IAI – Dokumentation: LED Stab

Basierend auf dem Konzept des POV (="Persistence Of Vision") Stick von Amanda Ghassaei. Gefunden auf der Website von instructables.com

LINK zur Anleitung:
http://www.instructables.com/id/Persistence-of-Vision-Wand/?ALLSTEPS

Extras sind:
Potentiometer, der die Blinkgeschwindigkeit reguliert
Button, der zwischen zwei Texten wechselt

LINK zu den Dateien:
https://github.com/aziralc/IAI---LED-Stab.git

Video (30s):
https://www.youtube.com/watch?v=G3mAqPA_sjQ&feature=youtu.be

Materialien:
Arduino Uno R3
Proto Shield
18x LEDs
18x 100Ohm Widerstände
Verbindungskabel
Wippschalter
Batterieclip
9V Batterie
Potentiometer
Druckknopf

Holzstab (aus dem Bauhaus)

Wie bin ich vorgegangen?

Zunächst baute ich das gesamte Konstrukt auf einer Steckplatine auf, um es auszutesten. Nachdem alles funktionierte, wie es in der Anleitung steht, überlegte ich mir als nächstes wie der Potentiometer eingefügt werden soll.

Dazu habe ich mir das Beispiel „AnalogReadSerial“ von Arduino angeschaut, um zu verstehen wie ich den Potentiometer an das Arduino anstecken und im Code ansprechen sollte.
Link: http://arduino.cc/en/Tutorial/AnalogReadSerial

Die nächste Hürde kam als ich den Code vom AnalogReadSerial zum das Basiskonzept hinzufügen wollte: Wie werden eigentlich die einzelnen Pins im Code angesteuert?

Nach langem Suchen gab es glücklicherweise ein Kommentar im Code, welches das Ansteuern der einzelnen Pins kurz erläutert. Um es kurz zu erklären: bestimmte Pins werden in 3 Ports (B, C und D) zusammengefasst und können somit als gesamtes oder teilweise angesprochen werden.

Ein Beispiel: DDRD
Port D sind die digitalen Pins von 0 bis 7. DDR (Data Direction Register) Port Register können sowohl beschrieben als auch gelesen werden. Um alle Pins von DDRD anzusprechen schreibt man:

```
DDRD = 0xFF;//port d- digital pins 0-7
```

Wichtig zu beachten ist der Hexadezimal Code, um zu wissen wie viele Pins des jeweiligen Ports angesprochen werden.

Dementsprechend habe ich zwei Pins für den Potentiometer (Pin A0) und den Button (Pin 13) frei gelassen.

```
  //port/pin assignments- set all pins to output
 DDRB = 0x1F;//port b- digital pins 8-13
 DDRC = 0x3E;//port c- anlog pins 0-5
 DDRD = 0xFF;//port d- digital pins 0-7
```

LINK für nähere Informationen:
http://www.arduino.cc/en/Reference/PortManipulation

Um die Refreshrate mittels des Potentiometer anzusteuern, brauchte es dazu nur folgende zwei Codezeilen im loop() :

```
int sensorValue = analogRead(A0);
delay(sensorValue);
```

Der empfangene Wert des Potentiometer (von 0 bis 1024) wird als refreshrate generiert.

Als nächstes kam der Wechsel zwischen zwei Texten mittels eines Buttons.

Hierfür nahm ich das Arduino Code Beispiel „Debounce“ als Basis und erweiterte es entsprechend.
LINK: http://arduino.cc/en/Tutorial/Debounce

Dazu wurden folgende Variablen und Werte global deklariert:

```
String text1 = "GUTEN";
String text2 = "APPETIT";
String povtext = text1;

int switchPin = 13;
int buttonState;
int lastButtonState = LOW;
long lastDebounceTime = 0;
long debounceDelay = 50;
```

im loop()

```
int reading = digitalRead(switchPin);

if (reading != lastButtonState) {
        lastDebounceTime = millis();
}

if ((millis() - lastDebounceTime) > debounceDelay) {
        if (reading != buttonState){
                buttonState = reading;
                if (buttonState == LOW && povtext.equals(text1)){
                        povtext = text2;
                }
                else if (buttonState == LOW && povtext.equals(text2)){
                povtext = text1;
                }
        }
}

lastButtonState = reading;
```

Sobald der Button gedrückt wird, wird der Wert im reading gespeichert. Der lastButtonState verändert sich und der aktuelle Zeitwert wird im lastDebounceTime gespeichert. Wenn die Differenz zwischen dem millis() und der lastDebounceTime größer ist als der debounceDelay, wird des Weiteren überprüft was der aktuell angezeigte Text ist. Sobald der Button wieder los gelassen wird, ändert sich der Text.

Der debounceDelay ist dazu da, um zu vergewissern, dass das Signal durch Drücken passiert ist und nicht durch eine ungewollte Störung.