Reaktionsspiel

**AP-Nummer:** 7

**AP-Titel:** Spielende

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

**Raumtemperatur:** 24.2 °C  
**Revision:** 01

Inhaltsverzeichnis

[1. Zusammenfassung 2](#_Toc483380632)

[2. Aufgabenstellung 2](#_Toc483380633)

[2.1 Schnittstelleninformationen 2](#_Toc483380634)

[3. Begriffsverzeichnis 2](#_Toc483380635)

[4. Struktogramm 2](#_Toc483380636)

[5. Programm – Code 3](#_Toc483380637)

[6. Verwendete Programme 4](#_Toc483380638)

[7. Schlussfolgerung 4](#_Toc483380639)

[8. Identifikation 4](#_Toc483380640)

# 1. Zusammenfassung

Es wurde ein Software-Programm entwickelt, welches das Spielende auswertet. Das Programm gibt eine optische Meldung aus, wenn ein Spieler neun Punkte erhält.

# 2. Aufgabenstellung

Das Spiel ist beendet, sobald ein Spieler neun Punkte erhalten hat. Die optische Meldung «Gewinner» des Spielers geht an.

2.1 Schnittstelleninformationen  
Die optische Meldung «Gewinner» soll leuchten. Wenn die optische Meldung «Gewinner» ausgegeben wird soll ist Spielende abgeschlossen.

|  |  |
| --- | --- |
| Eingabe | Ausgabe |
| Punktzahl-Signal1 | Optische Meldung1 |
| Punktzahl-Signal2 | Optische Meldung2 |

# 3. Begriffsverzeichnis

|  |  |
| --- | --- |
| Variable | Speicher-Name |
| Punktzahl1 | Pu\_Za1 |
| Punktzahl2 | Pu\_Za2 |

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

# 4. Struktogramm

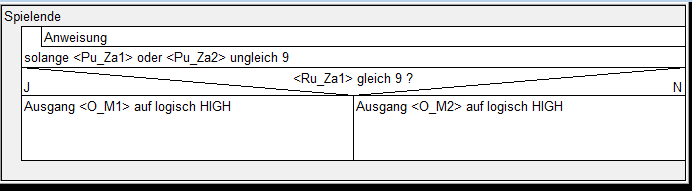


Abbildung , Struktogramm

# 5. Programm–Code

In den Variablen «Pu\_Za1» und «Pu\_Za1» werden die Punktzahlen der Spieler gespeichert. Da diese minimal 0 und maximal 9 sein können genügen Variablen vom Datentyp **char**.

**char** Pu\_Za1, // Variablen

Pu\_Za2;

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

**const char** O\_M1 = 8, // Konstanten

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.

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

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

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

}

**loop()** wird wiederholt ausgeführt. In der Funktion wird das Hauptprogramm ausgeführt.   
Die **do-while**-Schleife überprüft, ob «Pu-Za1» oder «Pu\_Za2» 9 entspricht. Trifft dies zu, fragt die **if**-Verzweigung, welche der beiden Variablen 9 entspricht. Danach wird eine optische Meldung ausgegeben.

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

**do**{ // Auswertung

// Spielerauswahl, Zeitmessung, Zeitmessung-Auswertung

}**while**(Pu\_Za1 == 9 || Pu\_Za2 == 9);

**if**(Pu\_Za1 == 9){

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

}

**else**{

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

}

}

# 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