



Arduino-Bibliothek MobaLEDLib

PROGRAMMIEREN MIT EXCEL



Dieser Musteraufbau zeigt die Vielfalt der MobaLedLib.
Alle Fotos: Heiko Herholz

Programmieren mit der Mikrocontroller-Plattform Arduino ist vergleichsweise einfach. Dennoch sind ein paar Grundkenntnisse des Programmierens vorab erforderlich und die Einstiegshürde entsprechend hoch. Bei der MobaLEDLib handelt es sich zwar auch um eine Arduino-Bibliothek, aber alle Einstellungen werden in einer Excel-Tabelle vorgenommen.

Vielleicht geht es Ihnen manchmal auch so wie mir: Zur Entspannung und aus Neugierde surfe ich gelegentlich bei einer guten Tasse Kaffee durch das eine oder andere Modellbahn-Forum im Internet. Ich bin zwar nicht wirklich der Foren-Aktivist, aber hier und da gibt es ja auch mal ganz interessante Sachen zu lesen. Im Stummiforum bin ich schon vor längerer Zeit auf die MobaLEDLib gestoßen. Beim schnellen Vorbeiflug fand ich es zwar ganz spannend, aber ich hatte nicht wirklich die Geduld 278 Foren-Beiträge und ein gut gefülltes Wiki zu studieren. Von einigen FREMO-Kollegen wurde ich vor einiger Zeit auch nochmal auf die MobaLEDLib aufmerksam gemacht und dann habe ich mich

doch mal mit dem Thema näher auseinandergesetzt. Schnell musste ich feststellen: Der erste Blick hat getäuscht: Man muss keine 278-Beiträge lesen, sondern nur eine kurze Anleitung. Auch die Hardware birgt Überraschungen: Man kann sowohl speziell für MobaLedLib gefertigte Platinen kaufen, als auch nur mit einem Arduino und ein paar LEDs starten.

ARDUINO

Auf den von mir benutzten Rechnern befindet sich üblicherweise bereits eine Arduino-Entwicklungsumgebung, von Profis auch Arduino-IDE (IDE= integrated development environment) gen-

nant. Wer es noch nicht hat, der kann auf www.arduino.cc diese IDE runterladen. Das System ist für Linux, macOS und Windows erhältlich. Leider ist es bei der MobaLedLib so, dass für einen der benötigten Bearbeitungsschritte unbedingt Windows erforderlich ist. Daher muss auch hier schon Windows als Betriebssystem verwendet werden.

MobaLedLib ist ursprünglich zur Verwendung mit einem Arduino Nano entwickelt worden. Da beim Arduino Nano und beim Arduino Uno der gleiche Prozessor verwendet wird, kann man auch diesen benutzen, wenn gerade kein Arduino Nano im Haus ist. Ich habe hier gleich einen Nano verwendet. .

Der Nano wird nach der Installation



Bitte Bild freistellen

Für die volle Funktion der MobaLedLib werden noch weitere Bibliotheken benötigt, die man hier installieren lassen sollte.

Die Installation der MobaLedLib erfolgt über die Bibliotheksverwaltung der Arduino-IDE.

der Arduino IDE über ein passendes USB-Kabel (liegt gelegentlich auch dem Nano schon bei) mit dem PC verbunden. Der passende Treiber wird automatisch installiert. Anschließend wird die Arduino IDE geöffnet. Im Menü Werkzeuge wird unter Board Arduino Nano und als Prozessor Atmega328 ausgewählt. Dann muss noch der Port ausgewählt werden. Das muss geschehen, während der Arduino per USB angesteckt ist. Ist hier in der Port-Auswahl nur eine Option vorhanden, dann ist diese richtig. Sind mehrere Optionen vorhanden, dann muss man zunächst rausfinden, welchen Port der Arduino benutzt. Dies kann man machen, in dem man den Arduino kurz abzieht. Der jetzt fehlende Port wird vom Arduino benutzt. Jetzt kann man kurz zu testen, ob die Verbindung zum Arduino funktioniert. Im Datei-Menü gibt es Beispieldateien. Ich habe aus den Basics-Beispielen den

Blink-Sketch geöffnet. Arduino-Programme nennt man übrigens Sketch. Auf dem Arduino Nano ist eine LED eingebaut. Diese kann man mit dem Blink-Sketch langsam blinken lassen. Man muss jetzt nur auf den Button mit dem Pfeil nach rechts klicken und das Programm wird kompiliert (übersetzt) und auf den Arduino geladen. Bei mir gab es hier eine Fehlermeldung. Nach kurzem Nachdenken habe ich unter Werkzeug bei den Einstellungen für den Prozessor „ATmega328P Old Bootloader“ ausgewählt und danach wieder auf den „Hochladen“-Button geklickt. Diesmal hat es geklappt, die LED blinkt langsam.

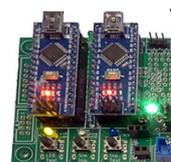
Dieser kleine Verbindungstest hat sich also gelohnt. Wer die direkte Programmierung von Arduinos lernen will, der kann jetzt im Blink-Sketch die Blinkdauer variieren und damit weiter experimentieren.

Damit die MobaLedLib funktioniert, müssen Makros und aktive Inhalte aktiviert werden.

Beleuchtungskonfiguration für Dummys

Einfach zu Bedienende Oberfläche zur Konfiguration von Beleuchtungen und anderen Effekten für eine Modelleisenbahn.

Mit diesem Programm können die Beleuchtungs- und andere Effekte für eine Modelleisenbahn ganz einfach erstellt werden. Es können bis zu 256 RGB LEDs oder 768 einzelne LEDs verwaltet werden.



MOBALEDLIB INSTALLIEREN

Der Erfolg der Arduino-Plattform ist vor allem den vielen Programm-Bibliotheken zu verdanken, die viele Dinge ganz einfach machen. Einen kleinen Teil dieser auch Libraries oder Lib genannten Erweiterungen bringt die Arduino-IDE direkt nach der Installation schon mit. Es gibt darüber hinaus aber noch weitere Bibliotheken, die sich vom Benutzer installieren lassen. Hiervon lässt sich ein Großteil komfortabel über die Bibliotheksverwaltung installieren. Diese findet man, wenn man im Sketch-Menü auf „Bibliothek einbinden“ geht. In dem Fenster, was sich jetzt öffnet ist oben rechts eine Suchfunktion vorhanden. Hier reicht es aus, einfach „moba“ einzugeben und die MobaLedLib wird sofort gefunden. Die Installation kann man mit dem entsprechenden Button starten. Die Bibliotheksverwaltung

Nach erfolgreicher Installation befinden sich auf dem Desktop drei MobaLedLib-Buttons. Mit dem Prog_Generator wird direkt die Konfiguration per Excel gestartet.



Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteilenummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte	Start LED	LEDS	InCh	Loc	LED Kanal
✓					Zeigt an, dass die LEDs angesteuert werden			RGB_Heartbeat(#LED)	0	1	0	0	0
✓					Test Led2			RGB_Heartbeat2(#LED, 5, 255)	1	1	0	0	0
✓					Test Haus			HOUSE(#LED, #InCh, 2, 5, ROOM_TV8_CHIMNEY, ROOM_D_RED, R	2	7	1	0	0
✓					Test Signal			HP_2_2Bin_RGB(#LED, #InCh, 16)	9	4	1	0	0

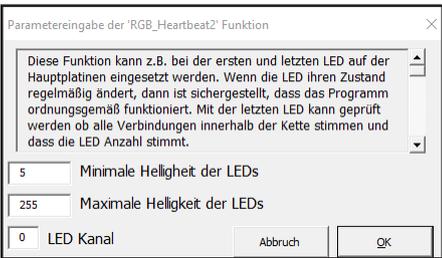
Mit dem Dialog-Button kann man menügeführt eine neue Zeile in die MobaLedLib eintragen. Der Button mit dem Arduino-Logo sorgt dann für die Übertragung der Konfiguration in den Arduino.

prüft dann zunächst, ob für die MobaLedLib noch weitere Bibliotheken benötigt werden und bietet diese zur Installation an. Das muss man machen, damit später auch alles funktioniert. Die gesamte Installation dauert einen kleinen Moment. Es wird jeweils die aktuellste Version aus dem Internet geladen. Die komplette Software ist jetzt installiert und wir können mit der MobaLedLib starten.

EXCEL

Das Tabellenkalkulationsprogramm Excel kennt wohl jeder, der schon mal mit einem Büro-PC gearbeitet hat. Damit die MobaLedLib funktioniert, muss ein Windows-Rechner eingesetzt werden. Die MobaLedLib benutzt einige Dinge, die bei Excel auf dem Mac nicht zur Verfügung stehen. Zur Konfiguration der MobaLedLib öffnet man die Datei „Prog_Generator_MobaLedLib.xslm“. Diese befindet sich im Library-Ordner der MobaLedLib unter „extras“. Der Library-Ordner ist üblicherweise ein Unterordner des Arduino-Ordners unter Dokumente.

Die Eingabe neuer Tabellenzeilen erfolgt mit geführten Dialogen zu den jeweiligen Einstellmöglichkeiten.



Beim Öffnen der Datei muss man zustimmen, dass Makros ausgeführt werden sollen. Beim ersten Start werden zunächst vielfältige Initialisierungsfunktionen durchgeführt. Danach sollten sich drei zusätzliche Buttons auf dem Desktop des PCs befinden. Mit diesen Buttons kann zukünftig die Konfiguration direkt gestartet werden. Außerdem bietet einer der Buttons einen direkter Link ins Internet zum MobaLedLib-Wiki. So kommt man schnell zu allen Anleitungen.

Die MobaLedLib ist jetzt einsatzbereit. Daher sollte jetzt die Hardware frisch zubereitet werden. Der Arduino Nano ist zwar schon startklar, aber wir wollen ja auch noch etwas mehr sehen.

RGB-LEDS

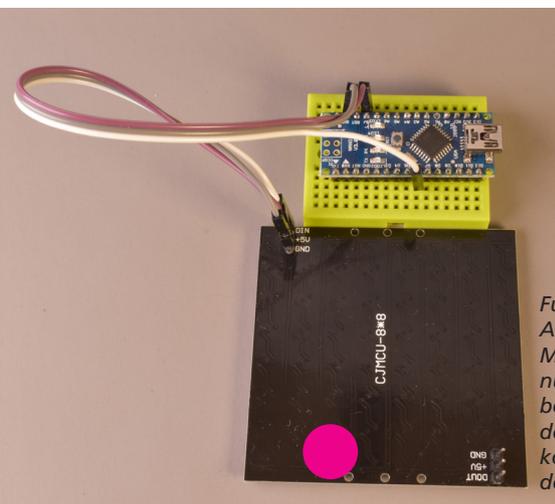
Kernkompetenz der MobaLedLib ist die RGB-Lichtsteuerung mit dem Steuerungsprozessoren WS2811 und WS2812. Diese Prozessoren befinden sich in nahezu jeder RGB-LED. Gemeint sind damit LEDs die in den Grundfarben rot, grün und blau leuchten können. Durch Variation der Farbanteile jeder

dieser LEDs lässt sich so praktisch jede Farbe darstellen. Die Steuerung dieser Farbe nimmt der integrierte Prozessor vor. Dieser bekommt auf einer 1-Draht-Datenleitung Farbinformationen geliefert. Jede Farbinformation besteht aus 24Bit, die sich in 8-Bit für jede Farbe aufteilen. Für die Benutzung solcher LEDs gibt es zahlreiche Arduino-Bibliotheken. Allerdings muss man hier jeweils etwas tiefer in die Arduino-Programmierung einsteigen.

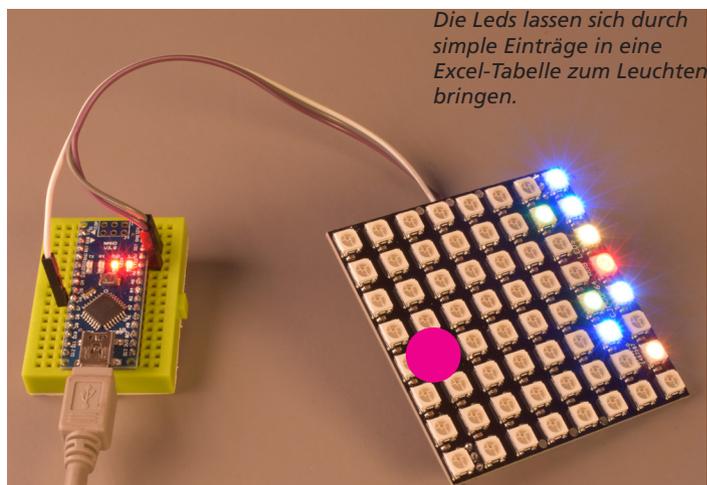
Für meinen ersten Test habe ich im Internet eine RGB-LED-Matrix mit insgesamt 64 LEDs bestellt. Der Matrix liegt ein dreipoliger Steckkontakt bei. Diesen habe ich in die drei Lötöffnungen bei DIN gesteckt und mittels dünnem Lötzinn und einem Elektronik-Lötkolben verlötet. Den Arduino Nano habe ich auf ein Steckbrett gesteckt und dann mit Steckkabeln Arduino und LEDs verbunden. Dabei werden +5V und GND mit den entsprechenden Kontakten des Arduino verbunden. DIN der Led-Platine wird mit dem Kontakt D6 des Arduino verbunden. Die Steckkabel sind unter dem Namen Jumper-Wires im Internet erhältlich.

Dieses Fenster ist quasi die Arduino-Fernsteuerung. Hier wird mittels der Arduino-Tools das Programm zusammengebaut, kompiliert und auf den Arduino übertragen. Das Piktogramm ist für die spezielle MobaLedLib-Hardware gedacht.





Für den Anschluss der Matrix werden nur drei Kabel benötigt. DIN der Matrix kommt an D6 des Arduino.



Die Leds lassen sich durch simple Einträge in eine Excel-Tabelle zum Leuchten bringen.

DIE ERSTE LED

Auf der Startseite der MobaLedLib-Excel-Tabelle gibt es einen Startbutton. Klickt man hier drauf, dann öffnet sich ein Fenster, in dem man auswählen kann, welches Modellbahn-Protokoll man einsetzt. Zur Auswahl stehen dabei DCC, Selectrix und Märklin CAN-Bus. Diese Protokolle werden von der MobaLedLib unterstützt. Allerdings benötigt man dafür auch die entsprechende Hardware. Der Arduino mit angeschlossener LED-Matrix kann diese Modellbahn-Systeme noch nicht. Daher ist es für den Moment erstmal egal, welches Protokoll ausgewählt wird. Ich habe DCC genommen und bin damit automatisch in der DCC-Tabelle gelandet. Hier ist bereits ein Mustereintrag in der ersten Zeile vorhanden.

Durch Klicken auf den Arduino-Button oben öffnet sich ein Dialog, bei dem der Arduino-Port ausgewählt wird. Bei

mir war bereits der richtige Port vorausgewählt. Anschließend öffnet sich ein blaues Fenster und es passiert eine kurze Zeit gar nichts. Das ist völlig normal, da im Hintergrund aus den Eintragungen in der Tabelle ein fertiges Programm für den Arduino generiert wird. Dieses wird dann auch zum Arduino übertragen, was man gut an den Fortschrittsbalken im blauen Fenster sehen kann. Das Fenster schließt sich nach erfolgreicher Programmierung von selbst und die erste LED auf der Matrix fängt an langsam blinkend die Farbe zu wechseln.

Als nächstes habe ich etwas rumgespielt. Zunächst habe ich eine zweite LED in Betrieb genommen. Dazu habe ich einfach auf den Dialog-Button geklickt. Hier wird dann mit mehreren Dialog-Fenstern eine weitere Zeile befüllt. Ich habe für meinen ersten eigenen Versuch in dem Makro-Auswahlfenster RGB_Heartbeat2 ausgewählt.

Den LED-Kanal habe ich bei 0 belassen. Die Nummerierung der LEDs erfolgt automatisch. Daher wird hier dann gleich die zweite LED auf der Matrix angesteuert. Durch Klick auf den Button mit dem Arduino-Logo wird wieder alles automatisch übertragen.

Für Tests mit der LED-Matrix eignen sich alle Makroeinträge mit dem Zusatz „RGB“. Besonders interessant ist das „House“-Makro für die Steuerung des Lichts in Modellhäusern. Hier gibt es viele Möglichkeiten etwas auszuprobieren und mit der LED-Matrix anzeigen zu lassen. Die Matrix dient dabei zur Simulation von bis zu 64 Räumen.

Die Möglichkeiten der MobaLedLib sind natürlich noch größer als hier gezeigt. Daher werden wir uns auch in der nächsten Ausgabe der DiMo damit beschäftigen und die Konfiguration des Musteraufbaus mit Lampen, Signalen, Servos und Sound genauer untersuchen.

Heiko Herholz

BEZUGSQUELLE

<https://wiki.mobaledlib.de>

RGB-Leds eignen sich hervorragend für die Beleuchtung von Modellhäusern. Jeder zu beleuchtende Raum bekommt eine LED. Einzel-RGB-LEDs mit WS2811 oder WS2812-Processor verfügen immer über einen Ein- und einen Ausgang. So kann man eine ganze Kette bauen. Im Konfigurationsfenster werden dann die Raum-Effekte in der Reihenfolge der Verkabelung durch anklicken eingetragen. Die MobaLedLib bringt eine große Auswahl an vorbereiteten Lichteffekten mit.

Mögliche Beleuchtungstypen:

ROOM_DARK	ROOM_BRIGHT	ROOM_WARM_W	ROOM_RED	ROOM_D_RED	* UNVERÄNDERBARE FARBEN Alle anderen Farben können mit dem Set_ColTab Befehl und dem Farbstab Programm angepasst werden. Die Candle Farben werden über Set_CandleTab angepasst.	ROOM_COL5	ROOM_COL345
ROOM_COLO	ROOM_COL1	ROOM_COL2	ROOM_COL3	ROOM_COL4			
FIRE	FIREA	FIREB	ROOM_CHIMNEY	ROOM_CHIMNEYD	ROOM_CHIMNEYB	*	
ROOM_TV0	ROOM_TV0_CHIMNEY	ROOM_TV0_CHIMNEY1	ROOM_TV0_CHIMNEY1	ROOM_TV1	ROOM_TV1_CHIMNEY	ROOM_TV1_CHIMNEY1	ROOM_TV1_CHIMNEY1
NEON_LIGHT	NEON_LIGHT1	NEON_LIGHT2	NEON_LIGHT3	NEON_LIGHTD	NEON_LIGHT1D	NEON_LIGHT2D	NEON_LIGHT3D
NEON_LIGHTH	NEON_LIGHT1H	NEON_LIGHT2H	NEON_LIGHT3H	NEON_LIGHTL	NEON_LIGHT1L	NEON_LIGHT2L	NEON_LIGHT3L
NEON_DEF_D	NEON_DEF1D	NEON_DEF2D	NEON_DEF3D	CANDLE	CANDLE1	CANDLE2	CANDLE3
SINGLE_LED1	SINGLE_LED2	SINGLE_LED3	SINGLE_LED1D	SINGLE_LED2D	SINGLE_LED3D		
GAS_LIGHT	GAS_LIGHT1	GAS_LIGHT2	GAS_LIGHT3	GAS_LIGHTD	GAS_LIGHT1D	GAS_LIGHT2D	GAS_LIGHT3D
SKIP_ROOM							

Ausgewählte Beleuchtungen: Mit einem Klick in das Feld unten kann die Position zum Einfügen / Löschen der Beleuchtungen gewählt werden. Anzahl: 7

ROOM_TV0_CHIMNEY, ROOM_D_RED, ROOM_TV0_CHIMNEYB, NEON_LIGHT3M, NEON_LIGHT1, FIRE, NEON_LIGHT1M

Lösche Raum

2 Minimale Anzahl der zufällig aktiven Beleuchtungen Individuelle Zeiten 1 Minimale Zeit bis zur nächsten Änderung [sek]

5 Maximale Anzahl der zufällig aktiven Beleuchtungen Eingang invertieren 255 Maximale Zeit bis zur nächsten Änderung [sek]

0 LED Kanal