
LeDi

Release 0.1

LeDi Team

18.12.2022

Inhaltsverzeichnis

1	LeDi.Docs	3
2	LeDi.Server	5
3	LeDi.WebClient	7
4	LeDi.Display	9
4.1	Inhalt	9

LeDi ist ein Open Source Software und Hardware Projekt, welches dazu dient Spiele, vornehmlich sportlicher Natur, mit einer Anzeigetafel zu begleiten und gleichzeitig das Spielmanagement an sich zu übernehmen (Zeitmessung, Spielerverwaltung uvm.).

Projekt auf GitHub: [LeDi](#)

Das Projekt unterteilt sich in fünf Teile:

KAPITEL 1

LeDi.Docs

Dieses Projekt stellt die Dokumentation des Projektes da. Hier ist alles zu finden was benötigt wird, um das Projekt nachzubauen und selbst zu betreiben sowie Überlegungen und Hintergründe. Diese Seite ist der Hauptbestandteil dieses Projekts. Auch die Installation und Konfiguration eines Raspberry Pis zum Betrieb der Softwarekomponenten ist Teil dieses Projektes. Außerdem beinhaltet es sowohl den Bau eines Gehäuses mit den LEDs und Netzteilen sowie die Gestaltung und den Zusammenbau der Elektronik. Letzteres sollte auf jeden Fall von einem Elektriker begleitet werden um Gefahr für Leib und Leben sowohl bei der Konstruktion als auch beim späteren Betrieb ausschließen zu können.

KAPITEL 2

LeDi.Server

Dies ist die Softwarekomponente, welche die Spielsteuerung übernimmt sowie eine API bereitstellt um den anderen Softwarekomponenten via REST API eine Datenbasis bereit zu stellen. Nur LeDi.Server hält Daten permanent vor und ist das führende und zentrale Glied in der Kette. Die Entwicklung geschieht in C# auf Basis von .Net 6.0.

KAPITEL 3

LeDi.WebClient

Dies ist die Softwarekomponente, welche den Schiedsrichtern und anderen Interessierten eine Oberfläche bietet, mit der die zentrale Komponente gesteuert werden kann. Die Entwicklung geschieht in C#, HTML, CSS und JavaScript auf Basis von Microsoft Blazor (Server) und .Net 6.0.

Dies ist die Softwarekomponente, welche ein LED Display ansteuert, welches auf Basis von WS2812B LEDs erstellt wurde. Gesteuert wird dies ebenfalls über die zentrale Softwarekomponente. Die Entwicklung geschieht in C# auf Basis von .Net 6.0

Bemerkung: Dieses Projekt und alle Unterprojekte sind noch in Entwicklung, dennoch gibt es bereits eine erste funktionsfähige und nutzbare Version. Es sind lediglich noch nicht alle angedachten Funktionen vorhanden. Siehe hierzu auch das Github Projekt.

4.1 Inhalt

4.1.1 Idee

Die Idee zu LeDi entstand bei einem Unterwasserhockey Turnier, bei dem eine Anzeigetafel mit Hilfe eines normalen Fernsehers umgesetzt wurde. Dies brachte jedoch drei erhebliche Probleme mit sich:

1. Wenn die Sonne auf den Bildschirm schien, war dieser nicht mehr lesbar.
2. Wenn man aus einem halbwegs spitzen Winkel auf den Bildschirm geguckt hat, war nichts zu erkennen.
3. Es gab Reflexionen von hellen Wänden und anderen Objekten, welche Teile des Bildschirms nicht mehr lesbar machten.

Es entstand die Idee, dass es doch möglich sein muss mit relativ wenig Aufwand ein Display auf Basis von LEDs zu erstellen, welches mit den drei genannten Punkten auf Grund der höheren Leuchtstärke keine Probleme haben sollte.

So wurde überlegt und recherchiert. Im Ergebniss kam heraus, dass wir auf Basis von WS2812B LED Strips ein ca. 1 Meter breites Display erschaffen wollen. Die Steuerung kann ein Raspberry Pi übernehmen, sodass hier kostengünstig und gleichzeitig flexibel gearbeitet werden kann. Das Sammeln von gewünschten Eigenschaften brachte folgende Liste hervor:

- Transportabel => nicht zu groß, nicht zu schwer

- Auch aus der Entfernung noch erkennbar => nicht zu klein
- Spritzwasser- und Regengeschützt => Keine Probleme mit Regen oder Spritzwasser bei Wassersportarten
- Keine 230V im Display selbst, da dies ggf. in Wassernähe aufgebaut wird und im schlimmsten Fall ins Wasser fallen kann => Trennung von Display und Schaltung
- Wasserdichte Netzteile (siehe Spritzwasser- und Regengeschützt)
- Bedienung via Browser von jedem beliebigen Gerät
- Verbindung via WLAN möglichst aufgespannt durch den Raspberry Pi selbst

Anmerkung: Siehe Materialliste für mehr Details zu den Komponenten.

So wurde dann eine erste Idee und ein erster Prototyp entwickelt, der die Machbarkeit belegen sollte. Hierzu wurden die LED Strips, 60 Stück á 96 LEDs, auf eine Polystyrol-Platte mit Sanitär silikon geklebt. Polystyrol ist leicht und robust (in entsprechender Dicke). Alle Anschlüsse der Strips wurden anschließend mit Kabeln verlötet und mit den 20 (!) Netzteilen verbunden. Wir haben uns dazu entschieden Wasserdichte Netzteile zu verwenden, welche allerdings auf Grund fehlender Belüftung in der Leistung reduziert sind. Daher benötigen wir ein paar mehr Netzteile und nicht nur eines. Wenn das Display ohnehin nur im Trockenen genutzt wird, kann hier auch ein (oder mehrere) offenes, nicht wasserdichtes, Netzteil genutzt werden. Nachdem auch die Netzteile angeschlossen und über extra Sicherungen mit dem Stromnetz verbunden waren, konnten wir den Prototypen testen. Hier stellte sich heraus, dass die Netzteile nacheinander mit Strom versorgt werden müssen, da der Einschaltstrom aller Netzteile zur gleichen Zeit zu hoch war und die Hauptsicherung (FI Schutzschalter) ausgelöst hat. Dies haben wir dann über den Einbau von Relais gelöst, die Gruppen von Netzteilen zeitversetzt eingeschaltet haben. Nachdem dies gelöst war, konnten die LED Strips über einen Raspberry Pi erfolgreich angesteuert werden. Der Prototyp war somit erfolgreich und es gab einige „Lessons-Learned“:

- Polystyrol als Träger ist zu unhandlich. Es ist nicht stabil genug um einen Rahmen drum herum zu tragen und zu unpraktisch um nur verkleidet zu werden. Außerdem ist es zu weich und würde schon bei kleineren Stößen Beschädigungen aufweisen.
- Es sind weniger Netzteile erforderlich, da die offiziellen Daten zu den LEDs einen höheren Stromverbrauch angeben, als tatsächlich anfällt. Zudem werden in der Regel nie alle LEDs mit maximaler Helligkeit in weiß angezeigt. Und falls doch: Im worst-case schaltet der Überlastschutz der Netzteile die LEDs temporär ab.
- Relais werden beim Einschalten benötigt um ein Auslösen der FI-Sicherung zu vermeiden

Daraus haben sich folgende Änderungen ergeben:

- Als Träger nutzen wir eine Kunststoff-Platte (z.B. Schreibtischstuhlunterlage oder Plexiglas)
- Als Schutz und Diffusor vor den LEDs nutzen wir eine weitere Schreibtischstuhlunterlage (oder Plexiglas mit Milchglasfolie)
- Als Rahmen nutzen wir Aluminium Doppel-Profil (30x60, Nut 8 B-Type)
- Als Verbinder zwischen Display und Schaltkasten nutzen wir zwei Anhängerkupplungsstecker und zwei Kabel mit 9*1,5mm² und 4*2,5mm² à 3m.
- Am Rahmen wird im oberen Bereich eine Kette befestigt zum Aufhängen
- Am Rahmen werden im unteren Bereich Füße installiert. Im Idealfall abnehmbar um die Gesamttiefe des Displays nicht zu erhöhen zwecks Transport.
- Die einzelnen Teile des Rahmes werden mit dünnen Stahlseilen oder Splints gegen auseinanderfallen gesichert (zusätzlich zu den regulären Schraubverbindern)

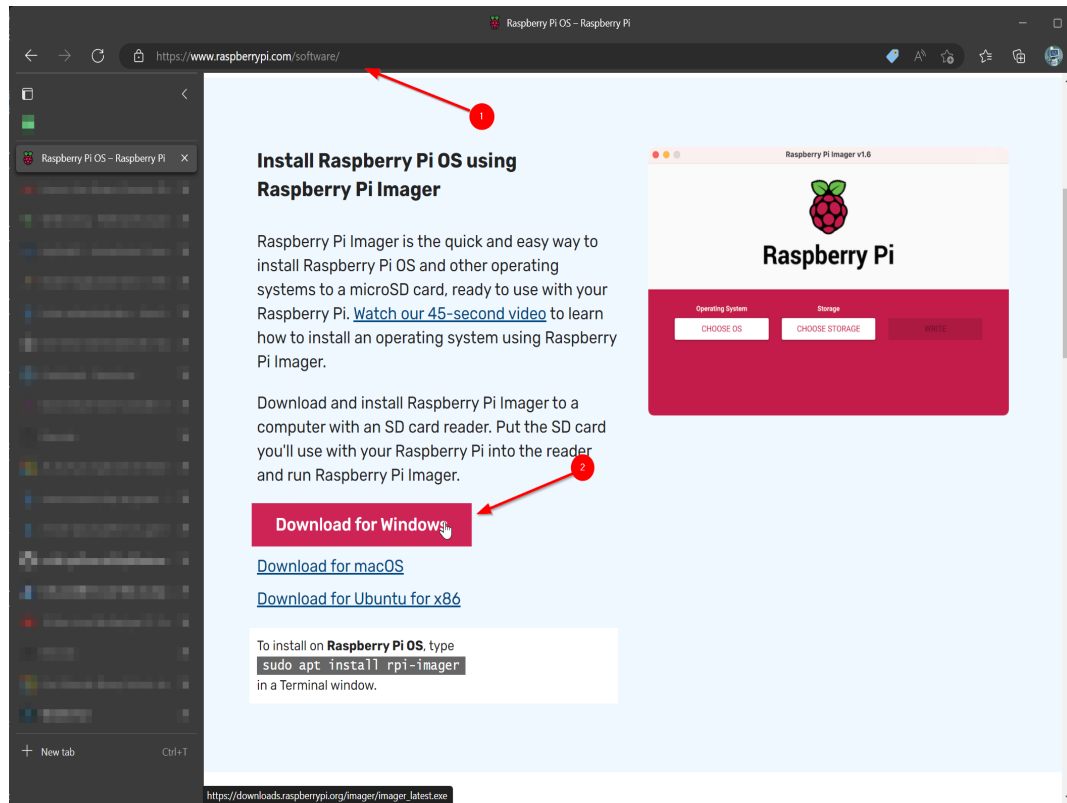
4.1.2 Voraussetzungen für LeDi auf einem RaspberryPi installieren

Diese Anleitung führt durch die Installation von LeDi auf einem RaspberryPi vom Start an. Für die Erstellung dieser Anleitung wurde ein RaspberryPi 3 genutzt, aber sie ist auch für einen RaspberryPi 4 anwendbar. Für einen RaspberryPi 2 werden zum Teil andere Lösungen benötigt wie z.B. für die Erstellung des Wifi Hotspots - z.B. durch Nutzung eines extra USB-WLAN-Sticks oder dem Hinzufügen eines WLAN Accesspoints zum Netzwerk des Displays. Wir empfehlen ausschließlich einen RaspberryPi 3 oder 4 mit mindestens 4GB RAM zu nutzen. Wir werden nicht auf nähere Details eingehen, wie eine Installation für andere Modelle möglich ist. Der RaspberryPi 1 wird nicht funktionieren, da dieser kein ARM7 Prozessor besitzt und somit .Net 6 nicht unterstützt.


RaspberryPi installieren

Als allererstes muss der RaspberryPi installiert werden. Dazu müssen wir zunächst den Raspberry Pi Imager herunterladen.

- Daher gehen wir auf <https://raspberrypi.com/software>: und klicken auf „Download for Windows“.



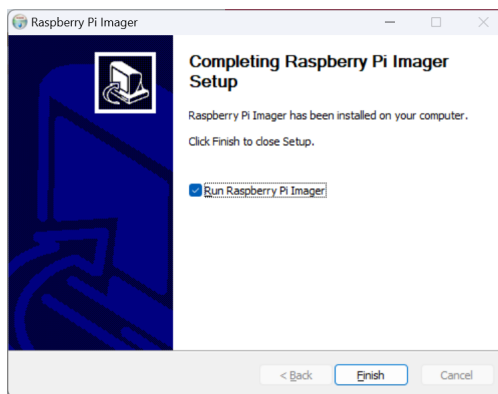
- Anschließend führen wir die heruntergeladene Datei `imager_x.y.z.exe` aus.

 imager_1.7.3	05.12.2022 09:21	Application	19.316 KB
--	------------------	-------------	-----------

- Nun führen wir die Datei aus und folgen dem sehr simplen Setup. Zunächst klicken wir einfach auf **Install**.



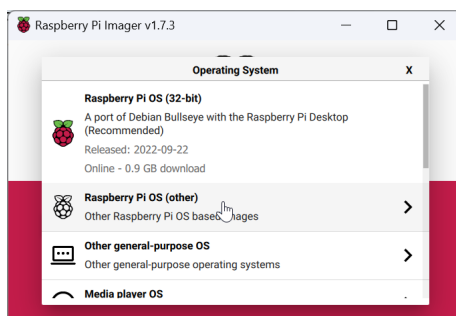
- Und das war es auch schon. Da wir direkt weiter machen wollen, wird der Haken Run Raspberry Pi Imager aktiviert und Finish geklickt.



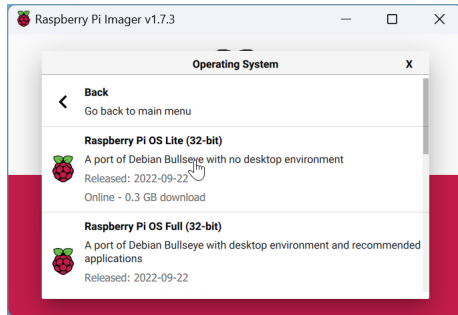
- Nun startet der Raspberry Pi Imager. Hier klicken wir zunächst auf Choose OS.



- In dem Dialog wählen wir zuerst Raspberry Pi OS (other) aus.



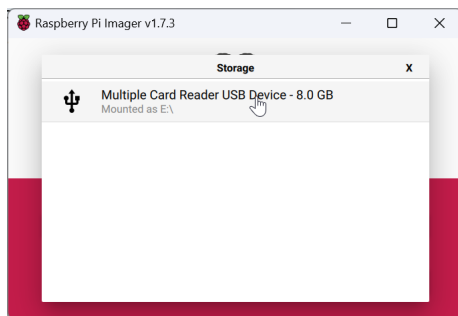
- Anschließend Raspberry Pi OS Lite (32-bit).



- Nun zeigt die Schaltfläche, die zuvor Choose OS angezeigt hat die ausgewählte Option an. Spätestens jetzt muss die Micro-SD-Karte in den Kartenleser des PCs eingeführt werden. Wenn dies geschehen ist wählen wir Choose Storage.



- Hier werden nun alle Wechselmedien angezeigt, die der Imager als SD-Karte identifiziert hat. Es kann gelegentlich vorkommen, dass auch andere Medien wie USB-Sticks hier aufgelistet werden. Diese können zwar im folgenden geflashed werden, aber können dann nicht am Raspberry Pi betrieben werden. Sollte in der Liste mehr als ein Element sein und nicht klar sein welches die SD-Karte ist, müssen alle USB-Sticks etc. vom PC getrennt werden. Bei der Identifizierung hilft meist zunächst die Größe des Mediums (hier im Beispiel 8GB). Erst in zweiter Reihe hilft ggf. der Name - leider auch nicht immer. Es muss jedenfalls nun die Micro-SD-Karte aus der Liste ausgewählt werden.

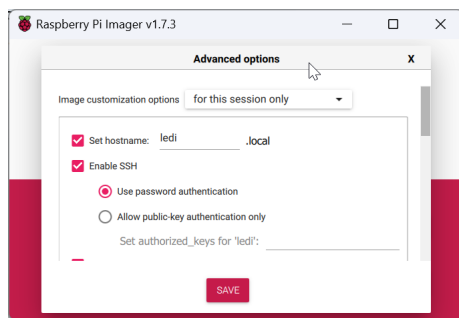


- Nachdem das Medium ausgewählt wurde, muss im Raspberry Pi Imager das Zahnrad-Symbol angeklickt werden um weitere Einstellungen vornehmen zu können.

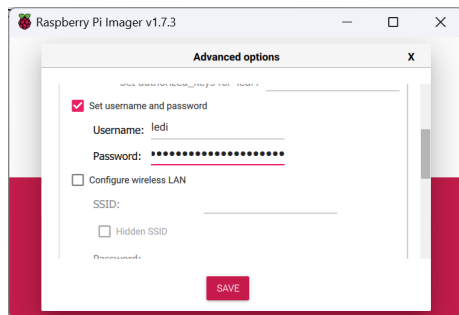


- Hier müssen nun die folgenden Einstellungen angepasst werden:

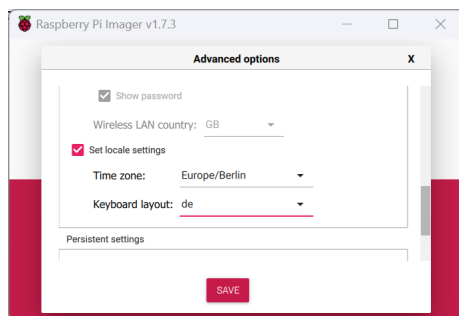
1. Der Hostname sollte auf einen passenden Namen gesetzt werden. ledi bietet sich hier an.
2. SSH sollte aktiviert werden. Die Passwort Authentifizierung ist ausreichend.



3. Der Benutzername und das Kennwort muss gesetzt werden. Als Benutzer bietet sich hier ledi an. Das Kennwort kann frei vergeben werden.



4. Zuletzt noch die passende Zeitzone und das passende Tastaturlayout setzen (z.B. Europe/Berlin und de).



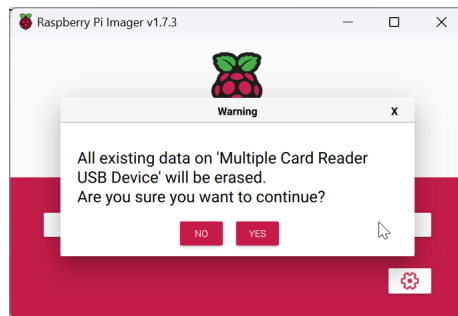
5. Abschließend den Dialog mit Save bestätigen.

- Nun befinden wir uns wieder im Raspberry Pi Imager Fenster. Dort starten wir mit einem Klick auf Write den Flash-Vorgang, in dem das Raspberry Pi OS auf die SD-Karte kopiert wird und die von uns getätigten

Einstellungen gesetzt werden.



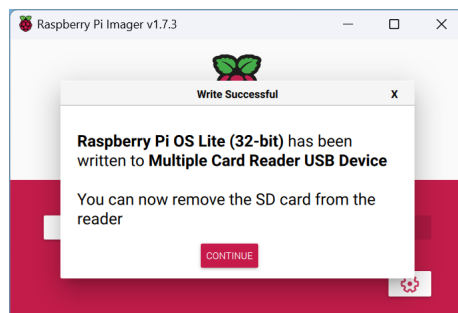
- Wir müssen nun noch einmal bestätigen, dass wir die eingelegte Micro-SD-Karte unwiderruflich mit dem Image überschreiben wollen.



- Nun warten wir, bis der Schreibvorgang abgeschlossen ist.



- Abschließend erscheint eine entsprechende Meldung.



- Nun ist die Installation des Raspberry Pis mit dem Raspberry Pi OS Lite Image auf die SD-Karte abgeschlossen.
- Lege nun die Micro-SD-Karte in den Raspberry Pi ein und start diesen. Der erste Start kann einige Minuten dauern. Sofern der Raspberry Pi via Netzkabel mit dem Netzwerk verbunden ist, kann dieser anschließend via SSH erreicht werden oder alternativ einfach über einen direkt angeschlossenen Bildschirm/Tastatur/Maus

bedient werden.

- Für Option via SSH: Um via SSH zu verbinden muss kann man via `ping -4 ledi` die IP-Adresse herausfinden. Alternativ kann man auch direkt via hostnamen verbinden. Dies geschieht mit dem Programm PuTTY ([Download hier](#): unter `putty-64bit-0.78-installer.msi`). Nach der Installation zu der IP-Adresse bzw. ledi verbinden und die im Raspberry Pi Imager angegebenen Benutzerdaten zum Anmelden nutzen.
- **Nachdem der erste Start erfolgreich durchgeführt wurden, aktualisieren wir noch das Betriebssystem mit folgenden Befehlen:**

```
sudo apt update
sudo apt upgrade
```

- Bei der ersten Ausführung eines `sudo`-Befehls erfolgt immer eine zusätzliche Passwortabfrage. Dies geschieht auch, wenn man einige Zeit keine `sudo`-Befehle ausgeführt hat. Ggf. erfolgt auch eine Rückfrage, ob man die Installation fortsetzen möchte, weil weitere X MB Speicherplatz benötigt werden. Diese kann mit einem Y bestätigt werden.
- Der `sudo apt upgrade` Befehl wird ggf. eine ganze Weile dauern.
- Nun ist der Raspberry Pi in der Basis installiert und wir können nun die Installation von LeDi starten.

.Net 6 installieren:

Jetzt werden wir .Net 6 installieren. Dazu gehen wir zunächst auf <https://dotnet.microsoft.com/en-us/download/dotnet/6.0>. Auf dieser Seite unter SDK 6.0.xx in der Tabelle den Eintrag Linux. Dort die Binary Arm32 anklicken. Nun die URL, die als Direktlink („direct link“) bezeichnet ist, kopieren (z.B. <https://download.visualstudio.microsoft.com/download/pr/5a24144e-0d7d-4cc9-b9d8-b4d32d6bb084/e882181e475e3c66f48a22fbfc7b19c0/dotnet-sdk-6.0.403-linux-arm.tar.gz>). Anschließend zur Raspberry Pi Konsole wechseln und folgendes ausführen:

```
wget <url from above>
```

Anschließend die folgenden Befehle ausführen. Dabei den Dateinamen in der ersten Zeile mit dem Dateinamen der zuvor heruntergeladenen Datei ersetzen.

```
DOTNET_FILE=dotnet-sdk-6.0.403-linux-arm.tar.gz
export DOTNET_ROOT=$(pwd)/.dotnet

mkdir -p "$DOTNET_ROOT" && tar xzf "$DOTNET_FILE" -C "$DOTNET_ROOT"

export PATH=$PATH:$DOTNET_ROOT:$DOTNET_ROOT/tools
```

Herzlichen Glückwunsch. .Net 6.0 ist nun installiert. Dies kann über den Befehl `dotnet --version` geprüft werden. Erfolgt dort eine Ausgabe mit der Versionsnummer, so war die installation erfolgreich.

Einrichten des WLAN AccessPoints

In diesem Absatz wird der RaspberryPi zu einem eigenständigen Hotspot. Dieser Schritt kann übersprungen werden, wenn ein externer WLAN AccessPoint genutzt werden soll oder kein AccessPoint benötigt wird da der RaspberryPi via Ethernet Kabel angeschlossen wird. Diesen Schritt zu überspringen macht LeDi jedoch weniger mobil, da nicht einfach der Stecker in die Steckdose gesteckt werden kann und anschließend losgelegt werden kann. Um den RaspberryPi als WLAN-AccessPoint einzurichten, müssen die folgenden Schritte durchgeführt werden. Um einen (permanenten) Verbindungsabbruch während der Einrichtung zu vermeiden, sollte der RaspberryPi via Netzkabel mit dem Netzwerk verbunden sein oder direkt via HDMI und Tastatur bedient werden. Zusammengefasst werden wir den RaspberryPi zu einem DNS Server (dnsmasq) und DHCP Server (dhcpd) machen und die Applikation hostapd installieren um die Hotspot/AccessPoint-Funktionalität bereit zu stellen, über den sich anschließend die Clients verbinden können.

Wi-Fi aktivieren

Sofern beim Login folgende Meldung angezeigt wurde, bitte noch den folgenden Abschnitt befolgen. Wenn nicht, diesen Abschnitt überspringen.

Wi-Fi **is** currently blocked by rfkill.
Use raspi-config to **set** the country before use.

Also machen wir, was dort steht. Wir führen folgenden Befehl aus:

```
sudo raspi-config
```

Anschließend öffnet sich das Raspberry Pi Software Configuration Tool. Hier gehen wir auf den ersten Punkt System Settings. Anschließend wählen wir S1 Wireless LAN aus. Nun DE Germany für Deutschland. Die anschließende Meldung bestätigen wir noch mit OK und wenn Please enter SSID erscheint, gehen wir auf Cancel. Im Hauptmenü dann auf Finish.

Hostapd und Dnsmasq installieren

Folgende Befehle ausführen um hostapd und dnsmasq zu installieren und den hostapd service zu konfigurieren. dhcpd ist in der Regel bereits vorinstalliert und braucht nicht installiert zu werden.

```
sudo apt install -y hostapd dnsmasq
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
```

Nachdem die installation ausgeführt wurde, müssen die Dienste einmal angehalten werden um diese zu konfigurieren:

```
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

DHCP konfigurieren (dhcpcd)

Nun installieren wir das private Netzwerk für den WLAN Netzwerkadapter des RaspberryPi. Wir wählen ein Subnetz in einem privaten IP Segment und ein /24 (255.255.255.0) Subnetz. In diesem Fall nutzen wir das Subnetz 10.10.100.0/24 und damit die IP 10.10.100.1 für die IP-Adresse des RaspberryPi. Es kann jedoch auch jedes andere Subnetz und jede andere darin liegende IP genutzt werden. Nun die IP Konfiguration wie folgt konfigurieren:

```
sudo nano /etc/dhcpcd.conf
```

Folgende Zeilen am Ende der Datei hinzufügen:

```
interface wlan0
nohook wpa_supplicant
static ip_address=10.10.100.1/24
static domain_name_servers=10.10.100.1
```

Die Datei mit Strg+O speichern und mit Strg+X die Datei schließen.

DHCP konfigurieren (dnsmasq)

Jetzt konfigurieren wir den DHCP Server in dnsmasq um den verbindenden Clients eine IP bereit zu stellen. Zunächst eine neue config erstellen:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
sudo nano /etc/dnsmasq.conf
```

Nun folgendes einfügen:

```
interface=wlan0
dhcp-range=10.10.100.11,10.10.100.254,255.255.255.0,1h
domain=intern
listen-address=10.10.100.1
listen-address=127.0.0.1
local=/intern/
```

Die Datei mit Strg+O speichern und mit Strg+X schließen. Diese Zeilen bedeuten, dass wir die IPs 10.10.100.11 bis 10.10.100.254 in dem Subnetz 255.255.255.0 mit einer Lease Time von einer Stunde an die verbindenden Clients vergeben. Die ersten 10 IPs lassen wir für Infrastrukturdienste frei. Eventuell brauchen wir die ja mal... Die Konfiguration definiert auch, dass andere Systeme als das lokale System den dnsmasq-service auf dem RaspberryPi nutzen um DNS Adressen aufzulösen. Auch die „intern“-Domain soll von niemand anderen als dem lokalen dnsmasq-Server aufgelöst werden für den Fall dass der RaspberryPi einen Uplink z.B. zum Internet bekommt.

DNS konfigurieren (dnsmasq)

Jetzt konfigurieren wir die DNS Auflösung. Dazu die hosts-Datei editieren:

```
sudo nano /etc/hosts
```

Folgende Zeilen am Ende hinzufügen:

```
10.10.100.1 ledi.intern
10.10.100.1 board board.intern
```

Die letzte Zeile bedeutet, dass das System den Alias „board“ erhält und so von den Clients via Browser aufgerufen werden kann. Es können auch andere Namen verwendet werden. Seid kreativ, so wie ihr es braucht oder haben möchtet. Die Datei mit Strg+O speichern und mit Strg+X schließen.

WLAN AccessPoint konfigurieren (hostap)

Im folgenden wird der WLAN AccessPoint selbst konfiguriert:

```
sudo nano /etc/hostapd/hostapd.conf
```

Folgende Zeilen in die Datei einfügen:

```
interface=wlan0
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=0
ssid=LeDi
```

Die Datei mit Strg+O speichern und mit Strg+X schließen. Dies konfiguriert den Netzwerkadapter wlan0. Der Name (SSID) des WLANs heißt „LeDi“ und die Verbindung wird unverschlüsselt sein. So kann sich jeder zum RaspberryPi verbinden.

Die neue Konfiguration muss nun noch beim init geladen werden:

```
sudo nano /etc/default/hostapd
```

Ersetze die Zeile **#DAEMON_CONF=""** mit **DAEMON_CONF="/etc/hostapd/hostapd.conf"** um das führende # zu entfernen und den Pfad zur neuen Konfigurationsdatei hinzuzufügen. Die Datei mit Strg+O speichern und mit Strg+X schließen.

Abschluss

Die Dienste (neu)starten:

```
sudo systemctl restart dhcpd
sudo systemctl start hostapd
sudo systemctl start dnsmasq
```

Wenn ihr nun mit einem Tablet oder Smartphone auf WLAN-Suche geht, findet ihr das WLAN „LeDi“, welches vom Raspberry Pi bereitgestellt wird.

4.1.3 LeDi Server installieren

Installation

Die Installation von LeDi besteht im Wesentlichen aus zwei Teilen:

1. .Net 6.0 als Framework
2. ws2811.so Datei kompilieren
3. LeDi selbst

.Net 6.0 installieren

.Net wurde bereits in dem Abschnitt „Voraussetzungen für LeDi auf einem RaspberryPi installieren“ installiert, sodass wir uns hier nun nur noch um LeDi kümmern müssen.

ws2811.so Datei kompilieren

die ws2811.so Datei ist eine Datei, die nachher die LEDs ansteuert. Dazu zunächst scon, git und gcc installieren:

```
sudo apt -y install scon gcc git
```

Anschließend das Github Repository herunterladen und die Datei aus den sources erstellen

```
cd ~
git clone https://github.com/jgarff/rpi_ws281x.git
cd ~/rpi_ws281x
scon
gcc -shared -o ws2811.so *.o
sudo mkdir /opt/LeDi.Display
sudo cp ws2811.so /opt/LeDi.Display/
```

LeDi herunterladen

LeDi kann auf der Github Projekt-Seite heruntergeladen werden. Dazu unter <https://github.com/Hunv/LeDi/releases> beim aktuellsten Release die „LeDi.zip“ herunterladen. Sofern der RaspberryPi Internetzugriff hat, kann diese Datei auch direkt via folgendem Befehl heruntergeladen werden (die # durch die Versionsnummer ersetzen):

```
wget https://github.com/Hunv/LeDi/releases/download/v#.#/LeDi.zip
```

LeDi installieren

Nun die Datei LeDi.zip nach /opt/ entpacken und die Berechtigungen auf Ordner und Ports setzen. Außerdem noch ein paar Konfigurationen für .Net:

```
sudo unzip-Befehl....
sudo chmod -R 744 /opt/LeDi*
sudo adduser ledi kmem
sudo setcap CAP_NET_BIND_SERVICE=+eip /home/ledi/.dotnet/dotnet
dotnet dev-certs https --trust
```


Jetzt erstellen wir ein Serverzertifikat für den Webserver (die roten Meldungen beim zweiten Befehl ignorieren):

```
sudo /home/ledi/.dotnet/dotnet dev-certs https --clean
sudo /home/ledi/.dotnet/dotnet dev-certs https --verbose
```

Anschließend müssen die einzelnen Komponenten nur noch als Service registriert werden. Dazu zunächst folgenden Befehl ausführen um eine neue Datei zu erstellen: `sudo nano /etc/systemd/system/ledi.server.service`. Dort dann folgenden Text einfügen:

```
[Unit]
Description=ledi.server
After=network.target

[Service]
ExecStart=/home/ledi/.dotnet/dotnet /opt/LeDi.Server/LeDi.Server.dll
WorkingDirectory=/opt/LeDi.Server/
StandardOutput=inherit
StandardError=inherit
Restart=always
User=ledi

[Install]
WantedBy=multi-user.target
```

Jetzt mit Strg+W die Datei speichern und mit Strg+X die Datei schließen. Dies wiederholen wir jetzt für das Display und den WebClient: `sudo nano /etc/systemd/system/ledi.display.service`. Dort dann folgenden Text einfügen:

```
[Unit]
Description=LeDi.Display
After=network.target

[Service]
ExecStart=/home/ledi/.dotnet/dotnet /opt/LeDi.Display/LeDi.Display.dll
WorkingDirectory=/opt/LeDi.Display/
StandardOutput=inherit
StandardError=inherit
Restart=always
User=ledi

[Install]
WantedBy=multi-user.target
```

Jetzt mit Strg+W die Datei speichern und mit Strg+X die Datei schließen und zuletzt `sudo nano /etc/systemd/system/ledi.webclient.service`. Dort dann folgenden Text einfügen:

```
[Unit]
Description=ledi.webclient
After=network.target

[Service]
ExecStart=/home/ledi/.dotnet/dotnet /opt/LeDi.WebClient/LeDi.WebClient.dll
WorkingDirectory=/opt/LeDi.WebClient/
StandardOutput=inherit
StandardError=inherit
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
Restart=always
User=ledi

[Install]
WantedBy=multi-user.target
```

Mit Strg+W die Datei speichern und mit Strg+X die Datei schließen. Nun sind die Konfigurationsdateien für die drei Services vorhanden. Jetzt müssen wir die Services nur noch registrieren und starten:

```
sudo systemctl daemon-reload
sudo systemctl enable ledi.server
sudo systemctl enable ledi.display
sudo systemctl enable ledi.webclient
sudo systemctl start ledi.server
sudo systemctl start ledi.display
sudo systemctl start ledi.webclient
```

Fertig ist die Installation von LeDi. Über einen Browser, der mit LeDi z.B. via WLAN Hotspot verbunden ist, kann LeDi nun unter <http://ledi.intern> aufgerufen werden.

4.1.4 LeDi Benutzeranleitung

Ersteinrichtung

Nachdem alle drei Dienste erfolgreich gestartet wurden, kann über die URL <https://ledi.intern> die Bedienoberfläche von LeDi aufgerufen werden, sofern das zugreifende Gerät über das WLAN verbunden ist, das LeDi bereit stellt. Von anderen Geräten muss über den hostnamen <https://ledi> oder die IP-Adresse zugegriffen werden.

Ledi ist direkt Betriebsbereit. Lediglich ein eventuell angeschlossenes LED Display muss zuvor noch konfiguriert werden.

Display einrichten

Um ein angeschlossenes LED Display zu konfigurieren, muss ggf. zunächst die `config.conf` im `/opt/LeDi`. Display-Verzeichniss bearbeitet werden - sofern nicht die Standardkonfiguration genutzt werden soll.

Standardmäßig sieht diese wie folgt aus

```
ServerUrl:https://localhost:5001/api/
DeviceType:LED Screen
DeviceModel:LeDiDisplay
GpioPin:18
PwmChannel:0
```

Dies heißt, dass die Schnittstelle zur Server-Komponente auf dem gleichen Gerät auf Port 5001 läuft. Außerdem meldet sich das Display als Gerätetyp `LED Screen` und dem Gerätemodell `LeDiDisplay`. Auf dem Raspberry Pi wird zudem der GPIO-Port 18 (Pin #12) genutzt. Hier ist auch der GPIO-Port 12 (Pin #32) möglich. Beide Pins sind für den PWM0 Channel vorgesehen. Wird der PWM1 Channel genutzt, muss die entsprechende Setting auf 1 geändert werden und entweder der GPIO Port 13 (Pin #33) oder der GPIO Port 19 (Pin #35) genutzt und konfiguriert werden.

Nach der Änderung der Konfiguration muss der Dienst einmal neu gestartet werden:

```
sudo systemctl restart ledi.display
```

Nun auf der LeDi-Benutzeroberfläche auf den Menüpunkt „Einstellungen“ wechseln. Dort wird das Display nun angezeigt.

The screenshot shows the LeDi web interface. On the left is a dark sidebar with a menu containing: Home, Dashboard, Start, Referee Match, Match Planning, Settings (highlighted), and About LeDi. The main content area has a light gray header with the title 'Settings'. Below the header, it says 'The following settings are configured:'. There are two rows of settings, each with a 'Setting' column and a 'Value' column. The first row shows 'timezone' with the value '(UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockh'. The second row shows 'wifi_password' with an empty text input field. Below the settings is a blue 'Save Changes' button. Underneath the settings section is a red-bordered box containing the 'Devices' section. It says 'The following devices are known:'. Below this is a table with columns: Device Name, Device Model, Device Type, Enabled, Default, and two action icons (a wrench and a trash can). The table has one row for 'LeDi' with values: LeDi, LeDiDisplay, LED Screen, True, False.

Setting	Value
timezone	(UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockh
wifi_password	

Save Changes

Device Name	Device Model	Device Type	Enabled	Default		
LeDi	LeDiDisplay	LED Screen	True	False		

Hier auf das kleine -Symbol klicken. Auf der nun erscheinenden Seite zu den Einstellungen gehen. Dort gibt es die Einstellungen height (= Höhe) und Width (= Breite). Diese müssen nun auf die Anzahl der LEDs gesetzt werden, die das Display in der Größe hat. z.B. 58x96. Anschließend auf „Einstellungen speichern“ klicken (es erfolgt keine Reaktion).

Bemerkung: Hinweis: Aktuell stehen Display-Layouts in folgenden Größen zur Verfügung (Breite x Höhe): 20x10, 60x10, 96x58, 96x60. Andere Größen müssen im Verzeichniss des Display Dienstes unter /Config/Layouts angelegt werden.

LeDi

Home

Dashboard

Start

Referee Match

Match Planning

Settings

About LeDi

Save Properties

The following settings are configured for device LeDi (5f8e5e8a-e33c-4749-9e8b-35464695037a):

Setting	Value
brightness	50
height	58
led_alternatingrows	true
led_firstledleft	true
led_toptobottom	true
mode	none
width	96

Save Settings

Des Weiteren gibt es noch die Optionen `led_alternatingrows`, `led_firstledleft` und `led_toptobottom`. Die erste Option ist in der Regel immer `true`, da die LEDs auf dem Display in der Regel immer abwechselnd von links nach rechts und wieder zurück laufen. Sollte man beim Zusammenbau alle immer z.B. von links nach rechts laufen lassen, dann muss diese Option auf `false`. `led_firstledleft` definiert, ob die Einspeisung der Daten auf der linken oder rechten Seite des Displays erfolgt. Ist dies auf der linken Seite, so ist der Wert auf `true` zu setzen. Ist dies auf der rechten Seite, so muss dieser Wert `false` lauten. Zuletzt kann mit der Option `led_toptobottom` noch eingestellt werden, ob die Einspeisung oben oder unten am Display erfolgt. Erfolgt diese unten, so ist der Wert auf `false` zu setzen. Erfolgt diese oben, dann auf `true`. Hier hilft es ggf. auch nach der ersten Konfiguration mit den Werte ein bisschen zu spielen, falls das Ergebniss nicht sofort das gewünschte Ergebniss ist (z.B. steht die Anzeige auf dem Kopf oder ist gespiegelt.).

Nun ein bisschen nach unten scrollen und unter „Befehl an Gerät übermitteln:“ in der Auswahlbox „Geräteeinstellung neu laden“ auswählen und auf „Befehl senden“ klicken.

Send command to device:

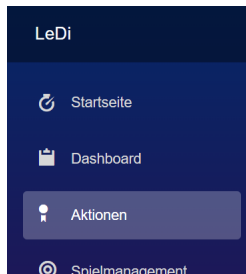
Reload Display Settings

Send Command

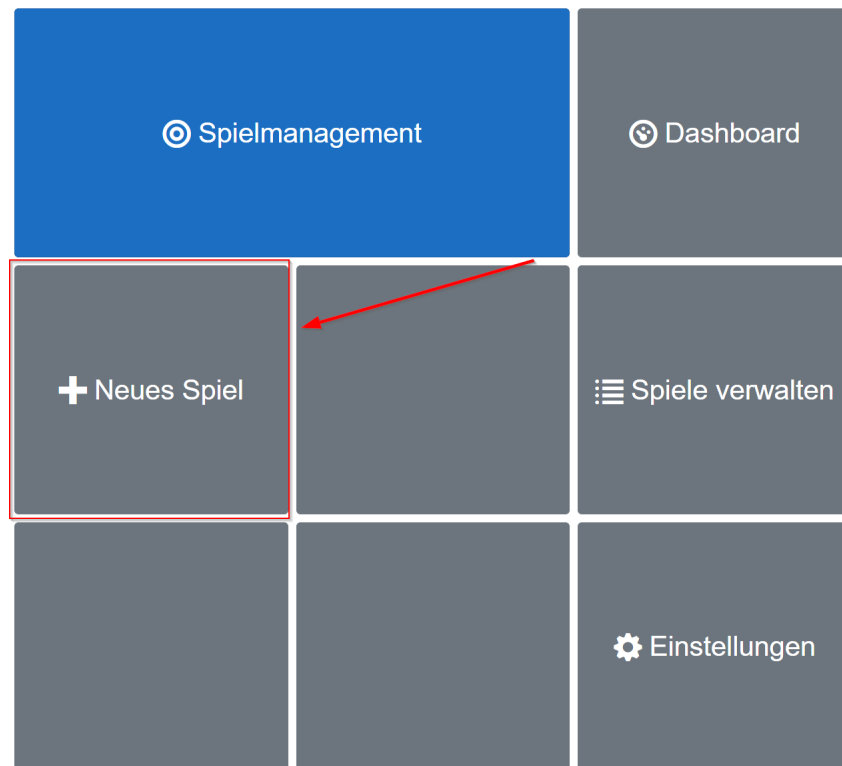
Wenn alles korrekt eingerichtet ist, ist die erste Zeile LEDs nun weiß und die letzte Zeile der LEDs nun grün. Somit ist die Einrichtung des Displays abgeschlossen.

Spiel erstellen

Um ein Spiel zu erstellen, folgende Schritte durchführen: 1. In der Navigation **Aktionen** auswählen.



2. Auf der Seite nun **Neues Spiel** auswählen.



3. Nun kann die Sportart ausgewählt werden. Aktuell gibt es lediglich **Unterwasserhockey**. Nach dem Auswählen der Sportart erscheinen weitere Einstellungsmöglichkeiten. Hier kann z.B. definiert werden wie die Teams heißen, die gegeneinander spielen. Auch die Anzahl der Halbzeiten, die Halbzeitlänge, der voraussichtliche Start des Spiels sowie die Regel, dass zwei Minuten vor Ende die Zeit bei einer Unterbrechung pausiert, können hier angegeben werden. Wenn alle Parameter eingestellt sind, kann das Spiel entweder gespeichert und sofort gestartet werden oder aber, z.B. für eine Turnierplanung, das Spiel nur gespeichert werden.

LeDi

Startseite

Dashboard

Aktionen

Spielmanagement

Spiele verwalten

Einstellungen

Über LeDi

Sportart wählen

Unterwasserhockey

Auf dem Display anzeigen:

Team 1

Team1

Team 2

Team2

Anzahl der Halbzeiten

2

Halbzeitlänge (Minuten : Sekunden)

15 : 0

Scheduled Start Date and Time

11.12.2022 00:25:00

☒ Das Spiel bei Unterbrechungen pausieren, wenn das Spiel fast beendet ist

Restzeit (Sekunden) vor Ende des Spiels, ab der das Spiel unterbrochen werden soll

120

☐ Spiel weiterführen bei Gleichstand nach abgelaufener Spielzeit

Speichern und Starten

Speichern

Spiel betreuen

Um ein Spiel zu begleiten und LeDi während eines Spiels zu steuern kann entweder in der Navigation direkt **Spielmanagement** ausgewählt werden oder über die **Aktionen**-Seite die Schaltfläche **Spielmanagement**. Hier muss nun das Spiel gewählt werden, welches nun stattfinden soll. Sofern zuvor **Speichern und Starten** gewählt wurde, ist dies nicht notwendig. Anschließend ist die Bedienoberfläche zu sehen, die das Display während eines Spiels steuert.

Team1

0

15:00

Halbzeit:
0 of 2

0

Team2

Strafe eintragen

Strafe eintragen

Tor

Start/Fortfahren

Tor

Tor zurücknehmen

Pause

Tor zurücknehmen

⬇ Zeige erweiterte Steuerung

Match log:

Zeit	Bemerkungen
------	-------------

Die Schaltflächen im Einzelnen:

Schaltfläche	Funktion
Tor	Fügt dem entsprechenden Team einen Punkt hinzu
Tor zurücknehmen	Zieht dem entsprechenden Team einen Punkt ab
Start/Fortfahren	Startet das Spiel/die Spielzeit
Pause	Stoppt das Spiel/die Spielzeit
Strafen	Ruft den Dialog auf um eine Strafe zu vergeben
Zeige erweiterte Steuerung	Zeigt weniger häufig genutzte Optionen
Spiel abbrechen	Bricht ein Spiel ab und beendet dieses dadurch
Spiel neustarten	Startet ein Spiel neu
Setze Spielzeit	Möglichkeit die Restspielzeit zu modifizieren
Zeige auf Display	Auf welchem Display wird das Spiel gezeigt
Strafe zurücknehmen	Eine eingetragene Strafe zurücknehmen

Unterhalb der Schaltflächen ist dann noch ein Logbuch zu sehen, in dem alle Ereignisse des Spiels noch einmal zusammengefasst abgebildet sind.

Ablauf eines Spiels

- Wenn ein Spiel beginnt, wird mit einem Klick auf **Start/Fortfahren** die Zeit gestartet.
- Wenn ein Timeout genommen wird, egal welcher Art, wird mit **Pause** die Zeit angehalten. Anschließend wieder mit **Start/Fortfahren** gestartet.
- Wenn ein Tor fällt, wird dies mit der linken Schaltfläche für das linke Team und mit der rechten Schaltfläche für das rechte Team notiert.
- Wenn ein Spieler oder ein Team eine Strafe, ins besondere Strafzeiten, bekommt, dann kann dies über **Strafen** eingetragen werden. Eine Strafzeit läuft dann auch in der Anzeige ab.

Spiele verwalten

In dem Menüpunkt **Spiele verwalten** können alle Spiele noch einmal in einer Übersicht angezeigt werden. Hier sind die anstehenden, laufenden und beendeten Spiele zu sehen. Über die Schaltfläche **Bearbeiten** können diese noch editiert werden.

4.1.5 LeDi Baubegleitung

Diese Seite ist im Aufbau

Diese Seite soll dabei unterstützen ein LED Display selbst zusammen zu bauen. Dies ist keine Schritt für Schritt Anleitung, aber hier ist dokumentiert wie wir unser Display aufgebaut haben und auch was wir ggf. beim nächsten Mal anders machen würden. Hilfreich für einen Nachbau ist auch unsere Zusammenfassung in folgendem YouTube-Video: [Link](#)

Display

Rahmen

Der Rahmen des Displays besteht im Wesentlichen aus einem Rahmen aus Standard-Aluprofilen mit 3cm Kantenlänge („3030 Nut8 I-Profil“). Der Rahmen, in den der Träger für das Display eingelassen ist, ist ein Doppelprofilrahmen. Also 3060-Profile mit 3cm bzw. 6cm Kantenlänge. Verbunden sind diese Profile mit Nut8 Winkelverbindern.

LED-Träger

Der Träger für die LED Strips besteht aus einer zurecht geschnittenen Schreibtischstuhlunterlage. Diese ist robust, leicht und günstig. Alternativ ginge auch jede andere Art von Trägermaterial, das nicht leitend ist. Vor den LEDs als Schutz und Diffusor haben wir eine weitere Schreibtischstuhlunterlage verwendet. Diese hat hier durch die „perlige“ Oberfläche die Wirkung eines Diffusors. Dadruch wirken die LEDs ein bisschen weniger Pixelig. Wer diesen Effekt noch weiter intensivieren möchte, kann eine grobe Milchglasfolie auf die glatte Seite der Blende kleben. Dies gilt auch, wenn statt eine Schreibtischstuhlunterlage ein einfaches transparentes Plexiglas genommen wird. Auch hier hilft Milchglasfolie. Gegebenenfalls diese sogar auf beiden Seiten der Plexiglasscheibe aufbringen.

LEDs

Die LEDs sind WS2812B LED-Strips mit 96 LEDs pro Meter. Es gibt diese auch mit 100 LEDs pro Meter. Die breite eines LED-Strips beträgt ca. 1,2cm. Hierdurch entsteht ein nahezu identischer Abstand zwischen den LEDs vertikal und horizontal, wenn mehrere LED-Strips übereinander gelegt werden. Unsere LED-Strips waren IP67 zertifiziert. Da durch die Silikonummantelung der LED-Strips jedoch weitere 2mm Breite der LED-Strips hinzu kommen, haben wir diese Ummantelung entfernt. Es gibt IP65 LED-Strips, welche ebenfalls eine gewisse Resistenz gegen Wasser haben, dabei aber nicht an Breite zunehmen. Diese haben lediglich einen Überzug aus Silikon. Diese Variante empfehle ich für alle, die LeDi mobil-, schwimmbad- oder outdoor-fähig machen möchten. Soll LeDi lediglich in trockenen indoor-Umgebungen eingesetzt werden, kann auf IP-Zertifizierungen auch verzichtet werden. Des Weiteren gibt es die LED-Strips in mehreren Längen zu kaufen. In der Regel bedeutet eine längere Variante, dass diese günstiger pro Meter wird. Es empfiehlt sich allerdings nicht nun die längste Variante zu kaufen und diese zurecht zu schneiden - sofern es die passende Größe für die angestrebte Displaygröße auch von der Stange gibt. Meist haben die gekauften LED-Strips am Anfang und am Ende einen Stecker bereits angelötet. Diese können die spätere Montage erheblich vereinfachen, da so stundenlange Lötarbeiten überflüssig werden. Die LED-Strips selbst haben wir mit einfachem Sanitär-Silikon auf dem LED-Träger aufgeklebt. Wir haben weißes Silikon genommen, jedoch würde ich eher transparentes Silikon empfehlen. Wenn ein bisschen Silikon zwischen den LED-Strips hervor quillt, ist transparent weniger auffällig als weiß. Die LED-Strips haben für die Datenleitung des Weiteren eine Richtung. Diese ist üblicherweise mit einem kleinen Pfeil auf den Strips bei jeder LED markiert. Die Strips müssen nun immer abwechselnd in der Richtung der Daten aufgeklebt werden. D.h. wenn die erste Reihe von rechts nach links verläuft, verläuft die zweite Reihe von links nach rechts. usw.

Verkabelung

Jeder LED-Strip hat 3 Leitungen: Plus, Minus und Daten. Plus und Minus müssen durch den LED-Träger hinter die Träger-Platte geführt werden. Hier bietet es sich an mit einem Bohrer, der einen möglichst passenden Durchmesser zum Kabel besitzt, ein Loch in den Träger zu bohren. So kann das Kabel anschließend von hinten nach vorne geführt werden und dort am LED-Strip festgelötet werden. 96 LEDs unserer WS2812B-LEDs brauchen ca. 2,5A Leistung wenn alle LEDs auf voller Helligkeit in weiß leuchten. Die Netzteile liefern 12A. Mit ein wenig Puffer von 2A kann ein Netzteil also 4 LED-Strips versorgen. Daher werden hinter dem LED-Träger 4 Kabel der LED-Strips zu einem zusammengeführt. Hierzu haben wir 5er-Klemmen benutzt. Ggf. eignen sich hier auch kurze Reihenklemmen. Da dies für Plus und Minus gemacht werden muss, kommen wir am Ende auf insgesamt 15 Klemmen mit Plus und 15 Klemmen mit Minus. Als Kabel haben wir von den LED-Strips zu den Klemmen 1,0mm²-Leitungen verwendet (entsprechend rot und schwarz). Ab den Klemmen zu den Steckdosen haben wir 1,5mm² verwendet. Dies entspricht der Dicke der Leitungen im Verbindungskabel. Eine weitere Beachtung finden muss, dass alle Masse-Kabel untereinander verbunden sein müssen. Wenn die LED-Strips keine gemeinsamen Masse haben, werden diese später nicht funktionieren.

Datenleitung

Die Zuleitung der Daten muss am ANFANG der LED-Strips an den ersten LED-Strip bei der ersten LED angelötet werden. Die Datenleitung ist der mittlere Pin. Anschließend müssen die einzelnen LED-Strips untereinander miteinander verbunden werden. Dazu wird dann das Ende des ersten Strips mit dem Anfang des zweiten Strips verbunden. Dazu braucht man ein kurzes, ca. 10cm langes Stück Kabel (bei uns: 1,0mm²), welches quasi einen U-Turn vom ersten zum zweiten Strip macht. Dies wird dann bis zum Ende fortgeführt (Ende von Strip 2 mit Anfang von Strip 3 verbinden, ...).

Stecker

Die Stecker sind Standard-Stecker, welche üblicherweise für das Verbinden von Anhängern mit Autos genutzt wird. Wir haben hier die 13-polige Variante verwendet. Nun haben wir jedoch insgesamt 31 Leitungen, die wir mit dem Schaltkasten verbinden müssen. D.h. selbst zwei Stecker (=26 Leitungen) würden Augenscheinlich nicht reichen. Das stimmt aber nicht ganz. Laut DIN-Norm sind bei den Anhänger-Steckern 9 Leitungen auf 1,5mm² und 4 Leitung mit 2,5mm² ausgelegt. Wenn wir jetzt also 8x 2 Leitungen zu einer zusammenfassen, reduziert sich die Gesamtzahl der Leitungen auf 23. Das passt. D.h. wir haben jeweils 8 Leitungen von Plus und Minus genommen und zu 4 Leitungen mit 3er-Klemmen zusammengefasst. Hier haben wir ab der 3er-Klemme zum Stecker dann Kabel in der 2,5mm²-Variante verwendet - wie es auch in der Spezifikation der Stecker vorgesehen ist. Wir haben daher dann die einzelnen Pins der Stecker wie folgt belegt:

Stecker 1

(Leitung#: Zahl = Leitungsnummer, +/- = Plus- oder Minus-Pol)

Pin	Ader-Farbe	Ltg.-Dicke	Leitung #
1	gelb	1,5 mm ²	5+
2	blau	1,5 mm ²	7+
3	weiß oder pink	1,5 mm ²	9+
4	grün	1,5 mm ²	11+
5	braun	1,5 mm ²	5-
6	rot	1,5 mm ²	7-
7	schwarz	1,5 mm ²	9-
8	grau	1,5 mm ²	Daten
9	braun/blau	2,5 mm ²	1+ und 3+
10	braun/rot	2,5 mm ²	13+ und 15+
11	weiß/rot	2,5 mm ²	1- und 3-
12	lila	1,5 mm ²	11-
13	weiß/schwarz	2,5 mm ²	13- und 15-

Stecker 2

(Leitung#: Zahl = Leitungsnummer, +/- = Plus- oder Minus-Pol)

Pin	Ader-Farbe	Ltg.-Dicke	Leitung #
1	gelb	1,5 mm ²	4+
2	blau	1,5 mm ²	6+
3	weiß oder pink	1,5 mm ²	8+
4	grün	1,5 mm ²	10+
5	braun	1,5 mm ²	4-
6	rot	1,5 mm ²	6-
7	schwarz	1,5 mm ²	8-
8	grau	1,5 mm ²	Daten
9	braun/blau	2,5 mm ²	2+
10	braun/rot	2,5 mm ²	12+ und 14+
11	weiß/rot	2,5 mm ²	2-
12	lila	1,5 mm ²	10-
13	weiß/schwarz	2,5 mm ²	12- und 14-

Schaltkasten

Der Schaltkasten ist das Gehirn von LeDi. Hier befindet sich im Schaltschrank die gesamte Steuerung und ein großer Teil der Elektronik.

Schaltschrank

Stromführung

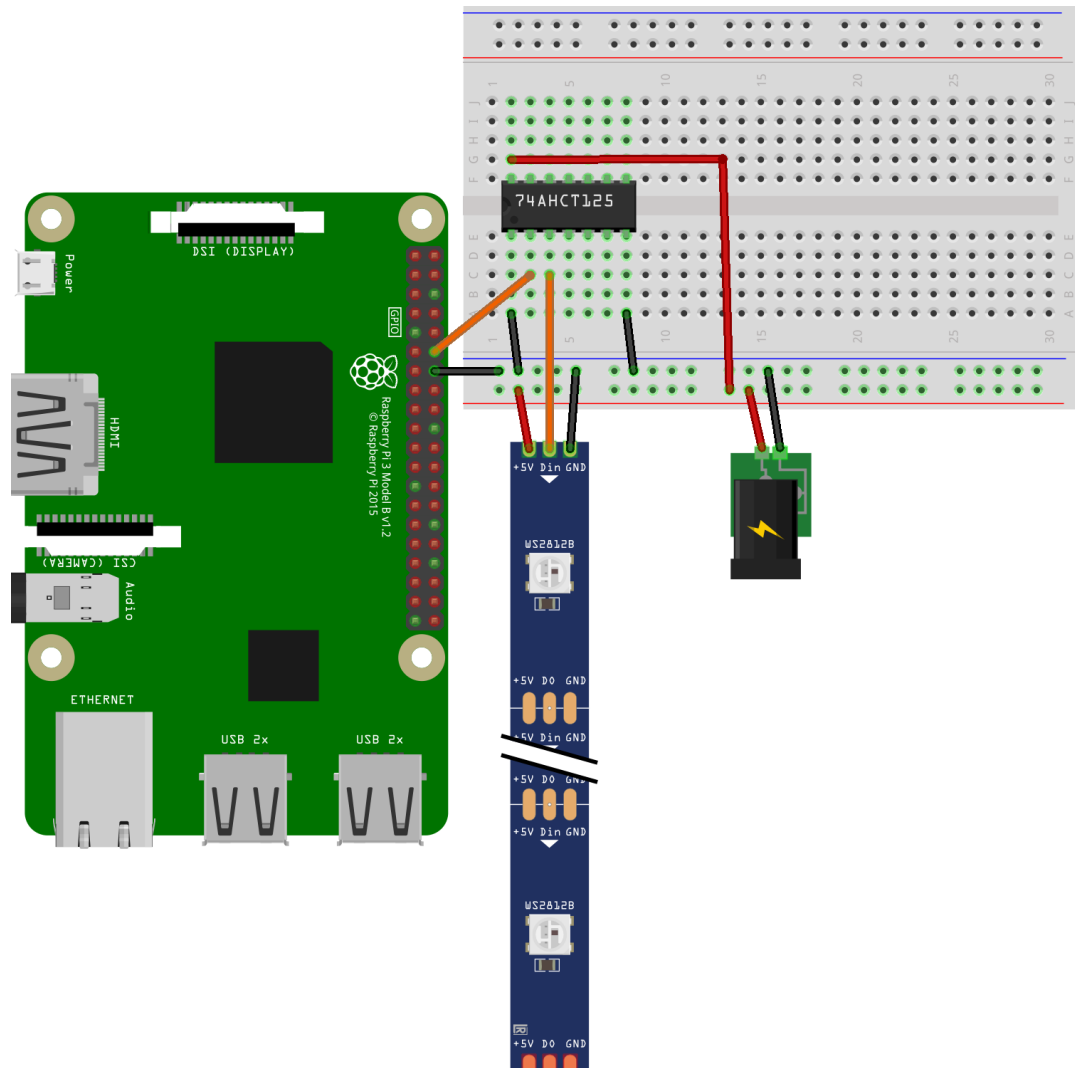
Im Schaltkasten wird der Strom durch ein normales Zuleitungskabel mit 230V eingeführt. Dieses Endet an einem FI-Schalter. Der FI-Schalter bringt für den Fall der Fälle eine letzte Sicherheit, dass keine Kurzschlüsse oder Stromschläge zu größeren Problemen oder Verletzungen führen. Am Ausgang des FI-Schalter ist zunächst eine Schuko-Steckdose angeschlossen, welche für das Netzteil des Raspberry Pi genutzt wird. Ebenfalls am Ausgang des FI-Schalters angeschlossen sind insgesamt fünf Sicherungsautomaten. Jeder dieser Sicherungsautomaten sichert drei Netzteile ab. Für jeden Sicherungsautomaten gilt anschließend, dass der Ausgang an einem Relais angeschlossen ist, welches ein verzögertes Einschalten der Netzteile ermöglicht. Die Relais werden im Abstand von 2 Sekunden nacheinander eingeschaltet. Hierdurch wird die Spannungsspitze beim Einschalten der Netzteile entzerrt und die Sicherung der Zuleitung zum Schaltkasten löst nicht mehr aus. Nach den Relais führt jedes Relais weiter in drei Schmelzsicherungen und anschließend zu je einem Netzteil. Der Materialliste ist zu entnehmen welche Arten von FI-Schalter, Sicherungsautomaten, Relais und Schmelzsicherungen verwendet wurden. Als Leitungen für die Schaltschrank-Verkabelung wurden 1,5mm² Leitungen verwendet. Die Ausführung zu den Netzteilen ist je ein „Gummikabel“ mit 3x1,5mm² wobei die Leitung zur Erdung ungenutzt ist.

Kasten

Der Kasten ist ein einfacher Schaltschrank mit IP65 Zertifizierung. Sämtliche Einführungen wurden von unten her durchgeführt und mit Quetsch-Einführungen wurde die Einführung dicht gemacht.

RaspberryPi

Der RaspberryPi ist an der erwähnte Schuko-Steckdose angeschlossen und wird von dort mit Strom versorgt. Auf dem Raspberry Pi selbst muss der GPIO-Pin 18 (Physischer Port 12) mit der Datenleitung der LED-Strips verbunden werden. Der physische Pin 14 mit der Masse. An dem GPIO Pin 18 ist dann eine Verbindung zu einem vom Typ SN74HCT125N. Dieser Chip kann das Eingangssignal, welches vom Raspberry Pi mit 3,3V gesendet wird, auf 5V hochtransformieren, sodass die LED-Stips damit „zufriedener“ sind. Dies ist ins Besondere unter dem Aspekt der Leitungslängen zu empfehlen. Es KANN auch ohne klappen - ins Besondere bei kurzen Leitungslängen. Es KANN jedoch auch nicht ausreichend sein. Sollte es nicht ausreichend sein, muss vor der ersten LED am Display noch ein Pull-Down-Widerstand eingebracht werden. Dieser zieht Störsignale dann runter und die LEDs können meist das Signal wieder verarbeiten. Das Schema der Verkabelung des SN74HCT125N ist hierbei wie folgt:



fritzing

Netzteile

Die verwendeten Netzteile, welche die LED-Strips und den SN74HCT125N mit 12A bei 5V mit Strom versorgen, sind im Schaltkasten hinter dem Schaltschrank verbaut. Die Netzteile sind IP67 zertifiziert, können also quasi beliebig viel Wasser ab. Ein Netzteil reicht mit den 12A Leistung für 4 LED Strips, wodurch wir auf eine Gesamtmenge von 15 Netzteilen kommen. Im Falle, dass das Display niemals in feuchten Umgebungen oder im Außenbereich eingesetzt wird, kann man auch ein ungeschütztes Netzteil verwenden. Diese gibt es mit deutlich mehr Leistung bei 5V und sind dadurch erheblich günstiger und vor allem auch leichter. Diese Netzteile sind aber offen mit Lüftungslöchern, wodurch selbst bei leichter Feuchtigkeit ein erhebliches Kurzschlussrisiko besteht. Die von uns verwendeten Netzteile haben neben der IP67 Zertifizierung auch einen Überlastschutz und einen Kurzschlusschutz. So werden in Regelfall größere Schäden verhindert.

Verkabelung

Die Verkabelung vom Schaltschrank zu den Netzteilen erfolgt in „Gummileitungen“. Die Verbindung zwischen Netzteil und den Adern der Leitung ist mit einem Quetschverbinder umgesetzt, welcher nach erfolgreichem Verbinden mit einem Heißluftföhn erhitzt wurde. Hierdurch sind die Quetschverbinder geschrumpft und sind dicht am Kabel anliegend. Nach Herstellerinformationen sollen die Quetschverbinder somit Wasserdicht sein. Zusätzlich zu den Quetschverbindern ist noch ein Schrumpfschlauch über die Verbinder gezogen. Dies dient zum Einen dem weiteren Schutz vor Feuchtigkeit und zum anderen der verbesserten Optik.

Rahmen

Der Rahmen des Schaltkastens besteht, wie auch das Display, aus 3030-Aluminiumprofilen (30mm x 30mm Kantenlänge). Diese wurden in entsprechender Größe bereits zugeschnitten bestellt und mit Winkel- und Eckverbindern verbunden.

Verbindungskabel

Das Verbindungskabel ist ein Kabel für die Verbindung von Anhängern zu Autos. Hierdurch ist es auch wasserfest. Die Kabel haben insgesamt 13 Adern wovon 9 Stück 1,5mm² dick sind und 4 Stück 2,5mm². Wichtig: Es gibt diese Kabel auch mit 13*1,5mm². Hier muss unbedingt auf die richtige Ausführung geachtet werden! An beiden Enden der Kabel ist jeweils ein Stecker montiert. Hierbei wurde die bereits in der Tabelle oben beschriebene Belegung umgesetzt. Durch die Belegung der beiden Kabel ist es auch möglich die Kabel zu vertauschen oder zu kreuzen, da z.B. auf Pin 1 in beiden Fällen Plus anliegt, auf Pin 5 in beiden Fällen Minus und die Pins 9, 10, 11 und 13 die dickeren Kabel sind sowie Pin 8 die Datenleitung ist. Die Datenleitung selbst ist im Schaltkasten an beiden Steckdosen angelegt, im Panel aber nur an einer. So läuft das Signal sicher immer durch eine Leitung zum Display ohne dass Interferenzen o.ä. entstehen. Wir haben zwei Ausführungen der Verbindungskabel hergestellt. Eine 1m-Version für den Fall, dass Display und Schaltkasten nahe beieinander stehen und eine 5m-Version für den Fall, dass diese etwas entfernt voneinander stehen (z.B. bei Schwimmbecken).

Materialliste

Folgt...