $44\mu\text{s}/20\mu\text{s}$ angesteuert. Analog verhält es sich beim Schließen des Ventils, dort muss PB4 LOW sein es wird aber ein Signal mit einem Puls/Pause Verhältnis von ca. $46\mu\text{s}/18\mu\text{s}$ erzeugt.

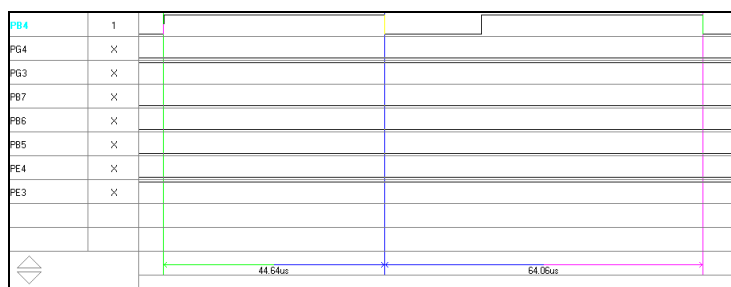


Abbildung 6: Signale Ventil öffnen

4.2 Inkrementalgeber am Einstellrad

Das große blaue Einstellrad vor Kopf des HR20 ist über einen Inkrementalgeber mit dem ATmega169 verbunden. Der Inkrementalgeber hat zwei Ausgänge, die mit den Eingängen PB5 und PB6 verbunden sind.



Abbildung 7: Signale Impulsgeber Drehrichtung abwärts

Wird an dem Rad gedreht wechseln bei jedem Einrasten des Drehrades beide Signale jeweils von LOW/LOW nach HIGH/HIGH und bei der nächsten Raste wieder zurück auf LOW/LOW. Die beiden Ausgänge wechseln ihren Zustand in der Abhängig der Drehrichtung zeitlich unterschiedlich zueinander. Das genauer Verhalten ist in Abbildung 7 und Abbildung 8, sowie in der Tabelle 8 zu sehen.

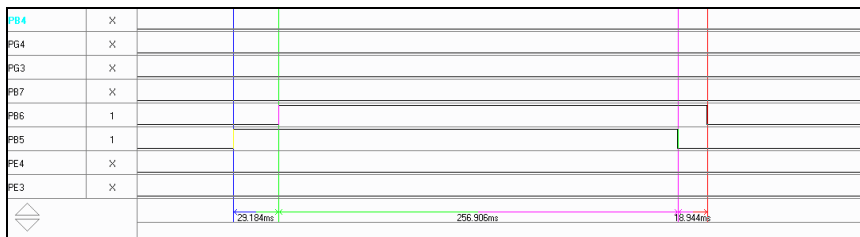


Abbildung 8: Signale Impulsgeber Drehrichtung aufwärts

Bei den dargestellten Messungen wurde das Rad mit maximaler Geschwindigkeit (ca. 1ms pro Rastung) gedreht. Dabei ist die Differenz zwischen der Wechsel der Signale 500µs. Bei moderaten Drehgeschwindigkeiten ist die Differenz ca. 20ms.

Richtung	Alter Zustand		Wechselreihenfolge
	PB 5	PB 6	
Auf	LOW	LOW	PB5 HIGH, PB6 HIGH
Auf	HIGH	HIGH	PB5 LOW, PB6 LOW
Ab	LOW	LOW	PB6 HIGH, PB5 HIGH
Ab	HIGH	HIGH	PB6 LOW, PB5 LOW

Tabelle 8: Signale am Inkrementalgeber

4.3 Thermofühler

Wie schon in Kapitel vermutet bestätigte die Messung, dass der Spannungsteiler aus R3 und dem NTC nicht ständig betrieben wird. Eine Messung findet alle 500ms statt. Während einer Messung wird der Spannungsteiler dreimal kurz hintereinander für ca. 100-200µs eingeschaltet, der Grund dafür ist unklar. Abbildung zeigt die drei Schaltvorgänge, welche sich periodisch alle 500ms wiederholen.

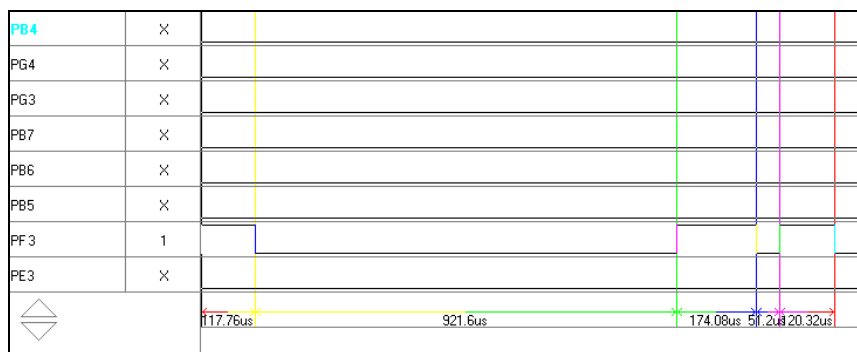


Abbildung 9: Aktivierung des Thermofühlers

Um das Verhalten der Spannung in Abhängigkeit der Temperatur zu messen, wurde die Funktion der Originalsoftware genutzt, welche die Ist-Temperatur kontinuierlich auf dem Display anzeigt. Anstelle des NTC wurde ein Trimpotentiometer angebracht und damit Temperaturen in Schritten von 0,5°C eingestellt. Danach wurde das Potentiometer an einen anderen HR20 angeschlossen und dort der ADC Wert ermittelt und abgelesen. Tabelle 9 zeigt die so ermittelten Werte.

°C	ADC	°C	ADC	°C	ADC	°C	ADC	°C	ADC
5,0	675	11,0	604	17,0	517	23,0	424	29,0	352
5,5	670	11,5	595	17,5	510	23,5	418	29,5	347
6,0	664	12,0	587	18,0	504	24,0	412	30,0	340
6,5	658	12,5	584	18,5	496	24,5	404	30,5	335
7,0	652	13,0	575	19,0	488	25,0	397	31,0	332
7,5	645	13,5	572	19,5	479	25,5	389	31,5	325
8,0	639	14,0	564	20,0	472	26,0	384	32,0	319
8,5	633	14,5	556	20,5	462	26,5	379	33,0	312
9,0	628	15,0	549	21,0	455	27,0	376	34,0	304
9,5	620	15,5	541	21,5	448	27,5	368		
10,0	614	16,0	536	22,0	440	28,0	363		
10,5	608	16,5	526	22,5	431	28,5	359		

Tabelle 9: ADC-Werte in Abhängigkeit der Ist-Temperatur

Abbildung zeigt die grafische Auswertung der Messwerte, sowie die sieben Punkte, welche linear verbunden die Kennlinie annähern.

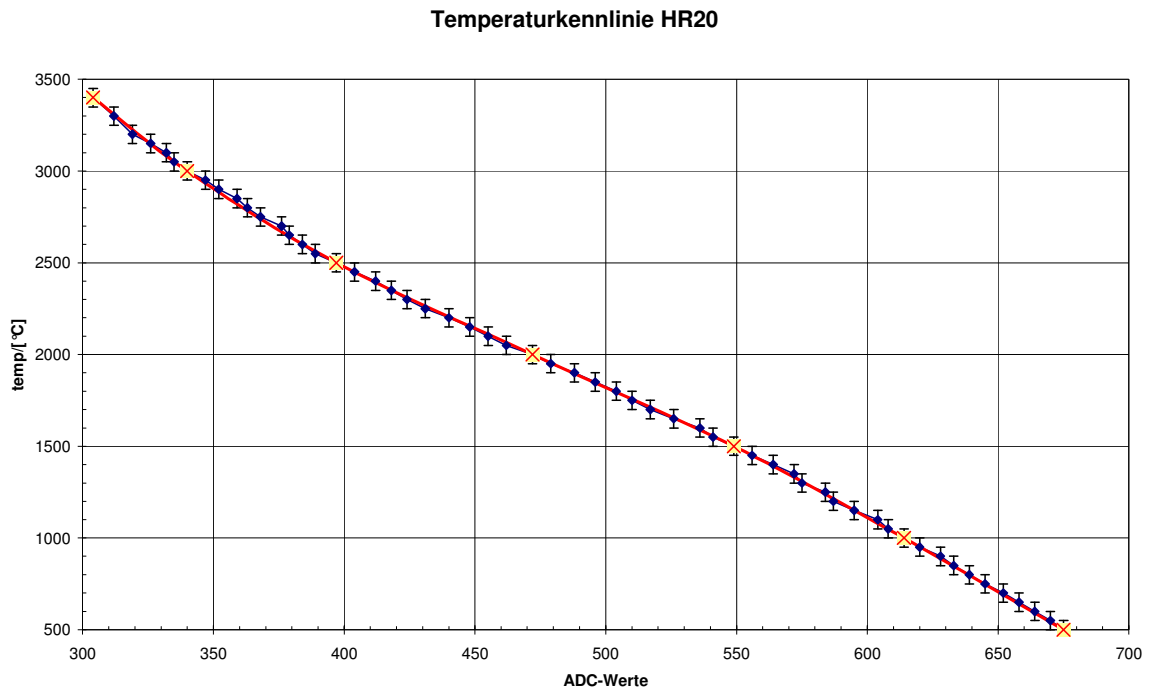


Abbildung 10: Kennlinie des Thermofühlers

Mit diesen Informationen ist es leicht möglich die die Kennlinie mit sieben Wertepaaren abzubilden. Die verwendeten Werte sind in Tabelle 9 hervorgehoben.

4.4 LC-Display

Aus dem Schaltplan ist ersichtlich, dass das LC-Display an den Leitungen *Seg0-Seg21*, sowie *Com0-Com2* angeschlossen ist.

Um möglichst viele Informationen über Programmierung des internen LCD Controllers zu erhalten wurde der Spannungsverlauf an *Com0* und *Seg0* gemessen, Abbildung 11 zeigt diesen Spannungsverlauf.

Hieraus lassen sich folgende Aussagen treffen:

Das Display wird im *1/3 Duty* und *1/3 Bias* Modus betrieben, welches laut Datenblatt auch der bevorzugte Modus für LCDs mit drei gemeinsamen Anschlüssen ist.

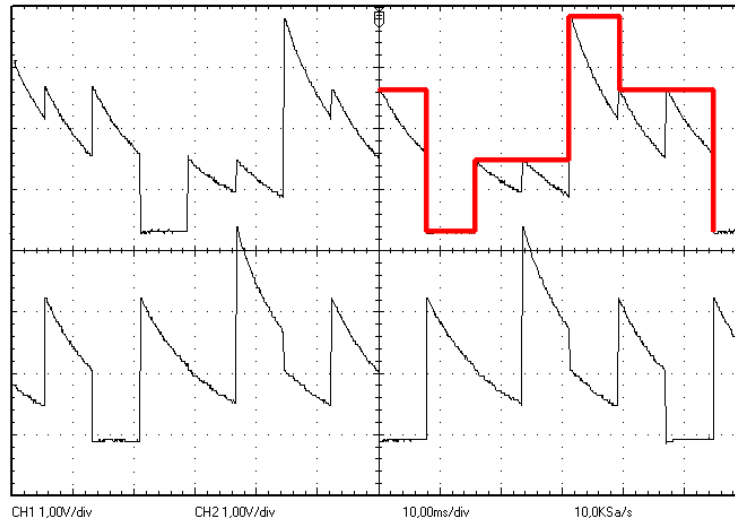


Abbildung 11: Spannungsverlauf an Com0 und Seg0

Um Strom zu sparen wird der *Low Power Waveform* Modus genutzt.

Das LCD wird im asynchronen Modus (LCDCS-Bit=1) mit der Frequenz des Quarzes TOSC1 = 32.768Hz als LCD Clock betrieben.

Nach folgender Formel können die K, D und N gewählt werden:

$$f_{frame} = \frac{f_{cls_{LCD}}}{(K \cdot N \cdot D)}$$

Für K=6 (1/3 Duty), N=16 und D=6 erhält man eine Frame Rate von 56 Hz, welches ein ausreichend hoher Wert ist.

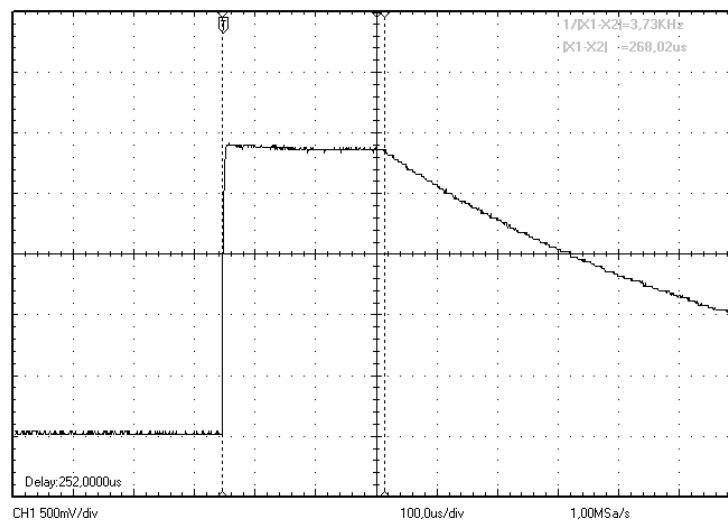


Abbildung 12: Spannungsverlauf an Seg0

Eine weitere Einstellung ist die *LCD Display Configuration*, welche angibt, wie lange die LCD Treiber des Controllers bei jeder Spannungsän-

derung eingeschaltet werden sollen. Diese Einstellung erfolgt über die Bits *LCDC0-2* im Register *LCDCCR*. Aus Abbildung 12 ist zu erkennen, dass hier 300µS eingestellt sind.

Um Das Display zu aktivieren müssen die folgenden Einstellungen in den entsprechenden Registern erfolgen.

LCD Control and Status Register A - LCDCRA:

- Bit 7 - LCDEN: LCD Enable = 1
- Bit 6 - LCDAB: LCD Low Power Waveform = 1
- Bit 4 - LCDIF: LCD Interrupt Flag = 1
- Bit 3 - LCDIE: LCD Interrupt Enable = 1
- Bit 0 - LCDBL: LCD Blanking = 0

LCD Control and Status Register B - LCDCRB:

- Bit 7 - LCDCS: LCD Clock Select = 1
- Bit 6 - LCD2B: LCD 1/2 Bias Select = 0
- Bit 5 - LCDMUX1: LCD Mux Select 1 = 1
- Bit 4 - LCDMUX0: LCD Mux Select 0 = 0
- Bit 2 - LCDPM2: LCD Port Mask 2 = 1
- Bit 1 - LCDPM1: LCD Port Mask 1 = 0
- Bit 0 - LCDPM0: LCD Port Mask 0 = 1

LCD Frame Rate Register - LCDFRR:

- Bit 6 - LCDPS2: LCD Prescaler Select 2 = 0
- Bit 5 - LCDPS1: LCD Prescaler Select 1 = 0
- Bit 4 - LCDPS0: LCD Prescaler Select 0 = 0
- Bit 2 - LCDCD2: LCD Clock Divide 2 = 1
- Bit 1 - LCDCD1: LCD Clock Divide 1 = 0
- Bit 0 - LCDCD0: LCD Clock Divide 0 = 1

LCD Contrast Control Register - LCDCCR:

- Bit 7 - LCDDC2: LCD Display Configuration 2 = 0
- Bit 6 - LCDDC1: LCD Display Configuration 1 = 0
- Bit 5 - LCDDC0: LCD Display Configuration 0 = 0
- Bit 3 - LCDCC3: LCD Contrast Control 3 = nach Bedarf
- Bit 2 - LCDCC2: LCD Contrast Control 2 = nach Bedarf
- Bit 1 - LCDCC1: LCD Contrast Control 1 = nach Bedarf
- Bit 0 - LCDCC0: LCD Contrast Control 0 = nach Bedarf

Um Informationen im LCD anzuzeigen, werden die zu aktivierenden Segmente in die entsprechenden LCDDR Register geschrieben. Der Anschlusskonfiguration nach sind die Register LCDDR13-LCDDR19 ohne Funktion.

Um die Zuordnung der einzelnen Symbole zu den Segmenten des LCD zu ermitteln wurde ein Programm erstellt, welches die einzelnen Segmente einzeln aktiviert, so konnte die in Abbildung 13 gezeigte Zuordnung erstellt werden.

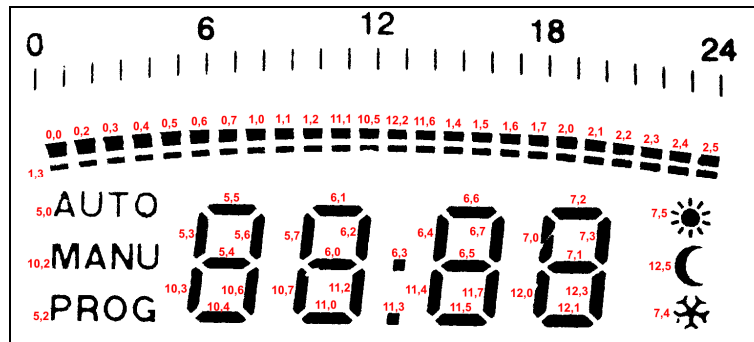


Abbildung 13: Zuordnung der Segmente nach Ports

Die Grafik zeigt, welche Segmente an dem entsprechenden Registern angeschlossen sind. Dabei steht die erste Ziffer für die Nummer des LCDDR Registers und die zweite Ziffer für das Bit innerhalb des Registers. 5,0 bedeutet also LCDDR5, Bit0 mit $LCDDR5 \mid = (1 \ll 0)$ kann man folglich das AUTO Symbol einschalten.

Diese Anordnung ist etwas unübersichtlich zudem werden nicht alle LCDDR Register genutzt, sondern nur LCDDR0-2, LCDDR5-7 und LCDDR10-12, Nummeriert man nun die Segmente innerhalb der Register in aufsteigender Reihenfolge (Seg0=0,0; Seg1=0,1; Seg8=1,0; Seg23=3,7; Seg24=5,0; Seg47=7,7; Seg47=10,0; Seg71=12,7) dann erhält man die in Abbildung 14 dargestellte Zuordnung.

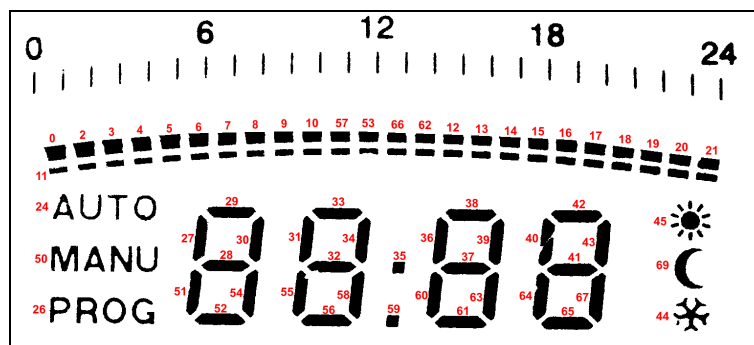


Abbildung 14: Zuordnung der Segmente nach Segmentnummern

Atmel bietet Programmieren ein kostenfreies Tool mit dem Namen „AVR LCD Visualizer“ [LCD] an. Die Daten dieses LCD wurden dort eingegeben

und so kann nun das LCD simuliert werden. Die Datei dazu befindet sich im [SVN]

5 Serielle Schnittstelle

Über den externen Anschluss lässt sich der Thermostat mit einem PC verbinden. Dazu wird eine Serielle Schnittstelle (COM) benötigt, sowie ein Spannungswandler, der die Signalpegel auf 3V anpasst. Die einfachste Möglichkeit beides an einen zeitgemäßen PC anzuschließen ist die Verwendung eines USB Handy-Kabels, für ein Handy welches nur über einen Seriellen Port verfügt (z.B. Siemens S25). Handelt es sich um ein 3V Kabel, dann können die Leitungen einfach verbunden werden, handelt es sich um ein 5V Kabel, dann sollte man einen Spannungsteiler in die TxD Leitung einbauen (z.B.: 9k1 und 5k6)

Am PC nutzt man das Terminal Programm seiner Wahl und stellt folgendes ein: 9600 BAUD, 8 Datenbit, 1 Stopbit, keine Parität. Als Zeilenende sendet der HR20 immer CR+LF, also 0x0D, 0x0A.

Jetzt solle nach dem Einlegen der Batterien in den Thermostaten folgendes angezeigt werden:

```
>HR20 SW Version 204 vom 08. Mar. 2006 12:00 HW Version 2  
>
```

Alle 4 Minuten und anscheinend auch immer wenn die Soll-Temperatur geändert wird, gibt der HR20 folgende Meldung aus:

```
>D: 28.1.2008 U: 22:27 V: 100 I: 2069 S: 2500  
>
```

Hierbei wird sind die ersten beiden Werte, wie unschwer zu erkennen Datum und Uhrzeit, V ist anscheinend die Ventilöffnung in %, I Die Ist-Temperatur und S die Soll Temperatur jeweils in 1/100°C.

Trennt man den Thermostaten von dem Stellventil und öffnet somit den Kontakt, der PB0 ansteuert sendet der HR20:

```
>G: 0b0
```

nach dem Abziehen und

```
>G: 0b1
```

nachdem er wieder montiert wurde.

Wenn man nun den Thermostat bedient, so sendet er bei jedem Tastendruck die Tastencodes der jeweiligen Tasten. Immer wenn eine Änderung des Tastenzu-

standes erkannt wird, sendet der Thermostat ein „K:“ gefolgt von dem entsprechenden Code. Sind mehrere Tasten gedrückt, werden die Codes addiert.

Die Codierung der einzelnen Tasten sind in Tabelle 10 dargestellt.

Taste	Code
Auto/Manu	0b100
Prog	0b10
°C	0b1
keine Taste	0b0

Tabelle 10: Tastencodes

Werden zum Beispiel nacheinander alle Tasten von links nach rechts gedrückt und gehalten und dann wieder von links nach rechts losgelassen sendet der HR20 folgendes:

```
>K: 0b100
>K: 0b101
>K: 0b111
>K: 0b11
>K: 0b10
>K: 0b0
>
```

6 Funktionsweise des HR20

6.1 Regelfunktion

Bei der Regelfunktion handelt es sich aller Wahrscheinlichkeit um einen PID Regler. Aus dem Protokoll der seriell ausgegebenen Daten ist zu erkennen, wie Stellgröße, Soll- und Ist-Temperatur über die Zeit zusammenhängen.

Hier ein Beispiel für das morgendliche Aufheizen eines Raumes (Uhrzeiten von Hand formatiert):

```
>D: 29.1.2008 U: 8:54 V: 0 I: 1867 S: 1600
>D: 29.1.2008 U: 8:58 V: 0 I: 1867 S: 1600
>D: 29.1.2008 U: 9:00 V: 100 I: 1867 S: 2000
>D: 29.1.2008 U: 9:04 V: 100 I: 1867 S: 2000
>D: 29.1.2008 U: 9:08 V: 100 I: 1867 S: 2000
>D: 29.1.2008 U: 9:12 V: 100 I: 1873 S: 2000
>D: 29.1.2008 U: 9:16 V: 100 I: 1885 S: 2000
>D: 29.1.2008 U: 9:20 V: 92 I: 1910 S: 2000
```

```
>D: 29.1.2008 U: 9:24 V: 79 I: 1922 S: 2000
>D: 29.1.2008 U: 9:28 V: 58 I: 1947 S: 2000
>D: 29.1.2008 U: 9:32 V: 51 I: 1966 S: 2000
>D: 29.1.2008 U: 9:36 V: 45 I: 1985 S: 2000
>
```

Wenn keine Verränderung der Soll-Temperatur stattfindet, dann wird die neue Stellgröße alle 4 Minuten lang neu berechnet.

6.2 Fenster Offen Erkennung

Wenn ein Fenster geöffnet wird und dadurch die Temperatur am Thermostat sinkt, dann schließt dieser das Ventil um Energie zu sparen. Im Display wird OPEN angezeigt, über die serielle Schnittstelle werden in diesem Modus keine Meldungen mehr ausgegeben. Auch die 4-minütigen Meldungen entfallen.

6.3 Ventilschutz

Wird innerhalb einer Woche das Ventil nicht einmal vollständig geöffnet, so öffnet der HR20 Montag kurzzeitig das Ventil um der Verkalkung vorzubeugen. Im Display wird CYCL angezeigt.

6.4 Frostschutz

Sinkt die Temperatur unter 4°C, öffnet der HR20 das Ventil, bis eine Temperatur von 6°C erreicht wird. So wird verhindert, dass der Heizkörper einfriert.

7 Auslesen über JTAG

Aus dem Schaltplan ist zu erkennen, dass alle Leitungen des JTAG von außen zugänglich sind. Durch Anschluss eines JTAG-Programmers konnte so die Verbindung zum ATmega169 hergestellt werden.

Zuerst wurde die Device Signature ausgelesen:

```
Reading signature .. 0x1E 0x94 0x05 .. OK
```

Dieser ist der für einen ATmega169 erwartete Wert, wir haben es also nicht mit einem umgelabelten ARM zu tun.

Nun konnten die LOCK- und FUSE- Bits ausgelesen werden.

```
Reading lock bits.. 0xFC .. OK
```

```
Reading fuse bits.. 0xFD, 0x9162 .. OK
```

Erwartungsgemäß sind die LOCK-Bits LB0 und LB1 gesetzt, sodass ein Auslesen des Programms nicht möglich ist:

- LB Mode: 3
Further programming and verification of the Flash and EEPROM is disabled in Parallel and Serial Programming mode.
- BLB0 Mode: 1
No restrictions for SPM or LPM accessing the Application section.
- BLB1 Mode: 1
No restrictions for SPM or LPM accessing the Boot Loader section.

Aus den Fuses ergeben sich die folgenden Einstellungen:

- Brown-out Detection enabled
- Brown-out Detector trigger level at VCC = 1.8V
- Reset enabled
- On-Chip debug disabled
- JTAG Interface enabled
- Serial Program and Data Downloading (SPI) enabled
- Watchdog disabled
- preserve EEPROM memory through the Chip Erase cycle
- BOOT Reset Vector disabled
- Divide Clock by 8 internally
- No Clock output on PortE7
- Internal RC Oscillator Startup Time 6 CK, + 65ms

Aus Stromspargründen läuft der ATmega169 anscheinend mit einer Taktfrequenz von 1MHz, theoretisch ist es möglich durch das Programm den Takt zur Laufzeit über die Clock Prescaler Select Bits (CLKPS3-0) zu ändern, es wurden jedoch bei der Analyse keine Hinweise gefunden, dass davon Gebrauch gemacht wird.

8 Referenzen

[AT169] Datasheet ATmega169, Version 2514P-AVR-07/06
www.atmel.com/dyn/resources/prod_documents/doc2514.pdf

[LCD] AVR LCD Visualizer
http://atmel.com/dyn/products/tools.asp?family_id=607
http://atmel.com/dyn/resources/prod_documents/AvrLcd.msi

[SVN] HR20 SVN
Repository Location: <https://OpenSVN.csie.org/hr20>
Browse: <https://OpenSVN.csie.org/viewcvs.cgi/?root=hr20>
TRAC: <https://opensvn.csie.org/traccgi/hr20>