

---

## Klausur

Prüfungsfach: Embedded Systems 2  
Datum/Uhrzeit: 28. Januar 2016 / 14:30 Uhr  
Raum: J3.19  
Prüfer: Dr. Hubert Högl  
Dauer: 60 Minuten  
Hilfsmittel: keine

### Hinweise:

1. Diese Angabenblatt hat auch eine **Rückseite**. Bitte sofort überprüfen.
2. **Sie dürfen dieses Angabenblatt behalten**. Ihre Lösungen schreiben Sie bitte auf die separat ausgeteilten karierten Bögen.
3. Schreiben Sie bitte nicht mit Bleistift und nicht mit roter Farbe.

---

Viel Glück!

---

### Aufgabe 1 (4 Punkte)

Warum ist die **Embedded Programmierung schwieriger** als die Programmierung eines Desktop PC?

### Aufgabe 2 (4 Punkte)

Hier sind ein paar Fragen zur **GNU Toolchain** für ARM Controller.

- a) Wie nennt man die Kompilierung von ARM Programmen auf dem PC, der bekanntlich keinen ARM Prozessor hat?
- b) Wie kommt man zur Toolchain? Nennen Sie zwei mögliche Wege.
- c) Aus welchen wesentlichen Programmen und Bibliotheken besteht die GNU Toolchain?
- d) Wozu dient das Makefile?

### Aufgabe 3 (4 Punkte)

Wie sieht ein **Arbeitsplatz** für die Entwicklung von Programmen für Embedded Systems aus? Geben Sie in Stichpunkten die **nötigen Geräte** und auch die zu **installierenden Programme** auf dem Entwicklungsrechner an.

### Aufgabe 4 (13 Punkte)

Zwischen dem **Reset** und der Ausführung der **main()** Funktion wird der Mikrocontroller initialisiert. Der Quelltext ist in den Dateien **startupcode.c** und **startupcode.h**.

- (a) Welche Aufgaben werden im Startupcode nacheinander erledigt?
- (b) Was ist die **data section** eines Programmes und wie muss diese initialisiert werden?
- (c) Was ist die **bss section** eines Programmes und wie muss diese initialisiert werden?
- (d) Wozu wird folgendes Makro verwendet?

```
#define WEAK_ALIAS(a) __attribute__((weak, alias(#a)))
```

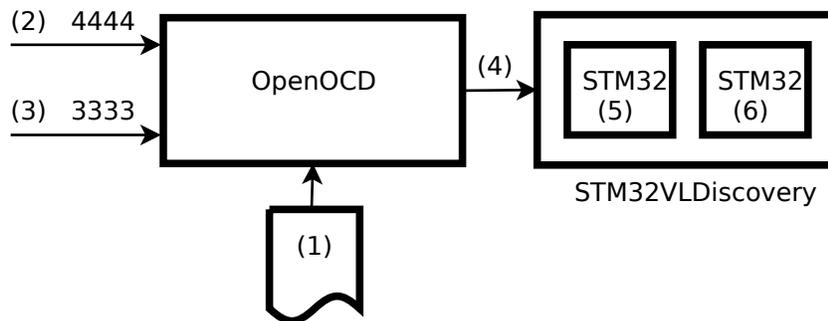
- (e) Wie steuert man, dass bestimmte Teile eines Programmes, z.B. die Interruptvektoren, an eine vordefinierte Adresse im Speicher kommen?
- (f) Was passiert nach der Rückkehr von `main()`?

Punkte: (a) 4, (b) 2, (c) 2, (d) 2, (e) 2, (f) 1

### Aufgabe 5 (6 Punkte)

Ordnen Sie die folgenden Begriffe den Zahlen (1) bis (6) im Diagramm zu:

JTAG Controller — USB Verbindung — Zielprozessor — Konfigurationsdatei —  
Telnet Port — GDB Debug-Port



### Aufgabe 6 (4 Punkte)

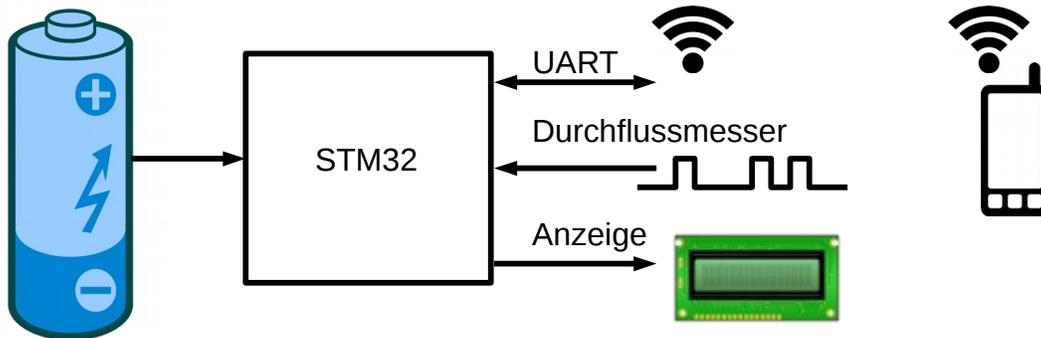
Was versteht man unter dem **SysTick** und wie nutzt man ihn in einem Programm?

### Aufgabe 7 (6 Punkte)

- (a) Beschreiben Sie die **zwei verschiedenen Arten**, wie wir im Praktikum eine zeitliche Verzögerung `delay(int msec)` programmiert haben. Skizzieren Sie die Programme mit Pseudo-Code.
- (b) Was macht der Prozessor während `delay()` ausgeführt wird?
- (c) Wenn man ein Echtzeit-Betriebssystem mit Tasks verwendet, gibt es auch eine `delay(int msec)` Funktion. Was macht in diesem Fall der Prozessor, wenn `delay()` ausgeführt wird?

### Aufgabe 8 (12 Punkte)

Sie möchten mit einem Baustein aus der STM32 Familie einen Wassermengenzähler aufbauen:



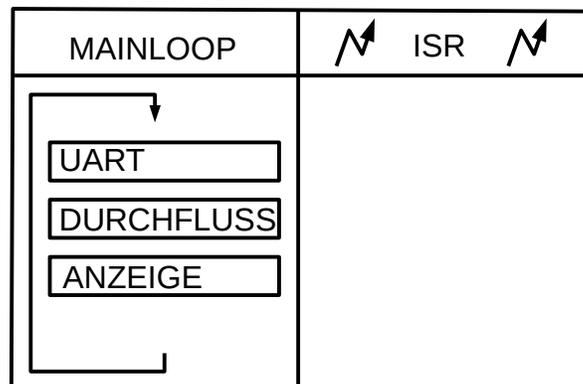
Im wesentlichen besteht das Gerät aus dem Mikrocontroller, der Batterie, einem Durchflusszähler und einer Anzeige.

Die Anzeige braucht nur zum Anzeigen eines neuen Wertes Strom, falls sich die Anzeige nicht ändert, braucht sie keine Energie. Die Anzeige wird nur **ein Mal pro Stunde** aktualisiert.

Der Durchflusszähler sendet digitale Impulse aus, deren Anzahl proportional zur Wassermenge sind.

Über eine Funkverbindung kann die serielle Schnittstelle (UART) des STM32 z.B. von einem Smartphone aus angesteuert werden, um den Zählerstand abzufragen.

Ein Anfänger in der Embedded-Programmierung programmiert eine „Mainloop“, die so schnell wie möglich durch alle drei Programmteile geht, so dass keine UART-Zeichen oder Impulse vom Durchflussmesser verloren gehen. Siehe folgende Programmstruktur:



(a) Erläutern Sie **alle Schwächen und Nachteile** dieser simplen Mainloop?

- (b) Nun stellen Sie Ihr Programm auf Interruptbetrieb um, so dass die Anwendung nur läuft, wenn der Durchflusszähler Impulse produziert oder der UART Daten empfängt. Zeichnen Sie eine neue Programmstruktur und erläutern Sie diese. Schieben Sie bestimmte Programmteile vom Mainloop Block in den ISR Block (ISR = Interrupt Service Routine).
- (c) Schliesslich möchten Sie so wenig Strom wie möglich verbrauchen, damit die Batterie mehrere Jahre zur Versorgung reicht. Dazu gibt es eine neue **SLEEP** Funktion. Wie muss der Programmablauf sein? Zeichnen Sie wieder eine neue Programmstruktur und erläutern Sie diese.

Punkte: (a) 4, (b) 4, (c) 4

### Aufgabe 9 (16 Punkte)

Füllen Sie folgende Tabelle zu verschiedenen Speicher-Technologien auf Ihrem Arbeitspapier aus.

	Funktionsprinzip	Flüchtig ja/nein	Typ. Grösse	Wo verwendet?
Statisches RAM (SRAM)				
Dynamisches RAM (DRAM)				
NOR Flash				
NAND Flash				