

Tag und Nacht Steuerung

Zur Steuerung der Beleuchtungen auf der Anlage können momentan drei Methoden verwendet werden.

1. Steuerung über einen Helligkeitssensor (LDR)
2. Steuerung per Modellbahnzeit
3. Synchronisation der Modellbahnzeit mit der Zentrale

Die Beleuchtungen der Häuser, Straßenlaternen, ... sollen möglichst realistisch Angehen, wenn es dunkel wird. Dabei sollen die Reihenfolge und die Zeitpunkte zufällig sein. Bei der MobaLedLib kann man das über die „Scheduler()“ Funktion steuern. Sie aktiviert oder deaktiviert die einzelnen Gruppen innerhalb eines vorgegebenen Bereichs Zufällig. Gesteuert wird das über eine Variable welche die Dunkelheit repräsentiert. Diese kann entweder über den Helligkeitssensor beeinflusst werden oder von einer Uhr. Beide Methoden haben Vor- und Nachteile.

Steuerung über den Helligkeitssensor:

Der Helligkeitssensor erfasst die Helligkeit im Raum automatisch und steuert damit die Beleuchtungen. Das ergibt eine sehr gute und flexible Anpassung an die Raumbeleuchtung. Allerdings kann man die unterschiedlichen Aktionen nur mit der Helligkeit verknüpfen. Es ist nicht möglich, dass man die Kirchenglocke zu jeder vollen Stunde aktiviert. Es ist auch nicht möglich, dass man speziellen Schaltungen mitten in der Nacht auslöst. Ein umschalten der Ampel vom normalen Betrieb auf das gelbe Blinklicht der späten Nachtstunden kann nicht konfiguriert werden.

Steuerung über eine Modellbahnzeit:

Anstelle eines Helligkeitssensors kann man auch eine Modellbahnzeit zur Steuerung der „Scheduler()“ Funktion verwenden. Hier kann man Aktionen zu jedem beliebigen Zeitpunkt oder innerhalb einer bestimmten Zeitspanne auslösen. Allerdings muss die Synchronisation mit der Umgebungshelligkeit über DCC, Selectrix, CAN oder einen Schalter gemacht werden. Die Dauer eines Modellbautages muss bei beiden Systemen gleich lang sein.

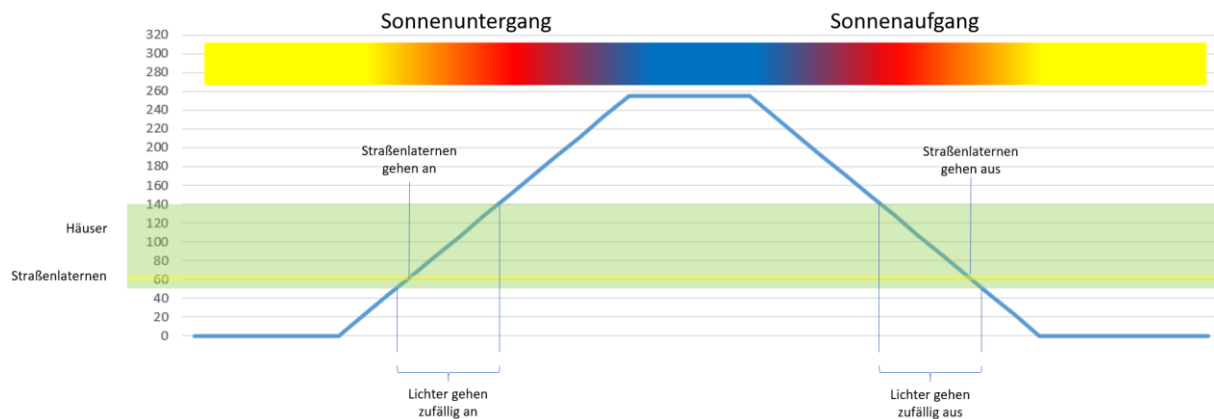
Aufbau der Variable:

Damit man die Konfiguration einfach zwischen Helligkeitssteuerung und der Zeitsteuerung umstellen kann wird die Zeit über die gleiche Variable abgebildet wie bei der Helligkeitssteuerung. Zunächst einmal wird gezeigt wie die Steuerung über den Helligkeitssensor funktioniert.

Helligkeitssensor:

Die Helligkeit wird über einen LDR gemessen. Sie wird als Zahl zwischen 0 und 255 abgelegt. 0 entspricht dabei der maximalen Helligkeit und die Variable ist 255, wenn es ganz dunkel ist. Die Variable enthält also die Dunkelheit. Zusätzlich zu diesem Wert erkennt das Programm noch ob es gerade dunkler oder heller wird. Im ersten Fall haben wir Sonnenuntergang und die Lichter sollen zufällig angeschaltet werden. Wenn der Arduino erkennt, dass es heller wird, dann haben wir Sonnenaufgang und die Lichter werden nach und nach wieder abgeschaltet.

Das sieht etwa so aus:



Die blaue Kurve zeigt den Verlauf der Dunkelheit. Während des Sonnenuntergangs gehen die Lichter in den Häusern langsam an. Dazu bekommen die verschiedenen Häuser nach und nach ihren Einschaltbefehl. Die Steuerung jedes Hauses kümmert sich dann selbstständig darum, dass in dem Haus die einzelnen Zimmer zufällig beleuchtet werden oder wieder aus gehen. Über die Parameter der „House()“ Funktion wird bestimmt wie viele Zimmer zufällig benutzt sind und wie schnell die ein und Ausschaltvorgänge ablaufen. Außerdem kümmert sich die Funktion um das Fernsehprogramm und das zufällige flackern des Kamins. Ebenso sorgt sie dafür, dass die Neonröhren beim Einschalten flackern. Wenn es dunkel genug ist sind alle Häuser aktiv. Aber nicht alle Zimmer sind gleichzeitig beleuchtet. Man kann im „House()“ Dialog auch einstellen, dass alle Lichter in einem Haus aus gehen dürfen.

Da es für den Betrachter langweilig wäre, wenn die Preiser schlafen gehen würden und es dabei auf der ganzen Anlage Stockfinster wird bleiben die Häuser so lange aktiv bis es wieder Morgen wird. Im Miniatur Wunderland verhalten sich die Bewohner genauso. Auch hier gibt es keine Nachtruhe. Im MiWuLa dauert ein Tag/Nacht Zyklus 16 Minuten. Davon ist es 11 Minuten lang Tag, Sonnenunter- und Aufgang dauern jeweils eine Minute und die Nacht ist 3 Minuten lang. Dieses Verhältnis entspricht einem Tag im Hochsommer.

Das Diagramm oben zeigt, dass die Lichter wieder zufällig ausgehen, wenn es wieder heller wird.

Konfiguriert wird dieses Verhalten über die „Scheduler()“ Funktion:

Parametereingabe der 'Schedule' Funktion

Dieser Plan gibt aber nur die groben Rahmenbedingungen vor. Wann die Ausgänge tatsächlich geschaltet werden, bestimmt das Programm zufällig damit ein realer Eindruck entsteht. Geschaltet werden die Ausgangsvariablen „DstVar1“ bis „DstVarN“. Sie werden zufällig zwischen dem Zeitpunkt „Start“ und „End“ eingeschaltet, wenn es „Abend“ ist und genauso wenn es „Morgen“ ist.

Haus1	Zielvariable 1
Haus20	Letzte Zielvariable
SL_1	Nummer der Enable Eingangs
50	Startwert
140	Endwert

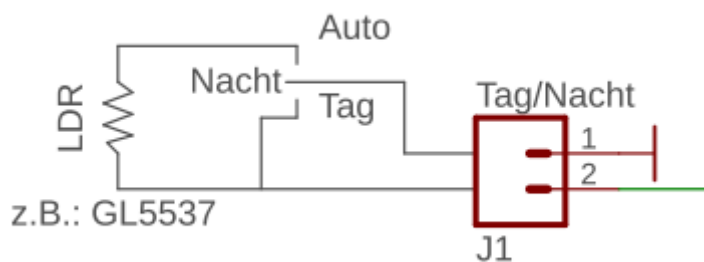
Abbruch OK

Die Funktion aktiviert verschiedene Variablen. Hier die Variablen „Haus1“ bis „Haus20“. Über den Parameter „Startwert“ wird bestimmt ab welcher Dunkelheit die Häuser aktiviert werden. Der Endwert bestimmt wann alle Häuser aktiviert sein sollen. Den genauen Zeitpunkt wann ein Haus eingeschaltet wird bestimmt der Zufall. Dieselben Parameter bestimmen auch wieder wann die Häuser bei Sonnenaufgang wieder abgeschaltet werden. Diese Symmetrie erscheint uns erst mal unlogisch, weil wir ja morgens nur schnell frühstücken und dann zur Arbeit gehen. Bei unseren

Preisern ist das anders. Da sie nicht schlafen gehen rennen sie die ganze Nacht durch das Haus und schalten abwechseln die Lichter an und aus damit wir Zuschauer unseren Spaß daran haben.

Es können mehrere Scheduler Funktionen mit unterschiedlichen Parametern verwendet werden. Das Diagramm oben zeigt noch die Funktion zum einschalten der Straßenbeleuchtungen. Hier ist der Bereich in dem die Lampen geschaltet werden viel kleiner gewählt. Die Zeiten wurden schließlich vom Bürgermeister vorgegeben.

Tatsächlich nutzt die Variable nicht den ganzen Bereich zwischen 0 und 255. Das hängt vom LDR, dem entsprechenden PullUp Widerstand (R27) und natürlich von den Helligkeitsverhältnissen ab.



Man kann zusätzlich zum Helligkeitssensor auch einen Schalter mit drei Stellungen Einsetzen. Wenn der Schalter den LDR mit dem Anschluss zum Arduino verbindet, dann ist der Automatik Modus aktiviert wie eben beschrieben. Wird der Schalter in die mittlere Stellung gebracht, dann erkennt das Programm das es Nacht werden soll. Dann wird es langsam „dunkler“. Dazu wird der Dunkelheitswert entsprechend erhöht. Wenn der Schalter die Anschlüsse des Tag/Nacht Steckers verbindet (Im Bild entspricht das der unteren Schalterstellung), dann ist wieder „Tag“ befohlen und der Dunkelheitswert wird verringert.

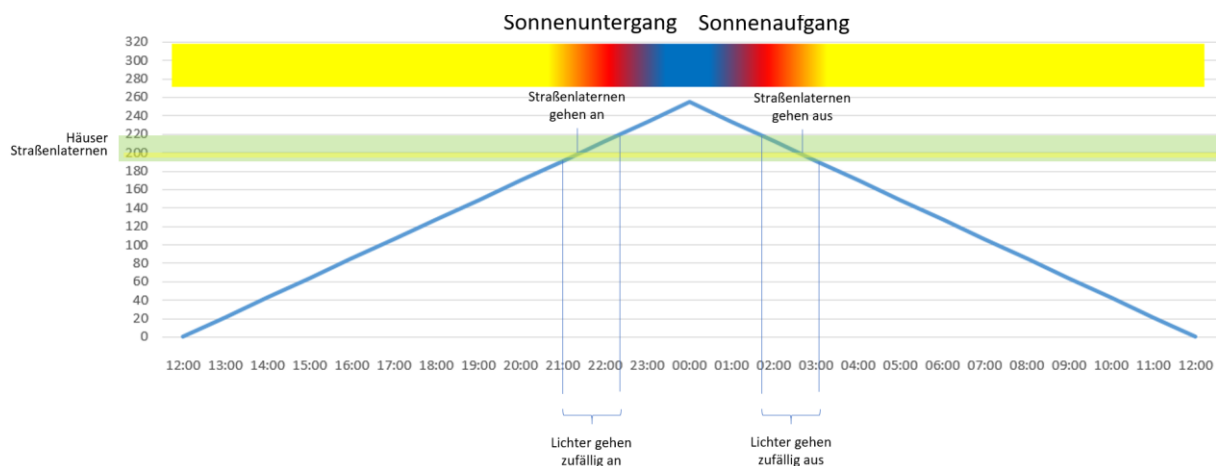
Es ist natürlich auch möglich, dass man den LDR ganz weglässt und einen Schalter mit zwei Stellungen verwendet. Wenn der Schalter geschlossen ist dann wird es „Tag“, und anders rum wird es langsam „Nacht“ wenn er geöffnet wird.

Die „Scheduler()“ Funktionen reagieren auf die „Helligkeitsänderungen“ per Schalter genau so wie bei einem echten Sensor.

In Pattern_Generator aktiviert man die Helligkeitssteuerung über den Befehl „READ_LDR“. Zur Anpassung der Parameter zu denen die „Scheduler()“ Funktion die Beleuchtungen aktivieren soll kann man zusätzlich den „READ_LDR_DEBUG“ Schalter aktivieren. Damit werden die aktuellen Dunkelheitswerte auf der seriellen Konsole der Arduino IDE Ausgegeben. Die gewünschten Werte prüft man indem man die Raumbeleuchtung modifiziert. Der Debug Schalter sollte im normalen Betrieb nicht mehr aktiviert sein, weil er unnötig Rechenzeit und Speicher verbraucht.

Modellbahnzeit:

Wenn eine automatisch ablaufende Modellbahnzeit verwendet wird, dann wird wieder ein ähnlicher Ablauf generiert. Die Modellbahnzeit wird auf einen Zahlenbereich von 0 bis 255 abgebildet. Zwischen 12 Uhr mittags und 0 Uhr nachts wachsen die Werte. Ab 0 Uhr ändert die Kurve ihre Steigung und die Zahlen werden wieder kleiner.



Hier ändert sich der Wert aber über den gesamten Tag/Nacht Zyklus. Dadurch kann man auch Ereignisse definieren welche zu Zeiten passieren an denen sich bei der Methode mit dem Helligkeitssensor nichts geändert hat. Auch hier sind die ein und Ausschaltzeiten symmetrisch.

Modellbahnzeit wird über dem Befehl „DayAndNightTimer()“ aktiviert. Die Dauer eines Tag/Nacht Zyklus wird über den Parameter der Funktion in Minuten bestimmt. Voreingestellt ist der Wert von 16 Minuten wie er im Wunderland benutzt wird. Die Jahreszeit, und damit die Zeiten wann die Sonne Auf- und Untergeht bestimmt man über die „Scheduler()“ Funktion.

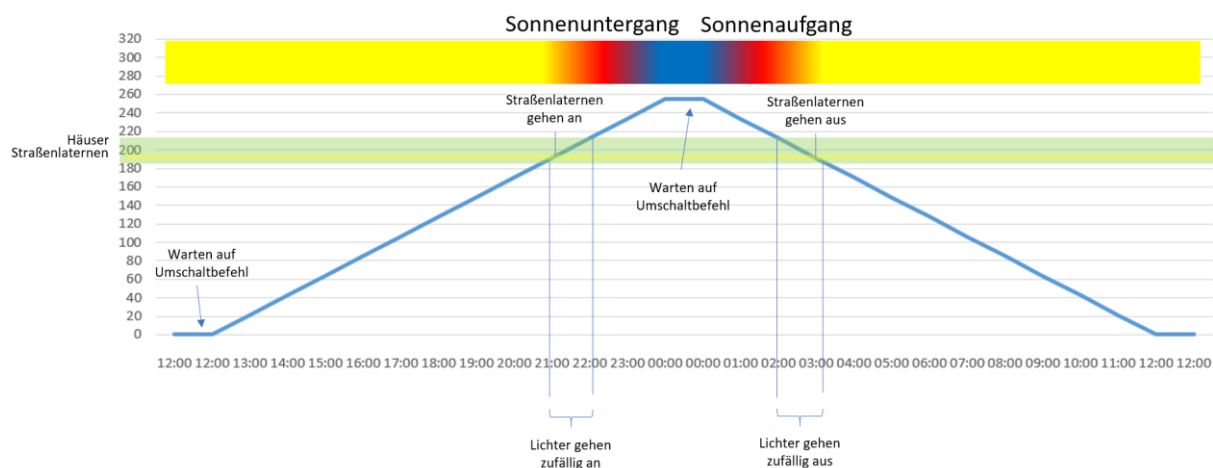
Bei einer Umstellung von LDR auf Modellbahnzeit müssen lediglich die Parameter der „Scheduler()“ Funktion angepasst werden, weil beide Methoden die gleiche Variable benutzen.

Modellbahnzeit synchronisiert mit Zentrale:

Die Helligkeit der Raumbelichtung muss natürlich mit den Einschaltzeiten der Beleuchtungen der MobaLedLib synchronisiert sein. Die Lichter in den Häusern sollen ja nur dann angehen, wenn es dunkel ist. Dazu kann man den „DayAndNightTimer()“ Befehl über DCC, Selectrix, CAN oder eine Schalter steuern. Hier wird der Wechsel von Tag auf Nacht über die DCC Adresse 300 gesteuert:

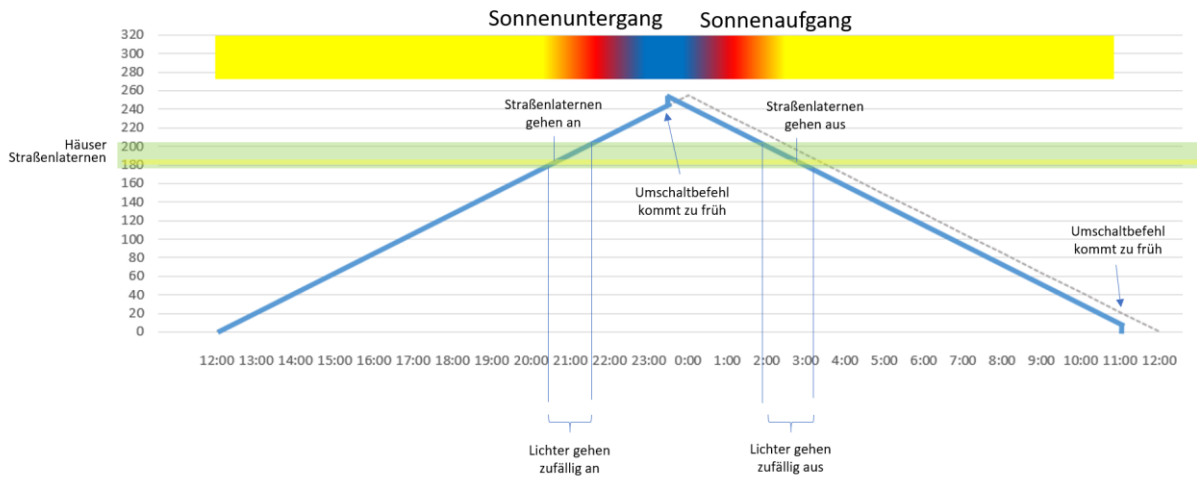
Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Beleuchtung, Sound, oder andere Effekte
✓		300	AnAus	0				DayAndNightTimer(#InCh, 16)

Wenn die DCC Funktion aktiviert wird dann beginnt der Nachmittag und anschließend der Abend. Im Diagramm unten wartet der Arduino so lange bis er einen Einschaltbefehl bekommt. Ab dem Zeitpunkt wird die „Dunkelheit“ hochgezählt.



Wenn der Maximalwert von 255 erreicht ist dann wartet das Programm wieder bis der DCC Befehl beenden der Nacht kommt indem das Signal auf 0 geht. Die Modellbahnzeiten von Zentrale und MobaLedLib sollten natürlich gleich eingestellt sein. In dem Bild oben ist ein Tag/Nacht Zyklus der Zentrale etwas größer die Einstellung der MobaLedLib. Dadurch wartet der Arduino bis der Umschaltbefehl der Zentrale kommt.

Wenn die Zykluszeit der Zentrale kleiner ist als die Zeit der MobaLedLib, dann kommt der Umschaltbefehl zu früh. Das bedeutet, dass der Dunkelheitswert springt.



Dadurch werden ein paar Werte „Dunkelheitswerte“ übersprungen. Im Bild die Werte zwischen 250 und 255. Solange diese Werte nicht genutzt werden ist das unkritisch. Wenn sie zur Steuerung einer Funktion benutzt werden, dann wird das entsprechende Ereignis zu früh ausgelöst. Liegen mehrere Ereignisse in dem übersprungenen Bereich, dann werden die Ereignisse gleichzeitig aktiviert.

:

ToDo:

- Bild wenn Zeit deutlich kürzer ist
- Eigentlich müssten die Zeitachse und der Helligkeitsverlauf oben auch angepasst werden
- Bilder wenn KeepDarknessCtr aktiv ist