Reaktionsspiel

**AP-Nummer:** 6

**AP-Titel:** Zeitmessung-Auswertung

**AP-Verantwortlicher:** Ioannis Christodoulakis, Thomas Ammann   
**Datum:** 29. Mai 2017

**Raumtemperatur:** 24.4 °C  
**Revision:** 01

Inhaltsverzeichnis

[1. Zusammenfassung 2](#_Toc483381013)

[2. Aufgabenstellung 2](#_Toc483381014)

[2.1 Schnittstelleninformationen 2](#_Toc483381015)

[3. Begriffserklärung 2](#_Toc483381016)

[4. Struktogramm 2](#_Toc483381017)

[5. Programm – Code 4](#_Toc483381018)

[6. Verwendete Programme 5](#_Toc483381019)

[7. Schlussfolgerung 5](#_Toc483381020)

[8. Identifikation 5](#_Toc483381021)

# 1. Zusammenfassung

Es wurde ein Software-Programm entwickelt, welches die Reaktionszeiten zweier Spieler vergleicht und auswertet. Derjenige Spieler mit der kleineren Reaktionszeit erhält einen Punkt.

# 2. Aufgabenstellung

Die Reaktionszeiten der Spieler werden verglichen. Der Spieler mit der kürzeren Reaktionszeit erhält einen Punkt.

Die Punktzahl der Spieler wird aus je einem Controllerausgang bestimmt. Das Punktzahl-Signal ist ein HIGH-Impuls, welcher auftritt, wenn die Punktzahl um einen Punkt erhöht werden soll.

2.1 Schnittstelleninformationen

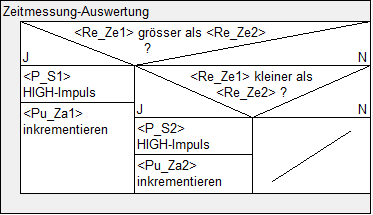
|  |  |
| --- | --- |
| Eingabe | Ausgabe |
| Reaktionszeit1 | Punktzahl-Signal1 |
| Reaktionszeit2 | Punktzahl-Signal2 |
|  | Punktzahl1 |
|  | Punktzahl2 |

# 3. Begriffserklärung

|  |  |
| --- | --- |
| Variable | Speicher-Name |
| Punktzahl1 | Pu\_Za1 |
| Punktzahl2 | Pu\_Za2 |
| Reaktionszeit1 | Re\_Ze1 |
| Reaktionszeit2 | Re\_Ze2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Ausgabe | Speicher-Name | Pin-Name | Pin |
| Punktzahl-Signal1 | Pu\_Si1 | P\_S1 | 6 |
| Punktzahl-Signal2 | Pu\_Si2 | P\_S2 | 7 |

# 4. Struktogramm



# 5. Programm–Code

Zunächst werden die Variablen definiert.

**int** Re\_Ze1, // Variablen

Re\_Ze2 ;

**char** Pu\_Za1 = 0,

Pu\_Za2 = 0 ;

Die Pin-Nummern der Ein- und Ausgänge sind den Pin-Namen gespeichert. Diese sind unveränderbar.

**const char** P\_S1 = 6, // Konstanten

P\_S2 = 7;

**setup()** wird einmalig beim Einschalten des Arduino ausgeführt. Dort definieren wir mit der Funktion **pinMode()**, ob die digitalen Pins Ein- oder Ausgänge sind.

**void setup()** { // I/O

**pinMode**(P\_S1, OUTPUT); // Ausgaenge

**pinMode**(P\_S2, OUTPUT);

}

**loop()** wird wiederholt ausgeführt. In der Funktion wird das Hauptprogramm ausgeführt.   
Die **if**-Verzweigung prüft, ob «Re\_Ze1» grösser als «Re\_Ze2» ist. Trifft dies zu wird ein Punktzahl-Signal ausgegeben und die Punktzahl des ersten Spielers wird inkrementiert. Die **else-if**-Verzweigung prüft, ob «Re\_Ze1» kleiner als «Re\_Ze2» ist. Trifft dies zu wird ein Punktzahl-Signal ausgegeben und die Punktzahl des zweiten Spielers wird inkrementiert.

**void loop()** { // Hauptprogramm

**if**(Re\_Ze1 > Re\_Ze2){ // Auswertung

**digitalWrite**(P\_S1, HIGH);

**delay**(10);

**digitalWrite**(P\_S1, LOW);

Pu\_Za2++;

}

**else if**(Re\_Ze1 < Re\_Ze2){

**digitalWrite**(P\_S2, HIGH);

**delay**(10);

**digitalWrite**(P\_S2, LOW);

Pu\_Za1++;

}

# 6. Verwendete Programme

* Arduino Software
* Struktogrammer

# 7. Schlussfolgerung

Das entwickelte Software-Programm erfüllt die Anforderungen der Aufgabenstellung  
(2. Aufgabenstellung/Seite 2).

# 8. Identifikation

Ort, Datum: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unterschrift: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Unterschrift: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Ioannis Christodoulakis Thomas Ammann