

Aufgabe 3: Reaktionstest

Aufgabenstellung: In dieser Aufgabe soll die eigene Reaktionszeit unter Zuhilfenahme einer Interrupt-Service-Routine (ISR) gemessen werden.

Das Spiel soll mit dem Drücken **beider Taster** gestartet werden können.

Nach Programmstart muss in einem **zufälligen Zeitintervall** (zwischen 1 und 5 Sekunden) die RGB-LED zufällig **grün** oder **rot** aufleuchten.

Daraufhin ist es Aufgabe des Nutzers schnellstmöglich den **rechten Taster** (LED ist grün) oder den **linken Taster** (LED ist rot) zu drücken.

Bei korrekter Eingabe wird nun die vergangene Zeit auf dem Display sowohl in dezimaler („MM:SS:MS“), als auch in binärer („SSSSSSS:MSMSMSM“) Form ausgegeben. Bei fehlerhafter Eingabe ist die Ausgabe stattdessen „Fehlerhafte Eingabe“.

Durch erneutes Drücken beider Taster kann eine neue Runde gestartet und das Display geleert werden.

Bei drücken der Taster in der Zeit zwischen Programmstart und Farbausgabe soll keine Reaktion erfolgen.

Vorgehensweise: 1. Öffnen Sie die Anwendung **Keil µVision 5** und öffnen über *Project und Open Project* die Projektdatei **Stopwatch.uvprojx**

2. Bearbeiten Sie lediglich die Methoden
void main(**void**),
void SystickInit(),

void SysTickIntHandler(**void**),
void GPIOIntHandler(**void**)
und
void printDecimalAsBinary(**decimal**)

in der Datei **Stopwatch.c**.

3. Aktivieren Sie die Interrupts für die Button und den SysTick in der **Startup.s** Datei. Beachten Sie dabei, dass es zusätzlich nötig ist, die SysTick-Periode so zu konfigurieren, dass diese nach einer Sekunde auslöst. Hierbei müssen Sie unter anderem den Systemtakt des Prozessors betrachten.

Tipp:

Bitte beachten Sie auch die Informationen, die Sie über das Board / den Prozessor erhalten haben (PDF Dokumente)

4. Implementieren Sie möglichst kurze und sinnvolle Anweisungen innerhalb der Interrupt-Service-Routinen. Beschränken Sie sich auf das Setzen von Flags (keine aufwändigen Schleifenaufrufe oder ähnliches!, if-Bedingungen oder switch-cases sind erlaubt)

5. Implementieren Sie die Spiellogik innerhalb der **main**-Methode.

6. Um den dezimalen Zeitwert im binären Datenformat auszugeben sollen Sie die Funktion printDecimalAsBinary(**decimal**) so implementieren, dass sie den eingehenden

Dezimalwert in binärer Form auf dem Display ausgibt.

Vorgabe:

Es dürfen keine fertigen Konvertierungsmethoden, wie beispielsweise itoa(), verwendet werden.

Benutzen Sie stattdessen den mathematischen Weg, um aus einer dezimalen Zahl ihre binäre Form zu errechnen.

7. Beachten Sie zu jedem Zeitpunkt die Vorgabe des Lösungsvideos im Stud.IP **„RST_Reaktionstest_LsgVid.mp4“**.

Abgabe:

1. Präsentieren Sie Ihr Programm im Labor einem Tutor
2. Markieren Sie den Quellcode mit Ihrem vollen Namen und Ihrer Matrikelnummer
3. Kommentieren Sie Ihre Codezeilen und laden diesen dann bei Stud.IP hoch (**Bitte auf korrekte Urheberrechtseinstellungen im Stud.IP achten**)