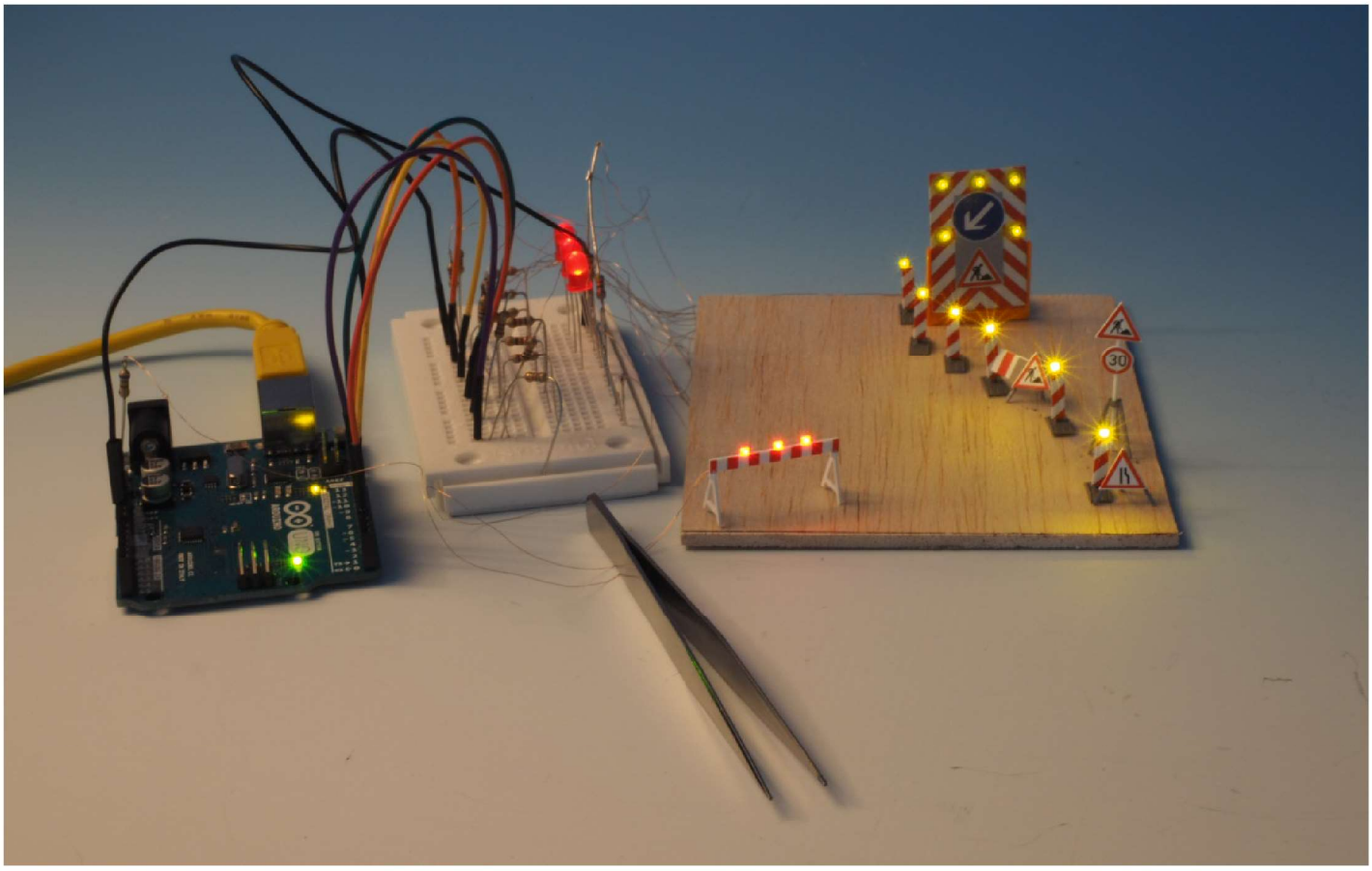
Die Modellbahnbande

Lesehappen Nr 8

Arduino Baustelle...

Die Modellbahnbande
Lesehappen Nr. 8

Baustellenlichter mit dem Arduino



Wer sich selbst einmal mit dem sogenannten „Hardware-“Programmieren beschäftigen möchte, der findet bei den Arduinos einen guten Einstieg.
Es wird kein großes elektronisches Wissen benötigt und wenn man nur einmal etwas probieren möchte, kann sogar der Lötkolben in der Werkzeugkiste verbleiben.

Sicherlich ist auch hier seit dem Artikel im Jahre 2016 vieles dazu gekommen, aber der Einstieg mit einem Blink- oder Lauflicht bietet sich für die Modellbahn auch heute noch an. Bei diesen Schaltungen erlebt man schnell seine ersten Erfolge, die einen dann zum weitermachen animieren.

Auch dieses Thema werden wir sicher noch einmal erneut aufgreifen.

Aber nun geht es auf der nächsten Seite weiter zum Artikel:

Mit Arduino und Co. lassen sich recht unkompliziert individuelle Beleuchtungen und spezielle Lichteffekte für die heimische Modellbahnanlage herstellen.

Lichteffekte leicht gemacht

Baustellenbeleuchtung mit Arduino

Arduino ist eine einfache Computer-Plattform, die preisgünstig und mit einfachster Programmierung die Automatisierung kleinerer Funktionen erlaubt. Britta Mumm hat mit nur wenig Aufwand damit die vorbildentsprechende Warnbeleuchtung einer Straßenbaustelle realisiert.

Wer seine Anlage ansprechend gestalten will, kommt um beleuchtete Accessoires, welche die Szenerie beleben, in der Regel nicht herum. Es gibt dabei viele Wege, die dafür notwendige elektronische Steuerung auch als nicht so eingefleischter „Elektro-Spezi“ in den Griff zu bekommen. Manches lässt sich zum Beispiel einfach fertig kaufen, aber auch auf eine individuellere – selbstgemachte – Beleuchtung muss nicht unbedingt verzichtet werden, seit es die netten kleinen Helfer „Arduino“ (beziehungsweise „Genuino“) und Co. kostengünstig auf dem Markt gibt.

Wer also, wie ich vor einiger Zeit, zum Geburtstag das unbeleuchtete Motiv-Set „Baustelle“ von Busch (Art.-Nr.: 6048, UvP.: 13,79 €) geschenkt bekommen hat, kann dieses, nachdem alles zusammengebaut wurde, einfach so auf seiner Anlage platzieren. Er kann aber auch noch ein wenig mit dem Arduino spielen und das Ganze entsprechend beleuchtet installieren.

Per Arduino lässt sich das Baustellen-Set von Busch mit wenig Aufwand individuell beleuchten.

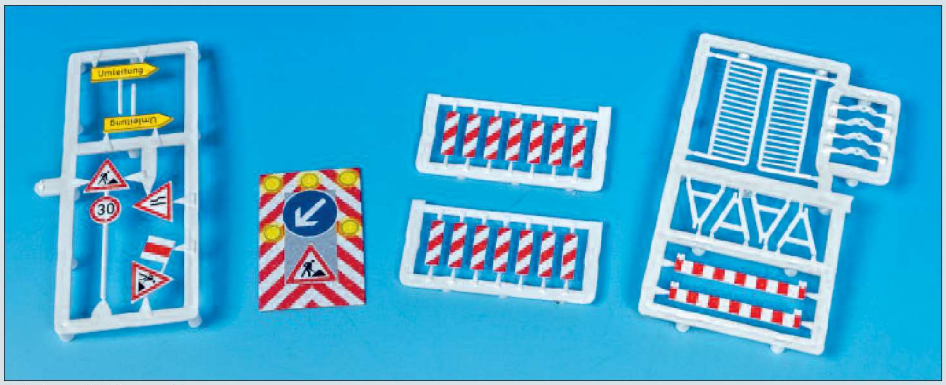
Vorbild Straßenbaustelle

Bevor es ans Werk geht, ist es zunächst sinnvoll, sich etwas eingehender mit dem Vorbild zu beschäftigen, denn gleich am Anfang kam bei mir die Frage auf: *„Und wie leuchtet das nun tatsächlich auf der Straße?“* Ich bin zwar schon so oft durch diverse Baustellen gefahren, aber bis auf ein paar markante Einzelheiten, wie einem „Doppelblitz“ links und rechts oben an einem dem aus der Packung ähnlichen Anhänger, war ich mir plötzlich gar nicht mehr so sicher, ob ich wirklich alles richtig in Erinnerung hatte. Leuchten die Baken gleichzeitig, wenn die Fahrbahn verschwenkt wird, oder wandert dort etwa ein Lauflicht? Nun, der Phantasie sollten eigentlich keine Grenzen gesetzt werden, wer es aber genau wissen möchte, der kann sich

Baustellenbeleuchtung mit Arduino

z. B. im Internet bei rsa-95.de unter „Warnleuchten“ informieren. Er wird dort auch gleich eine Entschuldigung finden, falls etwas doch nicht ganz so regelgerecht dargestellt wird: *Die Bauarbeiter halten sich selbst auch nicht immer an die Vorschriften …* Was ich jedoch unbedingt vorbildgerecht umsetzen wollte war, dass bei der Fahrbahnverschwenkung die Leuchten auf den Baken nacheinander entzündet werden, aber nicht wieder einzeln verlöschen, sondern erst alle zusammen.

Bei dem Anhänger hingegen, der offensichtlich einem älteren Vorbild entspricht, blitzen die oberen Leuchten wohl nicht, wie es bei der aktuellen Ausführung der Fall wäre, sondern sie blinken vermutlich nur gleichzeitig – jedenfalls habe ich bei youtube.com kein Video mit einer erhellenden Information gefunden, die meine Annahme widerlegen würde. Ich weiß nur: Früher wurde im Straßenverkehr – im Gegensatz zu heute – generell sparsamer mit Licht umgegangen, um Blendeffekte zu vermeiden, so dass ich einfach bei meiner Annahme bleibe.



Aufnahmen (13): Thorsten Mumm

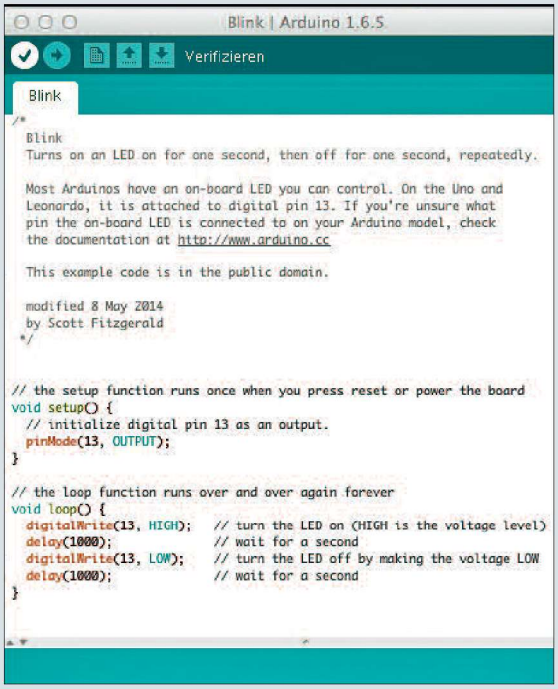
Die Einzelteile des Baustellensets müssen sowieso zusammengebaut werden, bevor sie aufgestellt werden können, da bedeutet ein bisschen Kabelziehen kaum zusätzlichen Aufwand.

Blinken mit Arduino

Ein gleichzeitiges Blinken ist mit einem Arduino leicht hergestellt, denn das Beispielprogramm „Blink“ aus dem Ordner „Basics“ der Entwicklungsumgebung kann ohne jegliche Änderung übernommen werden. Es müssen lediglich alle LEDs mit demselben Pin (hier Pin 13) verbunden werden – fertig! Allerdings ist dabei zu beachten, wieviel Strom so ein Pin liefern kann, damit die LEDs einerseits vernünftig leuchten und andererseits der Controller keinen Schaden nimmt, weil die LEDs in Summe u. U. zu viel Strom ziehen. Wird beispielsweise der Atmel Atmega 328P-PU (Preis: 2,50 € bei reichelt.de bzw. mit Bootloader 3,81 € bei watterott.com) oder das Arduino-Uno-Board (Preis: 19,95 € bei reichelt.de) genommen, dann sind pro Pin bis zu 40 mA erlaubt, wobei für alle Pins insgesamt 200 mA nicht überschritten werden sollten.

Nun müssen aber auch noch die Datenblätter der LEDs befragt werden, um herauszubekommen, wieviel Strom die LEDs maximal vertragen. Meist sind um die 20 mA für eine LED angegeben, aber dann leuchtet diese auch mit voller Stärke. Manch einen blendet diese Helligkeit bereits, was dazu führt, dass man einen größeren Vorwiderstand verwenden möchte, als es eine regelkonforme Berechnung mit den Werten des Datenblatts vorgibt. Bei den von mir verwendeten LEDs reichen z. B. 2 mA/LED aus, was bei einer Betriebsspannung von 5 V etwa 1,5 kΩ als Vorwiderstand und eine angenehme Helligkeit ergibt. Somit ist es gar kein Problem, alle fünf LEDs des Baustellen-Anhängers parallel an einen einzigen Pin zu hängen.

Dieses Ergebnis ist nicht nur für die Elektrifizierung des Anhängers ver-



Um ein einfaches Blinklicht zu erzeugen, kann der Beispiel-Sketch „Blink“ aus der Arduino-IDE unverändert übernommen werden.

Bei „roten“ Leuchten auf der Absperrung sollte man an dieser tunlichst nicht mehr vorbeifahren.

Die kleinen 0805-LEDs passen exakt in die aufgedruckten Scheinwerfer-Attrappen des Baustellen-Anhängers.

Modellbahn-Elektronik

Wird das Lauflicht mit einem „Array" programmiert, können Anpassungen leichter vorgenommen werden. In unserem Beispiel wurden die ursprünglichen Pins 2, 4 und 7 gegen die Pins 9, 10 und 11 getauscht.

… und ab geht es zur Baustelle. Der LKW bringt schon mal die ersten Verkehrszeichen hin.

wendbar, sondern auch für eine temporär aufgestellte Absperrschranke. Wie ich bei „rsa-95.de" erfahren habe, unterscheiden sich die Absperrungen in den Farben der Leuchten, die darauf angebracht sind: Bei „Gelb" darf an der Seite noch vorbeigefahren werden, „Rot" bedeutet hingegen, dass die Fahrbahn für alle komplett gesperrt ist. Hierbei leuchtet dann ein Dauerlicht, so dass kein Blinken erzeugt werden muss. Für so ein Dauerlicht benötigt man allerdings keinen Arduino – da reicht der Anschluss an die Betriebsspannung völlig aus.

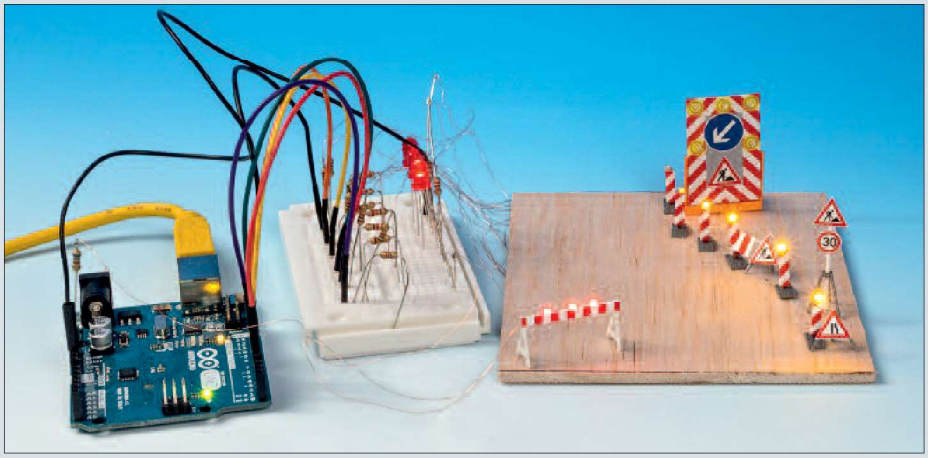
Lauflicht mit Arduino

Um nun das Lauflicht für eine Fahrbahnverschwenkung zu programmieren, habe ich mich zuerst einmal auf verschiedenen Internet-Seiten umgesehen. Dazu muss nur „arduino lauflicht" in eine Suchmaschine eingegeben werden, und schon kann man sich anschauen, wie andere es gelöst haben. Zwei Seiten haben mir dabei besonders geholfen: popovic.info – (Kapitel Arrays [eindimensional]) sowie die Youtube-Videos von „IMDFHTrier" (Kapitel 2.4 – Felder und Schleifen). Beide Quellen erklären sehr gut, wie ein Array (Feld) funktioniert und wie sich damit letztendlich auf einfache Weise ein Lauflicht realisieren lässt.

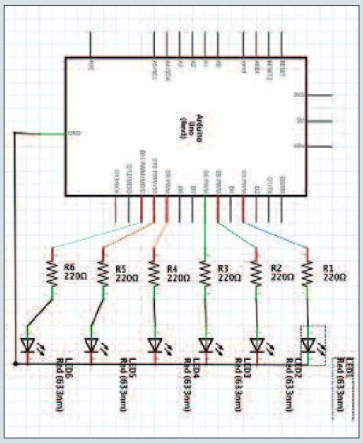
Dabei habe ich die Ausgabe am „Serial Monitor" in meinem Sketch bewusst stehen gelassen, obwohl diese für das Lauflicht eigentlich nicht benötigt wird. Sie kann nämlich gut als Analysetool genutzt werden, um zu prüfen, wie sich eine Änderung im Programm unmittelbar auswirkt. Das war sehr hilfreich, als ich das „08/15"-Lauflicht, das jede Leuchte einzeln aufleuchten und gleich wieder verlöschen lässt, dem Vorbild anpasste. Ich musste dazu den Sketch nur ein klein wenig ändern, indem ich das Delay – das ist eine Zeit, in der der Controller nichts weiter macht, als zu warten – eine Klammer-Ebene weiter (also aus der Zählfunktion hinaus) nach hinten geschoben habe, so dass erst alle LEDs nacheinander angehen, durch das versetzte Delay dann aber gleichzeitig – nach Ablauf der Zeit – zum Schluss erlöschen.

Solch ein Array ist nicht nur sehr praktisch, um Schreibarbeit zu sparen – ansonsten müsste man nämlich jede Leuchtdiode für jede Aktion stets in einer Extrazeile einzeln ansprechen –, sondern auch, um bei Änderungen statt der Verkabelung nur den Sketch anpassen zu können. So hatte ich ursprünglich die Pins 2, 3, 4, 5, 6. 7 für das Lauflicht verwendet. Als ich später auch noch an das Wohlergehen der Bauarbeiter dachte und ihnen zusätzlich ein „Lagerfeuer" – gefunden auf kreativkiste.de (unter elektro/Arduino Projekte) – zum Aufwärmen gönnen wollte, benötigte ich andere Pins, nämlich jene, die sich mit PWMs ansteuern lassen, um die Helligkeit der LEDs für ein Flackern einstellen zu können. Also steckte ich kurzerhand die Kabelbrücken von den Pins 2, 4 und 7 auf die Pins 9, 10 und 11 um, wobei danach an meinem Lauflicht nur noch die LEDs der Pins 3, 5 und 6 leuchteten. Dank des Arrays konnte ich auf ein erneutes Umstecken der Kabelbrücken verzichten, ich musste einfach nur den Inhalt des Arrays im Sketch anpassen – schon konnte ich abwechselnd entweder das Lauflicht oder das Lagerfeuer leuchten lassen, je nachdem, welchen Sketch ich in den Controller hochgeladen hatte.

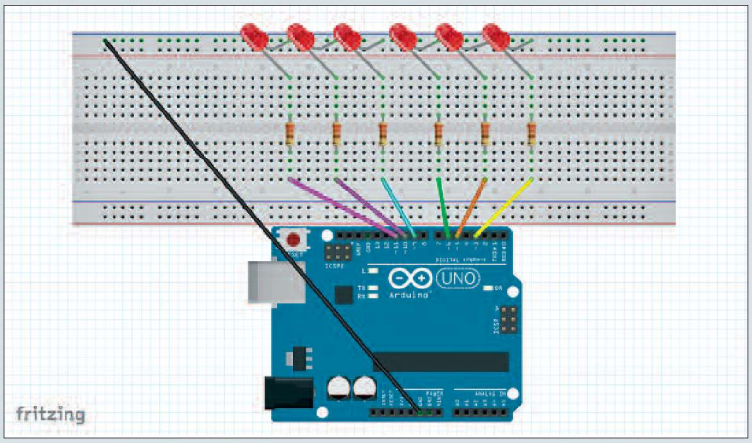
Für einen Probelauf, bevor alles fest eingebaut wird, lässt sich das Arduino-Uno-Board (hier mit SMD-Controller, der nicht austauschbar ist) gut nutzen.



Baustellenbeleuchtung mit Arduino



Mit dem kostenlos herunterladbaren Tool von fritzing.org lassen sich die Schaltpläne der eigenen Arduino-Projekte am PC einfach zeichnen. Der Schaltplan zeigt die nebenstehende Schaltung für das Lauflicht.



Die Schaltung für das Lauflicht ist ganz einfach. Hier sieht man das Steckbrett, sechs LEDs mit den jeweiligen Vorwiderstände und das Arduino Uno-Board mit THT-Controller, der sich, da er im echten Board gesockelt ist, auch austauschen lässt.

Was jetzt noch fehlte, war die Übertragung der Sketche auf drei einzelne Controller, denn ich wollte nicht mein Arduino-Board in die Anlage einbauen – was natürlich auch möglich wäre. Der einfachste Weg ist die Verwendung von bereits vorbereiteten Controllern mit Bootloader, wie sie z. B. bei watterott.com erhältlich sind. Mehr Aufwand bedeutet es, ganz leere Controller mit dem Code zu laden. Hierzu muss die Betriebsweise des Arduino-Boards geändert werden, so dass dieses zum Programmer wird. Wer diesen Weg beschreiten möchte, findet eine ausführliche Anleitung auf der Make-Seite von Heise: heise.de/make/artikel/Arduino-Uno-als-in-System-Programmer-2769246.html

Einbau der LEDs

Nachdem ich alle nötigen Sketche abgespeichert hatte, ging es an das Zusammenbauen der Baustelle 6048 von Busch, hier natürlich besonders an das Löten und Anbringen der Leuchtdioden an die Baken, Absperrungen sowie den Anhänger. Ich habe hierfür SMD-LEDs der Größe 0805 verwendet. Diese passen im Verhältnis recht gut, auch wenn sie kein gelb oder rot eingefärbtes Gehäuse besitzen. Wenn sie leuchten – und das sollen sie ja – dann sieht man das Gehäuse sowieso nicht …

Für die Bedrahtung habe ich den feinen Kupferdraht von einem alten Weichenantrieb „stibitzt“. Um die daraus entstandenen Drähtchen leichter an die LEDs anlöten zu können, habe ich mir einfach einen Tesa-Posterstrip auf meine Schreibtischunterlage geklebt. Diese kleben ja beidseitig, so dass ich