Reaktionsspiel

**AP-Nummer:** 5

**AP-Titel:** Zeitmessung

**AP-Verantwortlicher:** Ioannis Christodoulakis, Thomas Ammann   
**Datum:** 24. Mai 2017

**Raumtemperatur:** 24.3 °C  
**Revision:** 01

Inhaltsverzeichnis

[1. Zusammenfassung 2](#_Toc483391459)

[2. Aufgabenstellung 2](#_Toc483391460)

[2.1 Schnittstelleninformationen 2](#_Toc483391461)

[3. Begriffserklärung 2](#_Toc483391462)

[4. Struktogramm 3](#_Toc483391463)

[5. Programm – Code 3](#_Toc483391464)

[6.Verwendelte Programme 5](#_Toc483391465)

[7. Schlussfolgerung 5](#_Toc483391466)

[8. Identifikation 6](#_Toc483391467)

# 1. Zusammenfassung

Es wurde ein Software-Programm entwickelt, welches die Reaktionszeit der beiden Spieler misst.

# 2. Aufgabenstellung

Die Reaktionszeit wird zwischen dem optischen Zeitmessung-Start-Signal und dem Quittierungs-Signal gemessen.

Um die Reaktion richtig zu testen wird die Reaktionszeitmessung zufällig 1 bis 5 Sekundenverspätet gestartet.

2.1 Schnittstelleninformationen

|  |  |
| --- | --- |
| Eingabe | Ausgabe |
| Quittierungs-Signal1 | Reaktionszeit1 |
| Quittierungs-Signal2 | Reaktionszeit2 |
| Zeitmessungstart1 | Optische Meldung1 |
| Zeitmessungstart2 | Optische Meldung2 |
| EingangsSpieler1 |  |
| EingangsSpieler2 |  |
| Spieler1 |  |
| Spieler2 |  |

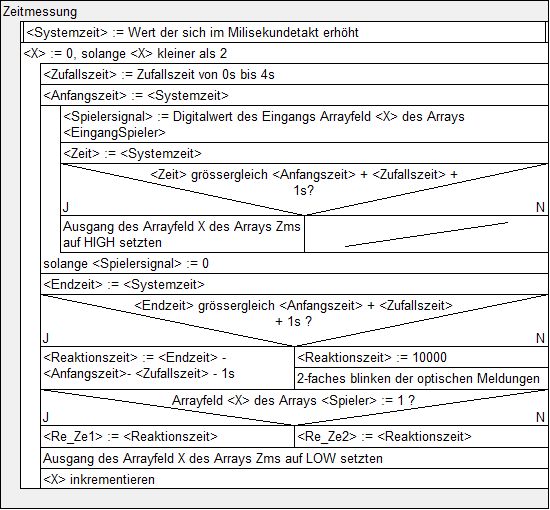
# 3. Begriffserklärung

|  |  |
| --- | --- |
| Variable | Speicher-Name |
| Spieler1 | Spieler[0] |
| Spieler2 | Spieler[1] |
| Zeitmessungstart1 | Zms[0] |
| Zeitmessungstart2 | Zms[1] |
| EingangSpieler1 | EingangSpieler[0] |
| EingangSpieler2 | EingangSpieler[1] |
| Reaktionszeit1 | Re\_Ze1 |
| Reaktionszeit2 | Re\_Ze2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Eingabe | Speicher-Name | Pin-Name | Pin |
| Quittieruns-Signal1 | Qu\_Si1 | Q\_S1 | 4 |
| Quittieruns-Signal2 | Qu\_Si2 | Q\_S2 | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Ausgabe | Speicher-Name | Pin-Name | Pin |
| Optische Meldung1 | Op\_Me1 | O\_M1 | 8 |
| Optische Meldung2 | Op\_Me2 | O\_M2 | 9 |

# 4. Struktogramm



# 5. Programm – Code

Zunächst werden die Variablen definiert.

**char** X,// Variablen

Spielersignal,

Zms[2], // Zeitmessungstart

Spieler[2];

**int** Zufallszeit,

Re\_Ze1,

Re\_Ze2;

**unsigned long** Anfangszeit, // Anfangszeit der Zeitmessung

Endzeit, // Endzeit der Zeitmessung

Zeit;

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

**const char** Q\_S1 = 4,// Konstanten

Q\_S2 = 5,

O\_M1 = 8,

O\_M2 = 9;

**setup()** wird einmalig beim Einschalten des Arduino ausgeführt. Dort definieren wir mit der Funktion **pinMode()**, ob die digitalen Pins Ein- oder Ausgänge sind. Mit der Funktion **randomSeed()** wird der Analogeingang definiert, welcher für die Erstellung des «zufälligen» Werts verwendet wird. Dieser Eingang muss logisch unbestimmt sein, darf also nicht angeschlossen sein.

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

**pinMode**(Q\_S1, INPUT); // Eingaenge

**pinMode**(Q\_S2, INPUT);

**pinMode**(O\_M1, OUTPUT); // Ausgaenge

**pinMode**(O\_M2, OUTPUT);

**randomSeed**(**analogRead**(0)); // Der analogInput 0 darf nicht angeschlossen werden.

}

**loop()** wird wiederholt ausgeführt. In der Funktion wird das Hauptprogramm ausgeführt.   
Die **for**-Schleife wird zweimal durchgeführt. Die Funktion **random()** erzeugt einen «zufälligen» Wert von 0 bis 4000. Die Funktion **millis()** erzeugt einen Wert, welcher jede Millisekunde inkrementiert wird. «Anfangszeit» entspricht dem Wert von **millis()**.

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

**for**(X = 0; X < 2; X++){ // Auswertung

Zufallszeit = **random**(4001);

Anfangszeit = **millis**();

«Zeit» entspricht dem aktuellen Wert von **millis()**. Die **if**-Verzweigung prüft, ob «Zeit» grössergleich «Anfangszeit» + «Zufallszeit» + 1000 entspricht. Trifft dies zu wird ein Zeitmessungsstart-Signal ausgegeben. Die **do-while**-Schleife wird wiederholt solange «Spielersignal» 0 entspricht. Danach wird der Wert von **millis()** «Endzeit» zugeordnet.

**do**{

Spielersignal = **digitalRead**(EingangSpieler[X]);

Zeit = **millis**();

if(Zeit >= Anfangszeit + Zufallszeit + 1000){

**digitalWrite**(Zms[X], HIGH);

}

}**while**(Spielersignal == 0);

Endzeit = **millis**();

Die **if**-Verzweigung prüft, ob «Endzeit» grössergleich «Anfangszeit» + «Zufallszeit» + 1000. Trifft dies zu wird die Reaktionszeit ausgerechnet und in «Reaktionszeit» gespeichert. Trifft der Vergleich nicht zu wird «Reaktionszeit» auf 10000 (10s) gesetzt. Dabei werden zwei optische Signale ausgegeben.

**if**(Endzeit >= Anfangszeit + Zufallszeit + 1000){

Reaktionszeit = Endzeit - Anfangszeit - Zufallszeit - 1000;

}

**else**{

Reaktionszeit = 10000;

**digitalWrite**(O\_M1, HIGH);

**digitalWrite**(O\_M2, HIGH);

**delay**(300);

**digitalWrite**(O\_M1, LOW);

**digitalWrite**(O\_M2, LOW);

**delay**(300);

**digitalWrite**(O\_M1, HIGH);

**digitalWrite**(O\_M2, HIGH);

**delay**(300);

**digitalWrite**(O\_M1, LOW);

**digitalWrite**(O\_M2, LOW);

}

Die **if**-Verzweigung prüft welchem Spieler «Reaktionszeit» zuzuordnen ist und speichert diese in die entsprechende Variable. Danach wird das Zeitmessungsstart-Signal auf logisch LOW gesetzt.

**if**(Spieler[X] = 1){

Re\_Ze1 = Reaktionszeit;

}

**else**{

Re\_Ze2 = Reaktionszeit;

}

**digitalWrite**(Zms[X], LOW);

}

}

# 6.Verwendelte Programme

* Arduino Software
* Struktogrammer

# 7. Schlussfolgerung

Das entwickelte Software-Programm erfüllt die Anforderungen der Aufgabenstellung.

# 8. Identifikation

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

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