

Die Arduino-Blöcke

Es gibt fünf verschiedene Arduino-Blöcke. Diese werden genutzt, um an den einzelnen Pins anliegende Signale der angeschlossenen Sensoren auszulesen und um Signale über die Pins nach außen an die Aktoren zu geben.

Setze Servo



Dieser Block dient dazu, einem angeschlossenen Servo einen Wert zuzuweisen.

Grove-Base-Shield: Der mitgelieferte Servo kann an allen digitalen Pins (D2-D8) genutzt werden. Es besteht zudem keine Notwendigkeit einer zusätzlichen Stromquelle.

MyIG-Shield: Die Servos dürfen nur an die dafür vorgesehenen Pins auf dem Shield (4, 7, 8, 12) angeschlossen werden, um Schäden am Board durch den höheren Stromverbrauch zu vermeiden. **Servos dürfen nur mit zusätzlicher Stromquelle (Netzteil oder Batterie) betrieben werden!**

Es gibt zwei Arten von Servos. Continuous Rotation (CR)-Servos können sich kontinuierlich mit oder gegen den Uhrzeigersinn drehen, Standard-Servos können auf einen bestimmten Winkel eingestellt werden. Die Auswahl im ersten, runden Feld muss dem Pin entsprechen, an dem der Servo angeschlossen ist. (Bsp. Grove: Servo an D4 → Auswahl „4“; Bsp. MyIG: Servo an 4 → Auswahl „4“)

Erklärung des Auswahlmenüs

Winkel (0-180): Der Winkel für einen angeschlossenen Standard-Servo wird hier eingestellt. Dazu wird ein Wert zwischen 0 und 180 eingegeben, was ungefähr einer Bewegung des Servos zwischen 0° bis 180° entspricht. Der genaue Minimal- und Maximalwert kann von Servo zu Servo unterschiedlich sein, sollte sich aber in der Nähe von 0 und 180 befinden. Hier muss experimentiert werden. Einige Servos lassen sich auch über ein kleines Stellrädchen mit einem Schraubendreher physisch justieren.

Folgende Menü-Punkte sind ausschließlich für CR-Servos:

gestoppt: Ein CR-Servo wird mit diesem Block angehalten. Sollte der Servo nicht komplett anhalten, muss der tatsächliche Stoppwert experimentell ermittelt

werden. Der Wert befindet sich in der Nähe von 90 und muss in diesem Fall manuell als Zahl eingegeben werden, indem man einmal auf das bereits ausgewählte Wort „gestoppt“ klickt. Auf diese Weise lässt sich auch die Drehgeschwindigkeit regulieren.

im Uhrzeigersinn: Ein CR-Servo wird mit diesem Block gestartet und dreht sich fortan kontinuierlich mit dem Uhrzeigersinn.

gegen den Uhrzeigersinn: Ein CR-Servo wird mit diesem Block gestartet und dreht sich fortan kontinuierlich gegen den Uhrzeigersinn.

Grove Starter Kit: enthält einen Standard-Servo

MyIG Toolbox: enthält einen Standard- und einen CR-Servo

Setze digitalen Pin



Dem angegebenen digitalen Ausgang wird ein Wahrheitswert (**wahr** oder **falsch**) zugewiesen. Das bedeutet, dass an den Pin nur „Strom soll fließen“ oder „Strom soll nicht fließen“ ausgegeben werden kann. In der Rundung hinter dem Wort „Pin“ wird die Nummer des Pins eingetragen. Im rechten Sechseck wird der Wahrheitswert per Drag & Drop eingetragen.

Grove-Base-Shield: Die digitalen Pins D2-D8 des Arduino Uno werden durchgereicht und können genutzt werden.

MyIG-Shield: Die digitalen Pins D2-D13 des Arduino Uno werden durchgereicht und können genutzt werden.

Setze PWM-Pin



Die digitalen Pins 3, 5, 6, 9, 10 und 11 des Arduino lassen sich per PWM nutzen. PWM steht für Pulsweitenmodulation und bedeutet, dass die digitalen Signale in pseudoanaloge Signale umgewandelt werden. So entsteht die Möglichkeit, 256 verschiedene Werte (0 bis 255) ausgeben zu können, die in der rechten Rundung des Blocks eingegeben werden. Im linken Auswahlfeld wählt man den entsprechenden PWM-Pin.

Am **Grove-Base-Shield** können nur die Ports D3, D5 und D6 für PWM genutzt werden (Shieldlimitierung).

Am **MyIG-Shield** können alle sechs vom Arduino zur Verfügung gestellten Pins benutzt werden.

Lies analogen Pin

lies analogen Pin



Mit diesem Block können analog angeschlossene Sensoren ausgelesen werden. Die am Pin anliegenden Spannungswerte (0 bis 5 Volt) werden hierbei in 1024 verschiedene Zahlenwerte (0 bis 1023) umgerechnet. Nur die im Bereich „ANALOG IN“ befindlichen Pins A0 bis A5 am Arduino können als analoge Eingänge verwendet werden. Der entsprechende Eingang wird im Auswahlménü gewählt.

Es handelt sich bei diesem Block um einen Reporter-Block (abgerundete Gestalt). Das bedeutet, dass bei Klick auf den Block der aktuelle Wert des gewählten Pins angezeigt wird.

Grove-Base-Shield: Pins A0-A3 werden vom Arduino durchgereicht und können genutzt werden.

MyIG-Shield: Alle analogen Pins des Arduino Uno werden durchgereicht und können genutzt werden.

Lies digitalen Pin

lies digitalen Pin



Mit diesem Block können digital angeschlossene Sensoren ausgelesen werden. Die am Pin anliegenden Spannungswerte (0 oder 5 Volt) werden hierbei in die Wahrheitswerte (**wahr** oder **falsch**) umgerechnet. Der entsprechende Eingang wird im Auswahlménü gewählt.

Es handelt sich bei diesem Block um einen Reporter-Block (abgerundete Gestalt). Das bedeutet, dass bei Klick auf den Block der aktuelle Wert des gewählten Pins angezeigt wird.

Grove-Base-Shield: Die digitalen Pins D2-D8 des Arduino Uno werden durchgereicht und können genutzt werden.

MyIG-Shield: Die digitalen Pins D2-D13 des Arduino Uno werden durchgereicht und können genutzt werden.

Beispiele



(Abb. 7)

Ein Programm wird gestartet, indem die grüne Flagge oben rechts im Fenster angeklickt wird. (Abb. 7) Ein einzelner Programmblock kann auch gestartet werden, indem mit der linken Maustaste einmal auf den Programmblock geklickt wird. Laufende Programme sind grün umrandet. Enthält ein Programm Fehler und kann nicht ausgeführt werden, so ist es rot umrandet. Um das Programm wieder zu beenden, wird der rote Punkt oben rechts in der Ecke des Programmfensters angeklickt oder erneut einmal mit der linken Maustaste auf den Programmblock geklickt.

Blinkende LED



Zunächst wird der Start-Block mit der grünen Fahne hinzugefügt.

Im Schleifen-Block werden dem digitalen Pin 2 im Wechsel die Werte **wahr** und **falsch** zugewiesen. Damit sich ein nicht zu schnelles Blinken ergibt, wurde zusätzlich ein Warteblock eingefügt, der die angeschlossene LED jeweils 5 Sekunden in ihrem Zustand belässt. Alles im „fortlaufend“-Block Befindliche wird in einer Dauerschleife solange ausgeführt, bis das Programm beendet oder die Verbindung zum Arduino getrennt wird.

