

Bogenampel mit Vollfarb-Display

Von Rudi Meyer

Motivation:

Unsere Bogensportabteilung wünschte sich für die Sommersaison 2 Bogenampeln. Da für unseren kleinen Verein eine Investition von rund 2000.-€ für Fertiggeräte keine Option war, hat man mich gebeten, mit der Selbstbau-Anleitung 2 entsprechende Anzeigen zu erstellen. Mit den Informationen und Hinweisen von Andres unter <https://www.bogenampel.de/> habe ich mich mit mäßigem Erfolg auf die Suche nach entsprechenden Displays gemacht. Bei der Suche nach P10 LED Modulen bin ich dann schnell auf die Vollfarb-Module gestoßen. Um sicher zu gehen, ob ich hier nicht Geld in eine am Ende unbrauchbare Lösung zu stecken, habe ich mit meiner Idee dann Andreas und Edmund von <http://www.purecontest.de> bzw. <http://bogenampel.de> angeschrieben, ob den eine Adaption meiner Ampeln an Edmunds PureContest möglich ist.

Da diese Art der Displays eigentlich für riesige Werbetafeln hergestellt werden, bieten die Hersteller in Fernost Controller und zugehörige SW an, die zwar in der Lage sind auch Videos abzuspielen, aber immer nur einmal programmiert werden und dann in Endlosschleife laufen. Das Datenübertragungsformat zwischen PC und den LED-Controllern wird auch nicht offengelegt. Somit war diese Idee eine Sackgasse.

Nach einigem Suchen bin ich dann auf ein Projekt auf <https://github.com> aufmerksam geworden, bei dem P10 vollfarb-Module mit einem Raspberry Pi angesteuert werden.

Nachdem Edmund mir seine Unterstützung durch seine SW zugesagt hat, wenn ich den die Ansteuerung der Displays in den Griff bekomme, habe ich mich entschieden diesen Weg zu verfolgen. Getrieben durch die Aussage unseres Bogensporttrainers, dass die Anzeigen aber auch bei hellem Sonnenschein noch gut erkennbar sein müssen, habe ich mich bei Alibaba auf die Suche nach preiswerten und hellen Modulen gemacht. Im ersten Ansatz bin ich auf preiswerte Module gestoßen (https://wholesaler.alibaba.com/product-detail/P10mm-Indoor-32x16pixel-SMD-Stage-LED_60226235102.html?spm=a2700.7724838.0.0.R3aEi0). Da diese Module explizit als Indoor ausgewiesen sind, habe ich nach entsprechenden Outdoor – Modulen gesucht, schließlich müssen die Ampeln auch mal bei Training oder Wettkampf im Regen draußen stehen. Nach einigen Recherchen bin ich dann auf dieses Angebot gestoßen: https://www.alibaba.com/product-detail/RGB-Full-Color-SMD-P10-Led_60526829057.html

Das sind Module, die alle meine Anforderungen erfüllen: P10 Vollfarbe, Temperatur- und Luftfeuchtebereiche, sowie die CE-Kennzeichnung, damit sie auch durch den Zoll kommen. Auf Anfrage haben die dann auch Kabelsätze für Stromversorgung und Datenkabel für je 0,50\$ sowie 2 MEANWELL – Netzteile 5V / 40A für je 13\$ mitgeliefert.

Aber Achtung: zu den Listenpreisen kommen nicht nur die nicht unerhebliche Transportkosten, sondern auch Zollgebühren, Einfuhrumsatzsteuer und die Bearbeitungsgebühren des Transporteurs für die Erledigung der Zollformalitäten.

In der Zwischenzeit, bis die Lieferung aus China ankommen würde (dafür sollte man ca. 3 – 4 Wochen einplanen), konnte ich mich dann schon um den Raspi und die Verdrahtung kümmern.

Dazu habe ich auf die Infos auf <https://github.com/hzeller/rpi-rgb-led-matrix> zurückgegriffen. Da in unserem Fall ja nur 2 Zeilen LED-Module angesteuert werden sollen, habe ich auch nur die ersten beiden Wannenstecker auf der Prototyp-Platine verdrahtet.

Die meisten Teile habe ich bei Reichelt bestellt. Dort sind so ziemlich alle Zubehörteile erhältlich und die Bestellung wird in der Regel innerhalb von 3 Tagen geliefert. Das RASP SHD PROTO ist hier aber für unsere Zwecke eher nicht brauchbar, das habe ich hier <http://www.ebay.de/itm/262515464263> deutlich günstiger und mit sehr viel mehr freiem Platz bestellt.

Aufbau der Elektronik

Als erstes ist die Prototyp-Platine mit den Buchsen für die Ansteuerung der LED-Displays zu verdrahten. Dazu sollte man schon in der Lage sein, mit dem Lötcolben kleinere Projekte problemlos zu meistern.

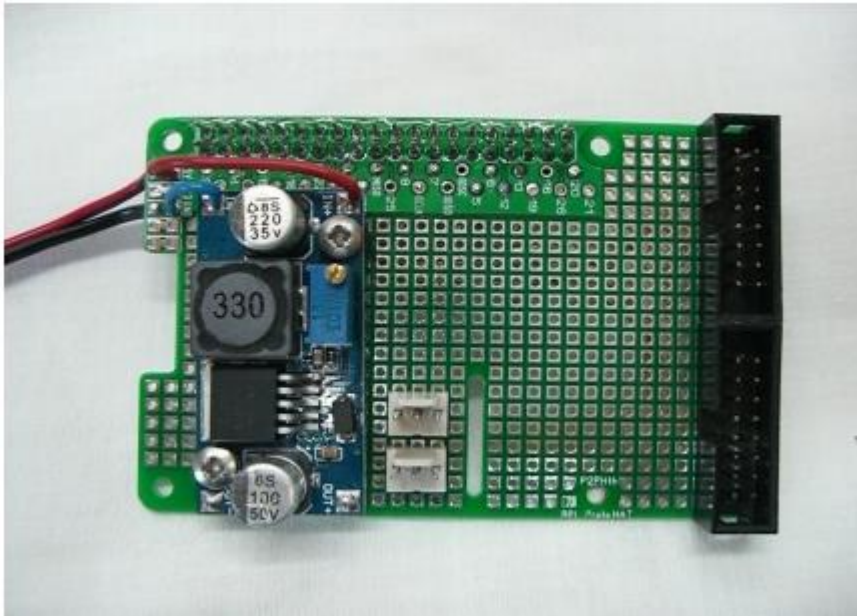
Schließlich haben beide Wannenstecker für die Ansteuerung der beiden Displayzeilen jeweils 16 Kontakte, die alle verdrahtet werden wollen. Dazu ist dann noch die Federleiste für den Anschluss auf den Raspi zu verlöten.

Die beiden 3-poligen Printstecker für die Lüfter und der Step-Up Konverter um aus den 5V des Netzteils 12V für die Lüfter zu generieren habe ich als letztes aufgebaut, nachdem die Funktion der Displays erreicht war.

Hier der Link zu der Verdrahtung der Display-Steuerleitungen:

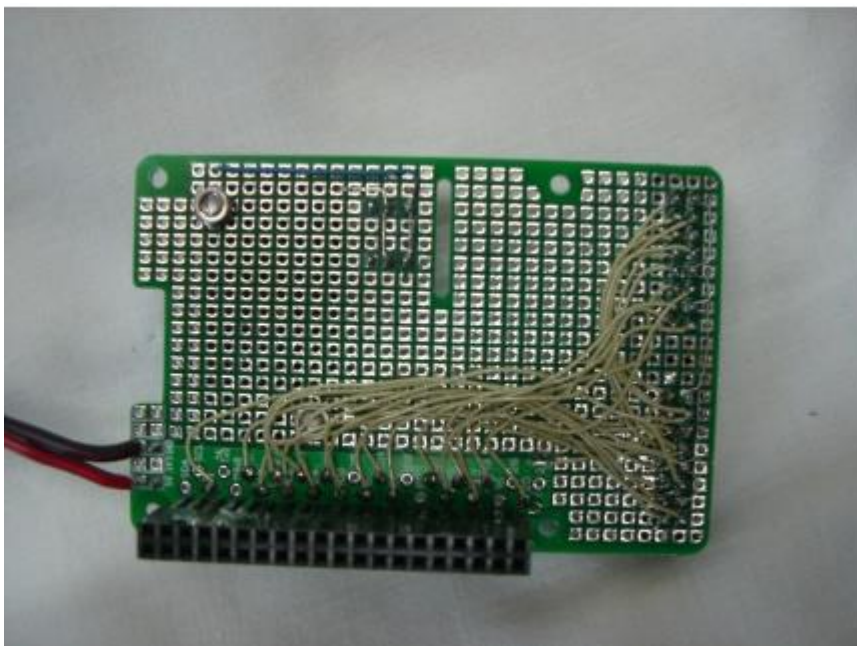
<https://github.com/hzeller/rpi-rgb-led-matrix/blob/master/wiring.md#chaining-parallel-chains-and-coordinate-system>

Und hier die Oberseite von meinem Prototyp:

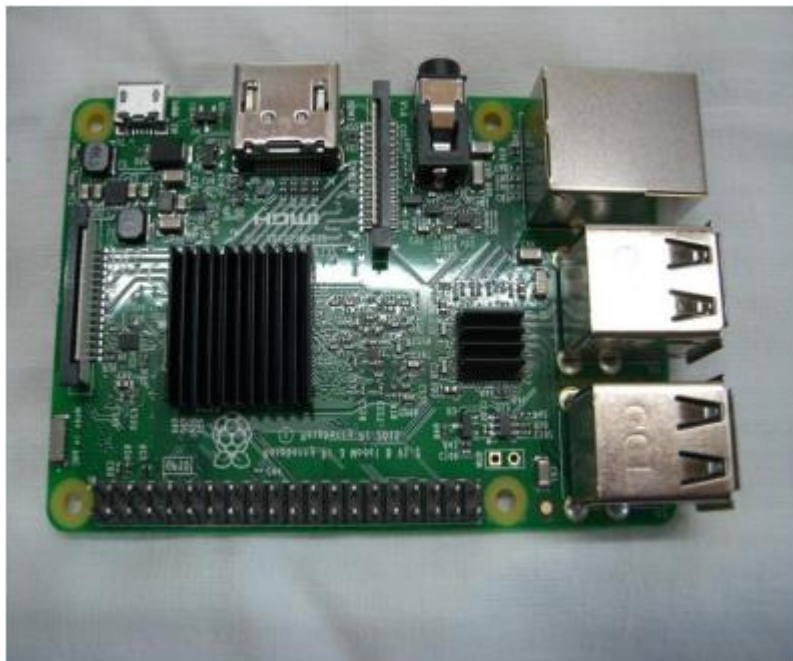


Links der Spannungsregler, um aus den 5V des Netzteils 12V für die Lüfter zu generieren, daneben die beiden Stecker für die Lüfter und am rechten Rand die Stecker für die Steuerleitungen der LED-Module.

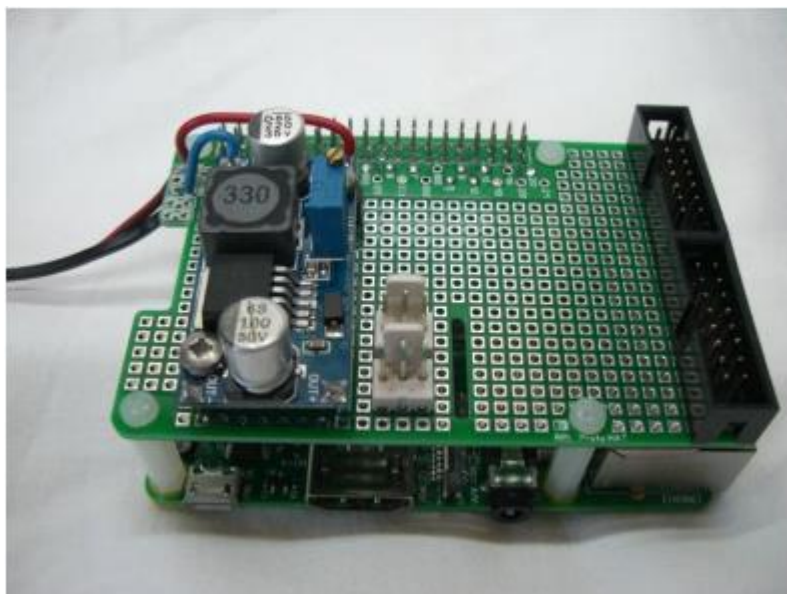
Auf der Unterseite ist der „Drahtverhau“ zu sehen:



Hier der Raspberry Pi 3B mit den aufgeklebten Kühlkörpern (der dritte sitzt auf der Unterseite)



Eine Anleitung für den Umbau mit externer WLAN-Antenne habe ich unter <https://hackaday.io/project/10091-raspberry-pi-3-external-antenna> gefunden. Beide Baugruppen zusammengesteckt und verschraubt sieht dann so aus:

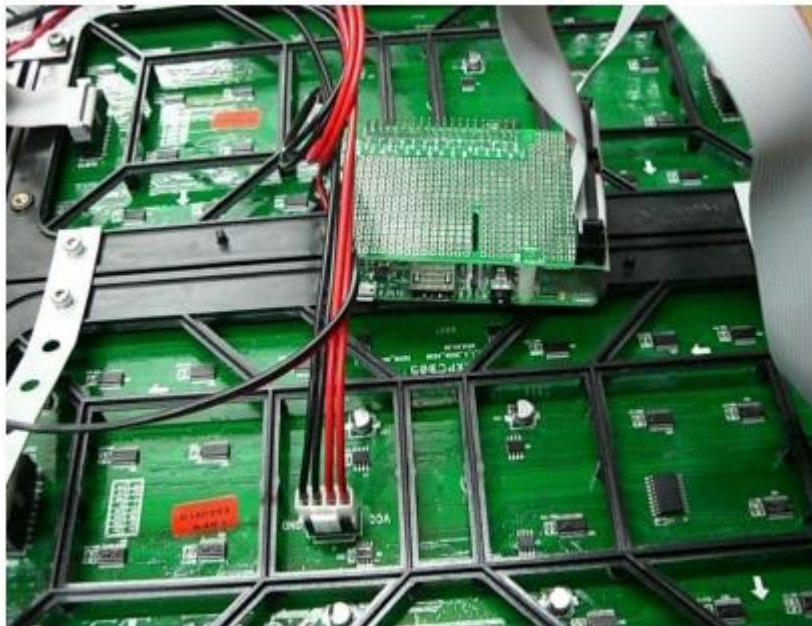


Die Verdrahtung der Module ist dank der vorgefertigten Kabel kein Problem, wenn man die Richtungspfeile und Beschriftung für Ein- und Ausgänge auf den Modulen berücksichtigt.



Da ich noch kein Gehäuse hatte, habe ich die Module mit Stücken von Lochblechstreifen zu einem kompletten Display zusammengeschaubt.

Mit den Steckern für die Daten und der Stromversorgung kann man eigentlich nichts falsch machen, lediglich auf die Polarität beim Verbinden mit dem Netzteil sollte man aufpassen, ein verpolen der Spannung nimmt einem die Elektronik meistens recht übel und signalisiert das dann häufig durch „Rauchzeichen“.

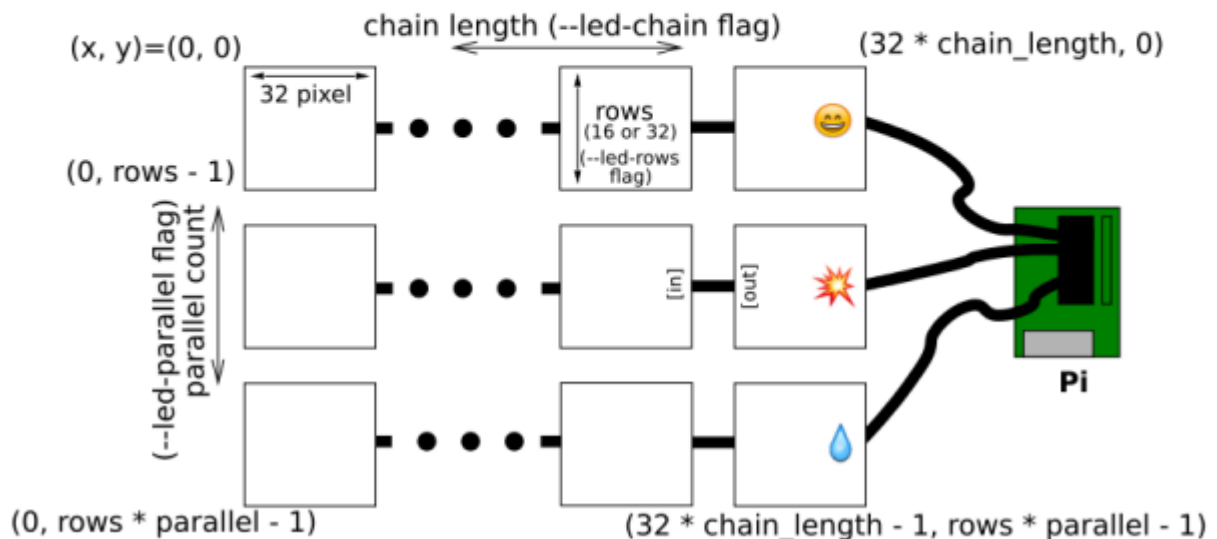


An den äußersten Dateneingangsstecker wird dann der Raspberry Pi angeschlossen.

Verdrahtung der Stecker auf der Prototyp-Platine

☺ = obere Panelreihe; ✨ = mittlere Panelreihe; 💧 = untere Panelreihe

Connection	Pin	Pin	Connection
-	1	2	-
💧 [3] G1	3	4	-
💧 [3] B1	5	6	GND ☺ ✨ 💧
☺ ✨ 💧 strobe	7	8	[3] R1 💧
-	9	10	E ☺ ✨ 💧 (for 64 row matrix, 1:32)
☺ ✨ 💧 clock	11	12	OE- ☺ ✨ 💧
☺ [1] G1	13	14	-
☺ ✨ 💧 A	15	16	B ☺ ✨ 💧
-	17	18	C ☺ ✨ 💧
☺ [1] B2	19	20	-
☺ [1] G2	21	22	D ☺ ✨ 💧 (for 32 row matrix, 1:16)
☺ [1] R1	23	24	[1] R2 ☺
-	25	26	[1] B1 ☺
-	27	28	-
✨ [2] G1	29	30	-
✨ [2] B1	31	32	[2] R1 ✨
✨ [2] G2	33	34	-
✨ [2] R2	35	36	[3] G2 💧
💧 [3] R2	37	38	[2] B2 ✨
-	39	40	[3] B2 💧



Für die hier verwendeten Module wird der Pin 10 des Raspis nicht benötigt, für viele Outdoor – Module mit „1/4 Scan“ wird auch der Pin 22 nicht benötigt.

Für die logische Ansteuerung haben die Panels dann die doppelte Länge (64) und die halbe Höhe (8). Das muss die SW dann mit einem Transformer wieder richtig hinbiegen.

Achtung: die Indoor-Module mit „1/16 Scan“ werden nativ von der SW unterstützt, da gibt es offensichtlich nur eine definierte Reihenfolge. Bei den „1/4 Scan“ – Modulen hat offensichtlich jeder Hersteller (ich vermute aus Lizenzgründen) eine andere Philosophie der Ansteuerung. Jedenfalls füllen die Anfragen wegen nicht richtig laufender Anzeigen mehrere Webseiten auf github. Meine Module zum Laufen zu bringen hat mich etliche Stunden mit suchen und probieren gekostet. Bei manchen Modulen sind zusätzlich dann auch noch Farben vertauscht.

Software – Installation

Als erstes gilt es, das Betriebssystem auf die SD-Karte zu bringen.

Als Betriebssystem habe ich mich für das Raspbian aus dem NOOBS -Packet von der offiziellen <https://www.raspberrypi.org/downloads/> Web-Seite entschieden.

Im dem heruntergeladenen ZIP-Ordner findet man eine Anleitung, wie das Betriebssystem auf die SD-Karte gelangt.

Während des Downloads kann man schon den Raspi mit Tastatur, Maus und Monitor sowie der Stromversorgung und dem Netzwerk verbinden, aber noch nicht einschalten.

Nachdem die SD-Karte nach Anweisung hergerichtet ist, kann dann der Anleitung folgend das Betriebssystem installiert werden.

Nach dem erfolgreichen ersten Start des Systems bringen wir dieses auf den neuesten Stand.

Dazu ein Konsolenfenster öffnen und mit dem Befehl `sudo apt-get update` das Update starten.

Bleibt noch das Upgrade der installierten Pakete mit `sudo apt-get dist-upgrade`.

Damit ich später evtl. Einstellungen oder SW-Updates am Raspi vornehmen kann, ohne das Gehäuse öffnen zu müssen, habe ich noch den SSH-Server eingerichtet bzw. aktiviert. Damit habe ich dann über die Netzwerkschnittstelle Zugriff auf das System.

Da die eingebaute Sound-Schnittstelle die gleichen Ressourcen wie die Displaysteuerung braucht, müssen wir den Treiber dafür stilllegen, außerdem soll ja die Software nach dem Einschalten automatisch starten. Hierfür sind 2 Dateien an die richtige Stelle im System zu erstellen, sowohl diese Script-Dateien als auch die ausführbare Datei für die Ansteuerung des Displays müssen ausführbar gemacht werden.

Die Befehle hierfür habe ich in einem Script zusammengefasst, das im SW-Paket mit enthalten ist.

Das entpackte SW-Paket kann einfach von einem USB-Stick ins Home-Verzeichnis des Standardbenutzers pi kopiert werden.

Gestartet wird das Script auch in einem Konsolenfenster mit `bash ampel/system/setup.sh`.

Am Ende des Scripts ist ein Neustart des Systems, damit alle Modifikationen, insbesondere die Einstellungen für die Sound-HW übernommen werden.

Nach ca. 45 Sek. Sollte die Anzeige dann gestartet sein und die IPs für Ethernet und WLAN anzeigen, sofern verbunden bzw. eingerichtet ist. Diese IP-Adresse ist dann bei PureContest in das entsprechende Feld mit der Portnummer 1337 einzutragen und der Haken für die Vollfarbdisplays zu setzen.

Sobald eine gültige IP-Adresse (Ethernet oder WLAN) gefunden wurde, geht die SW in den Servermode für die PureContest – Daten.

Unser Bogensporttrainer und die Schützen selbst waren jedenfalls vom Ergebnis restlos begeistert.