|  |  |  |
| --- | --- | --- |
| **Übung Nr. 1**  Jahrgang: BHME20  Gruppe: 3D |  | **Protokollabgabe**  Solldatum: 19.10.2023  Istdatum:  Note: |

PROTOKOLL

**Thema: Arduino - Wiederholung**

Tag: 12.10.2023

Zeit: 10:45-13:15 Uhr

Ort: HTBLA Kaindorf | PRR Labor

Anwesend: Traußnigg Jan, Ursnik Iwana

Abwesend: -

Schriftführer:

Betreuer: Dipl.-Ing. Steiner Walter

**Aufgabenstellung**

Schreiben eines Programmes, welches Ziffern auf der 7-Segmentanzeige darstellt. Erweiterung dieses Programmes laut Aufgabenstellung 7.

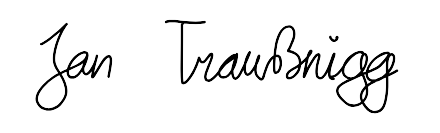
**Resümee**

In der heutigen Einheit haben wir die Grundlagen der Arduino-Programmierung in der Übung 7 wiederholt und uns wieder mit der Steuerung des Arduino UNO vertraut gemacht.

**Unterschriften**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Iwana Ursnik



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jan Traußnigg

[1 Zeitplan 2](#_Toc148370342)

[2 Thema 2](#_Toc148370343)

[2.1 Aufgabenstellung 2](#_Toc148370344)

[2.2 Verwendete Geräte und Hilfsmittel 2](#_Toc148370345)

[2.3 Vorgangsweise 2](#_Toc148370346)

[3 Messergebnisse 2](#_Toc148370347)

# Zeitplan

10:45 – 11:35 Wiederholung – Übung 7a)

11:35 – 12:25 Unterbrechung Labor

12:25 – 13:15 Wiederholung – Übung 7b)

# Thema

## Aufgabenstellung

**Übung 7**

1. Schreiben Sie ein Programm, das im Sekundentakt die Ziffern 0 bis 9 auf der 7-Segmentanzeige darstellt. Nach der Ziffer 9 soll mit der Ziffer 0 fortgesetzt werden.
2. Schreiben Sie eine Funktion *void setSeg7(uint8\_t nr)*, die die Zahl auf der 7 Segmentanzeige ausgibt
3. Ändern Sie das Programm so ab, dass zusätzlich zum Zählen Übung 5 implementiert wird, also die Nummer der gedrückten Taste auf den Leds abgebildet wird.

Ein Bild, das Elektronik, Schaltung, Elektronisches Bauteil, Text enthält.

Automatisch generierte Beschreibung

Abbildung 1: Angabe Arduino Mega Shield   
(Pin-Nummern von dieser Abbildung ablesbar)

## Verwendete Geräte und Hilfsmittel

* Arduino IDE (privat-Laptop oder Schul-PC)
* Arduino UNO

## Vorgangsweise

### Übung 7) 1)

void setup()

{

  for(int n=62; n<70; n++)

  {

    pinMode(n, OUTPUT);

  }

//Segmente der 7-Segmentanzeige auf Output setzen

}

int p = 0;

void loop() {

  for(int i=62; i<70; i++)

  {

    digitalWrite(i, HIGH);

  }

//alle Segmente “ausschalten”

  switch(p)

  {

    case 0: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            digitalWrite(66, LOW);

            digitalWrite(67, LOW);

            break;

    case 1: digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            break;

    case 2: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(68, LOW);

            digitalWrite(66, LOW);

            digitalWrite(65, LOW);

            break;

    case 3: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            digitalWrite(68, LOW);

            break;

    case 4: digitalWrite(67, LOW);

            digitalWrite(63, LOW);

            digitalWrite(68, LOW);

            digitalWrite(64, LOW);

            break;

    case 5: digitalWrite(62, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            break;

    case 6: digitalWrite(62, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            digitalWrite(66, LOW);

            digitalWrite(63, LOW);

            digitalWrite(65, LOW);

            break;

    case 7: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            break;

    case 8: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            digitalWrite(66, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            break;

    case 9: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            break;

  }

  delay(1000);

  if(p==10)

  p=0;

  p++;

}

### Übung 7) 2)

void setup() {

  // put your setup code here, to run once:

  for(int n=62; n<70; n++)

  {

    pinMode(n, OUTPUT);

  }

}

void setSeg(uint8\_t nr)

{

   switch(nr)

  {

    case 0: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            digitalWrite(66, LOW);

            digitalWrite(67, LOW);

            break;

    case 1: digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            break;

    case 2: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(68, LOW);

            digitalWrite(66, LOW);

            digitalWrite(65, LOW);

            break;

    case 3: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            digitalWrite(68, LOW);

            break;

    case 4: digitalWrite(67, LOW);

            digitalWrite(63, LOW);

            digitalWrite(68, LOW);

            digitalWrite(64, LOW);

            break;

    case 5: digitalWrite(62, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            break;

    case 6: digitalWrite(62, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            digitalWrite(66, LOW);

            digitalWrite(63, LOW);

            digitalWrite(65, LOW);

            break;

    case 7: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            break;

    case 8: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(65, LOW);

            digitalWrite(66, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            break;

    case 9: digitalWrite(62, LOW);

            digitalWrite(63, LOW);

            digitalWrite(64, LOW);

            digitalWrite(67, LOW);

            digitalWrite(68, LOW);

            break;

  }

  delay(1000);

}

void loop() {

  for(int i=62; i<70; i++)

  {

    digitalWrite(i, HIGH);

  }

  setSeg(7);

}

### Überarbeitung der Übung 7) 1) & Übung 7) 2)

Da Übung 7) 1) ziemlich unübersichtlich und primitiv programmiert ist, erweitern wir diesen nun so dass er übersichtlicher und auch leichter abänderbar machen (für die Pins der Segmente)

const int segmentPins[] =

{62, 63, 64, 65, 66, 67, 68, 69};

const int numbers[] = {

  B11000000,  // 0

  B11111001,  // 1

  B10100100,  // 2

  B10110000,  // 3

  B10011001,  // 4

  B10010010,  // 5

  B10000010,  // 6

  B11111000,  // 7

  B10000000,  // 8

  B10010000   // 9

};

void setup() {

  for (int i = 0; i < 8; i++) {

    pinMode(segmentPins[i], OUTPUT);

  }

}

void loop() {

  for (int i = 0; i <= 9; i++) {

    for (int j = 0; j < 8; j++) {

      digitalWrite(segmentPins[j],

      bitRead(numbers[i], j));

    }

    delay(1000);

  }

}

Erklärung: Im Feld numbers[] wird binär gesetzt welches Segment leuchten soll und welches nicht. 0 steht dabei für die Segmente, die leuchten sollen. In loop () wird dann mit bitRead gelesen, welchen Zustand der Pin [j] haben soll, wobei wichtig zu beachten ist dass die binäre Zahl dabei von hinten nach vorne gelesen wird und die letzte Stelle immer 1 ist, weil das die Punkte auf der Anzeige sind die nicht leuchten sollen.

Im gleichen Stil wurde die Übung 7) 2) anschließend auch verändert.

const int segmentPins[] =

{62, 63, 64, 65, 66, 67, 68, 69};

const int numbers[] = {

  B11000000,  // 0

  B11111001,  // 1

  B10100100,  // 2

  B10110000,  // 3

  B10011001,  // 4

  B10010010,  // 5

  B10000010,  // 6

  B11111000,  // 7

  B10000000,  // 8

  B10010000   // 9

};

void setSeg7(uint8\_t nr);

void setup() {

  for (int i = 0; i < 8; i++) {

    pinMode(segmentPins[i], OUTPUT);

  }

}

void loop() {

  for (int i = 1; i <= 9; i++) {

    setSeg7(i);

    delay(1000);

  }

}

void setSeg7(uint8\_t nr)

{

  for (int j = 0; j < 8; j++) {

    digitalWrite(segmentPins[j],

    bitRead(numbers[nr], j));

  }

}

### Übung 7) 3)

Da die Angabe falsch interpretiert wurde, wird der überarbeitete Code nun fälschlicherweise so erweitert, dass die Taste, die gedrückt wird am 7-Segment Display angezeigt wird. Eigentlich sollte die Anzeige dauerhaft die Zahlen weiterhin anzeigen und von der Taste, die man drückt, soll die zugehörige LED aufleuchten. Dies wird in der nächsten Übung realisiert.

const int segmentPins[] = {62, 63, 64, 65, 66, 67, 68, 69};

const int buttonPins[] = {9, 8, 7, 6, 38, 39, 40, 41};

const int numbers[] = {

  B11000000,  // 0

  B11111001,  // 1

  B10100100,  // 2

  B10110000,  // 3

  B10011001,  // 4

  B10010010,  // 5

  B10000010,  // 6

  B11111000,  // 7

  B10000000,  // 8

  B10010000   // 9

};

void setSeg7(uint8\_t nr);

void setup() {

  for (int i = 0; i < 8; i++) {

    pinMode(segmentPins[i], OUTPUT);

  }

  for (int i = 0; i < 8; i++) {

    pinMode(buttonPins[i], INPUT\_PULLUP);

  }

}

int buttonPressed;

void loop()

{

  buttonPressed = -1;

  for (int i = 0; i < 8; i++) {

    if (digitalRead(buttonPins[i]) == LOW) {

      buttonPressed = i;

      break;

    }

  }

  if (buttonPressed != -1) {

    setSeg7(buttonPressed);

  }

}

void setSeg7(uint8\_t nr)

{

  for (int j = 0; j < 8; j++) {

    digitalWrite(segmentPins[j],

    bitRead(numbers[nr], j));

  }

}

# Messergebnisse

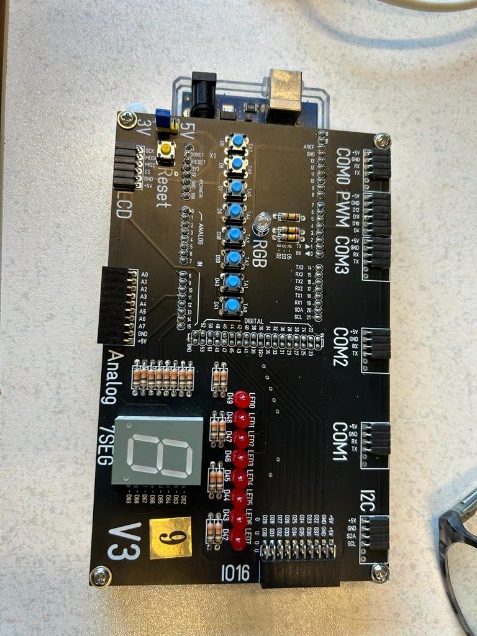


Abbildung 2: Arduino Mega Shield   
(Foto von Kollegin Anna-Aurora Schreiner)