

Software

Projektverwaltung

Anfangs stand die Idee im Raum, das Projekt mit Hilfe von Onedrive zu verwalten. Da OneDrive sich als umständlich erwies und man nur schwer Fehler korrigieren und das Projekt gar nicht zurückverfolgen konnte, wurde sich letzten Endes auf Rat unseres Projektleiters für Github entschieden.

Mithilfe von GitHub lassen sich ältere Versionen der Software wieder aufrufen, um beispielsweise Funktionen, die Anfangs nicht funktionieren, in die Aktuelle Software einzubauen.

Programmbeschreibung

Um die Lixie Uhr in Betrieb nehmen zu können, wird ein Programm benötigt. Dieser entstandene Code wurde in der Sprache C für den Attiny1606 (tinyAVR® 0-series) geschrieben.

Das Programm erlaubt dem Benutzer die Lixie-Uhr an einem Rechner mit einem Mini-USB zu USB-Kabel anzuschließen und über eine Konsole die Uhrzeit, die Farben und die Intensität des Lichts einzustellen.

Primär ist das Programm eine Uhrenansteuerung.

Der μ C sendet die Byteframes im Sekundentakt an die LED's.

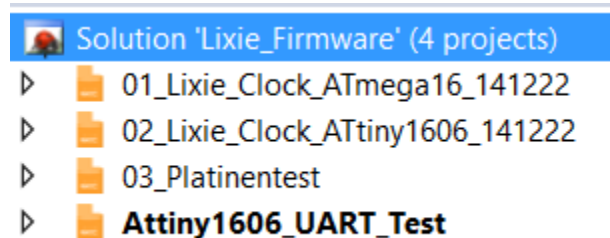
Sekundär erlaubt die Software eine individuelle Anpassung der Uhr.

Man kann beispielsweise das derzeitige Datum anzeigen oder ein Farbmuster generieren.

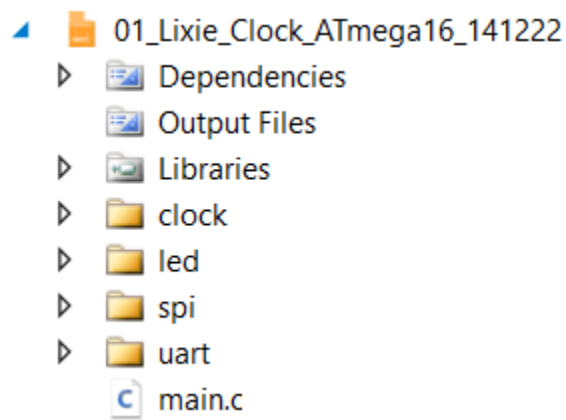
Entwicklungsablauf / Codebeschreibung

Da die LED's (APA102C) ausgetestet werden mussten, die für die Lixie Uhr verwendet werden, wurde die MEGACARD, die einen ATmega16 hat, verwendet. Im Unterricht wurde gemeinsam mit dem Projektleiter eine Bibliothek für die Datenverarbeitung der LED's erstellt, die anschließend für das Projekt verwendet wurde.

Die Solution „Lixie_Firmware“ besteht aus 4 Projekten:



Angefangen mit „01_Lixie_Clock_ATmega16_141222“, die als erstes bearbeitet wurde.



Konstanten und Bibliotheken

```
#define F_CPU 12000000UL //Systemtakt

#include <avr/io.h>
#include <avr/interrupt.h> //Systembibliotheken
#include <util/delay.h>

#include "uart/uart.h" //Benutzerbibliothek
#include "clock/clock.h"
```

Software Megacard

Struktogramme

Uhr Datentransfer

Uhr Daten

Timer Initialisierung

Takt Initialisierung

SPI Übertagung

LED-Zeit

LED-Initialisierung

LED-Startframe

LED Startframe

LED Startframe wird über SPI Übertragen

LED-Endframe

LED Endframe

LED Endframe wird über SPI Übertragen

Interrupt Service Routine

Interrupt Service Routine

Millisekunde wird hochgezählt			
T	Wenn Millisekunde größer als 9		F
Sekunde wird hochgezählt			
Millisekunde wird auf 0 gesetzt			
T	Wenn Sekunde größer als 59		F
Minute wird hochgezählt			
Sekunde wird auf 0 gesetzt			
T	Wenn Minute größer als 59		F
Stunde wird hochgezählt			Ø
Minute wird auf 0 gesetzt			
T	Wenn Stunde größer als 23		F
Stunde wird auf 0 gesetzt			Ø