

1. Funktionale Voraussetzungen

1.1. Anforderungen an die Uhr (Radio kontrollierte Uhr DCF77):

- Die aktuelle Zeit muss im Format HH:MM:SS auf dem LCD Display dargestellt werden.
- Aktualisierung der Zeit jede Sekunde durch time Clock
- Aktualisierung der Zeit jede Minute durch DCF77
- Falls die Informationen der DCF77 fehlschlägt, soll Uhrzeit der time Clock verwendet werden
- Die LED an Ausgang B.0 muss einmal pro Sekunde durch time Clock umgeschaltet werden
- Die LED an Ausgang B.1 muss an sein, wenn das DCF77 Signal niedrig ist und aus wenn das DCF77 Signal hoch ist. Wenn das komplette DCF Signal angekommen ist, soll auch diese LED einmal pro Sekunde umschalten
- Die LED an Ausgang B.2 muss an sein, wenn ein Fehler im Zeitformat durch Paritätsbitüberprüfung festgestellt wurde, solange bis gültige Informationen vorliegen
- Die LED an Ausgang B.3 muss an sein, wenn ein vollständiges und korrektes DCF77 Datum und Zeitinformation entschlüsselt wurde und aus sein wenn keine oder unvollständige Informationen angekommen sind
- Das DCF77 Signal ist über den Ausgang H.0 verbunden

1.2. Anforderungen an das LCD Display (Darstellung):

- Die erste Zeile des LCD Displays soll die Uhrzeit in folgendem Format darstellen:
hours :minutes :seconds Zeitzone (US/DE)
- Die zweite Zeile wird folgendermaßen Dargestellt:
Weekday day. month. year

1.3. Sonstige Anforderungen:

- Das Modul „ticker.asm“ aus Aufgabe 2.1 muss zur Taktgenerierung benutzt werden
- Für LCD und LED müssen die Module „LCD.asm“ und „LED.asm“ benutzt werden
- Die Unterbrechungsroutine soll sich nur mit der Unterbrechungshardware beschäftigen und nicht komplexe Routinen ausführen
- Die Kommunikation zwischen den Unterbrechungsrouinen und den anderen Teilen des Programms erfolgt über globale Variablen
- Um den Status der Knöpfe abzufragen, muss regelmäßiges Abfragen benutzt werden, keine Unterbrechungen

2. Die Benutzerschnittstelle des Programms

2.1. LCD Display

- Erste Zeile: hours:minutes:seconds Zeitzone (US/DE)
- Zweite Zeile: Wochentag day.month.year

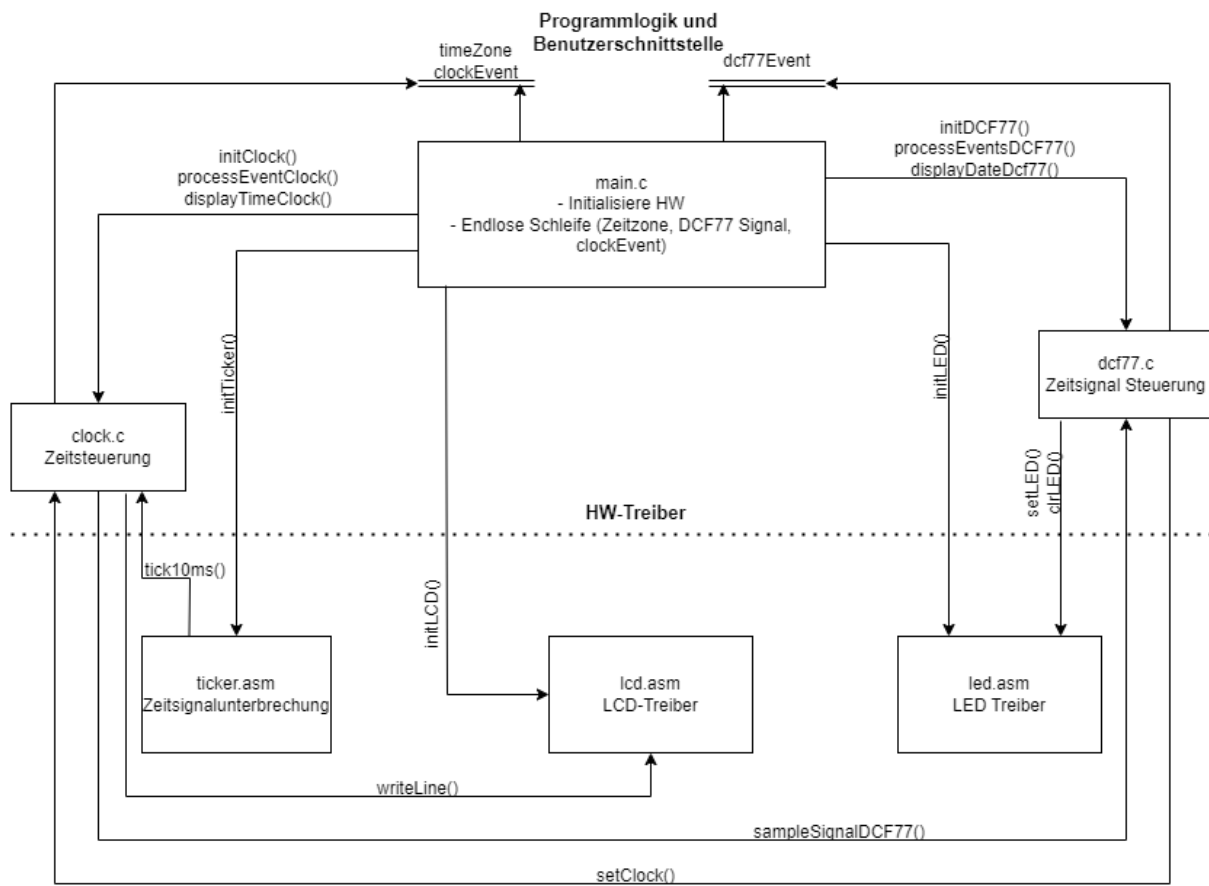
2.2. LEDs

- LED0: Wechsel einmal pro Sekunde
- LED1: An, wenn DCF77 Signal niedrig, aus wenn DCF77 Signal hoch.
Wechselnd sobald Signal vollständig erhalten
- LED2: An, wenn Error festgestellt, aus sobald gültige Daten angekommen
- LED3: An, wenn vollständiges und korrektes Zeit -und Datumssignal
entschlüsselt, aus wenn keine oder fehlerhafte Daten vorhanden

2.3. Knöpfe

- H3: Wechseln zwischen „DE“ und „US“ Zeitzone

3. Modulüberblick



4. Datenstruktur

4.1. Globale Variablen

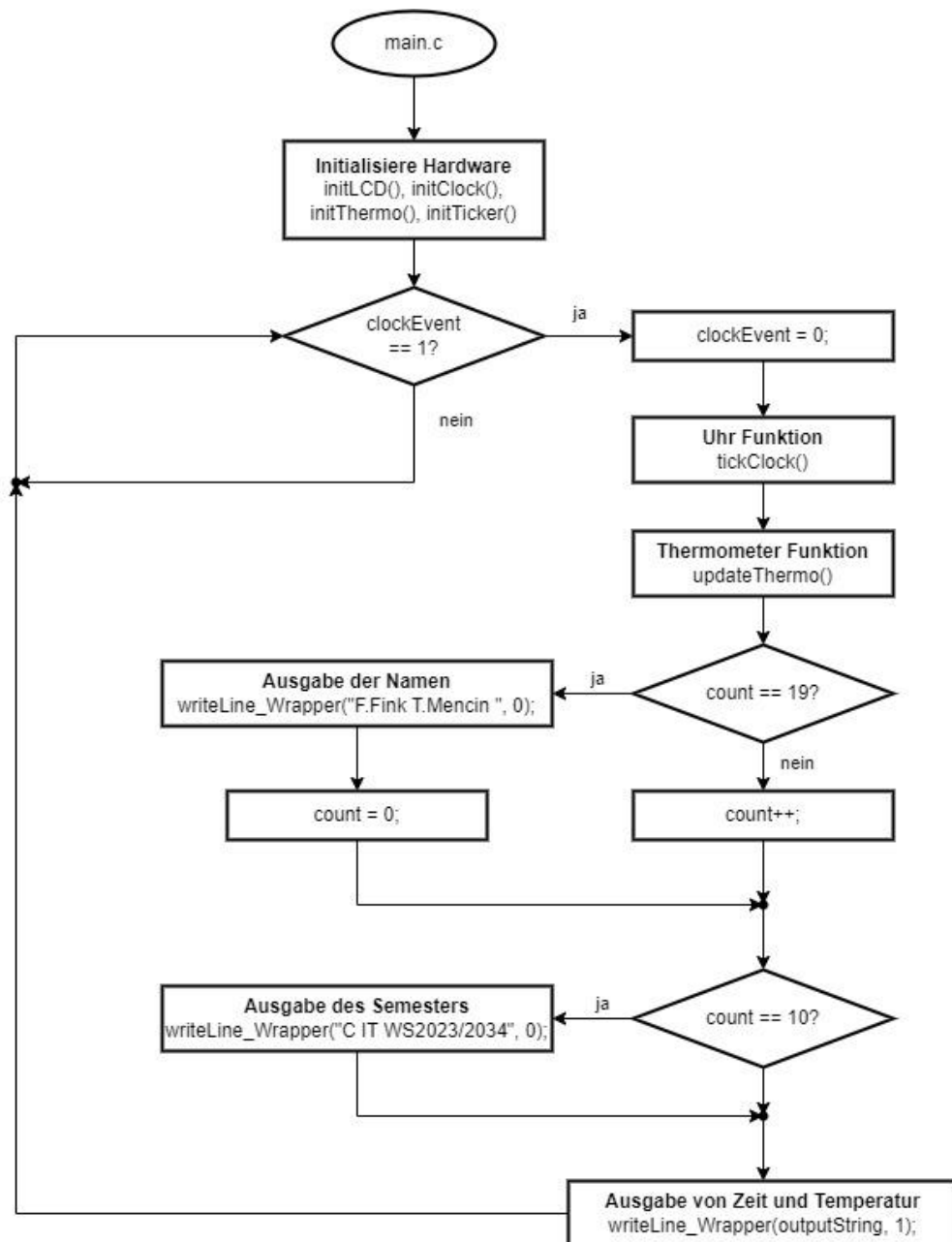
Modul	Variablenname	C Datentyp	Zweck
main [ticker]	clockEvent	unsigned char (0...1)	Regelmäßiges Anzeigen, das eine Sekunde rum ist
main	count	unsigned char (0...19)	Wechsel der ersten Zeile alle 10 Sekunden
wrappers [main, thermometer, clock]	outputString	char [17]	Speicherung der zweiten Ausgabezeile
clock	time.hour	char (struct) (0...23)	Stunden
clock	time.minute	char (struct) (0...59)	Minuten
clock	time.second	char (struct) (0...59)	Sekunden
clock	clockMode	enum (NORMAL,SET)	Speicherung des Uhrmodus
dcf77	dataBits	char[59]	Speicherung der erhaltenen dcf77 Daten Bits
dcf77	bitCount	unsigned char (0...1)	Speicherung des Decoder Bits
clock	timeZone	unsigned char (0...1)	Speicherung der Zeitzone (EU,US)
clock	clockEvent	CLOCKEVENT	Speicherung des letzten Clock Events
dcf77	decoderState	enum {DATAINVALID, WAITFORPOSEDGE, WAITFORNEGEDGE}	Speicherung des Decoder Status

4.2. Hardware Ressourcen

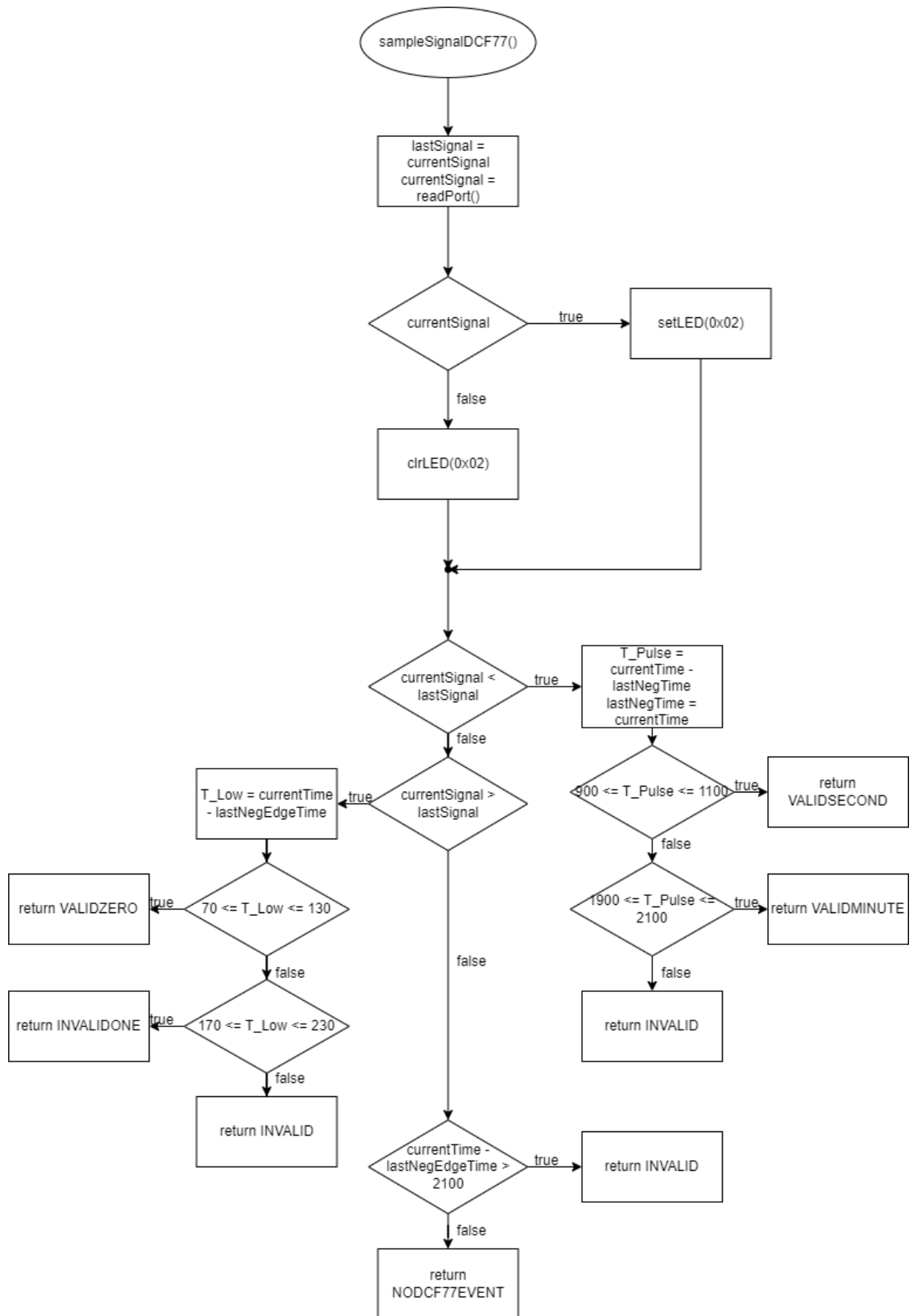
Module	HCS12 or Dragon12 HW resource	Purpose
LCD [Main]	LCD Anzeige	Erste Zeile: hours:minutes:seconds Zeitzone Zweite Zeile: Wochentag day.month.year
LED [Clock]	LEDs	Statussignale
Clock	Knopf H3	Wechsel zwischen US und DE
Ticker [Main]	Timer Kanal 4	10 ms Ticker

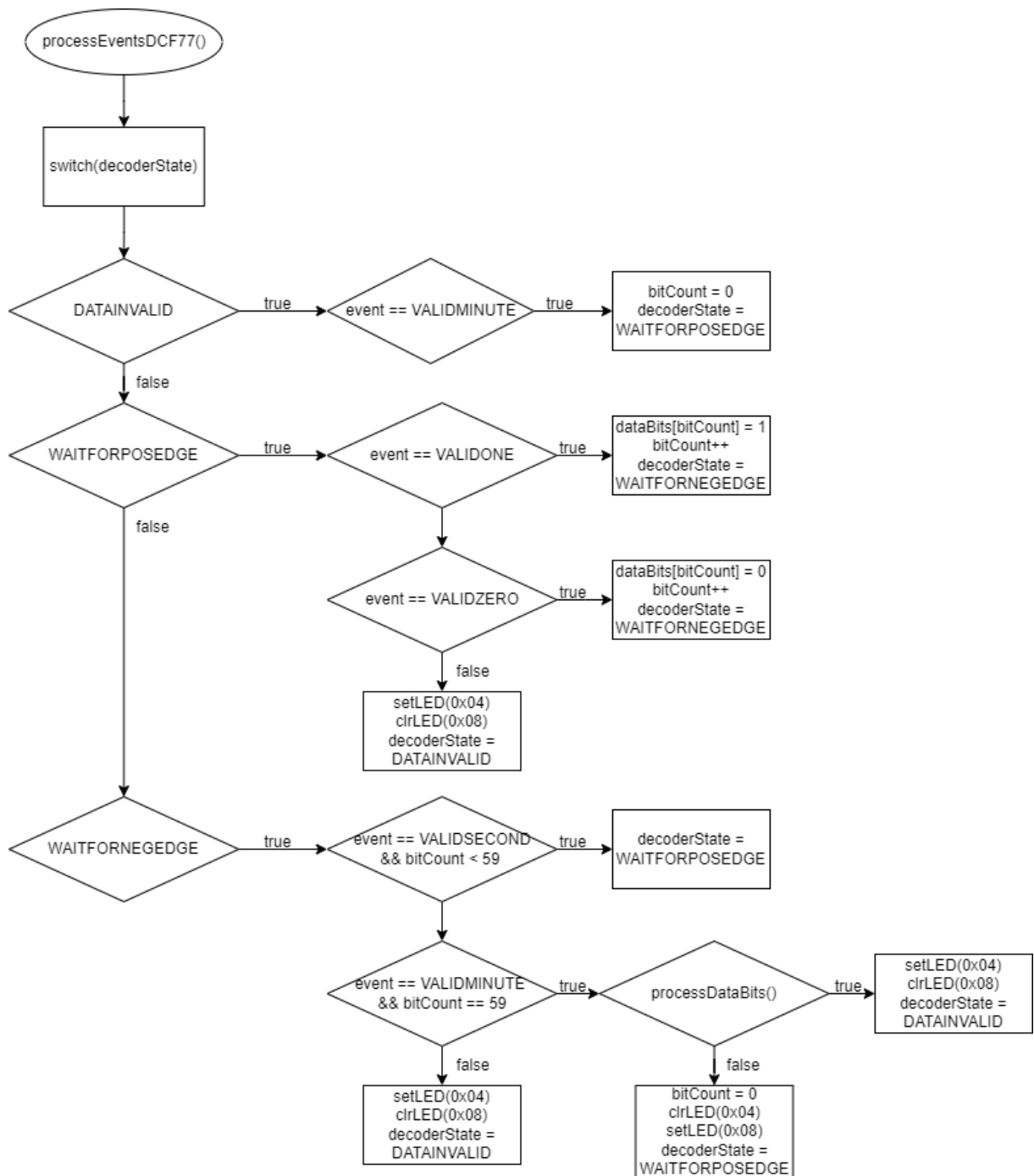
5. Flussdiagramm für alle Module

5.1. Main



5.2. DCF77





6. Schnittstellenbeschreibung

siehe Code