

LED Bank

Manuel Rickli, Lukas Stöckli, Michael Plüss

25. Januar 2018

1 Einleitung

Das Ziel unserer Gruppe war, eine LED Bank zu bauen. Diese besteht aus 48x8 grünen LEDs, welche als Pixel fungieren. Die LED Bank soll sich mit dem Internet verbinden können und Pixel Daten von dort entnehmen und anzeigen können. Was auch möglich sein soll, ist, einfach kurze Texte anzuzeigen und von rechts nach links scrollen zu lassen.

Die Vision ist, diese LED Bank im Zimmer aufhängen oder hinstellen zu können und darauf verschiedenste Informationen darstellen zu können. Ursprünglich waren Anwendungen wie eine Uhr oder eine Wetterangabe geplant, oder sogar kleine 48x8 Pixel Filmchen.

Es gibt mehrere Gründe, weshalb wir uns vorrangig für eine zweidimensionale LED Konstellation entschieden haben, anstatt für eine dreidimensionale. Einerseits können wir so gewohnt konventionell Text darstellen und zweitens versprochen wir uns davon eine wesentlich einfachere Verdrahtung. Das Hauptproblem so viele LEDs überhaupt mit dem Arduino anzusteuern haben wir so aber nicht umgangen.

2 Hauptteil

2.1 Umsetzung

2.1.1 Display

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! TODO!!!!

(Bild)

Die 384 LEDs befinden sich verteilt auf zwei Steckbrettern. Die acht Reihen der Bank werden direkt vom Arduino angesprochen. Die 48 Kolonnen steuern wir über drei Pins des Arduinos mittels Shift Registern. Ein Shift Register hat jeweils acht Output Pins, weshalb wir insgesamt sechs Shift Register benutzen, welche in Reihe geschaltet sind.

Wir erreichen die Darstellung eines komplexen Pixelmusters, indem wir jeweils einen Pin bei den acht Reihen einschalten und die anderen ausschalten und dann die Pixeldaten der Zeile in die Shift Register einfüttern. So wird jeweils eine horizontale Linie dargestellt. Diese lassen wir einen kurzen Moment leuchten und gehen dann zur nächsten Zeile über. Wenn wir alle Zeilen dargestellt haben,

haben wir ein Frame 'gezeichnet'. Mit einer genug schnellen Taktrate ist das so entstehende Flimmern auch kein grosses Problem mehr.

2.1.2 Code

Der Code für das Arduino haben wir in der Arduino IDE geschrieben und bildet die Kontrolle über den Display, sowie die Kommunikationsstelle zum Internet. Um Texte darstellen zu können haben wir viele Bitmuster für die einzelnen Buchstaben einprogrammiert, welche auf unsere Bedürfnisse angepasst sind.

2.1.3 Webseite

Die Webseite operiert auf einem Laptop via einem Python Skript. Dieses Skript ist ein einfacher Http Request Handler und dient eigentlich nur dazu, dem Arduino den Zugriff auf Dateien zu ermöglichen, in welchen Bitmuster gespeichert sind.

2.2 geänderte Ziele

Das Ethernet Shield für das Arduino Uno, welches wir benutzen um Daten aus dem Internet zu empfangen, kann leider nur mittels Http, nicht aber mittels Https kommunizieren. Dies hatte zur Folge, dass wir unseren Serverspace nicht nutzen konnten, da dieser auf dem Https Standard funktioniert. Die Lösung war ein privater Python Server auf einem Laptop laufen zu lassen, welches Http Requests entgegennehmen kann. Dies schränkt natürlich die Nutzung der Internetfunktion ein, da der Laptop dafür ständig angeschaltet sein muss.

2.3 technische Details

2.3.1 Display

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! TODO!!!!

2.3.2 Code

Der Code ist strukturiert in verschiedene Dateien, welche auch als Module dienen. Diese Module sind: "main", "internetInterface", "ledInterface" und "ledHardwareControl".

Das "main" Modul macht nicht viel mehr als die anderen Module zu initialisieren und die Program Loop zu steuern.

Das Modul "internetInterface" kümmert sich um den Aufbau einer Verbindung zur gefragten Internetseite und um die Interpretation der Antwort als Bitmuster, welches in den internen Grafikspeicher geladen wird.

Das "ledInterface" Modul enthält den Grafikspeicher, welcher aus einem 48 Felder grossen char Array besteht, mit welchem die Lichtzustände der Pixel gespeichert werden. Ausserdem bietet es viele Komfortfunktionen für den Zugriff auf

gegebenen Grafikspeicher und die Manipulation dessen.

Das Modul "ledHardwareControl" kümmert sich um die Ansprechung der Output Pins des Arduinos. Es benutzt dazu die Daten aus dem "ledInterface" Modul um zu entscheiden, welche Leitungen an- und welche abgeschaltet werden sollen.

2.3.3 Webseite

Auf der Webseite liegen Dateien, welche vom Arduino ausgelesen werden können. Diese Dateien sollten im .txt Dateiformat vorliegen und mit dem UTF-8 Encoding beschrieben sein. Dabei kann das Arduino mit vier verschiedenen Arten der Dateiweitergabe umgehen:

- 1) In der Datei liegt ein Text vor. Dieser wird dann auf das Display geschrieben.
- 2) Die Characters in der Datei werden als Bytes interpretiert. Das Display füllt sich dann Spalte für Spalte mit den Bitmustern, die den jeweiligen Bits in den Bytes (Character) entsprechen.
- 3+4) Die Datei ist gefüllt mit Zeilen aus Nullen und Einsen. Nullen entsprechen ausgeschalteten Pixeln und Einsen eingeschalteten. Diese Datei kann entweder horizontal (x=48, y=8) oder vertikal (x=8, y=48) orientiert sein. Daraus entstehen die zwei unterschiedlichen Formate.

3 Zusammenfassung

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! TODO!!!!

4 Quellen

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! TODO!!!!