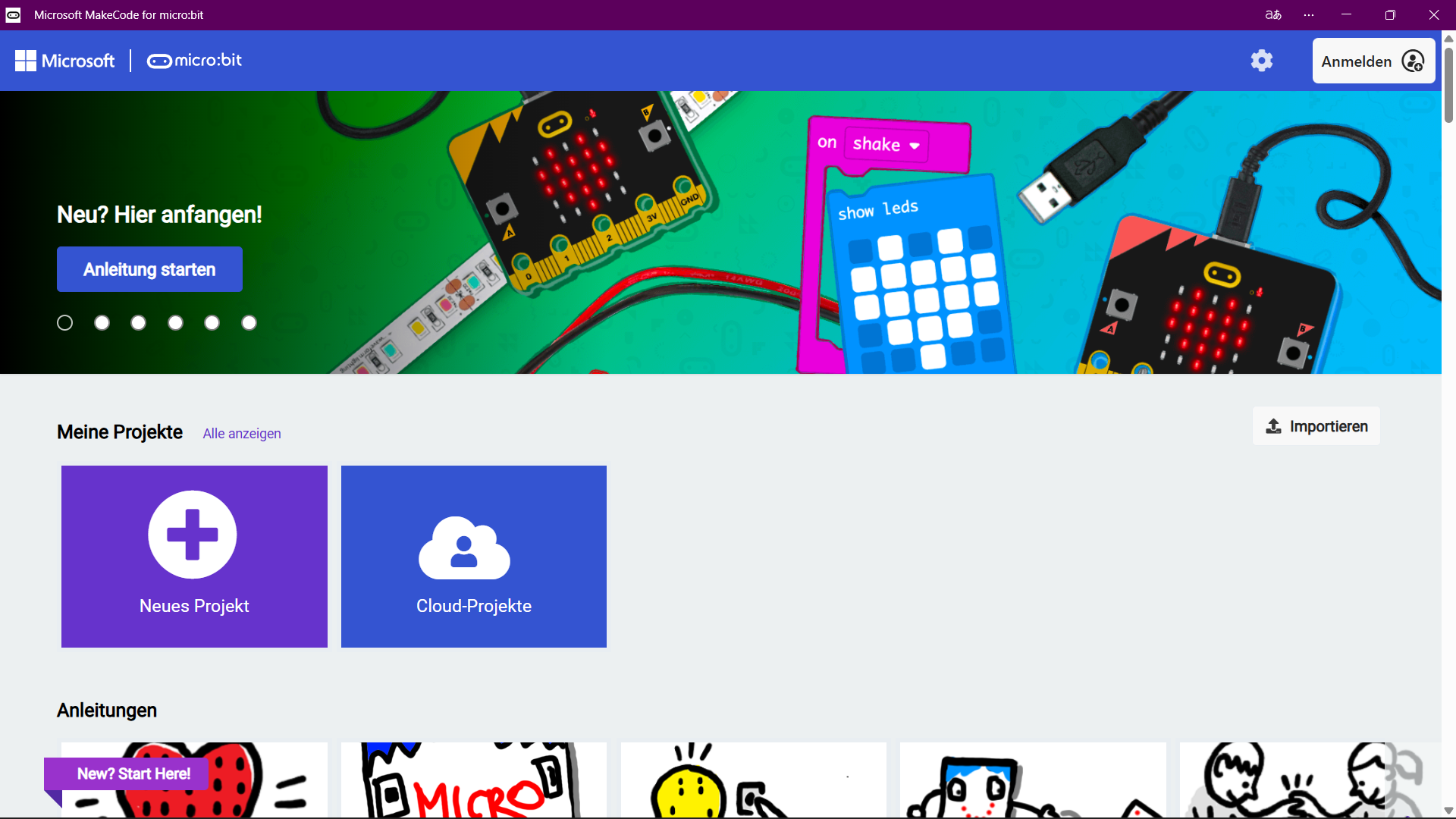
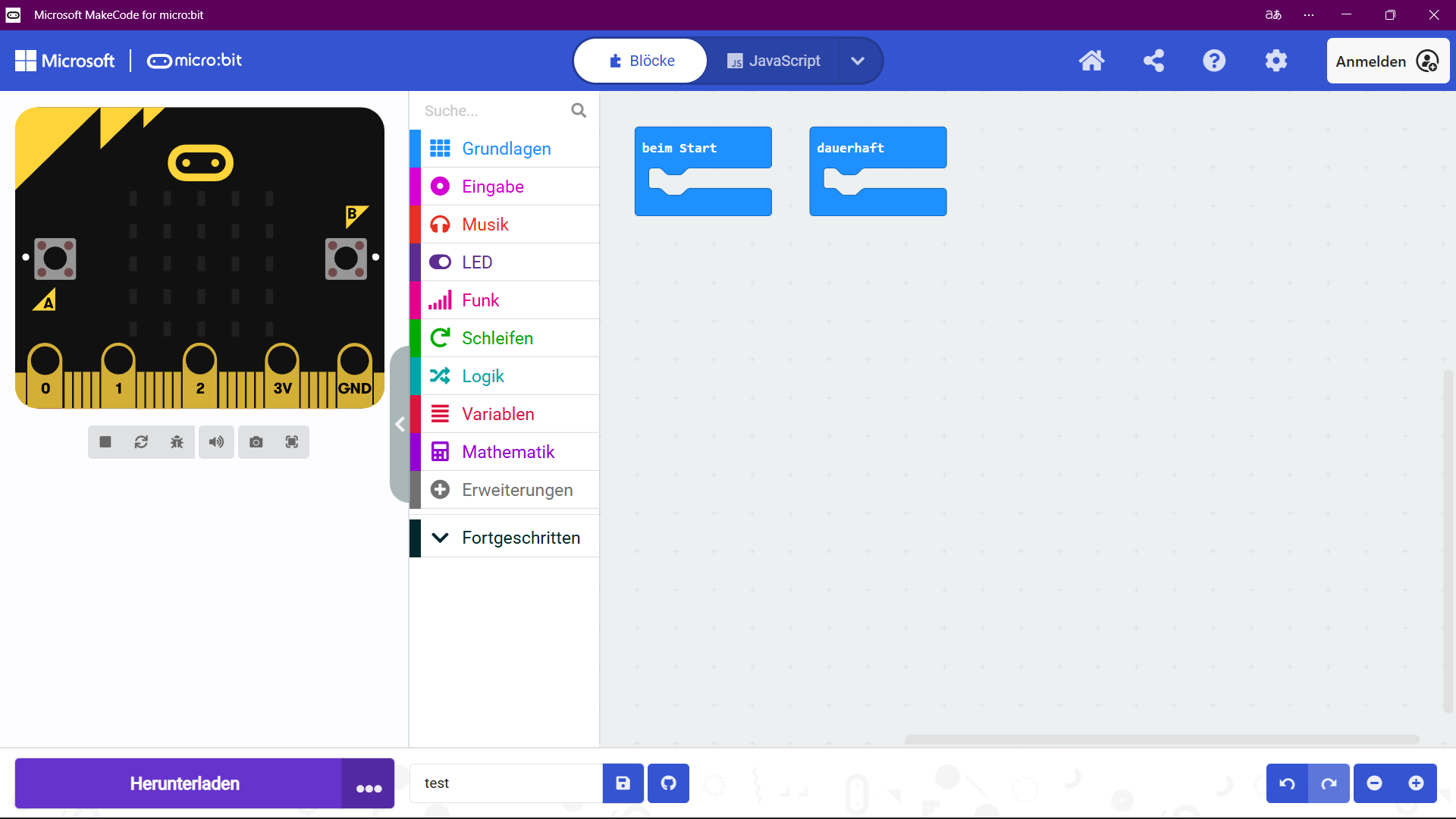
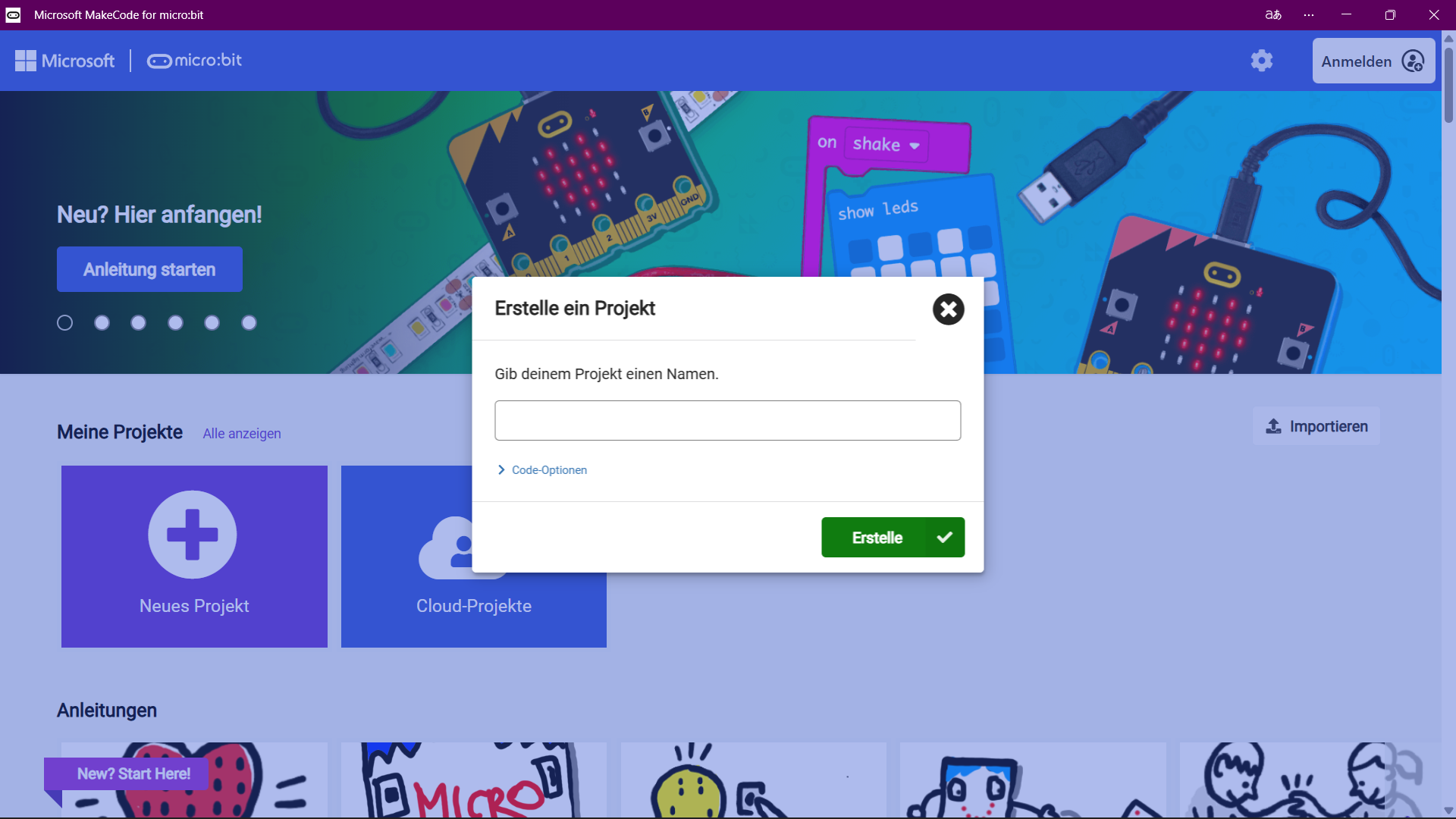
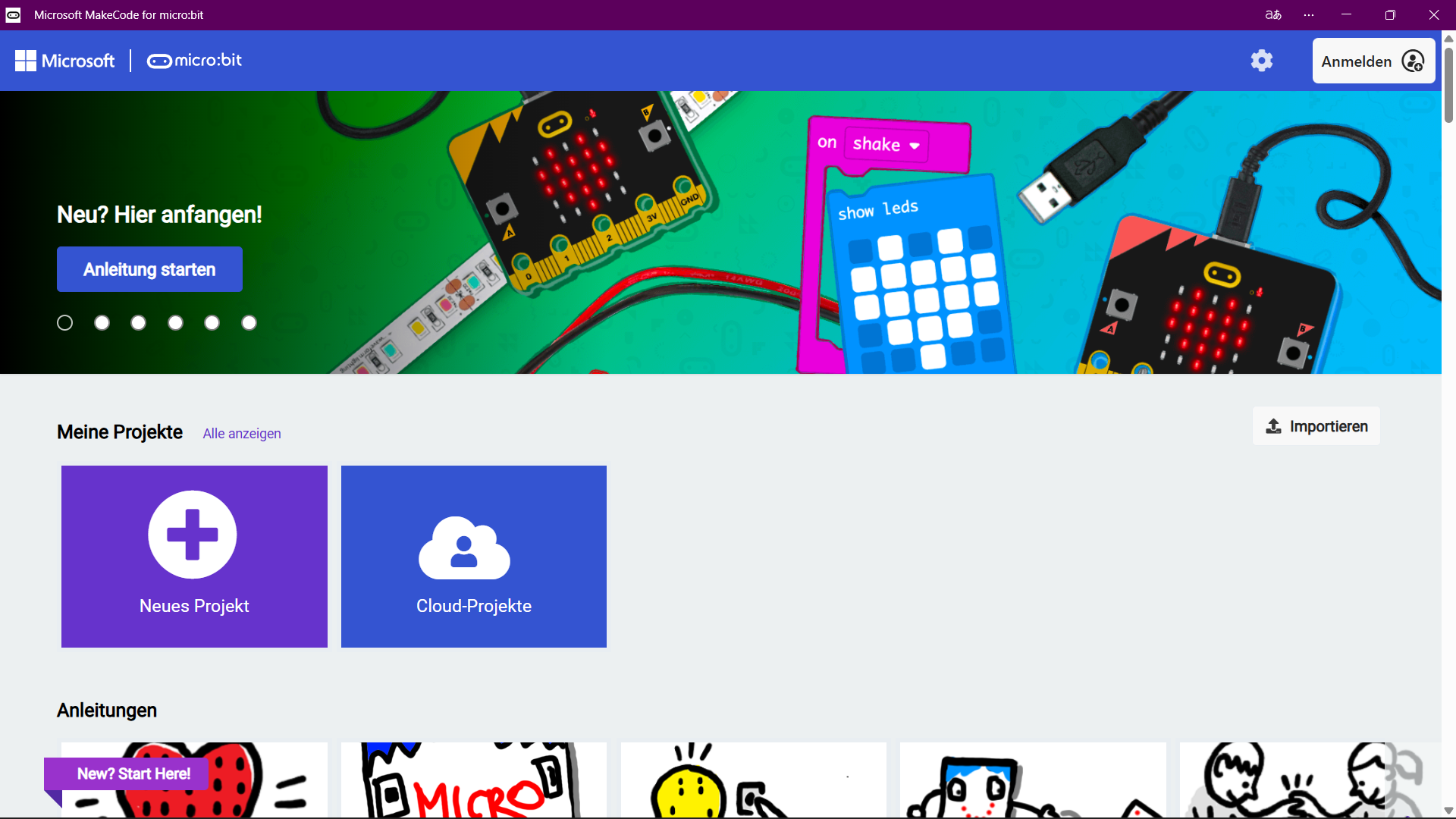
# Programmierung Micro-Bit

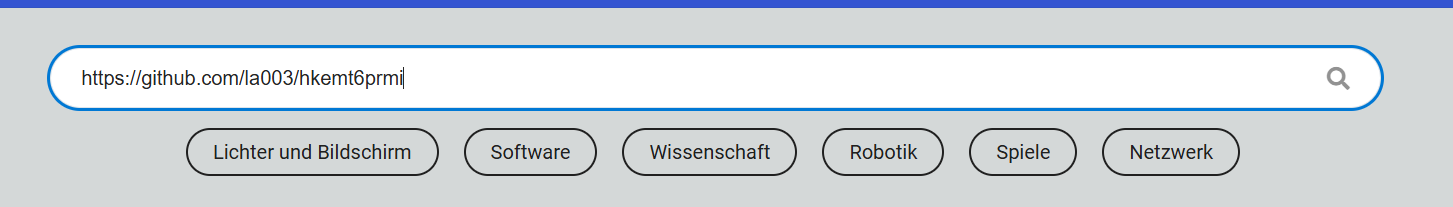
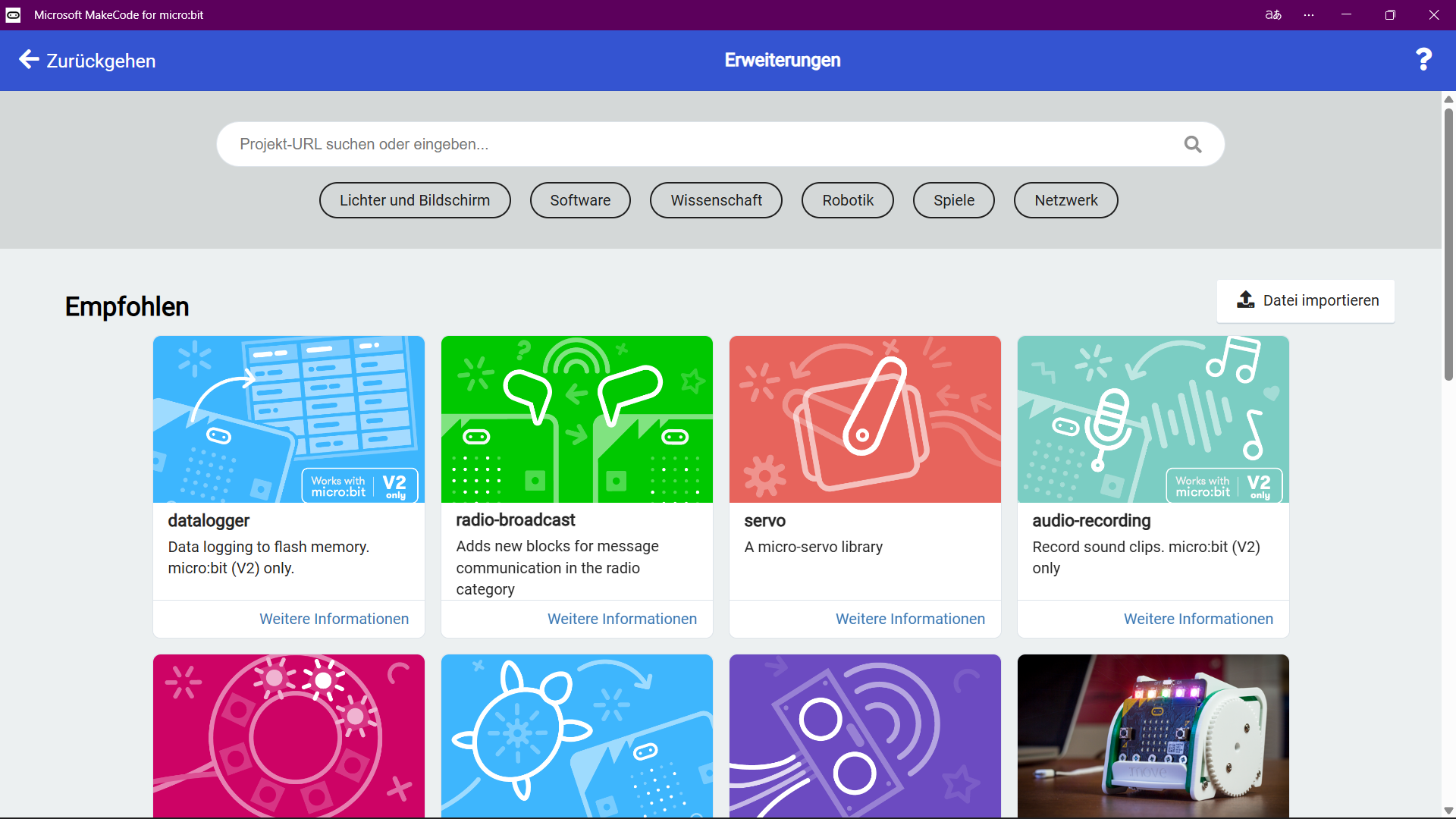
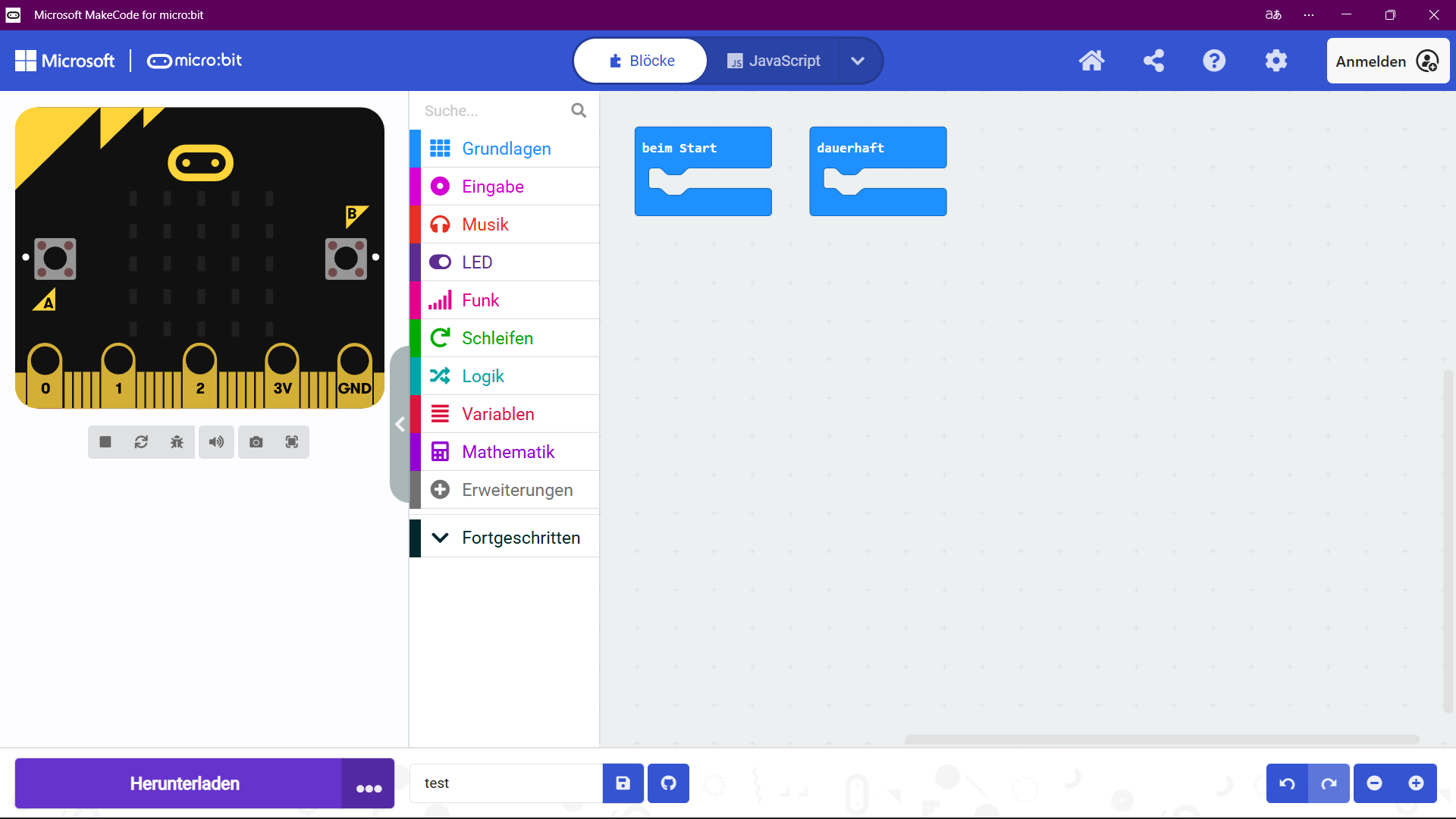
Für die Programmierung des Micro-Bit ist nicht viel notwendig. Ihr benötigt einen Laptop, den Micro-Bit und ein Micro USB Kabel sowie zu Beginn eine Internetverbindung. Entscheidet euch, ob ihr den Micro Bit über die Make Code App oder die Browser Anwendung programmieren wollt. Wir empfehlen die App, da die Bedienung einfacher ist, man mehrere Projekte parallel nutzen kann, die Umgebung offline verfügbar ist und die Umgebung für Kinder besser geeignet ist, als der Webbrowser. Die App kann man sich für Windows ganz einfach im Microsoft Store unter <https://apps.microsoft.com/detail/9NMQDQ2XZKWK?hl=neutral&gl=DE&ocid=pdpshare> herunterladen. Für Mac OS ist sie auch erhältlich. Die Browser Anwendung ist unter <https://makecode.com/playground> erreichbar.   
  
  
So sieht die Umgebung nach der Installation aus. Falls gewünscht kann man sich in der App anmelden um Fortschritte einfacher zu sichern.



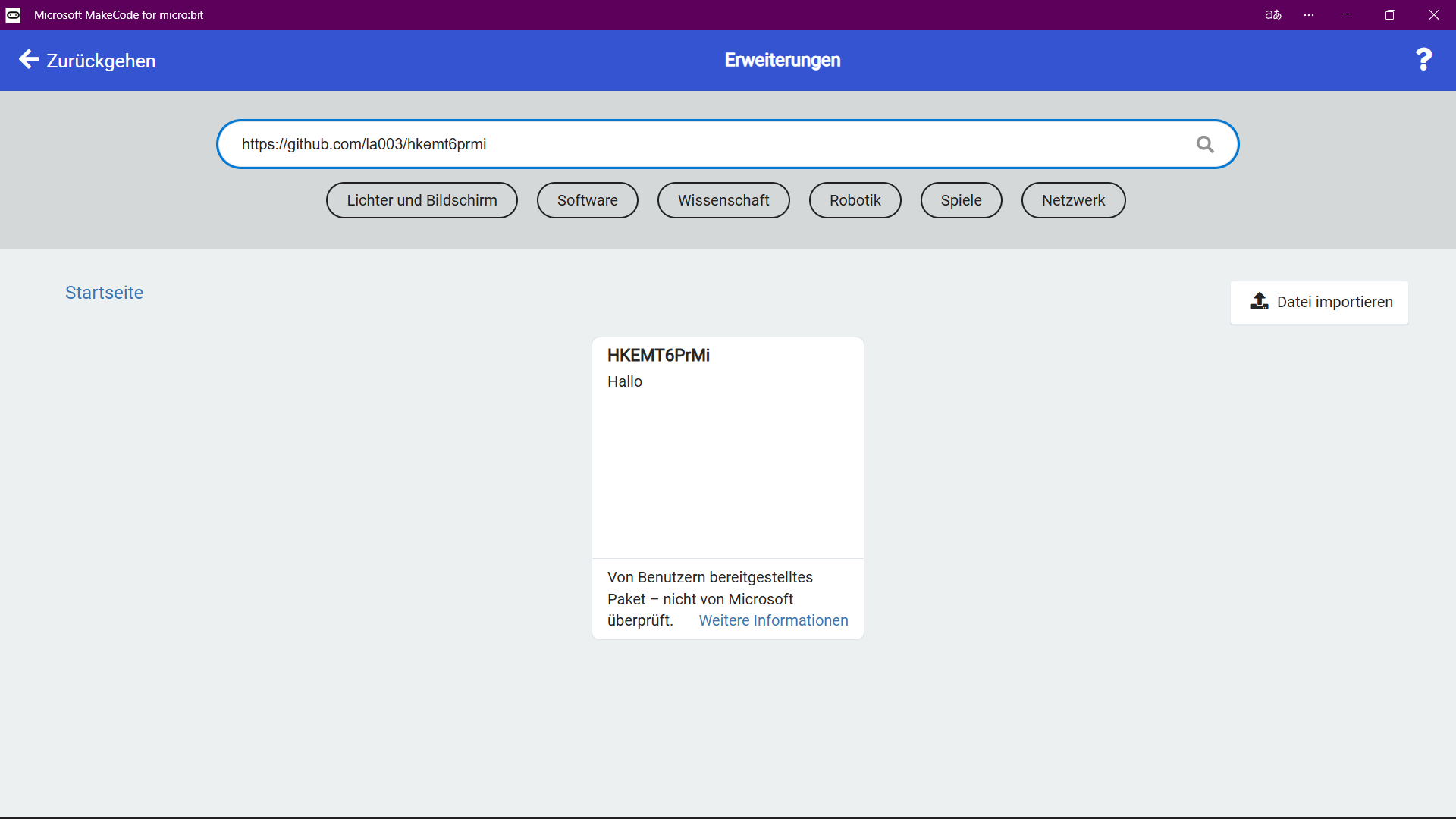
Nachdem man sich gegebenenfalls angemeldet hat oder wenn man so starten möchte, klickt man auf „Neues Projekt“   
  
Hier kann man einen beliebigen Namen vergeben und lässt das Projekt anschließend erstellen.  
Bei Erstnutzung wird eine kleine Einführung angeboten, mit der man die Umgebung kennenlernen kann.  
  
In der Mitte (1) sieht man die verschiedenen Bibliotheken für Blöcke, die man bei der Programmierung nutzen kann. Sollte man nicht grafisch sondern textbasiert programmieren wollten, kann man oben (2) zwischen der Blockumgebung, JavaScript und Python wechseln. Die Blöcke lassen sich einfach per Drag&Drop in die Übersicht ziehen und so zu einem Programm zusammensetzen. Um die Extension für die Anbauteile zu nutzen klickt man auf Erweiterungen. (3)



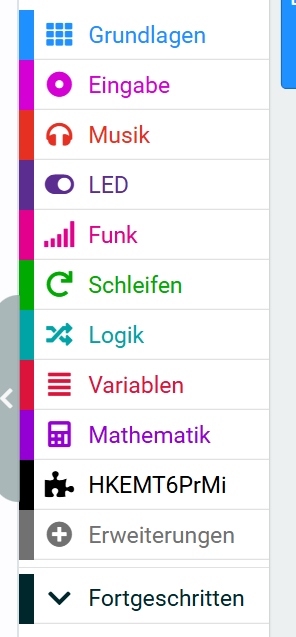
Anschließend gelangt man zu folgender Seite. Dort gibt man in die Suchzeile folgenden Link ein: <https://github.com/la003/hkemt6prmi> und klickt auf suchen.



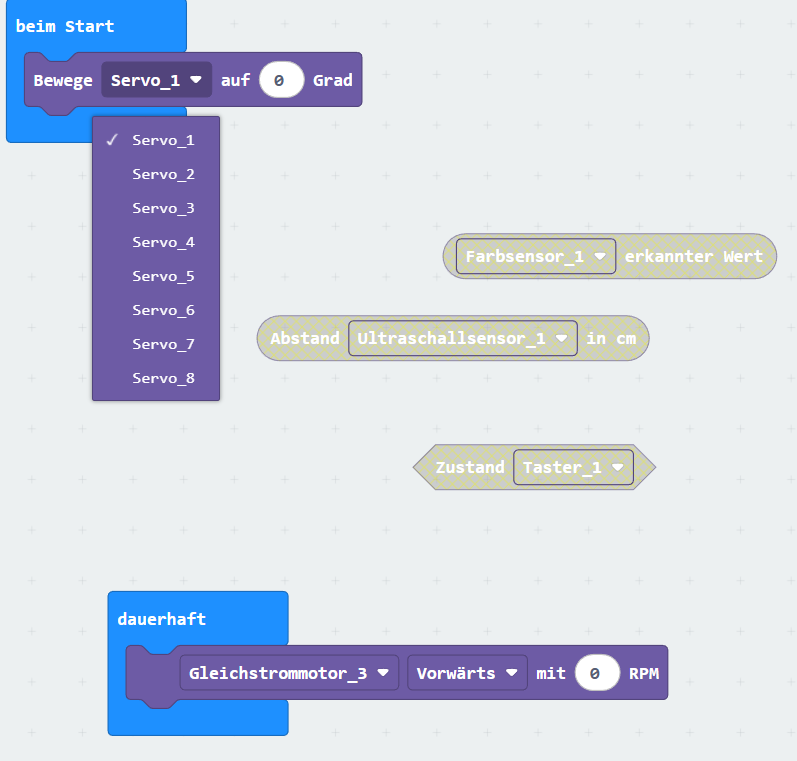
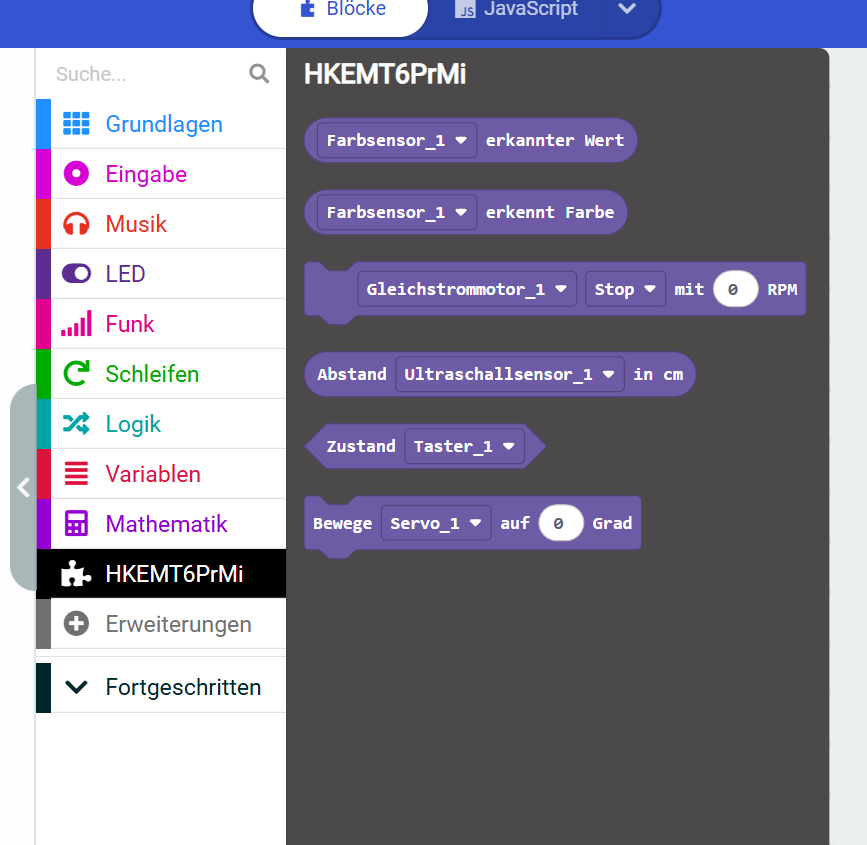
Als Suchergebnis sollte die Extension gefunden werden.



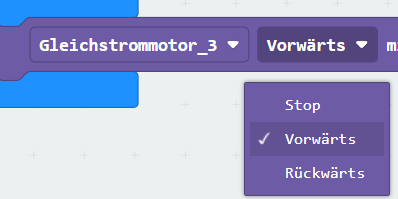
Diese klickt man an und anschließend wird sie automatisch in die Make Code Bibliothek importiert.



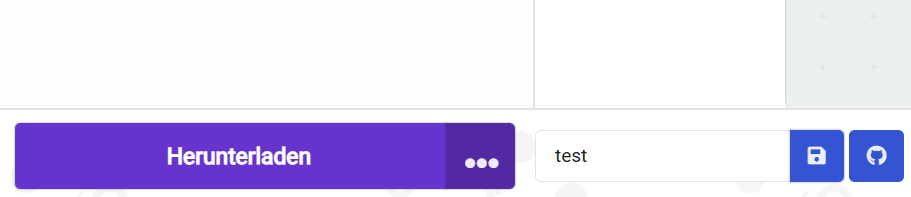
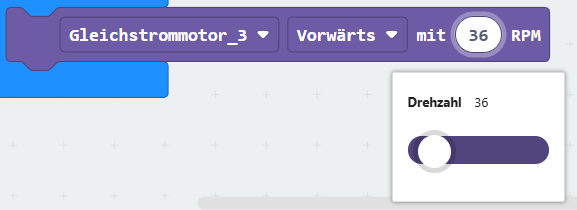
Dadurch sollte sie nun in der Übersicht mit auftauchen (4). Für die Programmierung klickt man die Bibliothek an und kann die Blöcke für die jeweiligen Anbauteile ganz normal mit verwenden.



In den einzelnen Blöcken gibt es immer Drop Down Menüs zur Auswahl von bestimmten Einstellungen. So kann man das entsprechende Gerät der Belegungstabelle folgend auswählen und die richtige Adresse ist automatisch hinterlegt (5). Auch andere Einstellungen sind abhängig vom Anbauteil möglich. Beispielsweise für den Gleichstrommotor lässt sich die Drehrichtung (6) sowie die Drehzahl (7) einstellen.



Sobald man mit der Programmierung fertig ist und das Programm auf den Micro Bit übertragen werden soll, steckt man diesen mit dem Kabel an den Laptop an. Dadurch, dass er automatisch erkannt wird, muss man keine weiteren Einstellungen durchführen.



Zum Übertragen des Programms muss man nur noch auf Herunterladen (8) klicken und der Micro:Bit wird mit dem erstellten Programm geflasht.   
  
  
Die Bilder sind nicht mehr ganz aktuell, in der aktuellen Funktion sieht es etwas anders aus. Die Funktionalität ist jedoch identisch.

Ein Bild, das Screenshot, Text, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

Funktion der verschiedenen Blöcke:

* Servomotor:
  + Servo und Gradzahl der Zielposition einstellbar
* Taster:
  + Ergebnis des Blocks ist Zustand von ausgewähltem Taster
* Ultraschallsensor:
  + Ergebnis des Blocks ist Abstand von ausgewähltem Ultraschallsensor
* Gleichstrommotor:
  + Gleichstrommotor und Bewegungsart einstellbar
* Helligkeitssensor:
  + Wert von 0 bis 255 abhängig von Helligkeit, 255 max. Helligkeit
* Farbsensor:
  + Ergebnis des Blocks ist erkannte Helligkeit
  + Ergebnis des Blocks ist erkannter Rotwert
  + Ergebnis des Blocks ist erkannter Grünwert
  + Ergebnis des Blocks ist erkannter Blauwert
  + Ergebnis des Blocks ist erkannter Abstand
  + Ergebnis des Blocks ist Zahl für Geste
    - 1=hoch, 2=runter, 3=links, 4=rechts
    - Wiederholung wahr bedeutet, dass gleiche Geste nacheinander erkannt werden darf -> bei Falsch und eingestelltem Timeout -> gleiche Geste für die Zeit gesperrt
  + Ergebnis des Blocks ist textuelle Ausgabe der Geste, analog zu dem Block vorher