JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll

IndInf03: Ampelsteuerung mit Interrupts

Systemtechnik Labor
5BHIT 2015/16, Gruppe z

Michael Weinberger

Version 1.0

Note:

Betreuer: Weiser

Begonnen am 20. November 2015

Beendet am 20. November 2015

Inhaltsverzeichnis

1Einführung 3
1.1Ziele 3
1.2Voraussetzungen 3
1.3Aufgabenstellung 3
2Ergebnisse 4

1 Einführung

1.1 Ziele

Via Interrupt soll die bereits implementierte Ampel in eine Nachtschaltung schalten können sowie auch per Knopfdruck wieder in den normalen Zustand zurückkehren.

1.2 Voraussetzungen

- STM32F3 Discovery-Mikrocontroller
- USB-Kabel
- OpenSTM32-Workbench
- Einarbeitung in Doxygen
- Grundwissen in C

1.3 Aufgabenstellung

Implementiere eine Ampel, welche rein mit Interrupts gesteuert wird:

a) Die Ampel möge von rot-orange-grün auf orange-blinken umschalten, wenn der Userbutton gedrückt wird.

b) Sowohl die rot/orange/grün-Phasen als auch die Phasen des Orange-Blinkens sollen mittels Timerinterrupts gesteuert werden.

Das Hauptprogramm besteht dann nur noch aus der Konfiguration des Systems, hernach folgt eine "Idle"-Phase (funktionslose Endlosschleife).

2 Ergebnisse

In der heruntergeladenen Workbench wird ein neues C-Projekt angelegt, mit folgenden Schritten:

- Projektnamen vergeben
- Toolchain ‘Ac6 STM32 MCU GCC’ wählen
- 2x Next
- Unter Series ‘STM32F3’ & Board ‘STM32F3DISCOVERY’ wählen + Next
- Bei Firmware wird der Hardware Abstraction Layer (Cube HAL) ausgewählt
- Dessen Firmware wird anschließend per Knopfdruck heruntergeladen (im Schulnetz zur Laborzeit hat dies viel Zeit in Anspruch genommen)
- Des Weiteren empfiehlt es sich die Option ‚As sources in the application project‘ auszuwählen, da es in meiner Umgebung mit anderen Einstellungen zu Fehlern gekommen ist.

Zur vorigen Aufgabe bzw. worauf in der 2. Übung aufgebaut wird:

Zu allererst habe ich Methoden zum Togglen der drei von uns benötigten LEDs erstellt und im Header-File definiert, und diese per Delay beliebig lang leuchten lassen bzw. wieder abgeschaltet. **– control.c & control.h**

Im File **ampel.h** habe ich nach einer Empfehlung über das konkrete Vorgehen seitens des Professors alle States & Events festgelegt. In der Datei **ampelsteuerung.c** ist der Kern dieses Programms untergebracht: Die Implementierung der State Machine. Ich verstehe die State Machine als Quasi-Design Pattern, die Sinn macht und sehr einfach zu implementieren ist.

Ein kurzer Einblick in den Code:

```
            case FAHREN:

                  if (repr->zustand == GRUEN) {

                        led_gruen();

                        repr->zustand = GRUEN_BLINKEN;
                        repr->event = VORBEREITUNG_HALT;

                  } else {

                        led_off();

                        repr->event = FALSE;

                  }

                  break;
```

In Worten: Wenn die Ampel grün leuchtet, wird der Übergang zu Gelb, also *grün blinken* ausgelöst, ansonsten tritt ein Fehler auf.

In der Hauptklasse **main.c** wird lediglich die Funktion mit den Ampelparametern versorgt aufgerufen und in einer Endlosschleife ausgeführt.

Zur Konfiguration von **Doxygen**:

Per Klick auf das ‚@'- Symbol wird ein example.Doxyfile erstellt. Hierbei besonders wichtig ist der gegebene Namen, die Version des Projekts, das Format (HTML, LaTeX) der Dokumentation sowie Input- und Outputfolder.

Der **Timer Interrupt** wurde als inkrementelle Variable implementiert. In der Switch Case-Anweisung wird wie im vorigen Beispiel überprüft, welcher Zustand derzeit aktuell ist und anhand dieses Zustandes wird dann die definierte Methode ausgeführt. Erreicht der Timer einen bestimmten Wert, geht das Programm in den nächsten Zustand über.

In der **main**-Klasse wird das gesamte System geladen und die interne Clock aktualisiert. Die Zeit, die zwischen den Ticks vergeht wird hierbei angepasst. Nach der Ausführung von *led_init()* verbleibt das Programm in einer „Idle"-Endlosschleife, solange kein Fehler auftritt und das Programm beendet.

Nach einem erfolgreichen Flash-Vorgang ist das Ergebnis sichtbar: Die Ampelschaltung des vorherigen Beispiels ist erhalten geblieben, bei Druck auf den User-Button schaltet das Board in eine Nachtschaltung über, die LED blinkt gelb und wartet darauf, in den normalen Betrieb zurückzukehren.

Während der Implementierungsphase gab es einige Probleme mit Doxygen, da erst nach einer langen Recherche ein gutes Tutorial gefunden wurde. Des Weiteren blieb der User-Knopf lange ohne Funktion, bis dieser ordnungsgemäß initialisiert wurde, was anfangs nicht gelingen wollte