FH University of Applied Sciences TECHNIKUM WIEN

WiFi-Clock mit Remote Display

EMBSY-Projekt

Roman Reiter;Stephan Gruber

www.technikum-wien.at

Inhaltsverzeichnis

Projektgruppe

REITER Roman (ic21b023@technikum-wien.at)

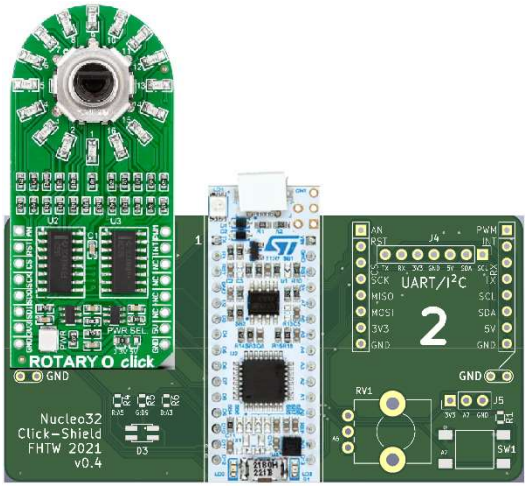
GRUBER Stephan (ic21b021@technikum-wien.at)

Projektbeschreibung

Im Zuge des EMBY-Projektes soll eine Real-Time-Clock mit Alarmfunktion entwickelt werden, welche aus einem Eingabe- und einem Ausgabe- Modul bestehen soll. Über das Eingabemodul (Rotary-Modul) hat der Benutzer die Möglichkeit die gewünschte Uhr- und Alarmzeit einzustellen, die Alarmfunktion ein- und auszuschalten und den Alarm zu resetten. Die Anzeige der Uhr- und Alarmzeit erfolgt über ein Ausgabemodul (OLED-Modul). Weiters besitzen das Ein- und Ausgabemodul jeweils ein WiFi-Modul, um die Uhr- und Alarmzeit, Alarmstatus und Parameter vom Eingabemodul zum Ausgabemodul zu transferieren.

Modulbeschreibung

Eingabemodul

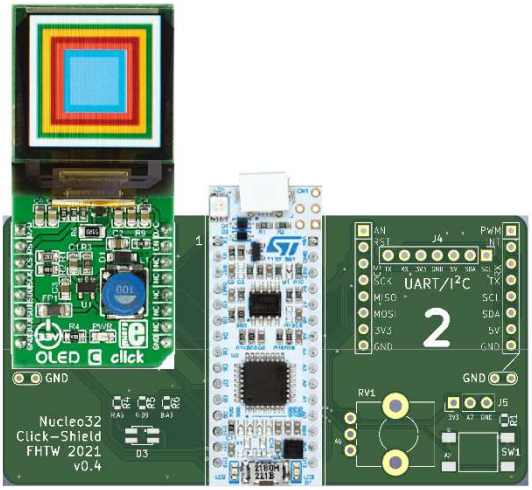


Das Eingabemodul besteht aus einem Rotary O click-Board und einem STM32-Mikrocontroller, welche über das Nucleo32Click-Shield der FH Technikum Wien miteinander gekoppelt sind.

Das Rotary O click™-Board verfügt über einen 15-Puls-Inkremental-Drehgeber (Encoder), welcher mit einem Ring aus 16 blauen LEDs umgeben ist. Die LED's werden mittels zwei 74HC595-Schieberegister angesteuert, welche sich auf der Hauptplatine befinden. Die Kommunikation mit den Schieberegistern erfolgt über eine SPI-Schnittstelle (CS, SCK, MISO, MOSI). Zusätzlich werden drei Leitungen herausgeführt, welche die Schaltinformationen (Drehrichtung und Tastendruck) des Encoders ausgeben (ENCB OUT, ENCA OUT und SW). Das Board ist für die

Verwendung von 3,3V und 5V ausgelegt. Eine grüne LED-Diode dient als Betriebsleuchte und signalisiert die Betriebsbereitschaft des Clipboards.

Ausgabemodul



Das Ausgabemodul besteht aus einem OLED C v101 click-Board und einem STM32-Mikrocontroller, welche über das Nucleo32Click-Shield der FH Technikum Wien miteinander gekoppelt sind.

Das OLED C v101 click™-Board verfügt über ein 128 RGB x 128 DotMatrixOLED-Segment mit einem SSD1351Z-Controller. Die Kommunikation mit dem SSD1351Z-Controller erfolgt über eine SPI-Schnittstelle (CS, SCK, MISO, MOSI). Das Board ist für die Verwendung von 3,3V und 5V ausgelegt. Eine grüne LED-Diode dient als Betriebsleuchte und signalisiert die Betriebsbereitschaft des Clipboards.

WiFi-Modul

Das WiFi BLE Click bietet WiFi- und BT/BLE-Konnektivität. Es verfügt über einen ESP32-WROOM-32, welches auf einen ESP32-D0WDQ6-Chip basiert. Der ESP32-WROOM-32 kann eine Datenraten von bis zu 150 Mbit/s erreichen. Die Firmware bietet eine UART-Kommunikationsschnittstelle, welche eine einfache Bedienung mit AT-Befehlen bietet. Das Board ist für die Verwendung von 3,3V und 5V ausgelegt.

Umsetzung

Funktionsbeschreibung

Nach dem Einschalten des jeweiligen Moduls (Ein- oder Ausgabemodul) wird dieses initialisiert. Das Eingabemodul (Rotary-Modul) wird als WiFi-Server initialisiert und das Ausgabemodul wird als WiFi-Client initialisiert.

Das Ausgabemodul verbindet sich nach erfolgreicher Konfiguration mit dem Eingabemodul und eröffnet eine TCP-Verbindung um Nachrichten vom Eingabemodul zu empfangen.

Das Eingabemodul sendet sekündlich einen Nachrichtenstring, welcher die aktuelle Uhr- und Alarmzeit, einen Parameter, welcher signalisiert welcher Uhrzeitparameter aktuell eingestellt wird und den Alarmstatus beinhaltet.

Das Ausgabemodul empfängt den Nachrichtenstring und zeigt die Zeiten des Nachrichtenstring an. Ebenso werden durch die mit übertragenen Parameter die jeweiligen Stunden, Minuten und Sekunden unterschiedliche eingefärbt, um die zu signalisieren, dass diese derzeit geändert werden.

Der Benutzer hat die Möglichkeit die Uhrzeit sowie die Alarmzeit einzustellen. Das Einstellen der Zeiten erfolgt durch das Aufrufen des Zeiteinstellungsmenüs.

Das Zeiteinstellungsmenü für die Uhrzeit erfolgt durch ein langes Drücken des Buttons am Click-Shield des Eingabemoduls und für die Alarmzeit muss der Encoder-Button lange gedrückt werden.

Der Ablauf des Zeiteinstellungsmenü für die aktuelle Uhrzeit und der Alarmzeit ist ident. Nachdem Eintritt in das Menü kann durch links- oder rechtsdrehen des Rotary's die Stunden der jeweiligen Zeit verändert werden. Ein Linksdreh verringert den Wert und ein Rechsdreh erhöht den Wert. Um die Minuten verändern zu können muss der Rotary-Button noch einmal kurz gedrückt werden, welche wieder durch einen links oder rechtsdreh verändert werden können. Durch nochmaliges kurzes drücken des Rotary-Buttons erfolgt die Einstellung der Sekunden, welche wieder durch einen links oder rechtsdreh verändert werden können. Durch nochmaliges kurzes Betätigen des Rotary-Buttons wird das Uhrzeiteinstellungsmenü beendet.

Um die aktuellen Werte der Stunden, Minuten und Sekunden zu erhalten, welche durch das Drehen des Rotary's verändert werden, werden diese über UART ausgegeben.

Das Aktivieren und Deaktivieren des Alarms erfolgt jeweils durch kurzes Betätigen des Rotary-Buttons. Wenn der Alarm aktiv ist, wird die grüne LED am Click-Shield eingeschaltet. Ebenso wird dies am UART ausgegeben.

Sollte ein Alarm ausgelöst werden, so leuchtet die LED rot und die Information, dass ein Alarm ausgelöst wurde, wird am UART ausgegeben.

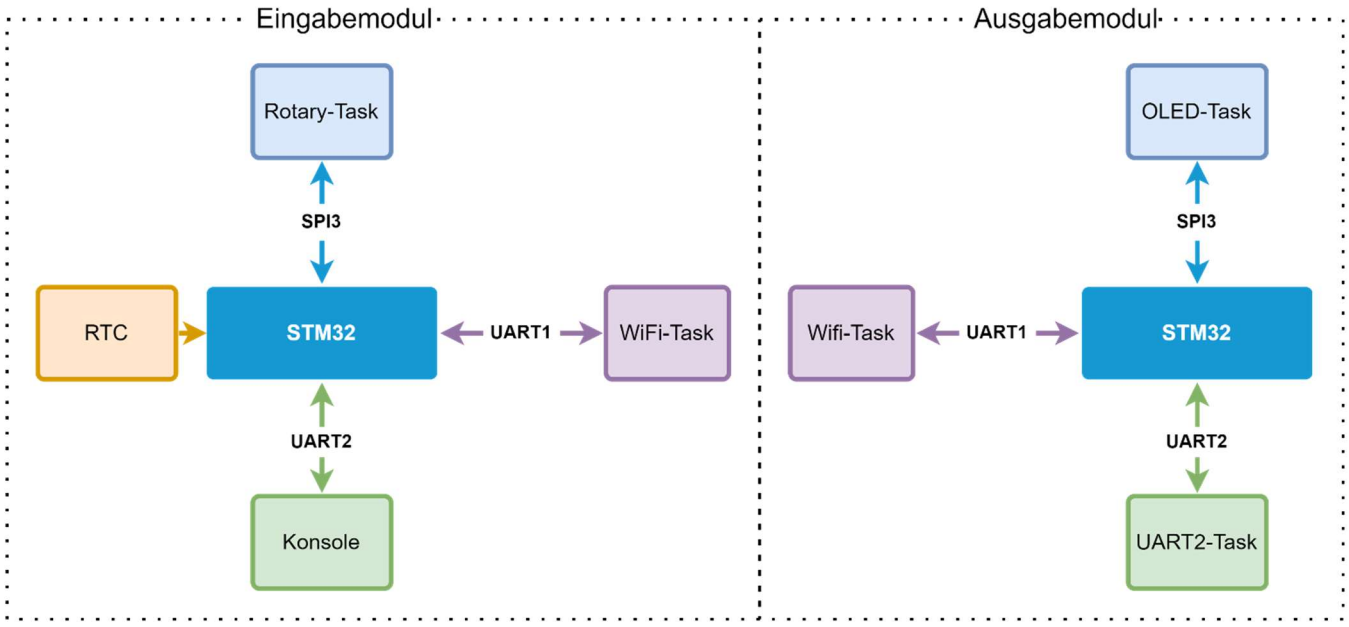
Der Alarm kann durch ein kurzes Betätigen des Rotary-Buttons resettet werden.

Ebenso besteht die Möglichkeit das Ein- und Ausgabemodul über UART (Putty-Konsole) zu steuern.

Blockdiagramme

Hardware

Hier werden die Schnittstellen zu den jeweiligen Hardwarekomponenten dargestellt.

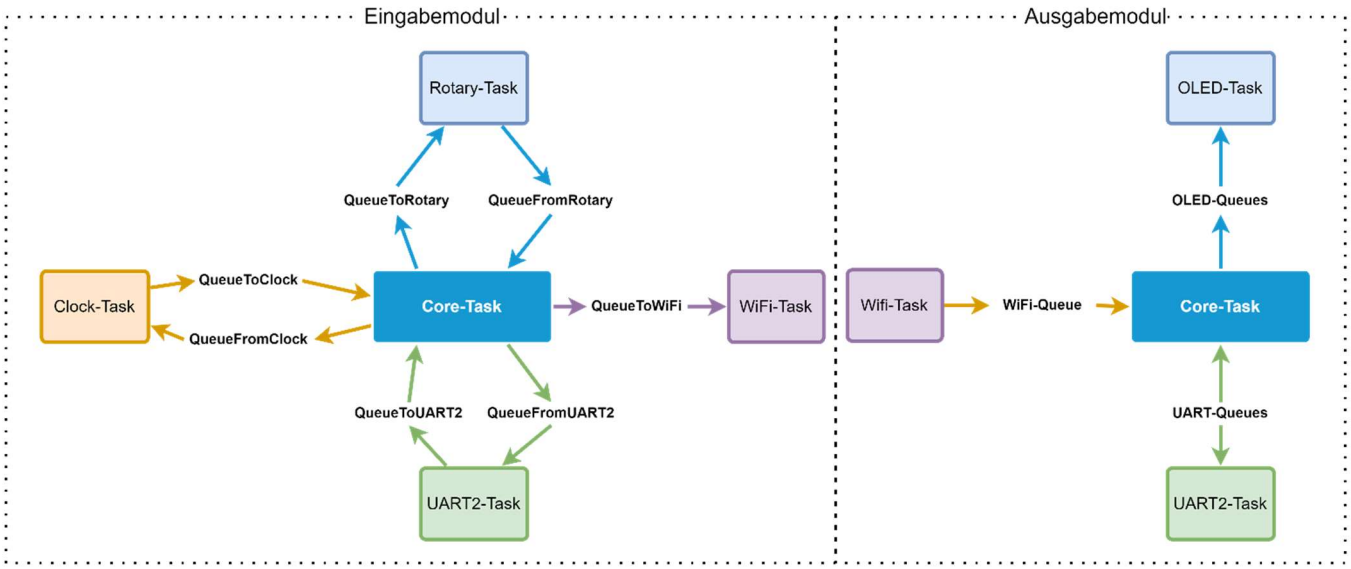


Im Eingabemodul wurde für die Uhrzeit und Alarm die interne RTC verwendet. Der Rotary dient dazu um die Uhr- und Alarmzeit sowie den Alarm einzustellen. Über die Konsole kann der Rotary gesteuert werden. Zusätzlich können Befehle verwendet werden, welche ebenfalls das Einstellen der Uhr- und Alarmzeit und den Alarm ermöglichen. Sollte das Eingabemodul ohne Ausgabemodul verwendet werden, so wird bei jeder Änderung der Uhr- und Alarmzeit, dem Alarmstatus und einem Alarm die Information am UART2 ausgegeben.

Das Ausgabemodul dient dazu, um die Uhr- und Alarmzeit und einen Alarm anzuzeigen.

Software

Zeigt den systematischen Softwareaufbau.



Im Eingabemodul erfolgt die Kommando/Nachrichten- Übertragung über Queues. Jeder außer der WiFi-Task verfügt über eine Sendequeue (QueueFrom#) und eine Empfangsqueue (QueueTo#). Sendequeues beinhalten Kommandos/Nachrichten, welche zu einem anderen Task transferiert werden. Empfangsqueues beinhalten Kommandos/Nachrichten, welche für den jeweiligen Task bestimmt sind. Der Core-Task ist der Verteiler der Kommandos/Nachrichten. Sekündlich wird eine Nachricht über WiFi zum Anzeigemodul gesendet. Das Ausgabemodul parst die empfangene Nachricht und zeigt je nach Parameter die Uhr- Alarmzeit, Zeiteinstellungskomponente und Alarm an.

Beschreibung der Softwarekomponenten

Eingabemodul

Kommunikation zwischen den Tasks (QueueItem_t)

Im Eingabemodul erfolgt die Kommunikation zwischen den Tasks mit Queues. Dazu wurden Empfängertask, Kommandos, Nachrichten und Status in einer Struktur zusammengefasst (QueueItem_t). Damit kann jeder Task eine Nachricht/Befehl/Status an einen anderen Task adressieren, indem er den Namen des Tasks (Empfängertask) angibt. Die QueueItems_t welche von einem Task zu einem anderen versendet werden, werden mit Queue#FromTask (#-Platzhalter für den Taskname) beziechnet. QueueItems_t welche für den jeweiligen Task zur Weiterverarbeitung bestimmt sind, befinden sich in der Queue#ToTask (#-Platzhalter für den Taskname).

```
typedef struct {
	ReceiverTask_t ReceiverTask;		/*!< Empfängertask */
	char Message[30];			/*!< Nachricht */
	int  Value;				/*!< Wert */
	State_t State;				/*!< State */
	Command_t Command;			/*!< Kommando */
} QueueItem_t;
```

Rotary-Task

Im Rotary-Task erfolgt die Verarbeitung der Evens, welche durch das Drehen und Drücken des Rotary-Buttons entstehen. Die Events werden ausgewertet und dementsprechend Handlungen gesetzt, welche in Form von Nachrichten an andere Tasks weitergeleitet werden. Die ausgehenden Nachrichten/Befehle werden in die QueueFromRotary eingefügt und vom Core-Task zum Empfängertask weitergeleitet. In der QueueToRotary befinden sich Nachrichten/Befehle, welche von anderen Tasks an den Rotary-Task gesendet wurden. Diese werden von dem Rotary-Task ausgelesen und dementsprechende Aktionen ausgeführt.

UART2-Task:

Im UART-Task werden eingegebene Befehle von der UART2-Schnittstelle (Putty-Konsole) verarbeitet. Je nach Befehl wird diese zu dem jeweiligen Task weitergeleitet, indem sie in die QueueFromUART eingefügt werden und vom Core-Task weitergeleitet werden. Nachrichten, welche sich in der QueueToUART befinden, werden auf dem UART2 (Konsole) ausgegeben.

WiFi-Task

Im WiFi-Task wird sekündlich eine Nachricht mit der aktuellen Uhr- und Alarmzeit über TCP gesendet. Die Nachricht wird im Clock-Task generiert und gelangt über den Core-Task in die QueueToWiFi, wo sie von WiFi-Task ausgelesen wird und über WiFi übertragen wird.

Clock-Task

Im Clock-Task werden Uhr- und Alarmzeit, Alarmstatus und Alarm verwaltet. Je nach Ereignis werden Nachrichten (QueueItems_t) zu anderen Tasks übertragen, welche in die QueueFromClock hinzugefügt werden. Änderungen und Abfragen von Uhrzeit, Alarmzeit, Alarmstatus und Alarm, befinden sich in der QueueToClock, welche von anderen Tasks stammen. Diese werden ausgelesen und je nach Anfrage abgearbeitet und ggf. mit einer Nachricht beantwortet.

Core-Task

Der Core-Task ist mit jeder Queue verbunden und leitet Nachrichten von einer Queue in die Queue weiter. Er nimmt das QueuItem_t aus der jeweiligen Queue, liest den Namen des Empfängertask und fügt das QueueItem_t in die entsprechende Queue des Empfängertasks.

Entwickelte Modul-Treiber

Im Zuge der Umsetzung wurden Bibliotheken bzw. Modultreiber für das Rotaty-Click Board, den Button des Click-Shields, die UART2-Schnittstelle und des Wifi-Moduls entwickelt. Die Modultreiber ermöglichen eine vereinfachte Benutzung der Komponenten. Die Beschreibung und Verwendung der Treiber befinden sich im jeweiligen c-Files des Modultreibes.

Ausgabemodul

OLED-Task:

Im OLED-Task werden die Ausgabewerte der OLED-Queue eingelesen und die Kommandos an das OLED C click-Board gesendet, um Werte anzuzeigen.

WiFi-Task:

Im WiFi-Task werden die empfangenen und gesendeten Nachrichten vom WiFi BLE click-Board verarbeitet, welche in den jeweiligen Sende- und Empfangsqueues gespeichert werden.

UART-Task:

Im UART-Task werden die empfangenen und gesendeten Nachrichten der Konsole verarbeitet, welche in den jeweiligen Sende- und Empfangsqueues gespeichert werden.

Core-Task:

Der Core Task sammelt alle Nachrichten (UART-Queue, WLAN-Queue) zusammen und leitet diese an den OLED Task weiter.

Protokollbeschreibung

Die WiFi-Clock besteht aus einem Serialprotocol und einem Wifiprotocol. Das Serialprotocol dient dazu Befehle und Abfragen über UART2 zu tätigen. Die eingegebenen Befehle werden analysiert und verarbeitet. Ebenso wird auch auf Falscheingaben reagiert. Das Wifiprotocol dient dazu Nachrichten mit dem Wifi-Modul über TCP zu versenden und zu empfangen.

Serialprotocol

Das Serialprotocol dient zur Kommunikation zwischen der UART2-Schnittstelle und dem STM32. Im Serialprotocol können Befehle definiert werden, welche validiert werden und je nach Definition verarbeitet werden.

Kommandoaufbau-Syntax

#<Command>=<Parameter1, ...>\r

Kommando zum Setzen von Parametern (Set-Command)

#<Command>?\r

Kommando zum Auslesen von Parametern (Get-Command)

#<Command>\r

Kommando zum Ausführen (Execute-Command)

Befehle des Eingabemoduls

#time?

Gibt die aktuelle Uhrzeit zurück (z.B. 12:45:23)

#time_hours=

Gibt die Stunden er aktuellen Uhrzeit zurück (z.B. #time_hours=12)

#time_hours?

Setzt die Stunden der aktuellen Uhrzeit (z.B. 12)

#time_ minutes =

Gibt die Minuten er aktuellen Uhrzeit zurück (z.B. #time_seconds=45)

#time_minutes?

Setzt die Minuten der aktuellen Uhrzeit (z.B. 45)

#time_ seconds =

Gibt die Sekunden der aktuellen Uhrzeit zurück (z.B. #time_seconds=23)

#time_seconds?

Setzt die Sekunden der aktuellen Uhrzeit (z.B. 23)

#alarm?

Gibt die aktuelle Alarmzeit zurück (z.B. 12:55:01)

 # alarm _hours=

Gibt die Stunden er aktuellen Alarmzeit zurück (z.B. #time_hours=12)

 # alarm _hours?

Setzt die Stunden der aktuellen Alarmzeit (z.B. 12)

 # alarm _ minutes =

Gibt die Minuten er aktuellen Alarmzeit zurück (z.B. #time_seconds=55)

 # alarm _minutes?

Setzt die Minuten der aktuellen Uhrzeit (z.B. 55)

 # alarm _ seconds =

Gibt die Sekunden der aktuellen Alarmzeit zurück (z.B. #time_seconds=01)

 # alarm _seconds?

Setzt die Sekunden der aktuellen Alarmzeit (z.B. 01)

 # alarm _state?

Gibt den Alarmstatus (Alarm ein oder aus) (z.B. 00 oder 01)

 # alarm _quit

Setzt den Alarm zurück

Befehle des Ausgabemoduls

#dc=<color>

Setzt die Zeichenfarbe (Angabe mittels Dezimalzahlen)

#dc?

Liest die Zeichenfarbe aus.

#drt=<text>

Zeichnet einen Text auf der Stelle (0,0). Die Texte werden hierbei ohne Hochkomma angegeben und enden bei dem nächsten Leerzeichen.

#drtp=<text>,<x>,<y>

Zeichnet einen Text auf der Stelle (x,y). Die Texte werden hierbei ohne Hochkomma angegeben und enden bei dem nächsten Leerzeichen.

#bgc=<color>

Ändert die Hintergrundfarbe des Displays. Farbwerte müssen mittels Dezimalzahlen angegeben werden.

#bgc=?

Liefert die Hintergrundfarbe des Displays zurück.

#tc=<color>

Setzt die Textfarbe des Oled Displays. Farbwert muss in Dezimalzahlen angegeben werden

#tc=?

Liefert die aktuelle Textfarbe zurück.

#drrp[s]=<start x>,<start y>,<end x>,<end y>

Zeichnet ein Rechteck, in der Zeichenfarbe (dc) an der Stelle ("start x","start y") mit der Größe "end x" - "start x" & "end y" - "start y"

Wifi-Protokoll

Das Wifiprotocol dient dazu, um Nachrichten über TCP zu senden und zu empfangen. Für die WiFi-Clock wird nur ein Nachrichtenformat verwendet, welche die Uhr- und Alarmzeit, sowie Anzeigeparameter enthalten. Grundsätzlich baut das Wifiprotocol auf den werksseitigen AT-Commands des ESP32 auf. Die AT-Commands des WiFi-Moduls befinden sich im Datenblatt.

Die Nachricht wird sekündlich vom Eingabemodul zum Ausgabemodul gesendet.

Aufbau des Nachrichtenframes

#time=<Uhrzeit>,<Alarmzeit>,<Index des einzustellenden Zeitparameters>,<Alarm-Anzeigeparameter>\r

Beschreibung des Übertragungsframes

#time=

Beginn des Frames

<Uhrzeit> (Format-00:00:00)

Stunden, Minuten und Sekunden der Uhrzeit werden zweistellig dargestellt.

<Alarmzeit> (Format-00:00:00)

Stunden, Minuten und Sekunden der Alarmzeit werden zweistellig dargestellt.

<Index des einzustellenden Zeitparameters>

Gibt bei einer Zeiteinstellung (Uhrzeit oder Alarmzeit) an, welcher Parameter eingestellt wird, um die zu ändernde Zeitkomponente (Stunden, Minuten oder Sekunden) farblich zu markieren.

0 – Standardfarbe (schwarz)

1 – Stunde-Uhrzeit (rot)

2 – Minute-Uhrzeit (rot)

3 – Sekunde-Uhrzeit (rot)

4 – Stunde-Alarmzeit (rot)

5 – Minute-Alarmzeit (rot)

6 – Sekunde-Alarmzeit (rot)

<Alarm-Anzeigeparameter>

Stunden, Minuten und Sekunden der Alarmzeit werden zweistellig dargestellt.

0 – Alarmzeit ausblenden

1 - Timer anzeigen

2 - Timer abgelaufen (andere Farbe)

\r

Ende des Frames

Arbeitsverteilung

Reiter Roman (ca. 50h):

Verantwortlich für das Eingabemodul (Rotary click)

Arbeitsschritt	Stunden
Rotary-Treibermodul	15
WiFi-Accespointmodul	15
UART-Kommunikationsmodul	10
Busineslogik (Clock, Clockmenü, Wifi, UART)	10

Stephan-Gruber (ca. 50h):

Verantwortlich für das Ausgabemodul (OLED C click)

Arbeitsschritt	Stunden
OLED-Treibermodul	15
WLAN-Clientmodul	15
UART-DebugLog (In- und Output)	5
Messageparsing	5
Busineslogik	10