

Syntax: Befehl_Parameter, Delimiter sind Leerstelle,Komma,Doppelpunkt

Befehl	Parameter	Erläuterung	Beispiel
t	?	time : gibt Uhrzeit aus im Format <i>hh:mm:ss</i> .	<i>t ?</i> 19:57:12 OK.
	hh:mm:ss	liest Uhrzeit als 3-Token-Kette ein (<i>Integer</i>) und setzt neue Uhrzeit.	<i>t 21:45:16</i> Time set to 21:45:16. OK.
d	?	date : gibt das mit führendem Wochentag als Drei-Buchstaben-Wort und das Datum im Format <i>tt.mm.jjjj</i> nach einem Komma aus.	<i>d ?</i> Sun, 01.01.2017 OK.
	wd,dd.mm.yyyy	liest den Wochentag als Zahl 1..7 (Mon = 1) als 4-Token-Kette ein und setzt Wochentag und neues Datum. <i>Hinweis: Der Wochentag wird nicht mit einem Kalenderalgorithmus überprüft.</i>	<i>d 5,12.5.2017</i> Date set to Fri, 12.05.2017. OK.
ts	?	timeshift : gibt Zeitzone -9..+9 aus.	<i>ts ?</i> +1 OK.
	[+]-x	liest Zeitzone als vorzeichenbehaftete Zeitzone -9..+9 ein und setzt neue Zeitzone. <i>Hinweise: Das Pluszeichen kann weggelassen werden. Die Zeitverschiebung in Stunden wird unter Beachtung eines eventuellen Datumswechsels berechnet und in der RTC abgelegt, das heißt, bei Vorverlegung oder Verzögerung zur Mitteleuropäischen Zeit (MEZ) wechselt auch das Datum zur angezeigten Mitternacht.</i>	<i>ts +5</i> Time shift set to +5 OK.
dcf	?	DCF77 : gibt den Einschaltzustand des DCF-Empfängers als Ausgabe <i>on off auto, hh:mm</i> aus	<i>dcf ?</i> DCF receiver off. OK.
	on oder 1	schaltet den DCF dauerhaft ein. <i>Hinweis: Der Zustand wird im EEPROM gespeichert bleibt auch nach einem Reset erhalten.</i>	<i>dcf 1</i> DCF receiver on. OK.
	off oder 0	schaltet den DCF dauerhaft aus. <i>Hinweis: Der Zustand wird im EEPROM gespeichert bleibt auch nach einem Reset erhalten.</i>	<i>dcf off</i> DCF receiver off. OK.
	auto[,hh:mm,wdmask]	schaltet die Automatik mit der im RTC-EEPROM hinterlegten Uhrzeit (Stunde, Minute und Wochentagsmaske) ein. <i>Hinweis: Die LED-Matrix bzw. LED-Kette wird während des Auto-Empfangs ausgeschaltet, bis ein gültiger Datensatz empfangen oder der Timeout (10 min) erreicht wurde.</i>	<i>dcf auto</i> DCF receiver in auto mode, start at 05:00, weekday mask: 127, DCF receiver down. OK.

	<p>Wenn eine Uhrzeit angegeben wird, liest der Befehl eine Uhrzeit, Wochentagsmaske als 3-Token-Kette ein und setzt die Startzeit <i>Stunde(24h):Minute, Wochentagsmaske</i> für DCF-Aktivierung. Der Automatikmodus und die Änderung der Zeit wird mit der Konfiguration im EEPROM gespeichert.</p> <p><i>Hinweis: Die Wochentagsmaske muss errechnet werden als Summe aus gesetzten Bits nach folgendem Schema:</i></p> <p><i>Tabelle Wochentagsmaske:</i></p> <table border="1"> <thead> <tr> <th>Bit 7</th> <th>Bit 6</th> <th>Bit 5</th> <th>Bit 4</th> <th>Bit 3</th> <th>Bit 2</th> <th>Bit 1</th> <th>Bit 0</th> </tr> </thead> <tbody> <tr> <td>128</td> <td>64</td> <td>32</td> <td>16</td> <td>8</td> <td>4</td> <td>2</td> <td>1</td> </tr> <tr> <td>-</td> <td>So</td> <td>Sa</td> <td>Fr</td> <td>Do</td> <td>Mi</td> <td>Di</td> <td>Mo</td> </tr> </tbody> </table>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	128	64	32	16	8	4	2	1	-	So	Sa	Fr	Do	Mi	Di	Mo	<p><i>dcf auto 12:15,63</i> DCF receiver in auto mode, start at 12:15, weekday mask: 63, DCF receiver down. OK.</p>
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0																			
128	64	32	16	8	4	2	1																			
-	So	Sa	Fr	Do	Mi	Di	Mo																			
	<p>debug on oder debug 1 debug off oder debug 0</p> <p>Schaltet die Debug-Funktion des DCF-Empfangsdecoder ein. Es werden 0 und 1 für erkannte Bits hintereinandergeschrieben, ein „-“ signalisiert eine zu kurze oder zu lange Bitlänge. Wird ein komplettes Telegramm als korrekt empfangen, wird „-sync“ ausgegeben, die Daten in den RTC übernommen und je nach Einstellung der DCF-Empfänger ausgeschaltet.</p> <p><i>Hinweis: Die Debug-Aktivierung wird im EEPROM gespeichert und ist auch nach einem Reset der Uhr aktiv.</i></p>	<p><i>dcf debug on</i> DCF receiver in auto mode, start at 05:00, weekday mask: 127, DCF receiver down. DCF debug is on. Changes saved in Configuration 0 OK. 00100001011101000 0100101100111011001101010...0-sync</p>																								
	<p>now</p> <p>Startet einen Empfangsversuch sofort. <i>Hinweis: Die LED-Matrix bzw. LED-Kette wird während des Auto-Empfangs ausgeschaltet, bis ein gültiger Datensatz empfangen oder der Timeout (10 min) erreicht wurde.</i></p>	<p><i>dcf now</i> DCF receiver in auto mode, start at 05:00, weekday mask: 127, DCF receiver down. DCF debug is on. Changes saved in Configuration 0 OK.</p>																								
gbr	<p>?</p> <p>general brightness: gibt die aktuelle Grundhelligkeit aus im Format <i>auto fix,Helligkeitswert</i>.</p>	<p><i>gbr ?</i> <i>auto.</i> OK.</p>																								
	<p>auto,l,h oder fix,x</p> <p>es werden Modus und Helligkeit als 2 oder 3-Token-Kette eingelesen. Der Modus <i>auto</i> ist bei der Classic Wanduhr bei <u>allen</u> Helligkeitsstufen wirksam. In der 5. Tastenstufe „Auto“ sind die mit dem Befehl <i>gbr auto,l,h</i> einstellbare Untergrenze <i>l</i> und die Obergrenze <i>h</i> als Begrenzung wirksam. Befindet sich der vom LDR gemessene Wert außerhalb dieser Grenzen, so wird die jeweils über- oder unterschrittene Grenze als Wert für die Helligkeit der Uhr genommen. In den Helligkeitsstufen 1 bis 4 wird der vom LDR gemessene Wert begrenzt mit typischen Helligkeitsbereichen für „Maximale Leuchtkraft“, „Normale Leuchtkraft“, „Mittlere Leuchtkraft“ und „Geringe Leuchtkraft“. Die Helligkeit der Uhr bewegt sich nur innerhalb dieser fest</p>	<p><i>gbr fix,200</i> Brightness set to fix, value 200 (0..255). OK. Changes saved in Configuration 0</p> <p><i>gbr auto,1,159</i> Brightness set to auto, 1 to 159. (0 <= b <=255). OK.</p>																								

eingestellten Grenzen. Beachten Sie, dass die mit dem Befehl *gbr auto,l,h* einstellbare Untergrenze *l* und die Obergrenze *h* als Begrenzung höhere Priorität haben und die festen Grenzen überschreiben.

Changes saved in Configuration 0

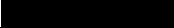









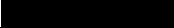









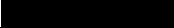









Wanduhr Classic Helligkeit (Taster)	Wert Untergrenze				LDR-Wert	Wert Obergrenze			
	1	33	65	97		159	191	223	255
1									
2									
3									
4									
Auto	frei*					frei* (Untergrenze < x ≤ 255)			






*) Hinweis: Die Werte der Auto-Einstellung haben höhere Priorität als die festen Werte der Helligkeitsstufen, das heißt, eine Auto-Untergrenze höher als die Untergrenze der aktiven Helligkeitsstufe wird nicht unterschritten.

Der Modus *auto* ist **beim Wecker (bei der Tischuhr)** bei den drei Helligkeitsstufen außer dem Nachtmodus wirksam. In den Helligkeitsstufen 1 bis 3 wird der vom LDR gemessene Wert „feinjustiert“ und es „stehen Ihnen dafür „drei verschiedene Helligkeitsstufen zur Verfügung“. Die Helligkeit der Uhr bewegt sich nur innerhalb dieser fest eingestellten Grenzen. Beachten Sie, dass die mit dem Befehl *gbr auto,l,h* einstellbare Untergrenze *l* und die Obergrenze *h* als Begrenzung höhere Priorität haben und die festen Grenzen überschreiben.

Wecker (Tischuhr) Helligkeit (Taster)	Wert Untergrenze			LDR-Wert	Wert Obergrenze		
	1	33	65		191	223	255
1							
2							
3							
(nur über <i>gbr-command</i> erreichbar)	frei*				frei* (Untergrenze < x ≤ 255)		

*) Hinweis: Die Werte der auto-Command-Einstellung haben höhere Priorität als die festen Werte der Helligkeitsstufen, das heißt, eine Auto-Untergrenze höher als die Untergrenze der aktiven Helligkeitsstufe wird nicht unterschritten.

		Mit <i>fix</i> wird ein fester Helligkeitswert <i>x</i> gesetzt. <i>Hinweis: Die Helligkeit hat einen Wertebereich von 0 bis 255. Die Änderung wird mit der Konfiguration im EEPROM gespeichert.</i>																																																																													
eep	r	EEPROM: gibt den gesamten EEPROM-Inhalt (0x0000 - 0x03FF Atmega328) als IntelHex aus.	eep r :10000000000010200000001C8FF0002F0303335009B ... :1003F0008870888870708888780810E001005AA595 :00000001FF OK.																																																																												
	w	startet eine Schreibprozedur: Q4 gibt ein ACK (ASCII 0x06) und erwartet eine Zeile IntelHex wenn die Checksumme stimmt, wird der Datensatz ins EEPROM geschrieben und mit 'ACK' die nächste Zeile angefragt. Timeout sind 10s pro ACK-Anfrage. Bei einem Fehler wird der serielle Buffer verworfen und NAK (ASCII 0x15) gesendet, um die Wiederholung der Zeile anzustoßen.																																																																													
c	x ?	<p>Color: zeigt die eingestellte Farbnummer [0..15] der LED-Gruppe x [0..3] an.</p> <p>Tabelle der vordefinierten LED-Gruppen:</p> <table border="1"> <thead> <tr> <th>LED-Gruppe</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Hintergrund. Alle LEDs, die nicht eingeschaltet sind, gehören zu dieser LED-Gruppe.</td> </tr> <tr> <td>1</td> <td>Matrix. Diese LED-Gruppe besteht aus den eingeschalteten LEDs der 10 x 11 Buchstaben-Matrix.</td> </tr> <tr> <td>2</td> <td>Minuten. Die eingeschalteten LEDs der vier Eck-Punkte bilden diese Gruppe.</td> </tr> <tr> <td>3</td> <td>Alarmsymbol. Die eingeschaltete LED hinter dem Alarmsymbol bildet eine eigene Gruppe.</td> </tr> </tbody> </table> <p>Tabelle der vordefinierten Farben:</p> <table border="1"> <thead> <tr> <th>Farbnummer</th> <th>Darstellung</th> <th>Beschreibung</th> <th>Rot-Wert</th> <th>Grün-Wert</th> <th>Blau-Wert</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>schwarz</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td></td> <td>weiß</td> <td>255</td> <td>238</td> <td>200</td> </tr> <tr> <td>2</td> <td></td> <td>rot</td> <td>255</td> <td>0</td> <td>0</td> </tr> <tr> <td>3</td> <td></td> <td>orangerot</td> <td>255</td> <td>109</td> <td>0</td> </tr> <tr> <td>4</td> <td></td> <td>orange</td> <td>255</td> <td>182</td> <td>0</td> </tr> <tr> <td>5</td> <td></td> <td></td> <td>255</td> <td>219</td> <td>0</td> </tr> <tr> <td>6</td> <td></td> <td>gelb</td> <td>255</td> <td>255</td> <td>0</td> </tr> <tr> <td>7</td> <td></td> <td></td> <td>200</td> <td>255</td> <td>0</td> </tr> <tr> <td>8</td> <td></td> <td>hellgrün</td> <td>128</td> <td>255</td> <td>0</td> </tr> <tr> <td>9</td> <td></td> <td>dunkelgrün</td> <td>0</td> <td>212</td> <td>28</td> </tr> </tbody> </table>	LED-Gruppe	Beschreibung	0	Hintergrund. Alle LEDs, die nicht eingeschaltet sind, gehören zu dieser LED-Gruppe.	1	Matrix. Diese LED-Gruppe besteht aus den eingeschalteten LEDs der 10 x 11 Buchstaben-Matrix.	2	Minuten. Die eingeschalteten LEDs der vier Eck-Punkte bilden diese Gruppe.	3	Alarmsymbol. Die eingeschaltete LED hinter dem Alarmsymbol bildet eine eigene Gruppe.	Farbnummer	Darstellung	Beschreibung	Rot-Wert	Grün-Wert	Blau-Wert	0		schwarz	0	0	0	1		weiß	255	238	200	2		rot	255	0	0	3		orangerot	255	109	0	4		orange	255	182	0	5			255	219	0	6		gelb	255	255	0	7			200	255	0	8		hellgrün	128	255	0	9		dunkelgrün	0	212	28	c 1 ? 1 OK.
LED-Gruppe	Beschreibung																																																																														
0	Hintergrund. Alle LEDs, die nicht eingeschaltet sind, gehören zu dieser LED-Gruppe.																																																																														
1	Matrix. Diese LED-Gruppe besteht aus den eingeschalteten LEDs der 10 x 11 Buchstaben-Matrix.																																																																														
2	Minuten. Die eingeschalteten LEDs der vier Eck-Punkte bilden diese Gruppe.																																																																														
3	Alarmsymbol. Die eingeschaltete LED hinter dem Alarmsymbol bildet eine eigene Gruppe.																																																																														
Farbnummer	Darstellung	Beschreibung	Rot-Wert	Grün-Wert	Blau-Wert																																																																										
0		schwarz	0	0	0																																																																										
1		weiß	255	238	200																																																																										
2		rot	255	0	0																																																																										
3		orangerot	255	109	0																																																																										
4		orange	255	182	0																																																																										
5			255	219	0																																																																										
6		gelb	255	255	0																																																																										
7			200	255	0																																																																										
8		hellgrün	128	255	0																																																																										
9		dunkelgrün	0	212	28																																																																										

10		hellblau	1	159	232
11			1	114	226
12		dunkelblau	4	0	208
13		violett	168	0	185
14		magenta	255	0	255
15		benutzerdefiniert			

Hinweis: Bei RGBW-LEDs wird der Weißanteil aus den RGB-Werten der vordefinierten Farben bestimmt und verwendet. Bei der benutzerdefinierten Farbe wird RGBW wie eingestellt verwendet und nicht umgerechnet. Dafür wird bei Verwendung von RGB-LEDs das Weiß ignoriert, RGB wie eingestellt verwendet.

	x y	setzt die LED-Gruppe x [0..3] auf die Farbe y [0..15]. Die Änderung wird mit der Konfiguration im EEPROM gespeichert.	c 2 15 Color 2 set to 15. OK.
cc	?	Custom Color: fragt die benutzerdefinierte Farbe Nummer 15 ab.	cc ? Custom Color (RGBW): 128,128,128,128. OK.
	r,g,b,w	setzt die benutzerdefinierte Farbe 15 auf die Farbwerte für Rot, Grün, Blau und Weiß. Die Änderung wird mit der Konfiguration im EEPROM gespeichert. <i>Hinweis: Der Weißanteil ist nur für RGBW-LEDs relevant.</i>	cc 255,0,0,10 Custom Color set to (RGBW): 255,0,0,10. OK.
lt		LEDTest: lässt für 10 s abwechselnd alle geraden und ungeraden Spalten / Minuten (und Alarm) LEDs leuchten und wechselt mit Überblenden. Bei farbigen LEDs werden die eingestellten Farben verwendet. <i>Hinweis: Sollen die Farbanteile von RGB(W)-LEDs getestet werden, so sollten alle 4 LED-Gruppen wie oben beschrieben auf Farbnummer 1 (weiß) gestellt werden.</i>	lt OK.
dfs		Default Settings: Setzt die Konfiguration zurück, welche beim nächsten Reset bzw. Aus- und Einschalten geladen wird. <i>Hinweis: Es gibt keinen Software-Befehl, um AtMegas zurückzusetzen. Atmel beschreibt offiziell ein beschriebene Verfahren, den Watchdog-Timer zu nutzen. In Q4 soll der Watchdog-Timer für eine zukünftige Erweiterung frei bleiben. Deswegen ist hier ein manueller Reset der Uhr nötig.</i>	
per	?	Show Day Period: gibt die aktuelle Einstellung der Anzeige von PM/AM aus im Format on off. <i>Hinweise: Die Buchstaben für AM und PM sind nur bei neueren Fronten-Designs so belegt, bei älteren Uhrenfronten findet man andere Buchstaben an diesen Stellen. Bei Fronten, die kein AM/PM enthalten (z.B. D3, Schwäbisch), leuchten keine LEDs. Die Funktion ist dann wirkungslos.</i>	per ? Show day period (AM/PM) is on. OK.
	on oder 1	schaltet die LEDs bei der Zeitdarstellung an den Buchstaben passend zur Uhrenfront ein.	per 1 Change: Show day period (AM/PM) is on.

			OK.
	off oder 0	schaltet die LEDs bei der Zeitdarstellung an den Buchstaben passend zur Uhrenfront aus.	per off Change: Show day period (AM/PM) is off. OK.
map	?	Fragt die aktive Abbildungsfunktion (Mapping) der Helligkeit auf die LEDs ab.	map ? LED mapping is lin. Br. debug on. OK.
	lin		map lin LED mapping changed to lin. Br. debug on. OK. Changes saved in Configuration 0
	exp		map exp LED mapping changed to exp. Br. debug on. OK. Changes saved in Configuration 0
	debug on oder debug 1 debug off oder debug 0	Schaltet die Debug-Funktion der Helligkeitssteuerung. <i>Hinweise:</i> <i>Die Debug-Aktivierung wird im EEPROM gespeichert und ist auch nach einem Reset der Uhr aktiv. Im Automatik-Modus wird die Helligkeit gelistet, sobald sie sich ändert. Im Fix-Modus wird die Umrechnung in RGBW einmalig bei jeder Änderung per „gbr fix,x“-Befehl.</i>	map debug on OK. Changes saved in Configuration 0 <i>Bei RGB(W)-Treiber:</i> Brightness: 159, Colorset 1 [RGBW]: 159,148,124,0. <i>Bei Matrix-Treiber:</i> Brightness: 159, Mapped: 159.
ldm	?	Fragt den aktiven Modus (LED Driver Mode) des LED-Treibers ab.	ldm ? LED driver is active. OK.
	on oder 1	schaltet den LED-Treiber ein.	ldm 1 Change: LED driver is active. OK.
	off oder 0	schaltet den LED-Treiber aus.	ldm off Change: LED driver is deactivated. OK.

		<i>Hinweis: Bei den Matrix-Treibern (GPIO, SPI) werden die Schieberegister nicht mehr bedient, bei den WS2812-LED-Ketten werden alle Farben auf 0 gesetzt und keine Schiebe-Vorgänge mehr ausgelöst.</i>	
Wenn die RTC DS323x statt DS1307 als Hardwarevariante eingestellt ist			
temp	?	Temperature: Gibt den gemessenen Wert des internen Temperatursensors des DS323x aus <i>Hinweise: Nachkommastellen sind bei der Messungenauigkeit nicht sinnvoll auszugeben. Die interne Temperatur der aktiven RTC wird höher sein als die Umgebungstemperatur.</i>	<i>temp ?</i> RTC temperature: 24°C.

ToDos:

- ~~— In dieser Liste: Beispielausgaben der Farbbefehle noch eintragen.~~
- Versionsausgabe für mögliche Fernsteuerprogramme
- ~~— Die Zeitzone wird in Q4 noch nicht berücksichtigt, denn da gibt es noch eine Hürde zu meistern: Mit der Zeitzone verschiebt sich auch die Datumsgrenze. Für ein softwareseitiges Mitführen des Datums müsste aber ein ewiger Kalender programmiert werden. Im DS1307 ist einer eingebaut, nur wie bekommt man ihn für das Anliegen genutzt? Einen Tag Verschiebung bekommt man auch mit einer Zeile Schaltjahresberechnung hin. Die Methode habe ich am 08.01.17 eingebaut und muss noch getestet werden. Ansatz: Im RTC ist die Lokalzeit gespeichert, die Zeitzone ist auch gespeichert, also bekannt. Beim ändern der Zeitzone wird die Differenz verschoben mit Datumkorrektur.~~
- ~~— Die neue Zeitzone wird zwar gespeichert, steht aber in der Hauptschleife noch auf der alten Einstellung.~~
- Fehlerhafte Eingaben werden noch nicht mit „unbekannter Befehl“ quittiert.
- ~~— Datumseingabe hat noch fehlerhaften Tag.~~
- Im Moment stört wohl noch die Debug-Ausgabe der Hauptschleife, Befehle muss man oft zweimal eingeben.
- Der Parameter Snoozetime soll veränderlich werden, noch wird er beim Kompilieren festgelegt auf 5 min (wie Originaluhr)
- ~~— DCF DEBUG (Ausgabe Empfangene Bits) soll noch fernsteuerbar gemacht werden (z.B. ‚dcf debug on|off|1|0‘), noch wird er beim Kompilieren festgelegt.~~
- tc ? und tc = xx|xx für die Oszillatorkorrektur des DC1307 (wenn verbaut), eventuell noch eine Automatik dazu im Vergleich zu DCF-Daten.
- ~~— Farbvorgaben mit C00-C15 für 14 Vordefinierte Farben (Regenbogen) und zwei frei definierbare Farben~~
- Man kann zwei (bis jetzt geplant) Presets für ein Schnellwechsel von Einstellungssätzen laden/speichern.
- ~~— Aufgabe des NVRAMs als Ablage für Zeit-Variablen, weil der DS3231 die Billig-Version des DS3232 ohne NVRAM ist. DS Forum favorisiert des DS3231. Die Variablen sind in die Konfiguration zu übertragen.~~
- ~~— Implementierung des DS3231/DS3232 als RTC~~
- Die Tastenreihenfolge der Platine *Du-kannst-mitspielen.de* „303“ und die elektrische Zuordnung sind gespiegelt verglichen zur Anordnung bei den *Du-kannst-mitspielen.de* Wortweckern. Deswegen sind die Zuordnungen der Zusatzfunktionen im Vergleich zum Original-Handbuch durcheinander geraten. Das wird mit dem nächsten Major-Change korrigiert
-