

# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)

Video: \_\_\_\_\_

## Vorwort

Meine Version 4 der LED-Steuerung war zentral aufgebaut, sprich alle schaltbaren LEDs mussten zentral im Zimmer zusammenlaufen und an der Steuerung angeschlossen werden. Das gab ein ziemliches Kabelchaos, also sollte eine neue Version mit Hilfe eines Bussystemes und dezentralen Clients auskommen. Die Industrie macht es ja vor, warum soll also nicht auch meine Steuerung so arbeiten. Außerdem konnte nur mit Hilfe einer RC5-Fernbedienung geschaltet werden. Da die Steuerung an sich recht unkomfortabel war (für Kanal 24 mussten zwei Tasten nacheinander gedrückt werden, was nicht immer funktionierte) wollte ich ein besseres komfortableres User-Interface. Die Steuerung hatte 24 Kanäle, die per Soft-PWM angesteuert wurden, es gab nur zwei Zustände der LEDs, an und aus, zwischen diesen Zuständen wurde gefadet. Das war die Motivation für die Entwicklung einer neuen Steuerung.

Projektstart:           Anfang 2007 (gedanklich)  
                              September 2008 (praktisch)

Projektende:            Februar 2009 (vorläufig)

Bitte nicht über „schlampige“ Ausführung beklagen, ich bin Maschinenbau-Student, d.h. ich hab nicht so viel Ahnung von professioneller Programmierung und Elektronik. Bei mir muss das Ding nur funktionieren, sollte z.B. ein Fehler auftreten dann gibt es keine Routinen die den Fehler melden oder entsprechend reagieren. Geht etwas nicht mehr, was allerdings sehr selten vorkommt, dann einfach kurz Strom weg nehmen. Bei mir muss es einfach nur grundsätzlich funktionieren. Hardwaremäßig kann es auch sein dass alles etwas hemdsärmelig ist (FET direkt an Pin von uC), allerdings hab ich das schon bei der Vorgängerversion dieses Projektes so gemacht. Diese Steuerung ist bei mir mehrere Jahre täglich gelaufen, und außer versagenden LEDs ist eigentlich nichts passiert.

Außerdem kann ich keine Garantie für die Richtigkeit des Codes übernehmen, es kann durch aus sein dass Fehler drin sind.

Die Webseiten sind für Safari designt, Firefox und Internet-Explorer haben Probleme bei der Darstellung.

## Spezifikation

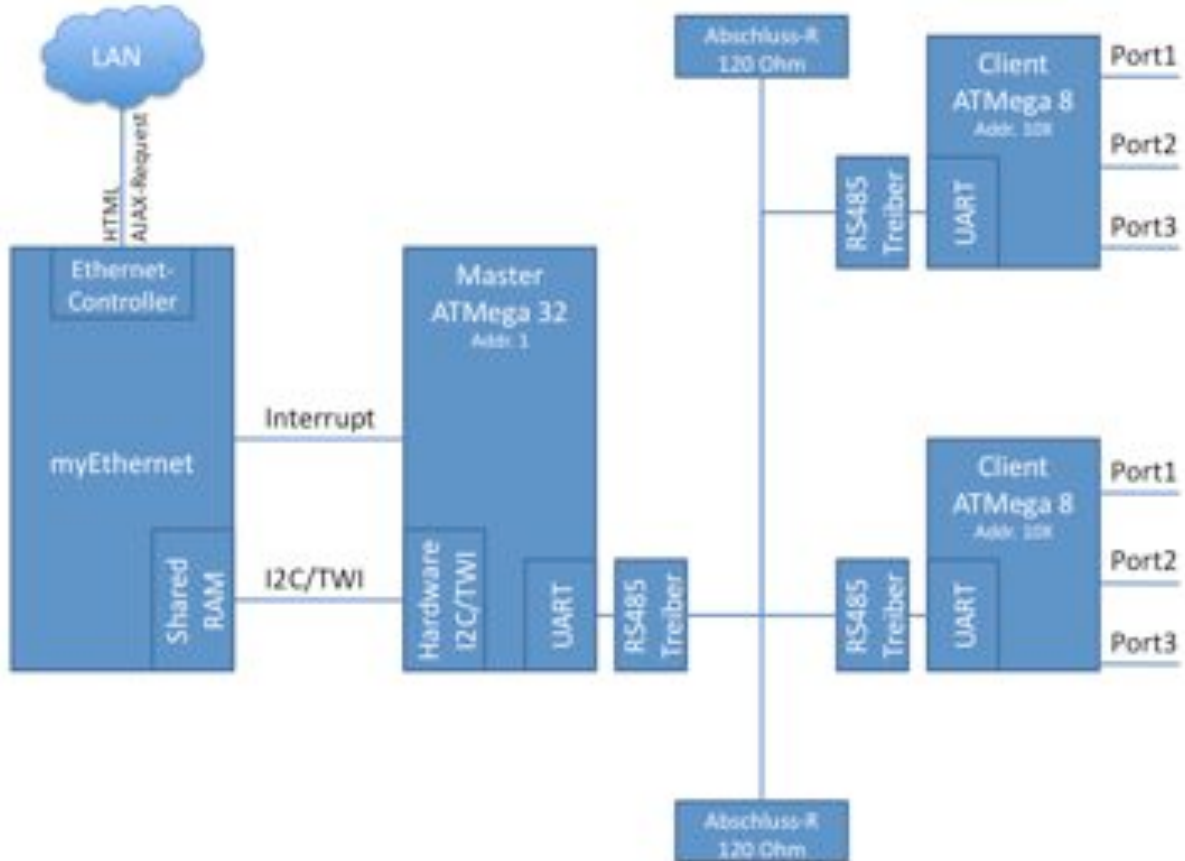
- Master/Client System, Bedienung über Webinterface
- bis zu 254 Clients
- 8 Bit LED-PWM-Dimmer mit intelligentem Ansatz (intelliPWM)

# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)

- 3 Software PWM-Kanäle
- integrierte HSV to RGB Umrechnung im Client
- 4 Speicherplätze im EEPROM für vordefinierte Szenen (dynamisch speicherbar)
- Aufbau komplett mit Lochrasterplatine und THT-Technik (für jeden machbar)
- geringe Gesamtkosten durch Standardbauteile
- einfache zu verstehende Programmierung
- Darstellung der Einstellungen der Clients im Webinterface

## Aufbau



## Aufbau Hardware

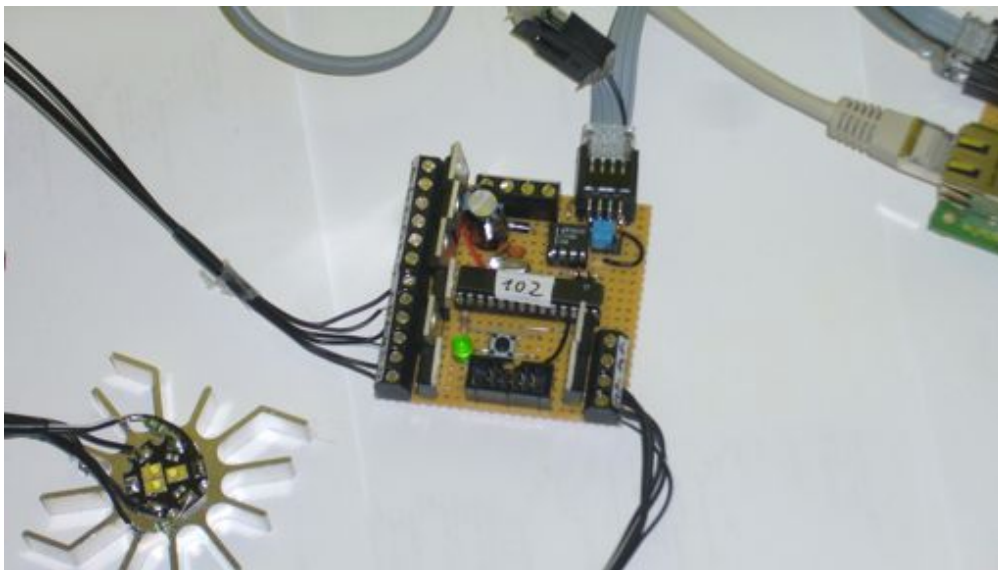
# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)

! " # \$ % & ' ( ) \* + von links nach rechts: Client, Master, Netzteil (5V, 20A)



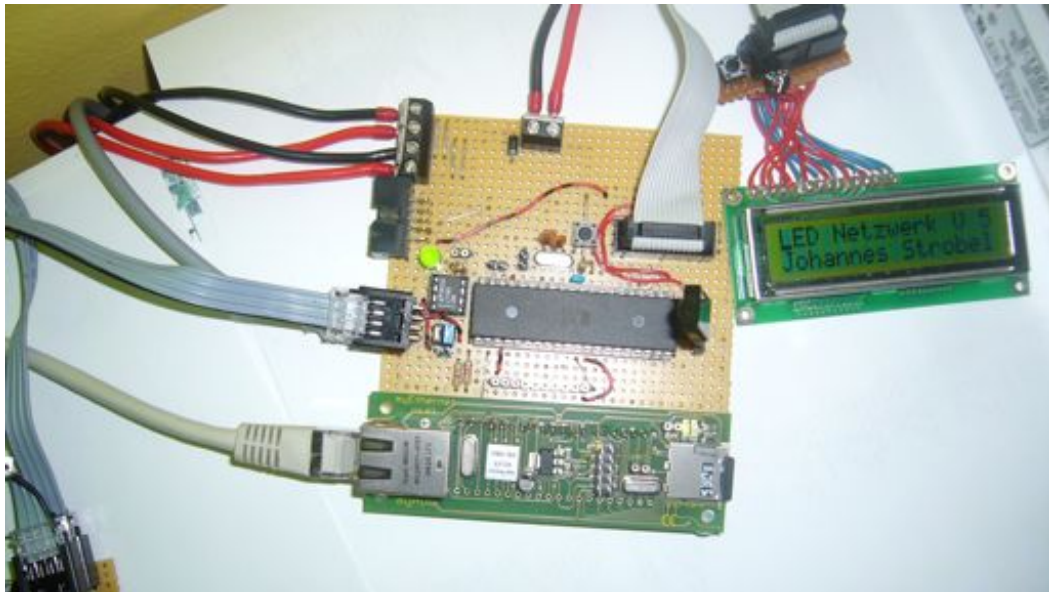
, - & # . ) Anschlüsse: oben RS485 mit Power; links RGB-LED 1; rechts LED Weiß



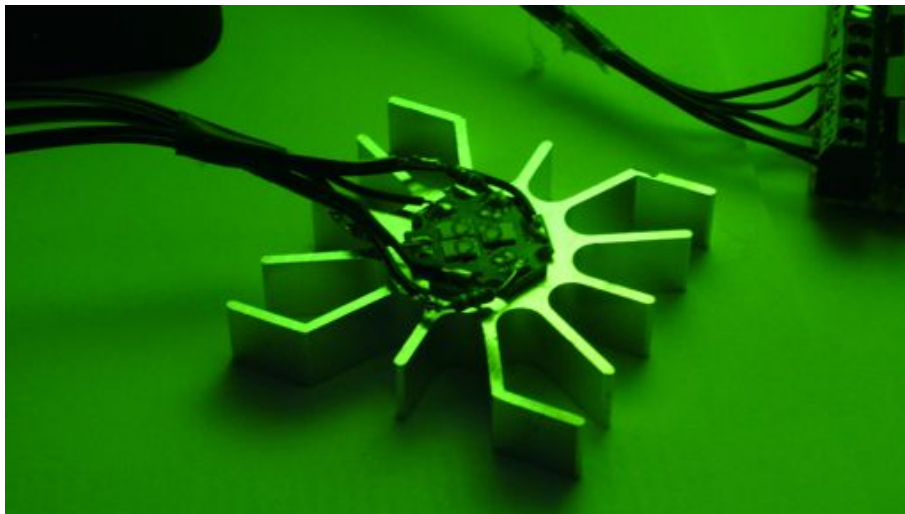
# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)

/0%)#\*\$ unten myEthernet (gesockelt) mit LAN Anschluss; mitte RS485; oben Power (2,5mm<sup>2</sup>)



Verwendete 1&2(3456#\$\$+789%: Luxeon Triple Rebel (ohne Konstantstromquelle, nur mit Vorwiderstand)

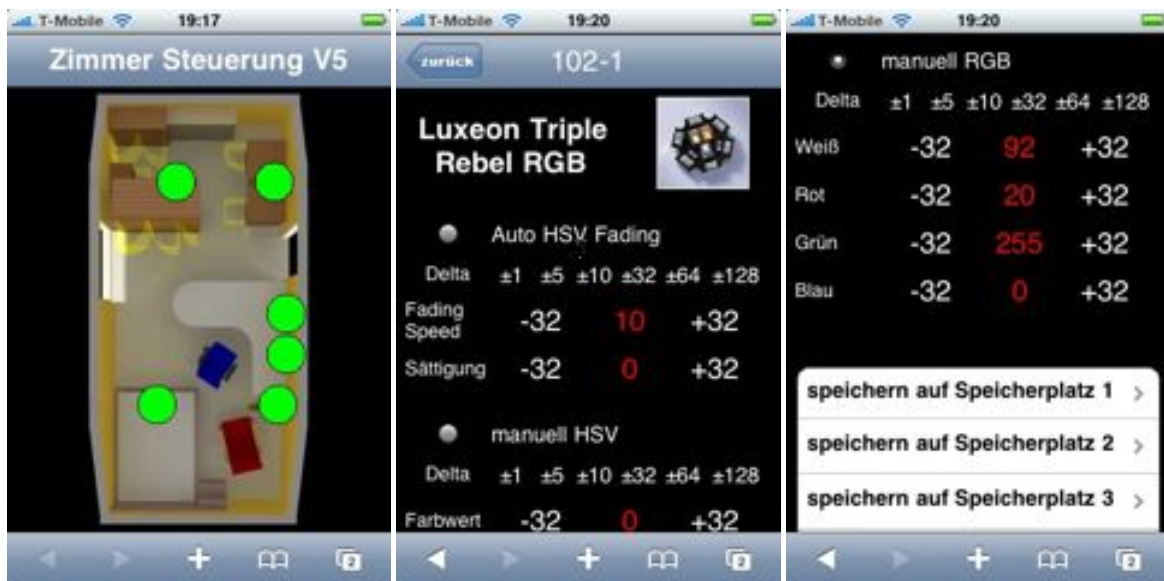


## Benutzer-Interface

Über das Benutzer-Interface lassen sich alle Einstellungen tätigen. Es wurde für die Verwendung auf einem iPhone/iPodTouch optimiert, betrifft vor allem die phys. Größe der Links. Auf dem normalen PC kann ein Pop-Up aufgerufen werden, dann entspricht die Fenster-Größe der Auflösung des iPhones.

# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)



Die Webseiten sind für Safari designt, Firefox und Internet-Explorer haben Probleme bei der Darstellung. (P.S. Safari gibt's auch für Windows ;-))

## Ablauf Kommunikation

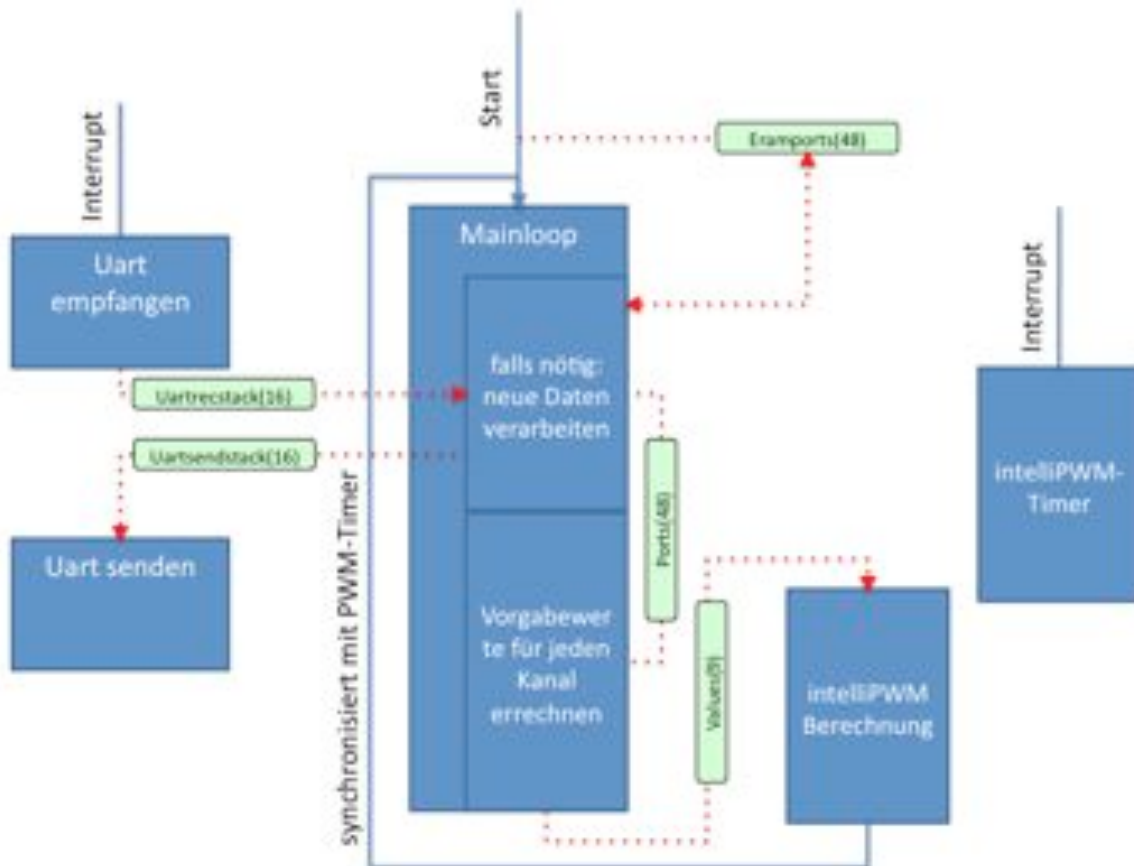
1. Webseite aufrufen (192.168.0.20)
2. Entsprechende LED in der Übersicht auswählen
3. Der Browser speichert nun über myEthernet jetzt die Adresse und den Port der LED ins SharedRAM , außerdem wird der Modus auf 2 gesetzt. Nun erzeugt er einen Interrupt.
4. Browser: der Aufruf der Detailseite wird nun um einige Hundert Millisekunden verzögert, so dass der Master sich die Informationen vom Client holen kann
5. Der Master spricht auf den Interrupt an, holt sich die Daten aus dem SharedRAM, leitet die Informationen 1:1 weiter ans RS485
6. Der entsprechende Client empfängt die Message (es sei denn es war ein Broadcast (Addr. 0))
7. Er reagiert darauf in dem er den Inhalt der „Ports“-Register ab Modus an die Adresse 1 sendet. Außerdem wird der Port bei dem Telegram auf 254 gesetzt.
8. Der Master empfängt das Telegram, weis dank Port=254 dass er die Daten in den SharedRAM schreiben soll. Also schreibt er ab Modus das Telegram in den SharedRAM
9. Nachdem die Verzögerung des Browsers abgelaufen ist lädt er die Detailseite.
10. Auf der Detailseite werden nun die Informationen mittels Ersetzungen ( $\%v1001^\circ$ , etc.) in das Array inputRAM gelesen.
11. Das Onload Javascript verarbeitet die Informationen (in unsigned konvertieren, den entsprechenden HTML-Elementen zuordnen)
12. Nun hat der Benutzer die Detailseite mit den aktuellen Einstellungen des Clients
13. Der Benutzer kann nun das Delta verändern, dabei wird nur über Javascript ersetzt

# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)

- Bei Betätigung einer Client-Einstellungsrelevanten Information wird mittels JavaScript die entsprechende Variable in den SharedRAM geschrieben. Dann wird wieder ein Interrupt erzeugt.
- Auf den Interrupt springt wiederum der Master an, er schickt den SharedRAM 1:1 auf den Bus
- Der Client empfängt das Telegramm und verarbeitet das Telegramm

## Programmablauf in den Clients



## Betriebs-Modi

Man kann zwischen drei Betriebsmodi wählen:

3+0: )51 ; <+

Fading entlang des HSV-Farbkreises mit einstellbarer Geschwindigkeit

3+=0. 1 ; <+

Einstellung von Helligkeit und Farbwert manuell

3+=0. >?@+

# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)

Einstellen der drei Farbwerte oder des Weißwertes manuell

Die Einstellungen können für jeden RGB-Kanal (Port) separat gemacht werden, auch die z.B. die Fading-Geschwindigkeit eines Ports ist unabhängig vom anderen Port des Clients.

## Beschreibung intelliPWM

Quelle: [http://www.mikrocontroller.net/articles/Soft-PWM#Intelligenter\\_L.C3.B6sungsansatz](http://www.mikrocontroller.net/articles/Soft-PWM#Intelligenter_L.C3.B6sungsansatz)

Um genug PWM Kanäle bereit stellen zu können, und außerdem flexibel und effizient genug die PWM integrieren zu können, entschied ich mich die Soft-PWM mit dem Ansatz wie in der Quelle umzusetzen. Allerdings musst dazu das Programm in BASCOM nachgebildet werden.

Die Funktion ist in der Quelle ja sehr gut beschrieben, deswegen hier nur kurz:

Der Grundansatz ist, die Anzahl der nötigen Timer-Interrupts auf die nur nötigen zu reduzieren, so hat man bei 8 Kanälen und 256 PWM-Stufen bei konventioneller Soft-PWM 256 Interrupts pro Zyklus. Die intelliPWM benötigt hier nur maximal 9 Interrupts, was dazu führt das der Prozessor insgesamt nicht so oft durch die Interrupts unterbrochen wird. Allerdings ist vor jedem PWM-Zyklus einen etwas längere Berechnung nötig. Allerdings ist der Hauptgrund für die Soft-PWM folgender: Möchte man die PWM mit 100 Zyklen pro Sekunde (100 Hz) betreiben, was nötig ist um selbst auf niedrigster Stufe (1/256 Helligkeit) kein Flimmern sehen, kommt man an die Grenzen des mit 16 MHz getakteten Mikrocontrollers. Es müsste bei konventioneller PWM der Timer mit 25 600 Hz laufen, sprich der uC hat zwischen jedem Zyklus 625 Takte zur Verfügung. Das ist zu wenig um solide auf die restlichen Interrupts, wie UART-Empfang, reagieren zu können. intelliPWM lässt hier im Durchschnitt wesentlich mehr Takte zwischen den Interrupts zu, und verringert insgesamt die Prozessorlast (siehe Quelle)

Die Umsetzung in BASCOM stellte sich als nicht ganz einfach heraus, und hat mich einige Nervenzellen gekostet. Aber die Mühe hat sich gelohnt, das intelliPWM funktioniert wunderbar.

## HSV to RGB

Quelle: <http://www.ledstyles.de/fpost57204.html#post57204>

Die Umrechnung ist im Client integriert.

## Protokoll

Komuniziert wird zwischen Master/Client über RS485

# Dokumentation LED-Netzwerk-V5

von Johannes Strobel (Kontakt: j.strobel (at) email.de)

Dabei wird MultiprocessorCommunication MPCM genutzt. Das heißt ein Byte besteht nicht aus 8 bit sondern 9 bit. Ist das 9. Bit gesetzt, so reagiert der Client darauf und löst einen Interrupt aus. Die Adresse und die Endbedingung (255) wird mit 9.Bit aktiv gesendet, die restlichen Daten mit 9.Bit = 0. Damit wird erreicht, dass nicht bei jedem gesendeten Byte in jedem Client ein Interrupt ausgelöst wird, sondern nur bei dem für den die Daten bestimmt sind.

RAM im Myethernet	0	1	2	3	4	5	6	7	8	9	10	11	12	
	Zustand	Adresse	Port	Modus	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Reserve	
<b>Programme</b>														
AutoHSV		Adresse	Port	50	Speed	Sat	0	0	0	0	0			
manuell HSV		Adresse	Port	60	0	Sat	Hue1	Hue2	0	0	0			
manuell RGB		Adresse	Port	70	0	0	0	0	Rot	Grün	Blau			
<b>Einstellungen</b>														
Werte anfordern		Adresse	Port	2										
Werte zurück		1	254	<>2										
Save to EEPROM		Adresse		1								Speicherplatz		
Load from EEPROM		Adresse		3								Speicherplatz		
		1	2	3	4	5	6	7	8	9	10	11	12	
<b>RS485 Protokoll</b>														
	Adresse	Port	Modus	Speed	Sat	Hue1	Hue2	Rot	Grün	Blau	Speicherplatz	Reserve		
				Port1										
				0	1	2	3	4	5	6	7	8	9	
<b>Speicherbereich Controller (Ports)</b>														
				Modus	Speed	Sat	Hue1	Hue2	Rot	Grün	Blau	Reserve	Reserve	

## Ansätze/Zukünftiges

- Steuerung von Steckdosen

## Code

siehe .zip im Anhang