

Station 1 – Elektrischer Rollladen

„Effizientes Sonnenbaden“

Besonders in Zeiten der „smarten“ Häuser gibt es immer mehr Dinge, die Menschen das Leben einfacher machen. Dazu gehören auch elektrische Rollläden, die sich öffnen und schließen, je nach der Helligkeit draußen.

Sobald es morgens hell wird, fahren sich die Rollläden hoch, damit man so viel Licht wie möglich reinlässt und die Wohnung sich etwas aufheizt. Wird es abends wieder dunkel, schließen sie sich wieder. Aber nicht nur für effiziente Sonnenausnutzung sind autonome Rollläden gut, sondern auch, um potenziellen Einbrechern vorzugaukeln, man sei zuhause, obwohl man eigentlich im Urlaub ist.

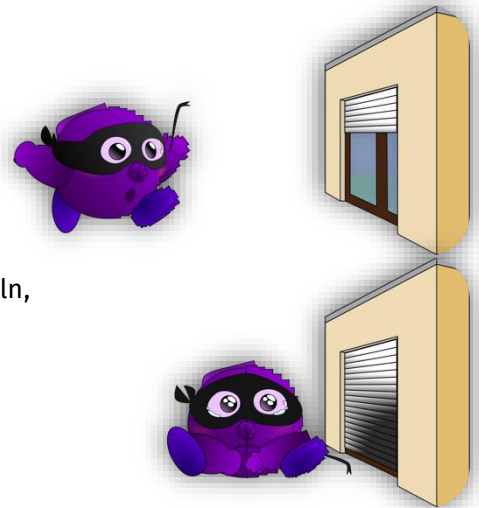
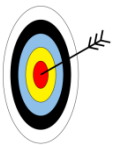


Abb. 1: Einbrecherabwehr Rollladen



Mit Hilfe dieser Arbeitsblätter...

- × baut ihr euren eigenen Rollladen,
- × lernt, wie sich elektronische Bauteile kombinieren lassen und
- × wie man sie kombiniert, um den Rollladen automatisch zu öffnen und zu schließen.

BENÖTIGTE BAUTEILE

Die finale Schaltung des Rollladens auf dem Arduino-Board besteht nur aus wenigen Bauteilen. Zusätzlich zum Arduino und den Steckbrettern benötigt ihr für dieses Arbeitsblatt:

- einen Umgebungslichtsensor
- einen 2 k Ω -Widerstand (rot-schwarz-rot)
- einen Servo-Motor mit Kreuzaufsatz
- 3 blaue, 3 rote, 2 gelbe Kabel

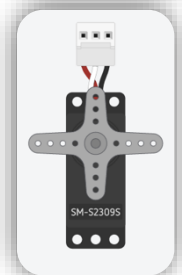
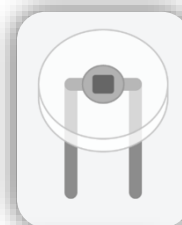


Abb. 2a-2c: Benötigte Bauteile



Damit euren Rollladen ihr gezielt bewegen könnt, mit jedem Bauteil vertraut euch machen ihr solltet. Dabei euch folgende Arbeitsblätter helfen werden.

Abb. 3: Foto von Stuart McGhee

Station 1 – Elektrischer Rollladen

Mit dem **UMGEBUNGSLICHTSENSOR** kann euer Rollladen erkennen, ob es heller oder dunkler wird und kann dadurch merken, ob er sich öffnen oder schließen sollte.

UMGEBUNGSLICHTSENSOR BZW. LICHTWIDERSTAND

Der Umgebungslichtsensor, den wir hier verwenden, ist ein Widerstand, der je nach Lichteinfall mehr oder weniger Strom durchlässt. Je mehr Licht auf den Sensor fällt, desto geringer wird sein Widerstand und umso mehr Strom kann fließen. Je weniger Licht auf den Sensor fällt, desto höher wird der Widerstand und umso weniger Strom fließt. Um einen Widerstand zu messen, muss man den Stromfluss untersuchen. Da ein digitaler Pin nur erkennen kann, ob Strom fließt (**HOCH**) oder nicht (**NIEDRIG**), ist dieser hier ungeeignet. Dafür benötigt man einen analogen Anschluss. Darum muss der Lichtwiderstand an einen analogen Pin angeschlossen werden.

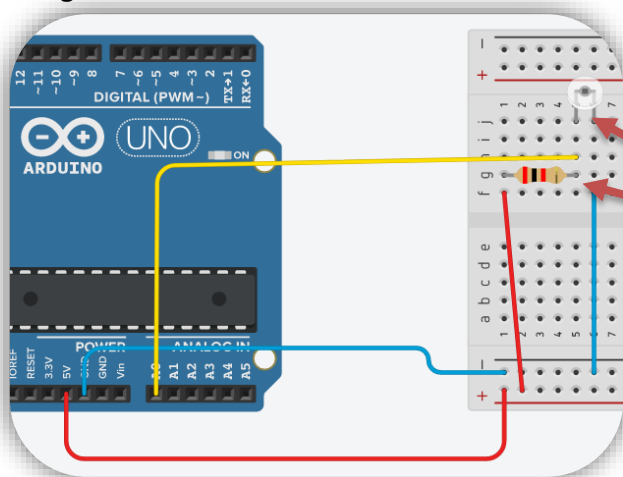
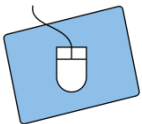


Abb. 4 Komplette Schaltung (Tinkercad)

BENÖTIGTE BAUTEILE

Um den Umgebungslichtsensor zu testen, benötigt ihr folgende Bauteile:

- einen Umgebungslichtsensor
- einen 2 k Ω -Widerstand (rot-schwarz-rot)
- 2 blaue, 2 rote, 1 gelbes Kabel

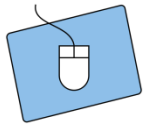


Schaut euch die Grafik genau an. Dort seht ihr, wie die Schaltung später aussehen soll.

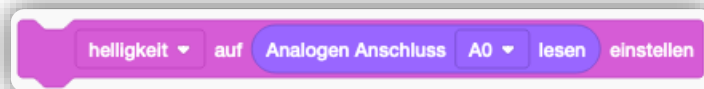
Die Schaltung funktioniert so ähnlich wie bei einer LED. Zusätzlich müsst ihr aber ein Kabel direkt zwischen Widerstand und Umgebungslichtsensor stecken und mit einem analogen Eingang verbinden.

Station 1 – Elektrischer Rollladen

Nun könnt ihr anfangen, zu programmieren und mit dem Umgebungslichtsensor zu arbeiten. Dazu erklären euch die folgenden Anweisungen, wie ihr mit Hilfe des **SERIELLEN MONITORS** herausfinden könnt, welche Signale am analogen Pin ankommen.



1. Öffnet das Code-Fenster in eurer Schaltung, und stellt es auf „Blöcke“.
2. Erstellt eine **Variable**, die eine Zahl für die Helligkeit speichern soll. Gebt ihr einen geeigneten Namen.
Helligkeitsvariablen-Name: _____
3. Ihr könnt nun den analogen Pin auslesen und den Inhalt in eurer Helligkeitsvariable aus 2.) speichern:



4. Damit ihr auch etwas sehen könnt, solltet ihr über den **SERIELLEN MONITOR** direkt die Variable für die Helligkeit ausgeben.
5. Das Programm wird schneller messen und Zahlen auf dem Bildschirm ausgeben, als ein Mensch lesen kann. Also müsst ihr noch eine Pause einfügen, damit ihr mit dem seriellen Monitor folgen könnt.
6. Jetzt könnt ihr mit dem Testen beginnen. Vermutlich müsst ihr dabei noch ein paar Flüchtigkeitsfehler beheben. Die passieren fast jedem und fallen erst beim Testen auf. Startet das Programm, und öffnet danach den **SERIELLEN MONITOR**. Werden die Zahlen immer noch zu schnell ausgegeben, erhöht den Wert für die Pause.
7. Ändert die Lichtstärke, indem ihr während der Simulation auf den Sensor klickt und den Schieberegler bewegt.



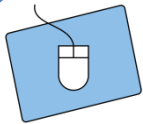
Abb. 5: Schieberegler Helligkeit

Notiert eure Erkenntnisse:

DUNKEL:	HOHE WERTE	<input type="checkbox"/>
	NIEDRIGE WERTE	<input type="checkbox"/>
HELL:	HOHE WERTE	<input type="checkbox"/>
	NIEDRIGE WERTE	<input type="checkbox"/>

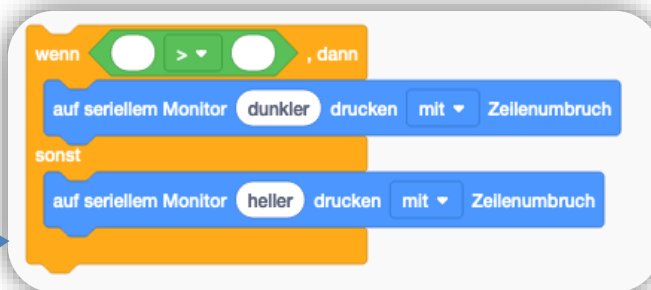
Station 1 – Elektrischer Rollladen

Nun wisst ihr, in welchem Bereich die Werte des Umgebungslichtsensors liegen und wie ihr erkennt, ob es hell oder dunkel ist. Aber der Rollladen muss auch feststellen können, ob es heller oder dunkler wird. Dabei hilft euch eine weitere **Variable**, die jeweils den alten Helligkeitswert speichert, und eine **Wenn-dann-Anweisung**, die überprüft, ob der aktuelle Helligkeitswert größer oder kleiner als der alte ist.



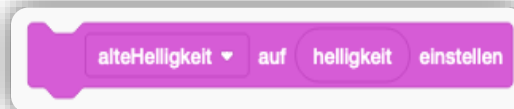
1. Erstellt eine neue **Variable**, die später den alten Helligkeitswert speichern soll. Gebt ihr einen sinnvollen Namen.
Variablen-Name: _____
2. Nun benötigt ihr eine **Wenn-dann**-Anweisung, die den alten Helligkeitswert mit dem aktuellen Helligkeitswert vergleicht und über den seriellen Monitor ausgibt, ob es heller oder dunkler geworden ist. Für die Bedingung könnt ihr wie im Matheunterricht die „größer“- und „kleiner“-Zeichen benutzen:

Vor oder hinter die Lichtmessung?
Vor oder hinter dem Speichern des Helligkeitswerts (siehe 3.)?



Tipp: Euer Programm wird von oben nach unten abgearbeitet. Überlegt, wo die **Wenn-dann**-Anweisung eingefügt werden muss, damit die Helligkeitswerte korrekt miteinander verglichen werden.

3. Speichert nun ganz am Ende den gemessenen Lichtwert in der neuen Variablen:



4. Testet euer Programm, und schaut, was auf dem seriellen Monitor geschieht. Falls es nicht direkt funktioniert, ist das nicht schlimm. Das passiert auch erfahrenen Programmierenden regelmäßig. Sucht nach dem Fehler, und versucht ihn zu beheben.

Station 1 – Elektrischer Rollladen

Sehr gut. Ihr wisst bereits, wie der Umgebungslichtsensor funktioniert und könnt mit dem Arduino Helligkeitsänderungen erkennen. Jetzt könnt ihr euch um die Drehbewegung kümmern.

BENÖTIGTE BAUTEILE

- 1 Servo-Motor mit Kreuzaufsatz
- 1 **blaues**, 1 **rotes**, 1 **gelbes** Kabel

Mit dem **SERVO-MOTOR** könnt ihr euren Rollladen bewegen. Hier wird euch seine Funktionsweise erklärt.

SERVO-MOTOR

Ein Servo-Motor ist ein Motor, der sich zu bestimmten Gradzahlen drehen kann. Euer Servo-Motor kann sich von 0° bis 179° drehen.



Ihr könnt dem Motor nur sagen, dass er sich **ZU** einem bestimmten Winkel drehen soll, **NICHT UM** einen bestimmten Winkel.

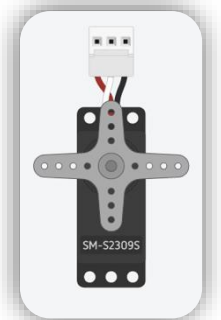
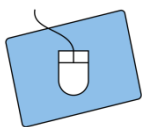


Abb. 6: Der Servo-Motor

AUFBAU DER SCHALTUNG

Eure alte Schaltung könnt ihr aufgebaut lassen, weil ihr sie für den Rollladen noch einmal brauchen werdet.



1. Schließt die Kabel an den Servo-Motor:

- **Blau** – Schwarz
- **Gelb** – Weiß
- **Rot** – Rot

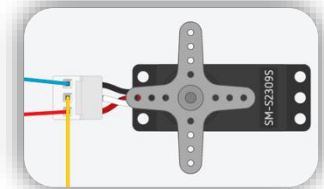


Abb. 7: Servo-Motor Kabelverbindung

2. Schließt die Kabel an den Arduino an:

- **Blau** – Minus-Leiste
- **Rot** – Plus-Leiste
- **Gelb** – Digital-Pin

Auf diesem Foto seht ihr, wie die beiden Schaltungen nebeneinander aufgebaut aussehen. Das **blaue** Kabel ganz rechts ist zusätzlich und dient nur dazu die beiden **Minus**-Leisten miteinander zu verbinden, ihr könnt das Kabel vom Servo-Motor auch einfach unten in die **Minus**-Leiste stecken.

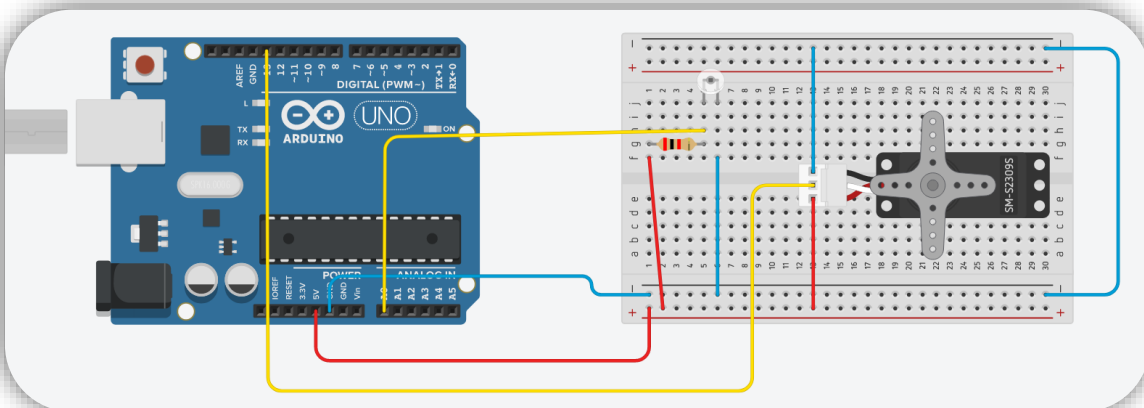
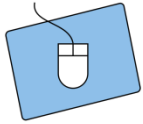


Abb. 8: Gesamte Schaltung (Tinkercad)

Station 1 – Elektrischer Rollladen

Nun könnt ihr damit anfangen, den Arduino so zu programmieren, dass er den Motor bewegen kann. Dazu werdet ihr zuerst lernen, wie ihr das Programm entsprechend vorbereitet und danach, wie ihr den Motor schrittweise bewegt.



1. Öffnet das Code-Fenster eurer Schaltung, und stellt es auf „Blöcke“.
2. Da ihr in einem blockbasierten Editor arbeitet, übernehmen die Blöcke schon einen sehr großen Teil der Arbeit, um den **SERVO** anzusprechen. Den einzigen Block, den ihr zum Bewegen benötigt, ist dieser:



Servo an Anschluss 0 auf 0 Grad drehen

Denkt daran, den richtigen Anschluss-Pin anzugeben, und gebt dann eine beliebige Zahl von 0 bis 179 für die Gradzahl an. [**Hinweis:** Gebt die Zahl ohne „°“ ein.]

3. Testet das Programm, und behebt eventuelle Flüchtigkeitsfehler in Schaltung oder Codeblöcken.
4. Wiederholt Schritt 2 mit verschiedenen Gradzahlen, und findet heraus, wo sich der Motor überall hinbewegen kann.

BEIM ERHÖHEN DER GRADZAHL DREHT SICH DER MOTOR:

IM UHRZEIGERSINN



GEGEN DEN UHRZEIGERSINN

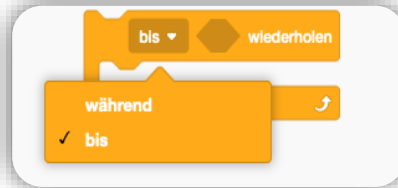


Station 1 – Elektrischer Rollladen

Super! Nun wisst ihr, wie sich der Motor bewegt und könnt lernen, wie man Variablen nutzen kann, um den Motor langsam und schrittweise von 0° bis 179° laufen zu lassen.



1. Erstellt eine neue **Variable**, die später die neue Gradzahl bzw. Position für den Motor speichern soll, und gebt ihr einen sinnvollen Namen.
Servo-Positions-Variablen-Name: _____
2. Ihr benötigt nun eine Schleife, die solange ausgeführt wird, bis eine bestimmte Bedingung erreicht ist. Überlegt euch, welche Bedingung ihr in das freie Feld des Blocks einfügen müsst.



3. In **der Schleife** soll nun die Position jedes Mal um 1 erhöht werden. Diese neue Position muss nun natürlich an euren Servo weitergegeben werden, damit der Motor sein neues Ziel auch ansteuert. Ersetzt die Gradzahl in eurem blauen „Servo auf Grad drehen“-Block deshalb durch den Namen eurer Positionsvariable.
4. Eine Schleifen-Wiederholung wird so schnell durchgeführt, dass der Motor keine Zeit hätte, sich zu einer neuen Position zu bewegen. Darum sollte hinter Servobewegung immer einen Moment gewartet werden. Fügt dazu einen Warte-Block ein.
5. Testet euer Programm.
6. Ändert den Wert des Warten-Blocks, und testet erneut. Wiederholt dies mehrmals, und vervollständigt die Regel.

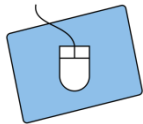
ERHÖHT MAN DEN WARTE-BLOCK, WIRD DER MOTOR:

LANGSAMER

SCHNELLER

Station 1 – Elektrischer Rollladen

Jetzt habt ihr gelernt, den Motor schrittweise in eine Richtung zu bewegen. Aber euch ist sicher aufgefallen, dass der Motor bei 179° stehen bleibt. Jetzt soll sich der Motor langsam zwischen 0° und 179° hin und her bewegen. Dies kann man mit einigen **Wenn-dann**-Anweisung und einer neuen **Variablen** für die Bewegungsrichtung lösen.



1. Erstellt eine **Variable**, die nun die Richtung speichern soll. Hat diese Variable den Wert 1, soll sich der Motor im Uhrzeigersinn bewegen. Hat sie den Wert -1, soll er sich entgegengesetzt bewegen.

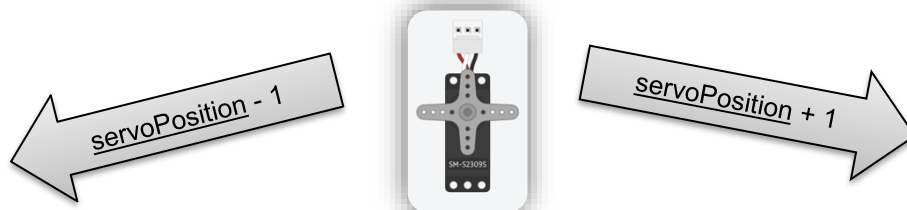


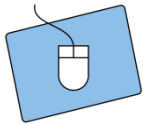
Abb. 9: Servo-Motor

2. Diese Variable solltet ihr nun in eurer Positionsberechnung nutzen, indem ihr einfach das „+1“ durch eure Richtungsvariable ersetzt.
3. Damit der Motor seine Richtung im richtigen Moment ändert, müsst ihr zwei **Wenn-dann**-Anweisungen einfügen, die die Richtungsvariable auf 1 oder -1 setzen, wenn sich der Motor zu weit drehen würde. Das funktioniert im Prinzip so:
 - Wenn** (neue Position über 178), **dann**:
Setze Richtungsvariable auf -1;
 - Wenn** (neue Position unter 1), **dann**:
Setzte Richtungsvariable auf 1;
4. Testet euer Programm. Wenn ihr alles richtiggemacht habt, dann sollte sich der Motor zwischen 0° und 179° hin und her bewegen.
5. Denkt daran, euch die Servoposition über den seriellen Monitor ausgeben zu lassen, um eure Werte genauer zu überprüfen.

Station 1 – Elektrischer Rollladen

FUNKTIONSWEISE DES ROLLADENS

Super, jetzt wisst ihr alles, um den Rollladen zu steuern. Der Rollladen soll sich immer in kleinen Stücken bewegen und seine Bewegungsrichtung ändern, wenn die Lichtverhältnisse sich ändern.



1. Lasst den Motor nun seine Bewegungsrichtung ändern, wenn es dunkler wird. Ansonsten soll die Richtung beibehalten werden.
 - a. Übernehmt die Variablen für die beiden Helligkeitswerte (neu und alt) aus dem Teil für den Umgebungslichtsensor.
 - b. Benutzt das, was ihr über den Umgebungslichtsensor gelernt habt, um eure **Schleife** anzupassen, sodass euer Programm grob so funktioniert:
 1. Messe und speichere Helligkeitswert
 2. Passe Richtung an
 3. Berechne neue Position für Servo
 4. Bewege Servo
 5. Wenn Position 179 oder 0: stoppe Bewegung
 6. Warte bis zu Ende gedreht
 7. Speichere Helligkeitswert in Variable für alten Helligkeitswert
2. Testet euer Programm. Beobachtet auch die Helligkeitswerte auf dem **SERIELLEN MONITOR**. Wenn es nicht funktioniert, kann das an einem Flüchtigkeitsfehler in euren **Wenn-dann**-Anweisungen liegen. Es kann aber auch an zu großen oder zu kleinen Werten eurer **Warte**-Blöcke liegen. Vielleicht habt ihr sogar irgendwo eine Pause zu viel oder zu wenig.

GESCHAFFT!

Klasse! Jetzt habt ihr euren eigenen automatischen Rollladen zusammengeschaubt und programmiert. Ihr könnt entweder versuchen, euren Rollladen zu optimieren (z. B.: schneller oder langsamer bei extremen Werten) oder ihr versucht euch an einer anderen Station.




Quellenverzeichnis:

Abb. 1 – Quelle: InfoSphere

Abb. 2a-c, 4, 5, 6, 7, 8, 9 – Quelle: Screenshots von Tinkercad <https://www.tinkercad.com/>

Abb. 3 – Quelle: Künstler Stuart McGhee

Alle Codeblock-Screenshots – Quelle: Screenshots von Tinkercad <https://www.tinkercad.com/>

 angefertigt vom InfoSphere-Team