

Praktikumsaufgabe 2 - Senso-Spiel II

Aufgabe

In diesem Praktikum implementieren Sie nun das Senso Spiel aus Praktikum 1 auf dem RED-V Board. Nutzen Sie dafür die im Labor vorhandene Hardware (siehe Abbildung 1) oder die senso-Konfiguration im Simulator. Ihre Implementierung muss die, im Folgenden nochmal aufgeführten Anforderungen, erfüllen. Bitte lesen Sie sich alle Informationen gründlich durch.

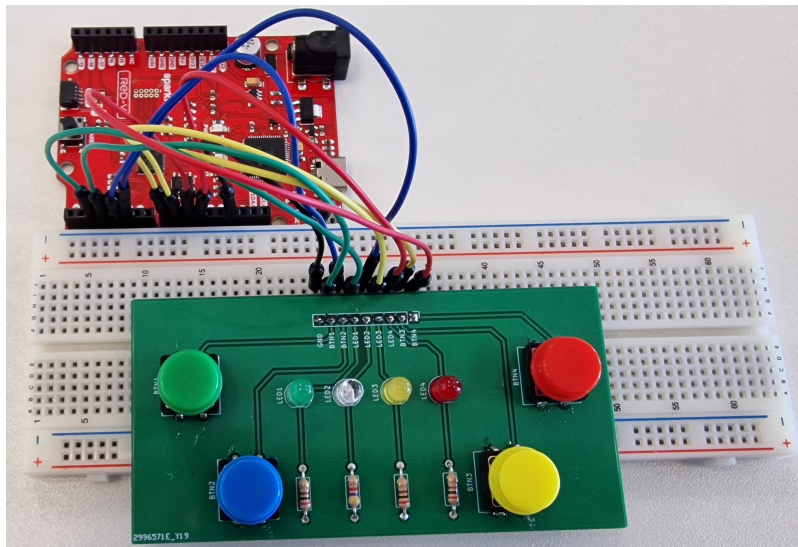


Abbildung 1: Senso Hardware mit vier LEDs und vier Buttons

Funktionsweise

Grundprinzip

Das Senso Spiel ist ein elektronisches Gedächtnisspiel, bei dem das Spiel eine zufällige farbliche Reihenfolge mit den LEDs vorgibt, die vom Spieler nachgeahmt werden muss. Für diese Nachahmung hat der Spieler nur eine limitierte Zeitspanne pro Taste. Wird eine falsche Taste/Reihenfolge eingegeben, ist das Spiel verloren und muss bei Level 1 gestartet werden. Wurde die Reihenfolge korrekt eingegeben, steigt man im Level auf. Mit jedem Levelaufstieg steigert sich der Schwierigkeitsgrad, indem die Anzeigegeschwindigkeit oder die Anzahl der Knöpfe gesteigert wird oder die limitierende Zeitspanne reduziert wird. Nach 10 Leveln ist das Spiel beendet und beginnt von vorne.

Detailbeschreibung

Das Programm startet im Bereitschaftsmodus.

Bereitschaftsmodus In diesem Modus laufen die LEDs in einem Muster durch. In diesem Muster blinken die LEDs für die Zeitspanne **T_SHORT** von links nach rechts auf. Durch einen Klick auf den grünen Knopf wird eine Partie beginnend bei Level 1 gestartet. Jedes Level besteht aus einer Vorführphase und einer Nachahmphase.

Vorführphase Zu Beginn jeder Vorführphase erfolgt eine LED-Sequenz, bei der zunächst die blaue und die gelbe LED für **T_SHORT** angehen und anschließend für **T_SHORT** wieder ausgehen. Anschließend werden nacheinander **n** LEDs angezeigt, wobei jede LED eine Zeitspanne **t** angezeigt wird. Zwischen den LEDs ist immer eine Zeitspanne **T_SHORT**, in der keine LED leuchtet. Die Reihenfolge der LEDs ist zufällig. Initial, also in Level 1, beträgt **n=3** und **t=T_LONG**. Nachdem ein Level vorgeführt wurde, gehen alle LEDs einmal an und nach der Zeitspanne **T_SHORT** aus. Dies markiert den Beginn der Nachahmphase.

Nachahmphase In der Nachahmphase muss der Spieler die korrekte Knopfreihefolge drücken. Dafür hat er pro Knopf die Zeitspanne **t** Zeit. Drückt er in dieser Zeitspanne keinen oder den falschen Knopf (bzw. mehrere Knöpfe), dann ist das Spiel verloren und zeigt die Verlorensequenz an. Drückt der Spieler innerhalb der Zeitspanne den korrekten Knopf, geht die entsprechende LED für die Zeit **T_SHORT** an und es werden keine neuen Knopfeingaben akzeptiert. Sobald die LED erlischt, wird der nächste Knopfdruck erwartet. Wird die Reihenfolge korrekt eingegeben, wechselt das Spiel in eine Zwischensequenz.

Verlorensequenz In dieser Sequenz blinken die grüne und rote LED 2 mal auf (**T_SHORT**). Anschließend wird das erreichte Level (also das vorherige) als binärkodierte Zahl dargestellt, wobei das LSB rechts (rote LED) ist. Nach der Zeitspanne **T_VERY_LONG** erlischt die Levelanzeige und das System wechselt in den Bereitschaftsmodus.

Zwischensequenz In der Zwischensequenz wird eine LED-Sequenz abgespielt. In dieser Sequenz leuchten abwechselnd die LED-Gruppen Grün-Gelb (leuchten gleichzeitig) und Blau-Rot (leuchten gleichzeitig) jeweils 2 mal für die Zeit **T_SHORT** auf. Zwischen den Gruppen ist keine Pause. Ist das finale Level nicht erreicht, wird der Schwierigkeitsgrad erhöht. Jedes Level wird **n** um 1 erhöht. **t** wird jedes zweite Level um 10% reduziert. Ist das Level > 10, dann ist das Spiel gewonnen und es erfolgt der Endmodus, andernfalls wird das nächste Level vorgemacht.

Endmodus Im Endmodus erfolgt eine LED-Sequenz, nach deren Anschluss das Spiel in den Bereitschaftsmodus wechselt. In dieser Leuchtsequenz blinken alle LEDs gleichzeitig im folgenden Rhythmus auf: **T-SHORT-T-LONG-T-SHORT-T_LONG**. Die Pausen zwischen den Blinken sind **T_SHORT**.

Technische Umsetzung

- Nutzen Sie für das Spiel eine zustandsbasierte Implementierung. Beachten Sie dabei die Detailbeschreibung der Funktionsweise.
- Nutzen Sie für das Abfragen der Buttons **keine** Interrupts, sondern nutzen Sie Polling und entprellen Sie die Knöpfe softwareseitig.
- Setzen Sie die Misra-Normen bei der Implementierung und evaluieren Sie ihren Code wie in der Vorlesung gezeigt und entfernen Sie die Warnungen durch Verbesserungen ihres Codes so weit wie möglich.
- Der abzugebende Code für das Spiel **muss** kompilierbar und in der Simulation sowie auf dem RED-V Board lauffähig sein. Kommentieren Sie notfalls fehlerhafte Codeabschnitte bzw. Funktionalitäten aus. Die Lauffähigkeit hat Vorrang.

Pinbelegung

Die einzelnen LEDs und Buttons der Senso Hardware sind wie folgt an das RED-V angeschlossen:

- Grüne LED => 2
- Blaue LED => 5
- Gelbe LED => 8
- Rote LED => 11
- Grüner Button => 3
- Blauer Button => 4
- Gelber Button => 9
- Roter Button => 10

Zeitbeschränkungen

Die Zeitbeschränkungen sind frei zu wählen, damit das Spiel ordnungsgemäß funktioniert sind die Zeitspannen wie folgt zu wählen:

- **T_SHORT** > 500 ms

- **T_LONG = 2* T_SHORT**
- **T_VERY_LONG = 2* T_LONG**

Abgabe für die Bewertung (Abgabe im Ilias als ZIP)

1. Der implementierte Code, der sowohl die Funktionsweise als auch die technische Umsetzung erfüllt.