

Joy-Pi



Joy-Pi

INHALTSVERZEICHNIS

1. Übersicht

2. Wechsel zwischen den Modulen

3. Scratch

4. Scratch mit GPIOs

5. Lektionen

Lektion 1: Buzzer ansteuern

Lektion 2: Buzzer mit Taster ansteuern

Lektion 3: Relais steuern

Lektion 4: Bewegung erkennen

Lektion 5: Neigung erkennen

Lektion 6: Berührung erkennen

Lektion 7: Vibrationsmotor steuern

Lektion 8: Blinkende LED

Lektion 9: Geräusche erkennen

6. Support



Die Anmeldedaten sind:

Username: pi

Passwort: 12345

1. ÜBERSICHT

Sehr geehrter Kunde,
vielen Dank, dass Sie sich für unser Produkt entschieden haben.

Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist. Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

Die folgenden Lektionen sind so konzipiert, dass Sie, unabhängig davon wie viel Vorwissen Sie bereits haben, alle Lektionen ohne Probleme ausführen können. Für die verschiedenen Lektionen müssen Sie Beispieldateien herunterladen und auf dem Joy-Pi ausführen. Wie Sie dies durchführen, können Sie ebenfalls dieser Anleitung entnehmen.

Doch diese Tutorials sind nur der Anfang.
Wir freuen uns darauf zu sehen, was Sie mit unserem Joy-Pi machen werden.

2. WECHSEL ZWISCHEN DEN MODULEN



Auf der Joy-Pi-Platine befinden sich zwei Schaltereinheiten mit je 8 Schaltern. Die Schalter ermöglichen es, zwischen verschiedenen Sensoren und Modulen zu wechseln. Da der Raspberry Pi nur eine begrenzte Anzahl an GPIO-Pins hat, werden diese Schalter benötigt um mehr Sensoren und Module verwenden zu können als GPIO-Pins vorhanden sind.

Die Verwendung dieser Schalter ist ziemlich einfach und wird in einigen der folgenden Lektionen benötigt.

In der Tabelle sehen Sie, welcher Schalter welchen Sensor bzw. welches Modul schaltet:

Sensoren / Module	Schalteinheit	Schalter
Taster-Matrix	Links	1 - 8
unabhängige Taster	Links	5 - 8
Vibrationsmodul	Rechts	1
Neigungssensor	Rechts	2
Schrittmotor	Rechts	3, 4, 5, 6
Servomotor	Rechts	7, 8

3. WAS IST SCRATCH



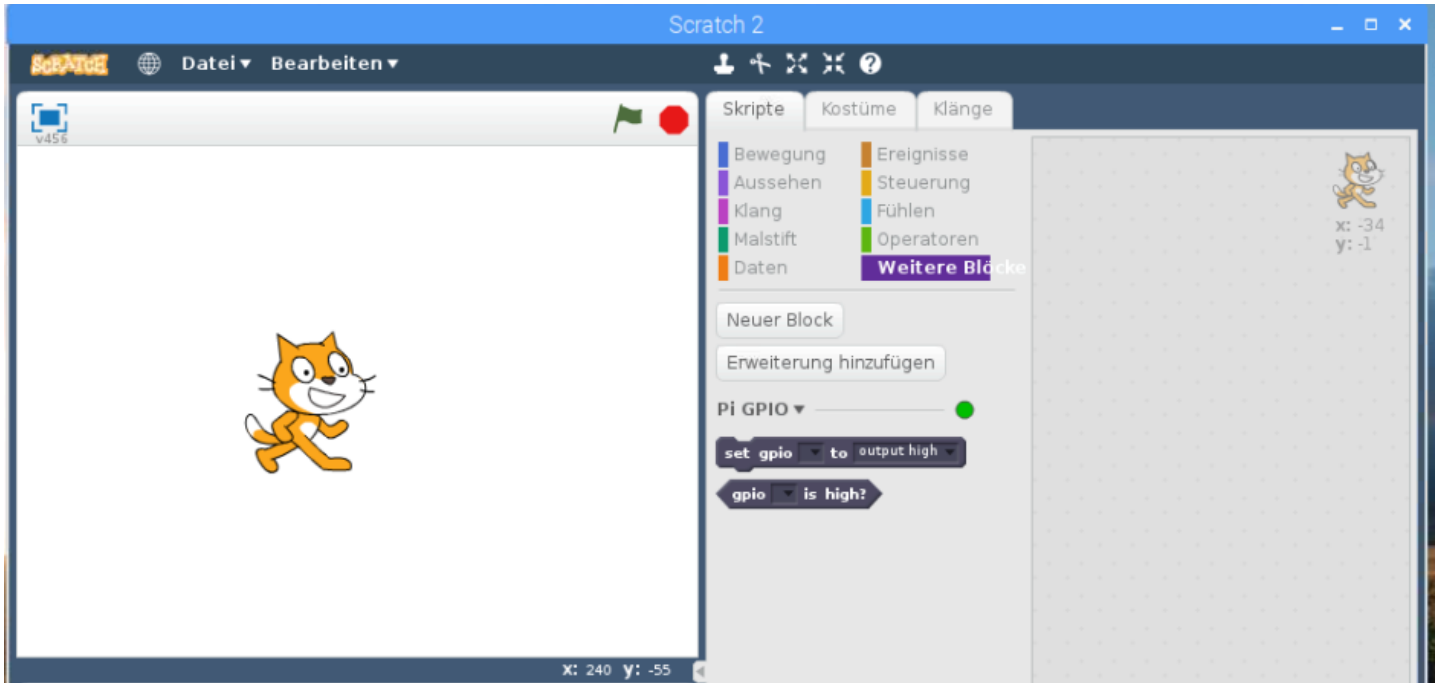
Scratch ist eine grafische Programmiersprache und gehört zu den sogenannten erziehungsorientierten Programmiersprachen. Sie wurde am MIT entwickelt und hat das Ziel Menschen die Grundlagen der Programmierung beizubringen. Im Vordergrund steht die sehr einfache Bedienung. Mit einem Editor lassen sich Scratchprojekte erstellen und auf der Scratch-Website veröffentlichen.

Der Editor ist in 3 Hauptbereiche aufgeteilt. Die Bühne, der Objektkatalog und eine Detailansicht in der sich der Skriptbereich versteckt. Man kann die Bühne und jedes Objekt über diesen Skriptbereich mittels vorgefertigter Blöcke programmieren. Die Befehle reichen von Bewegungen, Aussehen und Klängen bis hin zu Daten, Steuerung und Fühlen.

Bei den Rasbian-Betriebssystemen ist Scratch bereits vorinstalliert.

Zur offiziellen Scratch Website können Sie [hier](#) gelangen.

4. SCRATCH MIT GPIOS

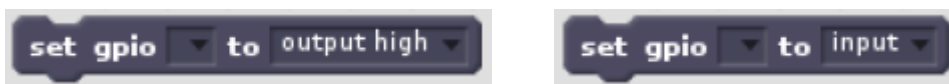


In dieser Einführung wird Ihnen erklärt, wie Sie die GPIO-Funktion in Scratch hinzufügen und wie Sie diese benutzen.

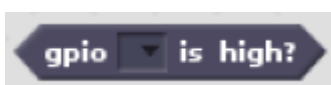
Öffnen Sie zunächst Scratch und klicken dann auf „Weitere Blöcke“. Klicken Sie nun auf „Erweiterung hinzufügen“ und dann auf „PI GPIO“.

Sie haben die GPIO-Funktion nun erfolgreich hinzugefügt. Jetzt sollten Sie, wie auch im Bild oben zusehen, eine neue Untergruppe mit dem Namen „PI GPIO“ auf der rechten Seite sehen können. Dort gibt es zwei Funktionen:

Mit der ersten Funktion können Sie Pins auf „high“ oder „low“ setzen. Außerdem können Sie damit einen Pin als Input festlegen.



Die zweite Funktion ist zum Abfragen des Zustands von einem Pin.



Achten Sie darauf, dass Sie in den beiden Funktionen immer die GPIO.BCM-Nummer angeben und nicht die Pin-Nummer.

4. SCRATCH MIT GPIOS

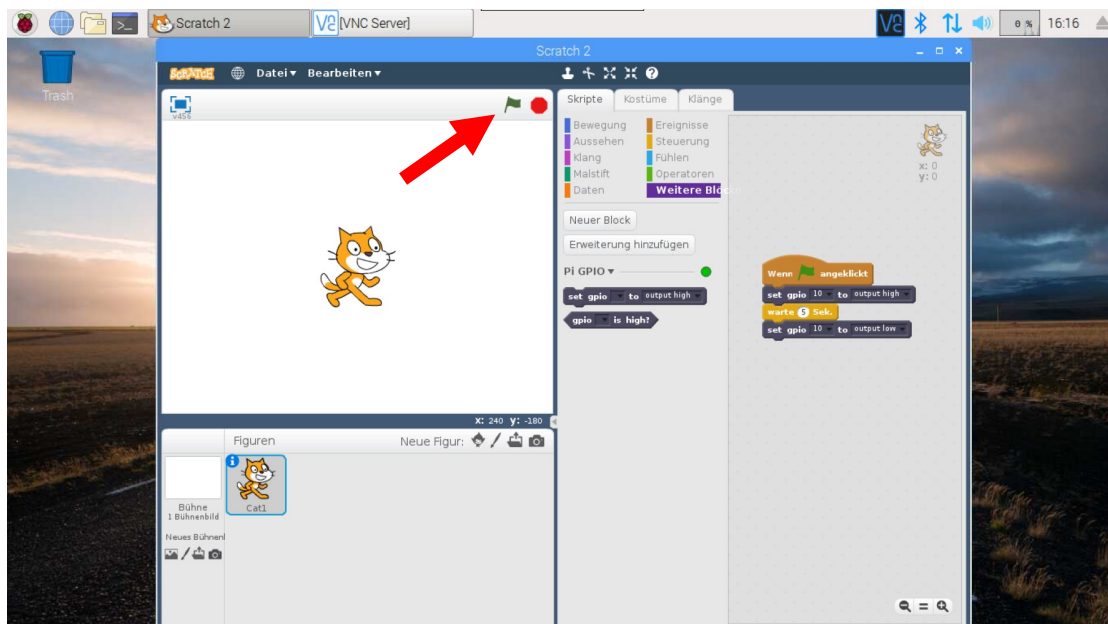
Um Scratch etwas besser kennen zulernen, bevor die erste Lektion beginnt, ist hier ein kleines Beispiel.



In diesem Beispiel wird, wenn die grüne Flagge angeklickt wird, GPIO 10 für 5 Sekunden auf „high“ gesetzt und danach wieder auf „low“.

Das bewirkt, wenn zum Beispiel eine LED an den Pin angeschlossen wird, dass die LED nach dem die Flagge angeklickt wurde für 5 Sekunden leuchtet und danach wieder ausgeht.

In dem folgendem Bild können Sie sehen, wo sich die Flagge befindet, mit der, in diesem Beispiel, das Programm gestartet wird.



5. LEKTION 1: BUZZER STEUERN

In dieser Lektion werden Sie lernen, wie der Buzzer funktioniert und wie man ihn mit Scratch programmieren kann.

Der Buzzer auf dem Joy-Pi ist über GPIO18 mit dem Raspberry Pi verbunden.

In dem Beispiel wird der Pin des Buzzers, wenn die Flagge angeklickt wird, für eine halbe Sekunde auf HIGH und danach wieder auf LOW gesetzt.

Entfernen Sie diesen Aufkleber vom Buzzer:



Wenn Sie wollen, dass das Programm endlos durchläuft, können Sie auch eine Schleife „wiederhole fortlaufend“ hinzufügen. In dem Fall kann das Programm nur durch klicken auf den roten Kreis neben der grünen Flagge in Scratch angehalten werden.

Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. Setze den Buzzer an GPIO18 auf HIGH
3. Warte 0,5 Sekunden
4. Setze den Buzzer an GPIO18 auf LOW



5. LEKTION 2: BUZZER MIT TASTER ANSTEUERN

In dieser Lektion werden Sie lernen, wie der Buzzer Tasterdruck gesteuert wird.

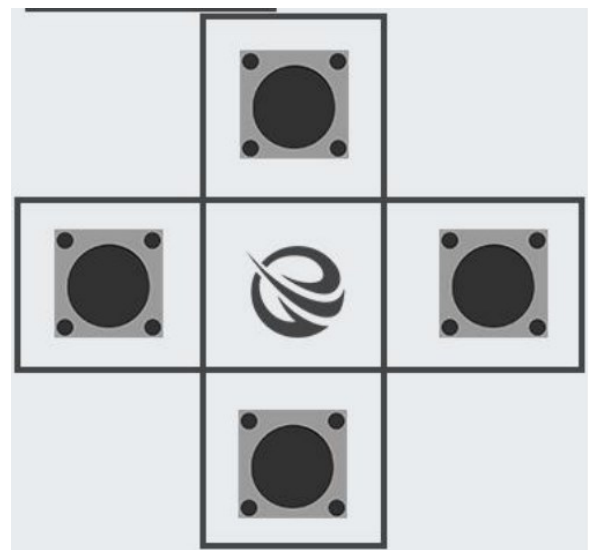
Der Buzzer ist an Pin GPIO18 (Ausgang) und der obere Taster an GPIO26 (Eingang) angeschlossen. Wenn der Taster gedrückt wird (Eingang), wird der Ausgang auf HIGH gesetzt (Buzzer piept). Wird der Taster losgelassen geht das Ausgangssignal wieder auf LOW (Buzzer geht aus).

Es wird der obere Taster des Taster Kreuzes benutzt:

Achtung:

Das Joy-Pi Board setzt das Signal bei einem gedrückten Taster auf LOW und wenn er losgelassen wird wieder auf HIGH. Berücksichtigen Sie dieses, wenn Sie eigene Programme erstellen.

Die Schalter des linken Schalterblocks müssen alle auf „on“ gestellt sein.



Block Struktur

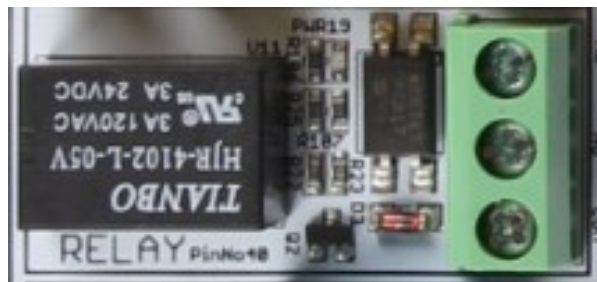
1. Mit Klick auf die grüne Flagge startet das Programm
2. GPIO 26 (Taster) als Eingang
3. Endlosschleife
4. Wenn GPIO26 HIGH (Taster nicht gedrückt)
5. Setze GPIO18 LOW (Buzzer kein Ton)
6. Sonst GPIO18 HIGH (Buzzer piept)



5. LEKTION 3: RELAIS STEUERN

In dieser Lektion werden Sie lernen, wie das Relais mit Hilfe von Scratch gesteuert wird.

Das Relais ist an Pin GPIO21 angeschlossen. GPIO21 wird auf LOW gesetzt um das Relais zu öffnen. Nach einer Pause von einer Sekunde wird GPIO21 auf HIGH gesetzt um das Relais zu schließen.



Achtung:

Es ist sehr wichtig, nicht zu versuchen, Hochspannungsgeräte an das Relais anzuschließen (z.B. Tischlampe, Kaffeemaschine usw.). Dies könnte zu Stromschlägen und schweren Verletzungen führen.

Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. GPIO21 LOW (Relais öffnet)
3. Warte eine Sekunde
4. GPIO21 HIGH (Relais schließt)



5. LEKTION 4: BEWEGUNG ERKENNEN

In dieser Lektion werden Sie lernen, wie Sie mit Hilfe des Bewegungssensors Bewegungen erkennen.

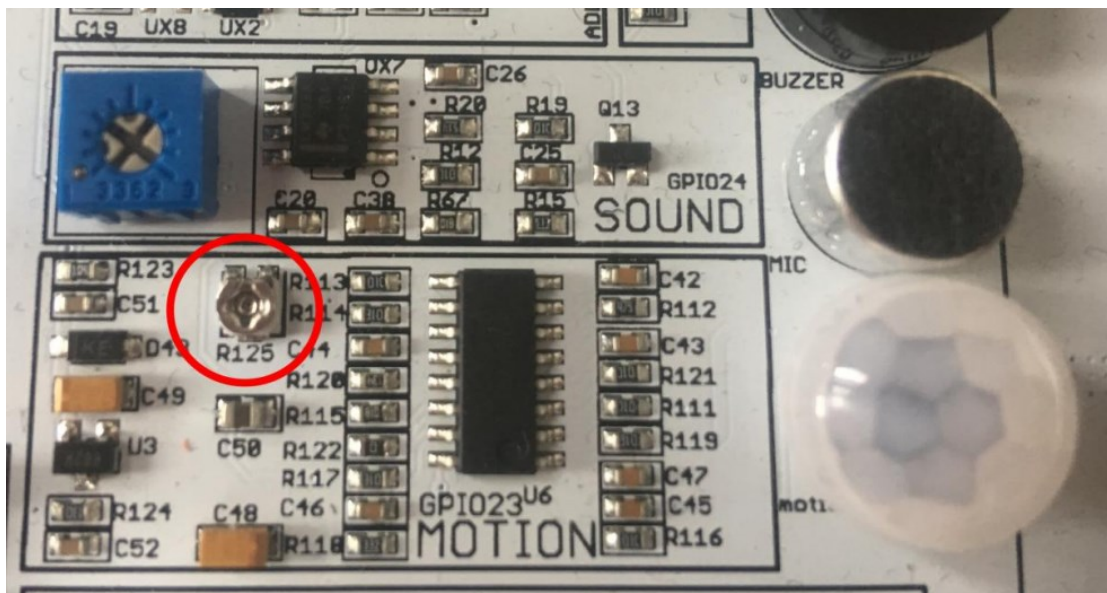
Der Bewegungssensor ist an GPIO23 angeschlossen. GPIO23 wird als Eingang konfiguriert. Das Signal des Sensors wird (wie in vorherigen Lektionen auch) durch eine wenn-dann-Abfrage verarbeitet und eine entsprechende Nachricht wird ausgegeben.

Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. GPIO23 als Eingang (Bewegungssensor)
3. Endlosschleife
4. Wenn GPIO23 HIGH
5. Sage „Bewegung erkannt!“ (etw. bewegt sich)
6. Sonst sage „Nichts bewegt sich...“ (keine Bewegung)



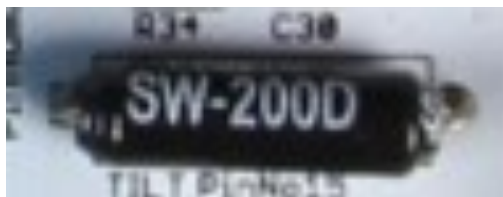
Die Empfindlichkeit des Bewegungssensors kann mit Hilfe des rot eingekreisten Potentiometers verstellt werden.



5. LEKTION 5: NEIGUNG ERKENNEN

In dieser Lektion werden Sie lernen, wie Sie mit Hilfe des Neigungssensors erkennen ob der Joy-Pi nach rechts oder nach links gekippt wird.

Der Neigungssensor ist an Pin GPIO22 angeschlossen. Wird der Sensor nach links gekippt gibt dieser ein HIGH Signal zurück. Wird der Sensor nach rechts gekippt wird das Signal LOW. In einer wenn-dann-Schleife wird das Signal anschließend verarbeitet.



Achtung:

Schalter 2 des rechten Schalterblocks muss auf „on“ gestellt sein.

Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. GPIO22 als Eingang (Bewegungssensor)
3. Endlosschleife
4. Wenn GPIO22 HIGH (Joy-Pi nach links gekippt)
5. Sage „left“
6. Sonst sage „right“ (Joy-Pi nach rechts gekippt)



5. LEKTION 6: BERÜHRUNG ERKENNEN

In dieser Lektion werden Sie lernen, wie Sie mit Hilfe des Touchsensors Berührungen erkennen.

Der Touchsensor ist an GPIO17 angeschlossen. Wie in vorherigen Lektionen auch, wird GPIO 17 als Eingang konfiguriert. Der Buzzer wird in diesem Beispiel auf HIGH gesetzt sobald der Touchsensor nicht mehr berührt wird, wenn der Sensor berührt wird, wird der Buzzer auf LOW gesetzt und geht aus.



Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. GPIO17 als Eingang (Touchsensor)
3. Endlosschleife
4. Wenn GPIO17 HIGH (Berührung erkannt)
5. Setze GPIO18 auf LOW
6. Sonst (keine Berührung erkannt)
7. Setze GPIO 18 auf HIGH



5. LEKTION 7: VIBRATIONSMOTOR STEUERN

In dieser Lektion werden Sie lernen, wie Sie den Vibrationsmotor ansteuern.

Der Vibrationsmotor ist an den Pin GPIO27 angeschlossen. Damit der Vibrationsmotor vibriert muss das Signal HIGH sein. Wird das Signal auf LOW geschaltet, stoppt der Motor wieder.

Um den Vibrationsmotor nutzen zu können müssen die Schalter des rechten Schalterblocks auf „On“ stehen.



Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. GPIO27 HIGH (Vibrationsmotor an)
3. Warte 0,5 Sekunden
4. GPIO27 LOW (Vibrationsmotor aus)

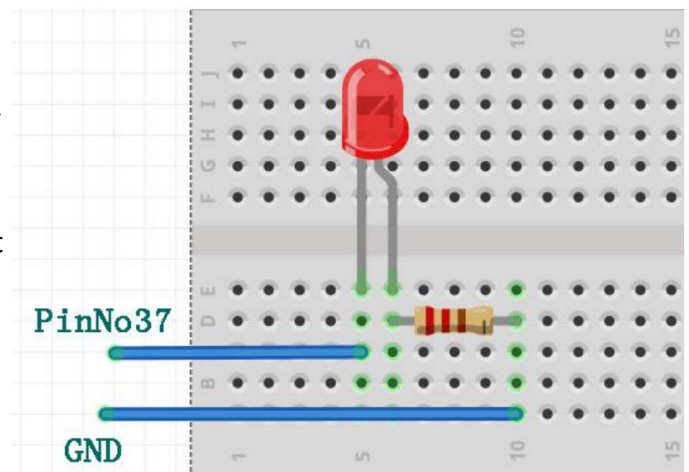


5. LEKTION 8: BLINKENDE LED

In dieser Lektion werden Sie lernen, wie Sie einen einfachen elektronischen Schaltkreis erstellen um eine LED zu steuern. Im Anschluss werden Sie mit Hilfe des Scratch Programms die LED blinken lassen.

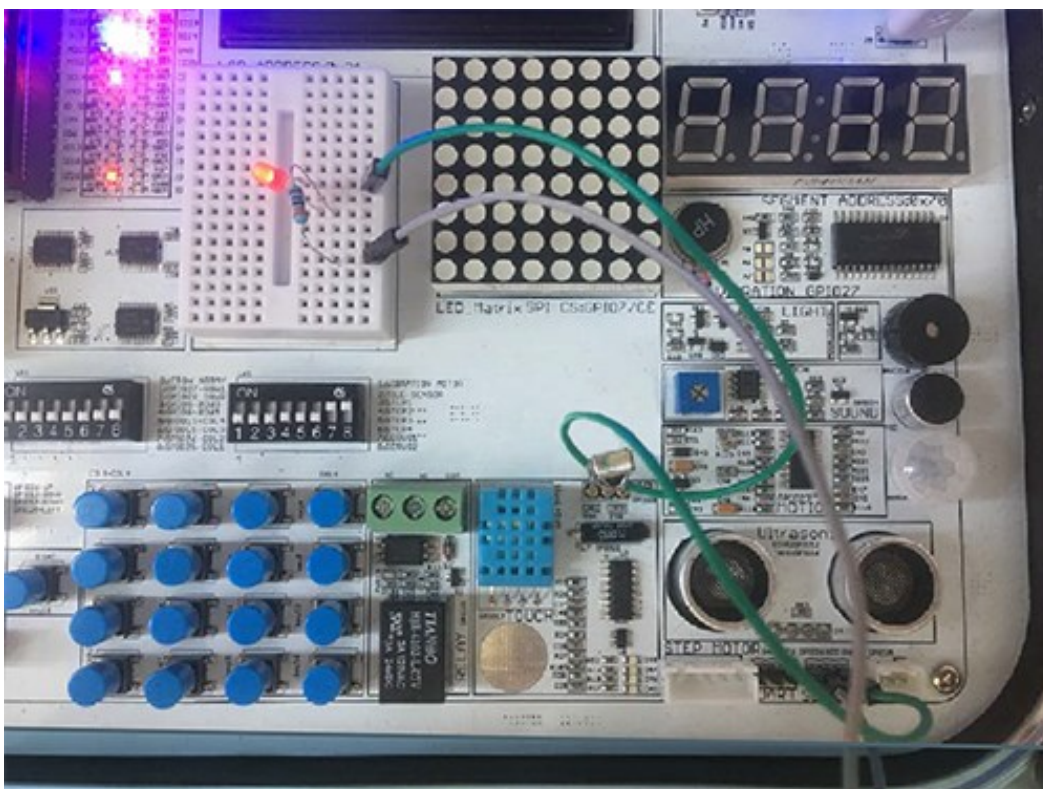
Für diese Lektion wird der Pin GPIO26 genutzt. Die Besonderheit ist, dass dieser Pin eigentlich zum Servoanschluss gehört, für diese Lektion aber zweckentfremdet wird.

Der Schaltkreis wird, wie auf der Abbildung rechts, auf dem Experimentierbrett aufgebaut. Pin 37 ist GPIO26 und ist am Servo 1 Anschluss zu finden. GND befindet sich am gleichen Stecker. Des Weiteren muss beachtet werden, dass die Leuchtdiode mit richtiger Polarität eingesetzt werden muss. Der Widerstand dient dazu die Diode vor zu großen Strömen zu schützen.



Für diese Lektion müssen die Schalter des rechten Schalterblocks auf „ON“ stehen.

Das Ergebnis sollte dann so aussehen:



5. LEKTION 8: BLINKENDE LED

Jetzt muss nur noch der Codeblock erstellt werden. Wir werden innerhalb einer Endlosschleife den Pin GPIO26 HIGH setzen um die LED einzuschalten. Im Anschluss folgt eine 0,3 sekundige Pause. Dann wird die LED durch GPIO26 LOW wieder ausgeschaltet und es folgt nochmal eine 0,3 sekundige Pause. Das Ergebnis ist eine blinkende LED.

Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. Endlosschleife
3. GPIO26 HIGH (LED an)
4. Warte 0,3 Sekunden
5. GPIO26 LOW (LED aus)
6. Warte 0,3 Sekunden



5. LEKTION 9: GERÄUSCHE ERKENNEN

In dieser Lektion werden Sie lernen, wie Sie mit Hilfe des Soundsensors Geräusche in der Umgebung erkennen.

Der Soundsensor ist an den Pin GPIO24 angeschlossen. Das Signal ist LOW, wenn ein Geräusch detektiert wurde. Ist die Umgebung ruhig ist das Signal HIGH.



Mit Hilfe des rot eingekreisten Potentiometers, kann die Empfindlichkeit des Soundsensors eingestellt werden. Dieses kann helfen falls der Sensor immer oder nie ein Geräusch erkennt.

Block Struktur

1. Mit Klick auf die grüne Flagge startet das Programm
2. GPIO24 als Eingang (Soundsensor)
3. Endlosschleife
4. Wenn GPIO22 HIGH (kein Geräusch erkannt)
5. Sage „Kein Geräusch erkannt...“
6. Sonst sage „Geräusch erkannt!“ (Geräusch erkannt)



6. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net
Ticket-System: <http://support.joy-it.net>
Telefon: +49 (0)2845 98469 – 66 (10- 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net