

Arduino-Projekt: LED-Armband vom Taylor-Swift-Konzert wieder leuchten lassen

| 07.11.2024 14:30 Uhr Jack Pop, Anne-Dorette Ziems



Blinkende LED-Armbänder sind bei vielen Events zu finden. Wer sein Armband auch zu Hause ansteuern möchte, braucht zwei Komponenten und wenige Minuten Zeit.

Die Fan-Blöcke funkeln wie Sterne, dann erstrahlt die gesamte Menschenmenge rot: Zahlreiche Videos der Publikumslightshow von Taylor Swifts "The Eras Tour" haben die sozialen Medien geflutet und neues Interesse für LED-Konzertarmbänder geweckt. Viele Besucherinnen und Besucher nehmen solche Gadgets als Erinnerungsstücke mit nach Hause. Das Problem: Die Steuersignale sind proprietär und die Armbänder scheinen außerhalb des Stadions nutzlos.

MEHR ZU ARDUINO-PROJEKTE ▲

Arduino-Projekt: LED-Armband vom Tailor-Swift-Konzert wieder leuchten lassen [1]

Tischtennisuhr mit Arduino und 3D-Drucker selbst bauen [2]

Bastel-Projekt mit Arduino Nano: Head-Up-Display für Skifahrer [3]

Spracherkennung mit Arduino nano connect realisieren [4]

Mit nur zwei Elektronikkomponenten und wenigen Arbeitsschritten kann man sie aber wieder zum Leuchten bringen. Denn seit zwei Jahren sammelt eine enthusiastische Maker-Community die nötigen Steuerbefehle im Netz zusammen.

C'T KOMPAKT

- Die leuchtenden Armbänder von Taylor-Swift-Konzerten oder anderen Großevents werden mithilfe von Infrarotstrahlung angesteuert und kontrolliert.
- Ein Infrarot-Transmitter gesteuert von einem Arduino bringt die Armbänder auch zu Hause wieder zum Leuchten.
- Skripte und sogar Chatbots, die beim Programmieren der Armbandsteuerung helfen, erstellt das PixMob-Reverse-Engineering-Projekt, eine globale Maker-Community.

Auf GitHub [5] stellt sie ein äußerst umfangreiches Repository mit verschiedensten Anwendungsmöglichkeiten zur Verfügung . Hier finden sich zum Beispiel simple Pythonskripte und Programmskripte für Arduino und ESP32, sogenannte Sketches.



Taylor Swifts "The Eras Tour" war die kommerziell erfolgreichste Tour überhaupt. Doch nicht nur sie nutzt Leuchtarmbänder, sondern auch Coldplay und die Veranstalter der Olympischen Spiele in Paris.
(Bild: Natasha Leake/PA Wire/dpa)

PROJEKTINFO

Mit wenigen Handgriffen das LED-Armband vom letzten Taylor-Swift-Konzert wieder zum Leuchten bringen

Benötigte Kenntnisse: Programmieren, Arduino-IDE

Zeitaufwand: 5 bis 15 Minuten

Kosten: 25 Euro

Materialien: Pixmob X2 Armband, Arduino Nano (oder ähnliches Board), 5 V-Infrarot-Transmittermodul

Werkzeug: Keins

Für Gelegenheitsbastler besteht die einfachste Variante aus einem Arduino, einem Infrarot-LED-Modul und einem auf [GitHub](#) verlinkten **Webinterface** [6], das die Steuersignale integriert hat. Damit konnten wir nach nur zehn Minuten die Armbänder wieder leuchten lassen. Zu Beginn noch ein Hinweis: Das GitHub-Projekt ist auf die Armbänder von Marktführer PixMob spezialisiert, die nicht nur bei Taylor Swift, sondern auch bei der aktuellen Coldplay-Tournee oder den Olympischen Spielen in Paris zum Einsatz gekommen sind. Wir haben das Projekt allerdings mit dem Eras-Tour-Modell, X2, getestet.



Die vor der Show verteilten LED-Armbänder nehmen viele Fans als Erinnerung mit nach Hause.

(Bild: Marius Becher, dpa)

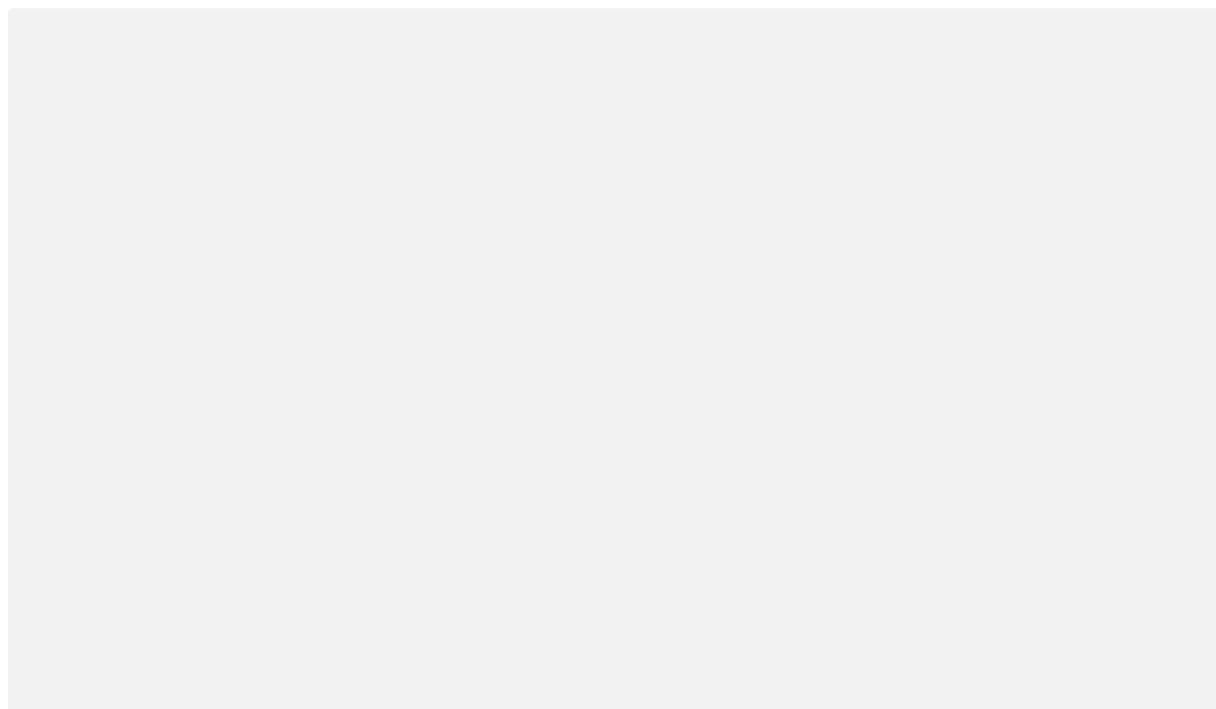
Reverse Engineering

Um herauszufinden, welche Infrarotsignale die Armbänder zum Leuchten bringen, haben die Mitglieder des **PixMob-Reverse-Engineering-Projekts** [7] die Signale, die während verschiedener Konzerte an die Armbänder gesendet wurden, mit speziellen Empfangsgeräten aufgezeichnet. Allerdings haben im Test von über 100 registrierten Signalen nur drei eine direkte Reaktion des Armbands hervorgerufen.

Die GitHub-Community vermutete, dass die meisten Signale, die während einer Show gesendet werden, keine direkten Befehle für bestimmte Farben sind, sondern Steuercodes. Diese signalisieren dem Armband, was zu tun ist, wenn es nur bruchstückhafte Farbbefehle erhält.

Immerhin konnte die Community so aber rekonstruieren, wie die Signale grundsätzlich aufgebaut sind. Mit einem Brute-Force-Ansatz – das heißt durch wildes Ausprobieren – hat die Gruppe dann weitere Befehle für verschiedene Farben und Modi herausgefunden.

Das PixMob-Reverse-Engineering-Projekt entwickelt sich ständig weiter, denn auch der Armbandhersteller PixMob veröffentlicht fortlaufend neue Generationen der Armbänder und implementiert neue Effekte. Wenn Sie jetzt die Motivation gepackt hat, können Sie selbst am PixMob-Reverse-Engineering-Projekt teilnehmen, indem Sie zum Beispiel Fotos ihres eigenen restaurierten Leuchtarmbands auf den Discord-Server hochladen oder Abwandlungen des Web-Interfaces im GitHub-Repository anbieten.

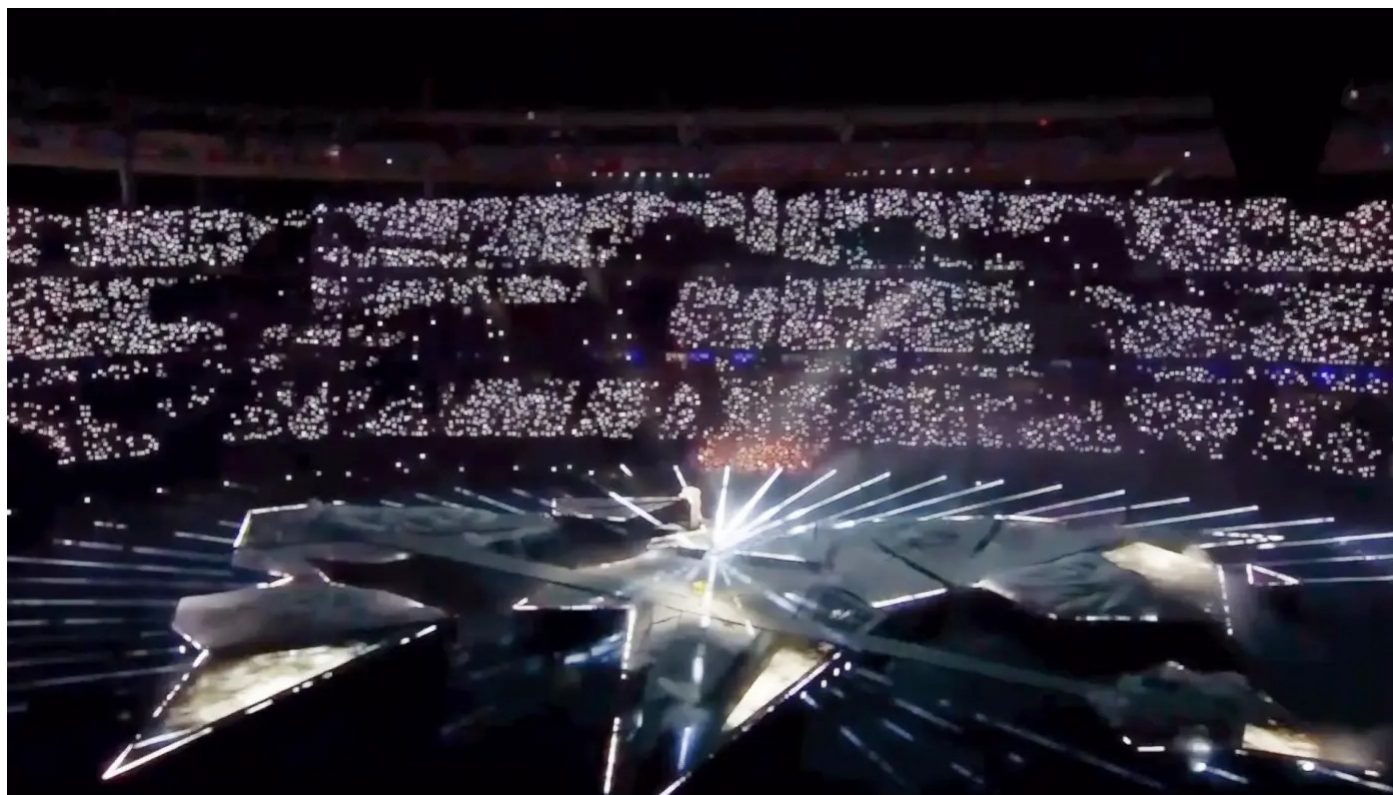


[8]

Zwei Hardwarekomponenten und Datenübertragung

Da die Übertragung der Farbsignale beim Konzert über Infrarotstrahler funktioniert, mussten wir einen solchen zunächst klein nachbauen. Dafür genügen zwei Komponenten: Als Steuereinheit

haben wir uns für einen Arduino Nano entschieden (ähnliche Modelle anderer Anbieter funktionieren in der Regel jedoch auch) und als Signalübermittler für ein 5-Volt-Infrarot-Transmittermodul.

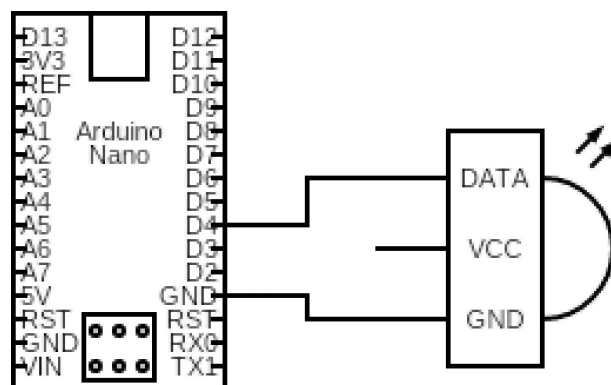


Während der Abschlusszeremonie der Olympischen Spiele in Paris 2024 wurde das Publikum dank Infrarotarmbändern zur lebenden Leinwand.

(Bild: Screenshot LinkedIn / PixMob)

Grundsätzlich lässt sich das Transmittermodul auch durch eine einzelne Infrarot-LED ersetzen, die als Cent-Artikel sogar noch preiswerter ist. Allerdings kommt diese Variante mit einigen technischen Hürden: Die Bastlerin oder der Bastler muss den Vorwiderstand berechnen und in die Schaltung integrieren, damit die LED im Betrieb nicht durchbrennt. Außerdem muss man die Stromstärke der LED berücksichtigen, um den Arduino nicht zu überlasten.

Wer sich das ersparen und den Griff zum Lötkolben vermeiden möchte, kommt mit dem Transmittermodul am schnellsten ans Ziel; dieses kann man direkt mit dem Arduino verbinden. Das dreipolige Verbindungskabel war bei unserem Modul im Lieferumfang enthalten, aber man kann Module mit freiliegenden Anschlüssen auch mit einzelnen Jumper-Kabeln verbinden.

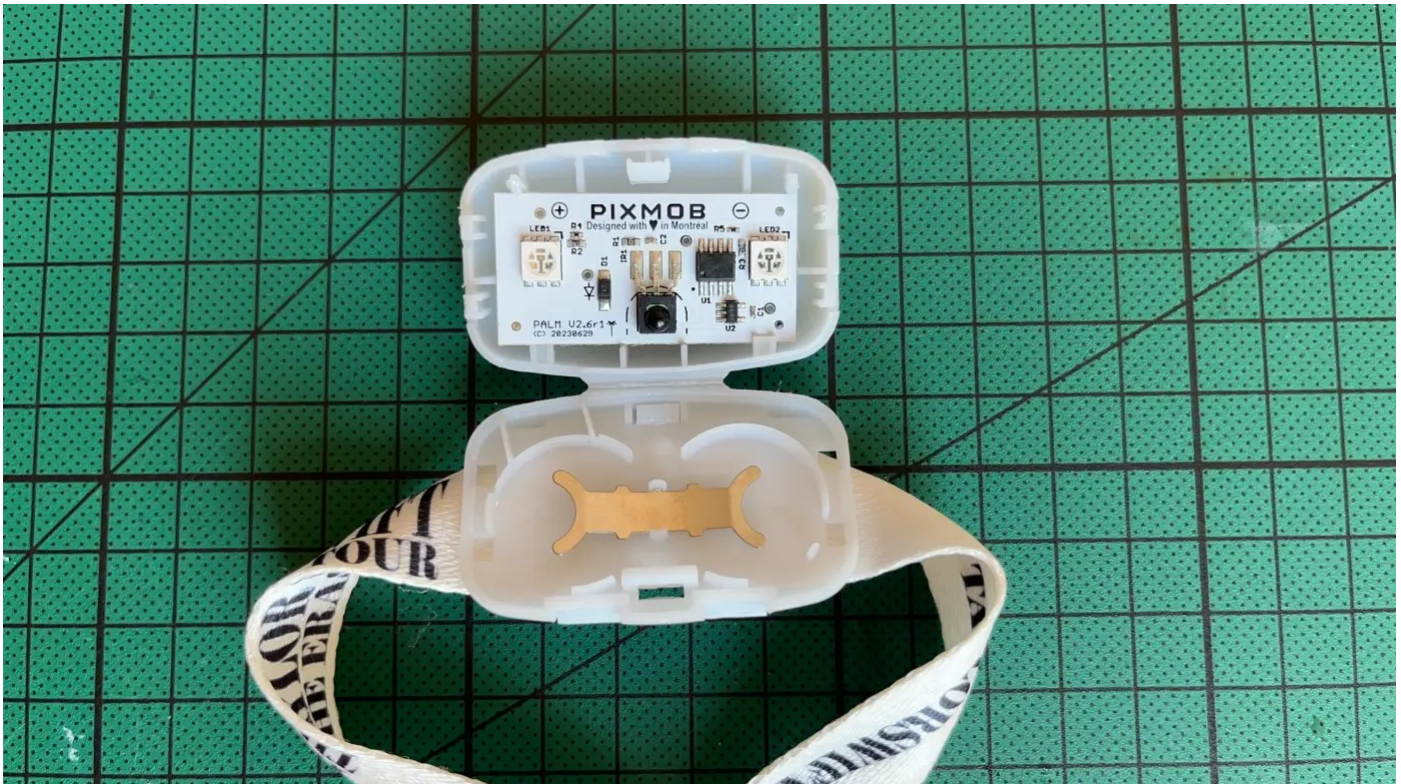


Die Schaltung zwischen Arduino und Transmittermodul benötigt nur zwei Kabel.
(Bild: Circuit-diagram.org)

Hat man beide Komponenten beisammen, ist der Anschluss fast selbsterklärend: Das Verbindungskabel wird so an das

Transmittermodul gesteckt, dass die Ader GND an einem der beiden GND-Pins des Arduino steckt. Die Ader VCC kann ignoriert werden. Diese regelt üblicherweise die Versorgungsspannung von Bauteilen mit größerer Last, der Stromfluss zur LED vom Arduino ist aber schon über die DAT-Ader gewährleistet.

Die DAT-Verbindung wird am Arduino auf den Pin D4 gesteckt. Prinzipiell können auch andere freie Output-Pins des Arduino genutzt werden, Pin 4 ist lediglich die Voreinstellung im Skript. Wer einen anderen Anschluss-Pin bevorzugt, muss später im Programmskript die Pin-Nummer im Code ändern.



Die LED-Armbänder sind im ständigen Empfangsmodus und verbrauchen somit auch Strom, wenn sie nicht leuchten. Im Zweifelsfall kann man das Plastikgehäuse öffnen und die Batterien tauschen.

(Bild: Jack Pop)

Datenübertragung per Infrarot

Infrarotlicht gehört zur elektromagnetischen Strahlung und hat weniger Energie als sichtbares rotes Licht. Infrarot-LEDs sind beispielsweise in Fernbedienungen für Fernseher und andere elektronische Geräte verbaut und funktionieren über kurze Distanzen. Wer also damit den Fernseher einschalten will, darf nicht allzu weit von ihm entfernt stehen. Gleiches gilt für die Eras-Tour-Armbänder – vor allem in dieser Bastelvariante für zu Hause.

In den Stadien steuern große Scheinwerfer mit Dutzenden Infrarot-LEDs die Lichtershow im Publikum. Wir nutzen hingegen nur eine einzelne LED, deren Reichweite geringer ist. In unserem Setup können wir das Armband aus fünf Metern Entfernung noch zuverlässig steuern. Ideal ist dabei direkter Sichtkontakt zwischen Sender und Empfänger.

Ob mit den großen Scheinwerfern auf den Konzerten oder im kleinen Setup zu Hause: Die grundlegende Funktionsweise der Armbänder ist gleich und wird schon seit vielen Jahren von unterschiedlichen Anbietern eingesetzt. Die Infrarot-LEDs verschicken die Befehle für die entsprechenden Farben; der Infrarotsensor im Armband empfängt sie. Grundsätzlich gibt es verschiedene Möglichkeiten, das Signal zu modulieren. Bei den Eras-Tour-Armbändern funktioniert das mit einer Art Morsecode, für den die Infrarot-LED im Millisekundentakt strahlt. Jedes Farbsignal entspricht einer bestimmten Abfolge.

In den Stadien werden dafür in zwei Rotationsachsen bewegliche Scheinwerfer genutzt. Diese sogenannten Moving Heads können das Infrarotsignal zielgerichtet in ganz bestimmte Bereiche im Publikum senden und so nicht nur alle Armbänder in der gleichen Farbe aufleuchten lassen, sondern räumliche Muster erzeugen.

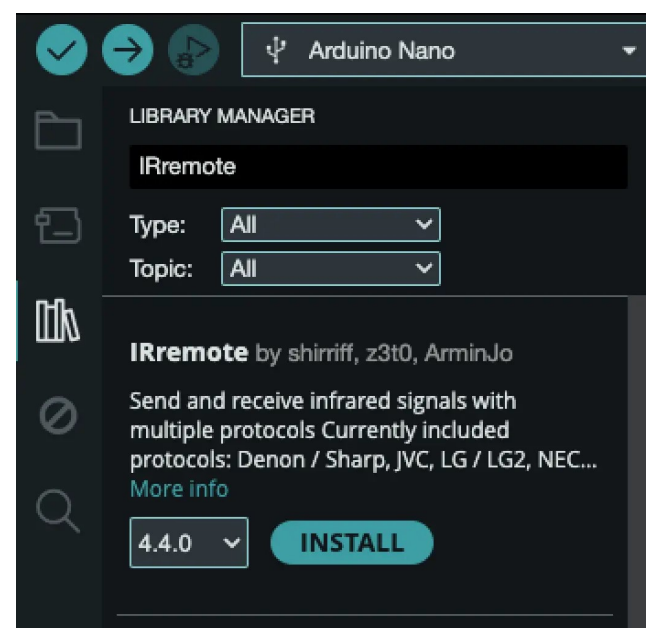
Die ganze Bandbreite an Möglichkeiten hat sich zuletzt bei den Olympischen Spielen in Paris gezeigt. Dort haben neuartige Moving Heads mehrere Infrarotsignale simultan übertragen und das Publikum dadurch wie ein Bildschirm mit "menschlichen Pixeln" genutzt: Hunderte Armbänder haben riesige Sportsilhouetten erzeugt, zum Beispiel einen Pferdewagen.

Der Arduino-Sketch und Steuerung

Ist alles zusammengesteckt, kann der Arduino per USB-Kabel mit dem Computer verbunden werden. Wer die frei verfügbare Arduino-IDE noch nicht auf dem Rechner hat, muss **diese zunächst herunterladen [9]** und installieren. Für Arduino-Neulinge gibt es **kurze, grundlegende Erklärungen [10]** für die Verbindung zwischen IDE und dem Arduino.

Nun fehlt noch der Arduino-Sketch: ein Skript, das die serielle Kommunikation zwischen Computer und Arduino überwacht, eingehende Befehle in Sendesignale für das Infrarotmodul übersetzt und dieses ansteuert. Dieses Skript findet man **im GitHub-Repository [11]** im Ordner "arduino_sender/PixMob_Transmitter_Arduino".

Bevor man dieses auf den Arduino lädt, gibt es allerdings noch zwei Dinge zu prüfen: zum einen, ob in Zeile 39 tatsächlich der korrekte Daten-Pin hinterlegt ist. Die Voreinstellung sollte hier "4" sein.



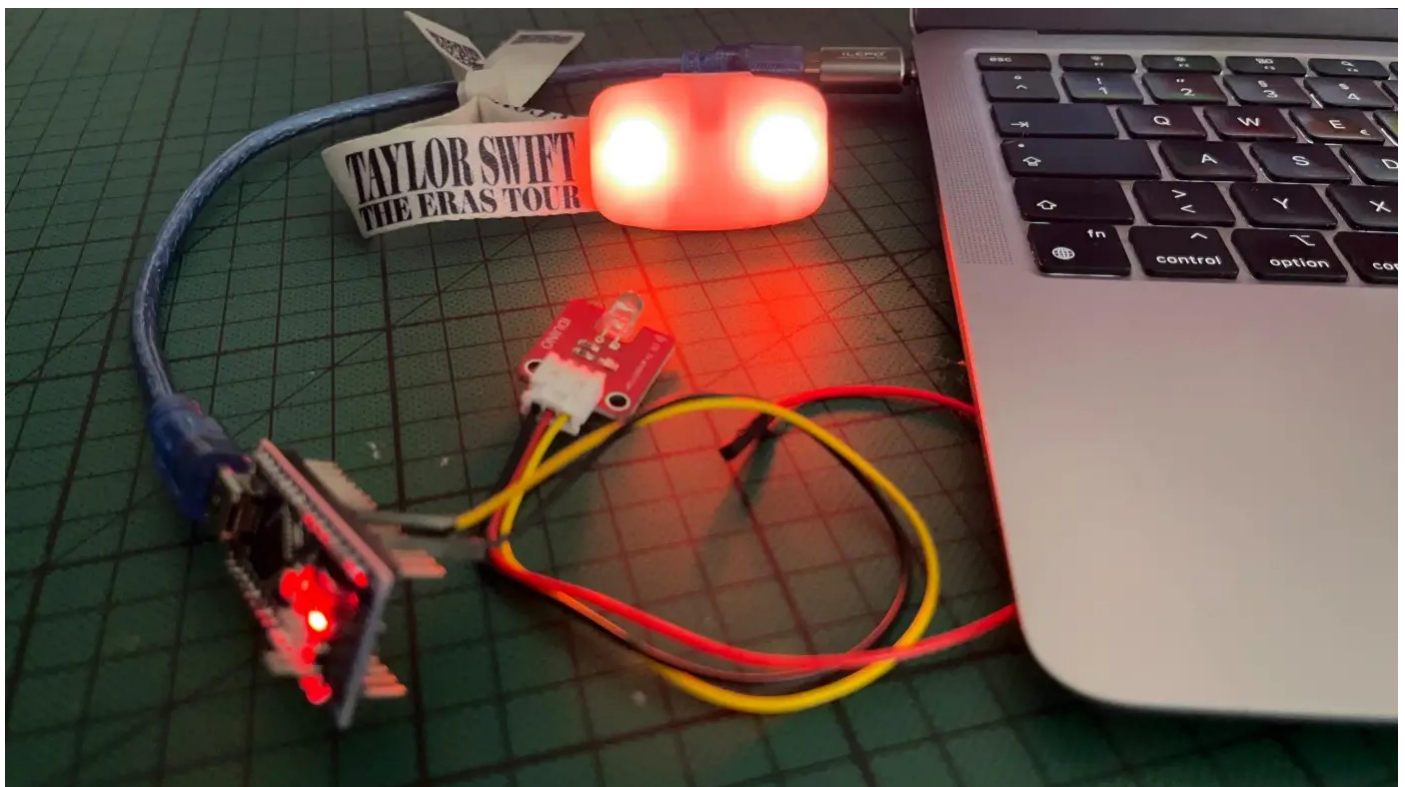
Arduino-IDE ist die Arduino-App, mit der Programmskripte erstellt, bearbeitet und anschließend auf den Arduino geladen werden, wie zum Beispiel die Bibliothek "IRremote".

```
38 // SET THIS TO THE DATA PIN USED FOR THE IR TRANSMITTER
39 const uint16_t kIrLed = 4;
```

Zum anderen fehlt noch eine Programmbibliothek, die innerhalb der Arduino-IDE für die Infrarotkommunikation verantwortlich ist. Dafür klickt man in der linken Seitenleiste auf das Bücher-Icon und gibt direkt oben in der Suchleiste "IRremote" ein. Es sollte die Bibliothek "IRremote by shiriff, z3t0, ArminJo" angezeigt werden, die nun installiert werden sollte. Danach kann man den Sketch per Upload-Button auf den Arduino laden.

Steuerung über ein Webinterface

Wie angenehm zugänglich das PixMob-Reverse-Engineering-Projekt ist, zeigt sich im letzten Schritt: **Online findet sich nämlich ein simples Webinterface [12]**, mit dem sich die Farbcodes für das Armband aus dem Browser heraus senden lassen. Wichtig: Der verwendete Browser muss "Web Serial API" unterstützen, das die serielle Kommunikation mit dem Arduino ermöglicht. In Chrome, Opera und Edge läuft die Funktion schon seit 2021; Safari und Firefox lassen sich für dieses Projekt (noch) nicht nutzen. Den Implementationsstatus für **gängige Browser findet man online [13]**.



Mit dem Webinterface können verschiedene Farben ausgewählt werden, in denen das Armband leuchtet.
(Bild: Jack Pop)

Im Webinterface stellt der Button "Connect to board" eine Verbindung zum Arduino her. Dann öffnet sich ein Pop-up, in dem man den Port "USB Serial" auswählt und auf "Verbinden" klickt – damit ist die Einrichtung komplett. Um das Armband anzusteuern, wählt man einen "Main Effect" aus der oberen Dropdown-Liste aus und schickt den Befehl über den "Send"-Button los. Das

Armband sollte nun aufblinken. Die zweite Dropdown-Liste lässt einen "Tail Code" auswählen: einen Zusatzbefehl, der – wie es der Name schon sagt – ans Ende des Farbbefehls geschrieben wird. Zuverlässig funktionieren vor allem die Fade-Effekte, bei denen die Farbe sanft aufleuchtet und abblendet.

Leuchtbefehle verbessern und Steuerung mit einem Android-Handy

An dieser Stelle wird ein kleiner Makel des PixMob-Reverse-Engineering-Projekts deutlich: Die aufgezeichneten, dokumentierten und im Webinterface implementierten Farbcodes stammen von unterschiedlichen Tourneen und Geräteversionen aus verschiedenen Jahren. Armbandhersteller PixMob aktualisiert und individualisiert seine Leucht-Gadgets ständig, sodass viele Befehle des Webinterface in unserem Test mit dem Taylor-Swift-Armband nicht (mehr) funktioniert haben. Außerdem erscheint die Bedienung – Liste aufmachen, durchscrollen, Effekt auswählen, senden – etwas umständlich.

Deshalb haben wir uns durch Dutzende Code-Listen, RGB-Tabellen, Demoversionen und Protokolle gearbeitet und eine übersichtlichere, erweiterte Weboberfläche zur Ansteuerung der Armbänder programmiert. **Wir haben 27 Eras-Tour-Farben und sechs Animations-Effekte hinterlegt [14].** Einer davon ist das sogenannte "letzte Signal", das die Veranstalter am Ende jedes Konzerts als eine Art Abschiedsgruß an die Armbänder schickten, um sie in einen dauerhaften Farbwechsel zu versetzen.

Wer selbst Hand anlegen möchte, sei hiermit ermutigt, unsere Version zu ergänzen und zu individualisieren: Unser Quellcode ist offen einsehbar, kommentiert und wie auch das PixMob-Reverse-Engineering-Projekt unter der MIT-Lizenz frei verfügbar. Wer lieber an den unbearbeiteten Originalcode möchte, findet diesen im GitHub-Repository **im Ordner "www" [15].**

Noch simpler mit Android-Handy

Mit etwas Glück geht es noch einfacher: Unter bestimmten Umständen lassen sich die LED-Armbänder nämlich auch mit dem Smartphone steuern. Dafür muss das Smartphone einen sogenannten Infrarot-Blaster besitzen.

Das ist ebenfalls eine Infrarot-LED – nur eben im Smartphone verbaut. **Auf Wikipedia findet sich eine Liste [16]** mit Mobiltelefonen, die damit ausgestattet sind. Dabei sind diverse Handys, die aus der Zeit vor der Smartphoneära stammen, aber auch einige aktuelle Android-Geräte.

Wer sein Handy in der Liste findet, kann sich im nächsten Schritt eine App herunterladen, um den Infrarot-Blaster zu steuern. Wir haben dazu die App "irplus Infrarot Fernbedienung" genutzt.

Analog zur Arduino-Variante haben wir **eine Datei hinterlegt [17]**, die man mit der App öffnet, um die Armbänder der Eras-Tour über farbige Buttons aufleuchten zu lassen.

Auch diese Datei basiert auf der **Arbeit des PixMob-Reverse-Engineering-Projekts [18].**

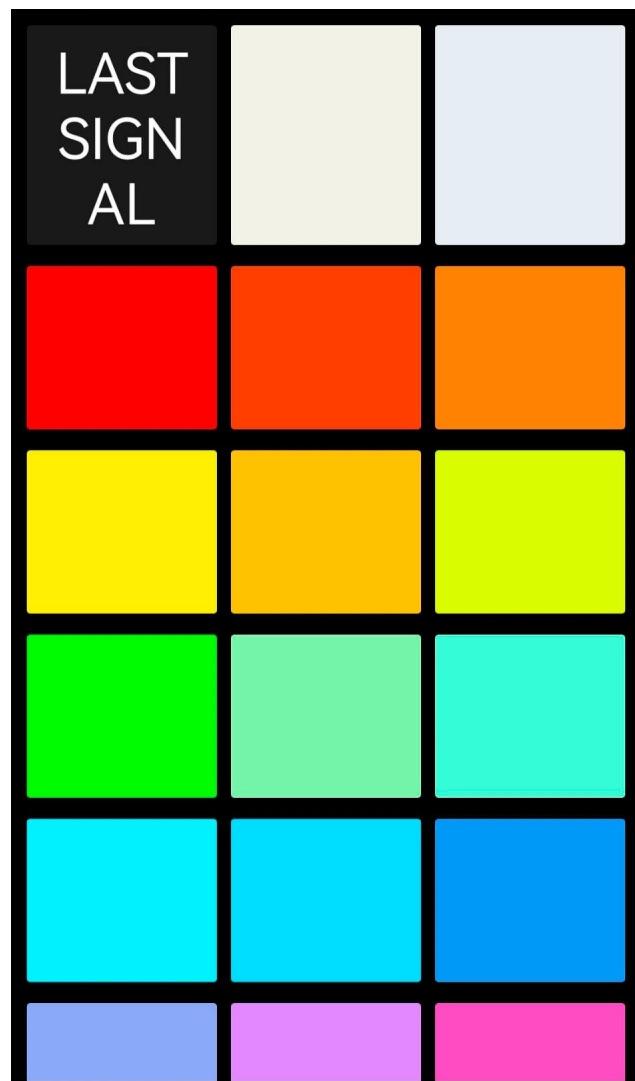
Armband per Musik ansteuern

Für Power-Bastler enden die Möglichkeiten damit noch nicht. Es wäre zum Beispiel möglich, das Armband per Musik anzusteuern. Dafür braucht der Arduino ein Schallerkennungsmodul, das genau wie das Infrarotmodul aus einer kleinen Platine mit einem aufgelöteten Bauteil besteht.

In diesem Fall ist aber statt der LED ein hochempfindliches Mikrofon als Schallsensor integriert. Wird ein Lied abgespielt, kann bei entsprechender Programmierung dann die Lautstärke unterschiedliche Farbsignale triggern.

Weitere Inspiration gibt es auf dem Discord-Server "PIXMOD: Reverse Engineering PixMob!". Er bietet als verlängerter Arm des Repository mehr dokumentierte Beispiele anderer Maker.

Ein begeisterter User hat sogar einen Chatbot mit dem Namen PIXMOB-GPT entwickelt, der unter anderem Quellcode zum Steuern der Armbänder erstellt. All das zeigt, wie Einzelne getrieben von einer Leidenschaft zusammenfinden können – wie auch bei der Eras-Tour. (spa [19])



Wenn das eigene Handy mit einem Infrarot-Blaster ausgestattet ist, kann man das LED-Armband auch mit einer App über das Smartphone ansteuern.

URL dieses Artikels:

<https://www.heise.de/-10003035>

Links in diesem Artikel:

- [1] <https://www.heise.de/ratgeber/Arduino-Projekt-LED-Armband-vom-Taylor-Swift-Konzert-wieder-leuchten-lassen-10003035.html>
- [2] <https://www.heise.de/ratgeber/Tischtennisuhr-mit-Arduino-und-3D-Drucker-selbst-bauen-9589967.html>
- [3] <https://www.heise.de/ratgeber/Bastel-Projekt-mit-Arduino-Nano-Head-Up-Display-fuer-Skifahrer-9535201.html>
- [4] <https://www.heise.de/ratgeber/Spracherkennung-mit-Arduino-nano-connect-realisieren-9308022.html>
- [5] <https://github.com/danielweidman/pixmob-ir-reverse-engineering>
- [6] <https://github.com/danielweidman/pixmob-ir-reverse-engineering>
- [7] <https://github.com/danielweidman/pixmob-ir-reverse-engineering>
- [8] <https://www.heise.de/ct>

- [9] <https://www.arduino.cc/en/software>
- [10] <https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide/>
- [11] https://github.com/danielweidman/pixmob-ir-reverse-engineering/tree/main/arduino_sender/PixMob_Transmitter_Arduino
- [12] <https://ivanr3d.com/tools/led-wristband/>
- [13] <https://caniuse.com/web-serial>
- [14] <https://wissenhinterdenkulissen.de/armband/>
- [15] <https://github.com/danielweidman/pixmob-ir-reverse-engineering/tree/main/www>
- [16] https://en.wikipedia.org/wiki/Category:Mobile_phones_with_infrared_transmitter
- [17] <https://wissenhinterdenkulissen.de/armband/>
- [18] <https://github.com/danielweidman/pixmob-ir-reverse-engineering/>
- [19] <mailto:spa@heise.de>

Copyright © 2024 Heise Medien