Software

# Projektkonzept / Programmkonzept

# 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.

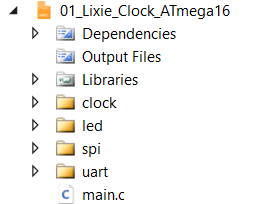
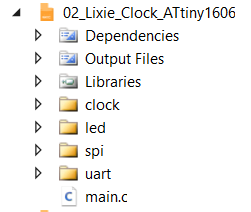
Primär ist das Programm eine Uhrenansteuerung.   
Der µC sendet die Byteframes im Sekundentakt an die LED’s.

Sekundär erlaubt die Software eine individuelle Anpassung der Uhr.   
Man kann beispielsweise das derzeitige Datum anzeigen oder ein Farbmuster generieren.

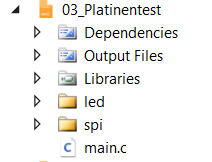
# Entwicklungsablauf / Codebeschreibung

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungDa 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:



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Projekt „01\_Lixie\_Clock\_ATmega16\_141222“

main.c

Konstanten und Bibliotheken



Uhr Voreinstellung  


Uhr Verarbeitung (Interruptgesteuert)



Timer Setup



Projekt „02\_Lixie\_Clock\_ATtiny1606“

main.c

Konstanten und Bibliotheken



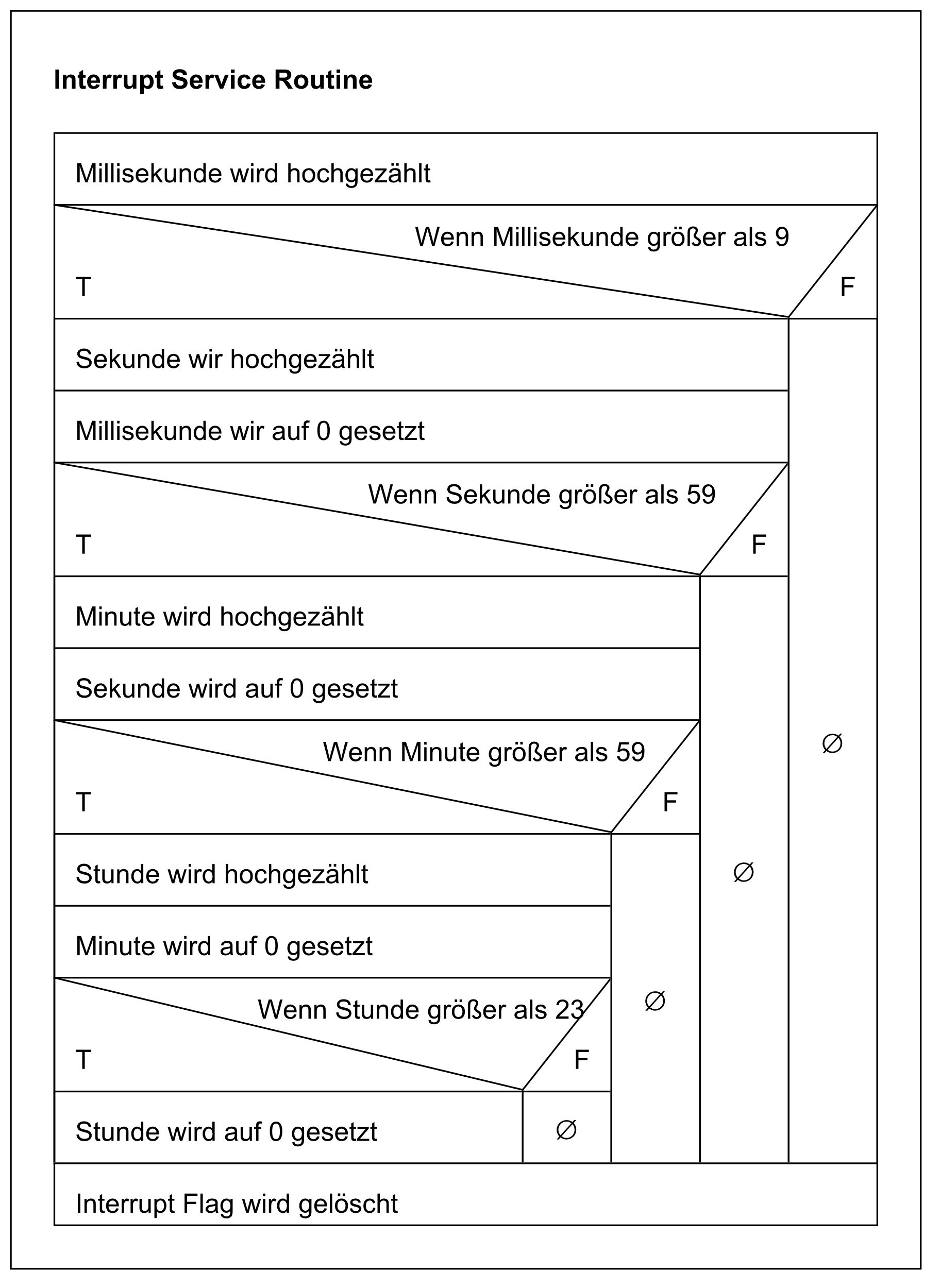
Uhr Voreinstellung  


Uhr Verarbeitung (Interruptgesteuert)

Code

Codebeschreibung  
Dies ist eine einfache Methode, um die Uhr hochzuzählen. Für eine Sekunde müssen 10 Mal 100 Millisekunden vergangen sein. Da das Interrupt jede 100 Millisekunden anspringt, muss eine Variable zählen, wie oft 100ms angesprungen ist. Erst ab dem 10. Mal wird die Variable zurückgesetzt und es wird um eine Sekunde raufgezählt

Struktogramm – Interrupt Service Routine



CPU-Initialisierung  


Timer Setup

Projekt „02\_Lixie\_Clock\_ATtiny1606“

clock.h



clock.c

Bibliotheken



Initialisierung



Daten

Datentransfer



Projekt „02\_Lixie\_Clock\_ATtiny1606“

led.h



Led.c

Bibliotheken



LED-Initialisierung



LED-Startframe



LED-Endframe



LED-Zeittransfer



Projekt „02\_Lixie\_Clock\_ATtiny1606“

spi.h



spi.c

Bibliotheken



SPI-Initialisierung



SPI-Übertragung



# Struktogramme