

CHEAT SHEET (CODE)

Serialle Ausgabe:

(zu finden) in:

Fortgeschritten -> Seriell

dauerhaft

Führt den Code im Inneren immer und immer wieder aus.

seriell Zeile ausgeben "A message to myself"

Gibt den Text, den wir darin geschrieben haben, aus

seriell Zeile ausgeben meineVariable ▼

Gibt den Wert, der Variablen mit dem Namen "meineVariable", aus.

||| Konsole anzeigen Gerät

Klicke hier, um die Konsole zu öffnen und zu sehen, was ausgegeben wird.

* "Konsole anzeigen Simulator" ist nur sichtbar, wenn eine Zeile ausgegeben wird.

Programm-Struktur:

(zu finden) in:

Grundlagen

beim Start

Führt den Code im Inneren einmal beim Start des Programms aus.

seriell Zeile ausgeben "Printed only at start"

Gibt die Nachricht "Printed only at start" aus

dauerhaft

Führt den Code im Inneren immer und immer wieder aus.

seriell Zeile ausgeben "Printed again and again"

Gibt die Nachricht "Printed again and again" aus.

Pausiere:

(zu finden) in:

Grundlagen

dauerhaft

Führt den Code im Inneren immer und immer wieder aus.

seriell Zeile ausgeben "Hello" Gibt "Hello" aus.

pausiere (ms) 1000 ▼ Pausiert das Programm für 1000ms (1 Sekunde).

CHEAT SHEET (CODE)

Variablen:

(zu finden) in:

Variablen

setze myNumber auf 4
Numerische Zahl: Voreinstellung, schreibe einfach eine Zahl.

setze myString auf ""
String (Text): Gehe zum Menu "Text" (unter "Fortgeschritten") und ziehe dieses Symbol hinein, dann schreibe den Text.



setze myBoolean auf wahr
Boolescher Wert (wahr/falsch): Gehe zum "Logik" Menu und ziehe dieses Symbol hinein, wähle dann wahr oder falsch



setze myNumber auf myNumber x 2
Mathematik: Ordne der Variablen einen errechneten, numerischen Wert (Zahl) zu, indem du Zahlen oder bereits existierende numerische Variablen benutzt, gehe zum Menu "Mathe" und wähle eine der Optionen.

setze myString auf verbinde myString "World" - +
Verbinde: Kombiniere zwei Strings (Texte) zu einem, indem du neue Werte oder existierende String Variablen verwendest, gehe zum "Text" Menu (unter "Fortgeschritten") und wähle die "verbinde" Option

ändere myNumber um 1
Ändere: Dieser Befehl nimmt eine Variable und addiert 1 (oder eine andere Zahl) hinzu.

CHEAT SHEET (CODE)

Bedingungen (Struktur)

(zu finden) in:

Logik

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

setze `myValue` **auf** `0` Erstellt die Variable "myValue" und gibt ihr den Wert 0.

wenn `myValue` **>** `6` **dann** Falls dies wahr ist, führe den Code im Inneren aus und überspringe alle "sonst wenn" und "ansonsten" Blöcke.

seriell Zeile ausgeben "If true, run this code"

sonst wenn `myValue` **=** `4` **dann** Falls das Vorherige falsch war, aber dies wahr ist, führe den Code im Inneren aus und überspringe alle weiteren "sonst wenn" und "ansonsten" Blöcke.

seriell Zeile ausgeben "Else, if this is true, run this code"

ansonsten Falls alles Vorherige falsch war, führe den Code im Inneren aus.

seriell Zeile ausgeben "Else, run this code instead"

Bedingungen (Operationen):

(zu finden) in:

Logik

`0` **=** `0` Vergleicht zwei numerische (Zahlen) oder boolsche (wahr/falsh) Werte, Beispiele:

`myValue` **=** `20`

`myValue` **+** `4` **≥** `10`

`myBoolean` **=** `wahr`

`" "` **=** `" "` Vergleicht zwei Strings (Text), Beispiele:

`myText` **=** `"Hello"`

und Überprüft, ob beide der zwei Vergleiche wahr sind

oder Überprüft, ob eine der zwei Vergleiche wahr ist

`myValue` **>** `200` **und** `myBoolean` **=** `wahr`

CHEAT SHEET (CODE)

Schleife - Wiederholung

(zu finden) in:

Schleife

beim Start

Führt den Code im Inneren einmal aus, wenn das Programm gestartet wird

für **index** von 0 bis 4
mache
 seriell Zeile ausgeben **index**

Wiederholt den Code im Inneren: "index" startet bei 0 und erhöht sich um 1 bei jedem Durchlauf. Die Wiederholung endet, wenn "index" den Wert der Zahl erreicht.

Gibt den momentanen Wert von "index" aus (0,1,2,3,4)

Schleife - für Element

(zu finden) in:

Schleife

beim Start

Führt den Code im Inneren einmal aus, wenn das Programm gestartet wird.

setze **myText** auf "Hello My Friend"
für Element **value** von **myText**
mache
 seriell Zeile ausgeben **value**
 pausiere (ms) 1000

Erstellt eine Variable "myText" und ordnet ihr den Wert "Hello My Friend" zu.

Durchläuft jedes Element (jeden Buchstaben) von "myText", einen Buchstaben nach dem anderen. Der Buchstabe wird in "value" gespeichert

Gibt den Wert der Variablen "value" (den momentanen Buchstaben) zurück

Macht eine Pause von 1000ms (1 Sekunde)

CHEAT SHEET (CODE)

Funktionen (ohne Parameter oder Rückgabe-Wert):

(zu finden) in:

Fortgeschritten -> Funktionen

dauerhaft

Führt den Code im Inneren immer und immer wieder aus.

Aufruf printHello

Ruft die Funktion "printHello" auf.

Funktion printHello

Führt den Code im Inneren aus, wenn die Funktion aufgerufen wird.

seriell Zeile ausgeben

"Howdy Friends! :D"

Gibt "Howdy Friends! :D" zurück.

Funktionen (mit Parametern und Rückgabe-Wert):

(zu finden) in:

Fortgeschritten -> Funktionen

dauerhaft

Führt den Code im Inneren immer und immer wieder aus.

setze newNumber auf Aufruf doubleThisNumber

7

Ruft die Funktion "doubleThisNumber" mit dem Parameter "7" auf und speichert den zurückgegeben Wert in der Variablen.

seriell Zeile ausgeben

newNumber

Gibt den Wert der Variablen zurück.

Funktion doubleThisNumber

num

Führt den Code im Inneren aus, wenn die Funktion aufgerufen wird, nimmt eine Zahl aus Parameter an.

seriell Zeile ausgeben

"This function doubles the number"

Gibt "This function doubles the number" aus.

num

x

2

zurückgeben

Multipliziert die Zahl mit 2 und gibt das Ergebnis zurück.

CHEAT SHEET (EFFEKTOR)

LED:

Leuchtdiode

dauerhaft

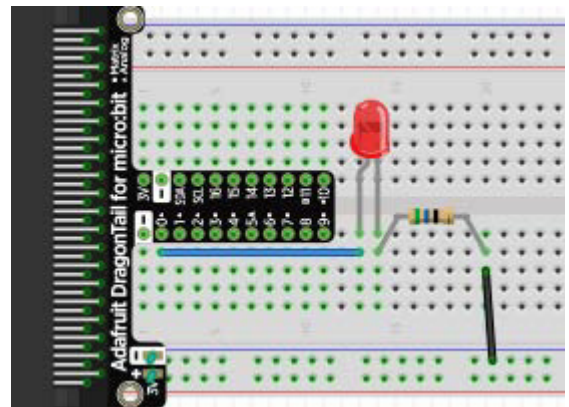
Führt den Code im Inneren immer und immer wieder aus.

schreibe digitalen Wert von Pin P0 auf 1

Schaltet die LED aus (0) oder an (1)

schreibe analogen Wert von Pin P0 auf 503

Schaltet die LED schrittweise von aus (0) auf ein (1023)



Benutzbare Pins: P0 - P16

Widerstand (grün, blau, schwarz): 56Ω



Neopixels:

Du musst das Neopixel-Menu installieren: Klicke auf Erweiterungen (ganz unten im Menu unter Fortgeschritten) -> suche "neopixel" -> klicke auf die erste Option "neopixel"

dauerhaft

Führt den Code im Inneren immer und immer wieder aus.

setze strip auf NeoPixels an Pin P0 mit 2 Pixeln und Modus RGB (GRB Format)

Variable, die die Neopixels enthält, zu finden im neopixels Menu

strip setze Helligkeit 255

Schaltet die Helligkeit nach und nach von niedrig (0) nach hoch (255)

strip setze Farbe von NeoPixel 0 auf rot

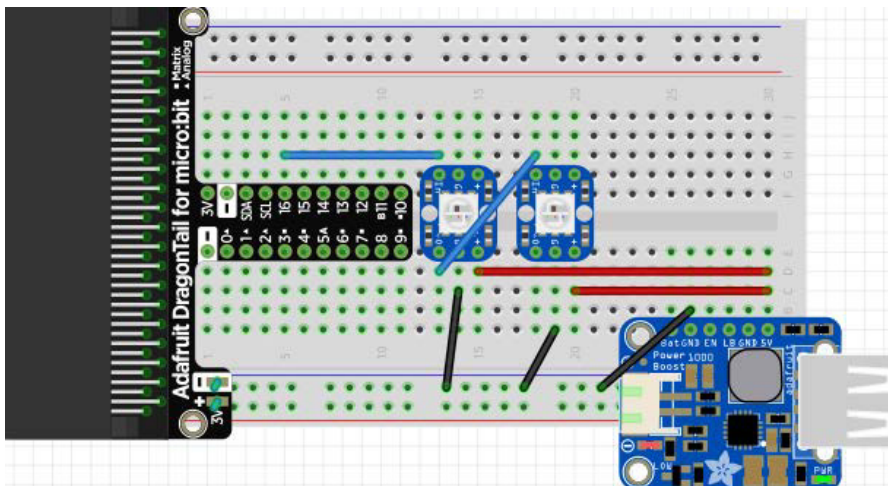
Stellt die Farbe des ersten Neopixels ein

strip setze Farbe von NeoPixel 1 auf blau

Stellt die Farbe des zweiten Neopixels ein

strip anzeigen

Aktiviert die Neopixels



Benutzbare Pins: P0 – P16

CHEAT SHEET (EFFEKTOR)

Servo Motor:

dauerhaft

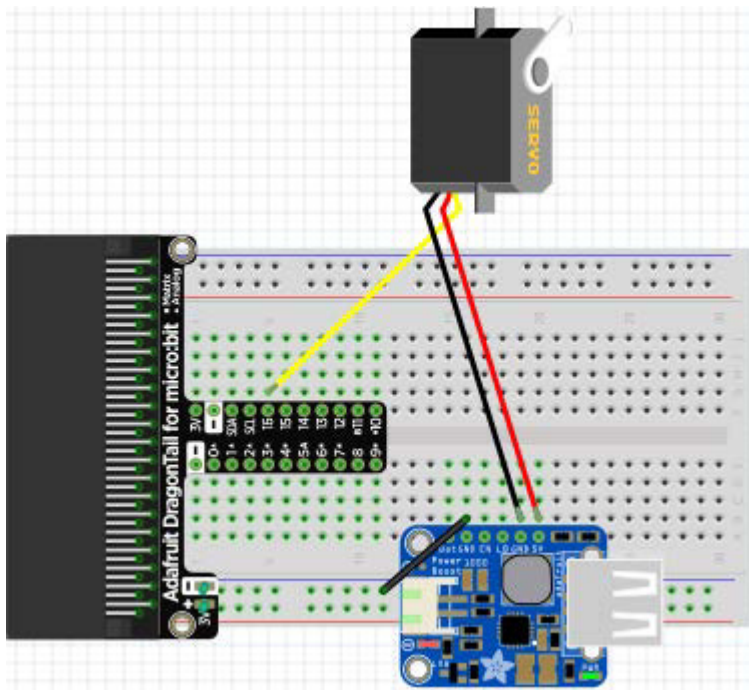
Führt den Code im Inneren immer und immer wieder aus.

setze Winkel von Servo an P16 (nur schreiben) ▼ auf 90 °

Stellt die Position des Servos ein: 0-179

pausiere (ms) 1000 ▼

Gibt dem Servo Zeit die Position zu erreichen.



Benutzbare Pins: P0 – P16

CHEAT SHEET (EFFEKTOR)

DC Motor:

Code (ohne Geschwindigkeitskontrolle):

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

schreibe digitalen Wert von Pin **P0** auf **0**

schreibe digitalen Wert von Pin **P1** auf **1**

Stellt die Richtung von Motor A ein, bei voller Geschwindigkeit

Motor A:	P0:	P1:
Stop	0	0
Links	1	0
Rechts	0	1

Motor B:	P15:	P16:
Stop	0	0
Links	0	1
Rechts	1	0

Code (mit Geschwindigkeitskontrolle):

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

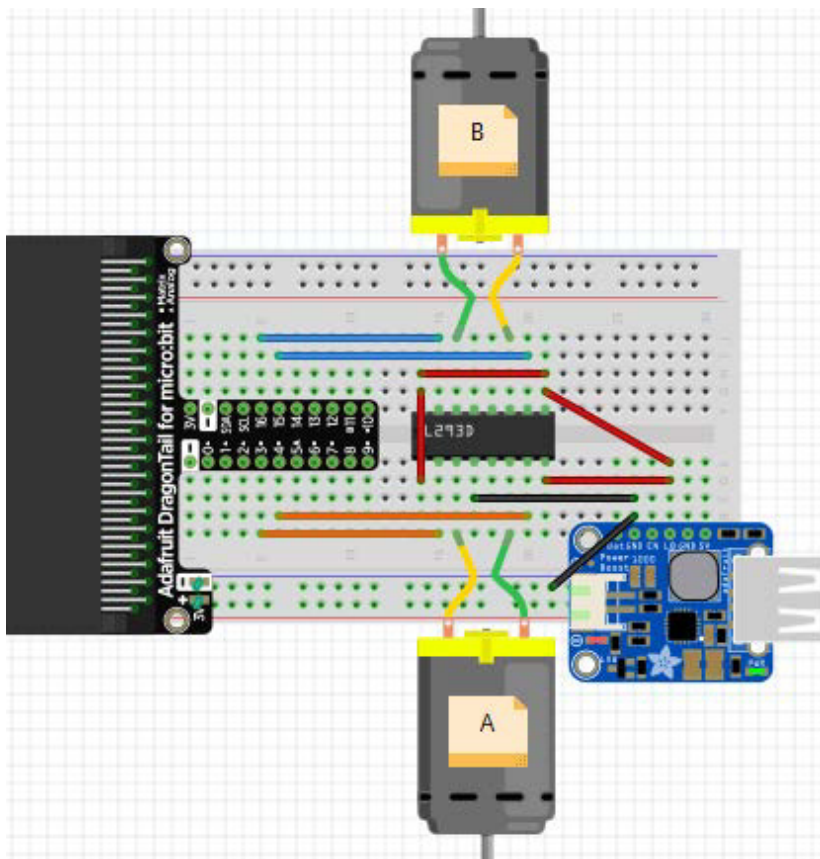
schreibe analogen Wert von Pin **P0** auf **0**

schreibe analogen Wert von Pin **P1** auf **510**

Stellt die Richtung von Motor A ein, bei halber Geschwindigkeit.

Motor A:	P0:	P1:
Stop	0	0
Links	1-1023	0
Rechts	0	1-1023

Motor B:	P15:	P16:
Stop	0	0
Links	1-1023	0
Rechts	0	1-1023



Benutzbare Pins: P0 - P16

CHEAT SHEET (SENSOREN - EXTERN)

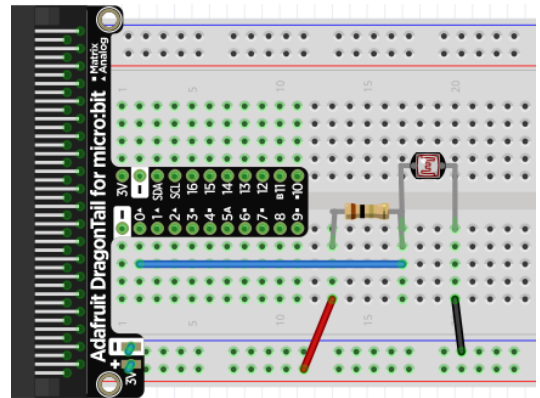
LDR

Fotowiderstand

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

setze wert auf analoge Werte von Pin P0 Liest den analogen Wert von P0 und speichert ihn

seriell Zeile ausgeben wert Gibt den Wert der Variablen aus.



Benutzbare Pins: P0-P4, P10

Widerstand (braun,schwarz,rot): 1K Ω

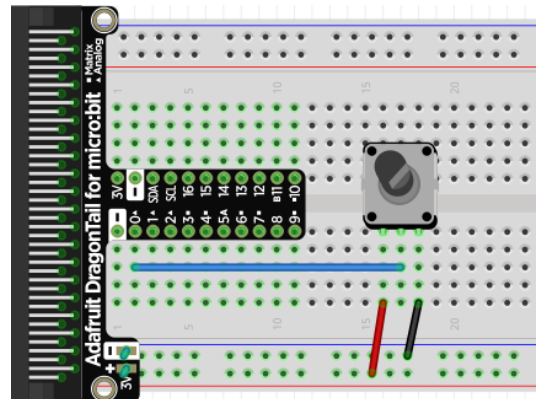


Potentiometer:

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

setze wert auf analoge Werte von Pin P0 Liest den analogen Wert von P0 und speichert ihn

seriell Zeile ausgeben wert Gibt den Wert der Variablen aus.



Benutzbare Pins: P0-P4, P10

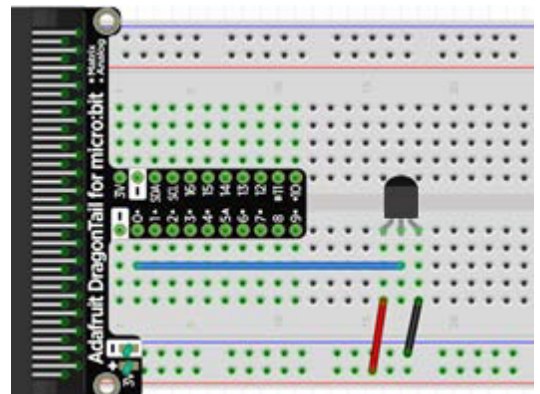
Temperature Sensor

MCP9700

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

setze wert auf analoge Werte von Pin P0 Liest den analogen Wert von P0 und speichert ihn

seriell Zeile ausgeben wert Gibt den Wert der Variablen aus.



Benutzbare Pins: P0-P4, P10

CHEAT SHEET (SENSOREN - EXTERN)

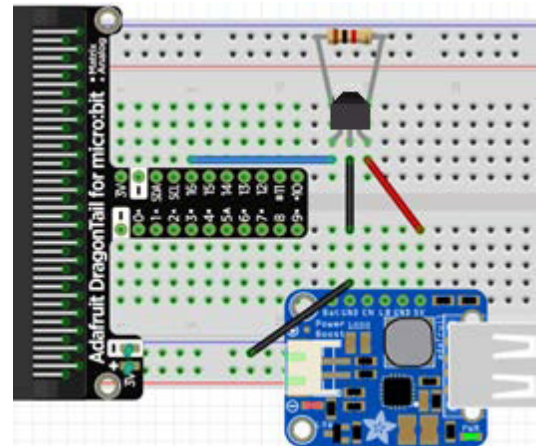
Hall Sensor

US5881

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

setze wert auf digitale Werte von Pin P0 Liest den analogen Wert von P0 und speichert ihn

seriell Zeile ausgeben wert Gibt den Wert der Variablen aus.



Benutzbare Pins: P0-P16

Widerstand (braun,schwarz,orange): 10K Ω



Knopf A/B

Mit einem Ereignis

wenn Knopf A gedrückt Führt den Code im Inneren aus, wenn der Knopf gedrückt wird.

seriell Zeile ausgeben "Pressed" Gibt "Pressed" zurück

Button A/B

Ohne ein Ereignis

dauerhaft Führt den Code im Inneren immer und immer wieder aus.

setze isPressed auf Knopf A ist gedrückt Liest den Wert, ob der Knopf gedrückt ist (wahr/falsch) und speichert ihn in der Variablen

wenn isPressed dann Falls der Wert der Variablen "wahr" ist, führe diesen Code aus:

seriell Zeile ausgeben "Pressed" Gib "Pressed" zurück.

ansonsten Falls der Wert der Variablen nicht "wahr" ist, führe diesen Code aus:

seriell Zeile ausgeben "Not Pressed" Gibt "Not Pressed" zurück.