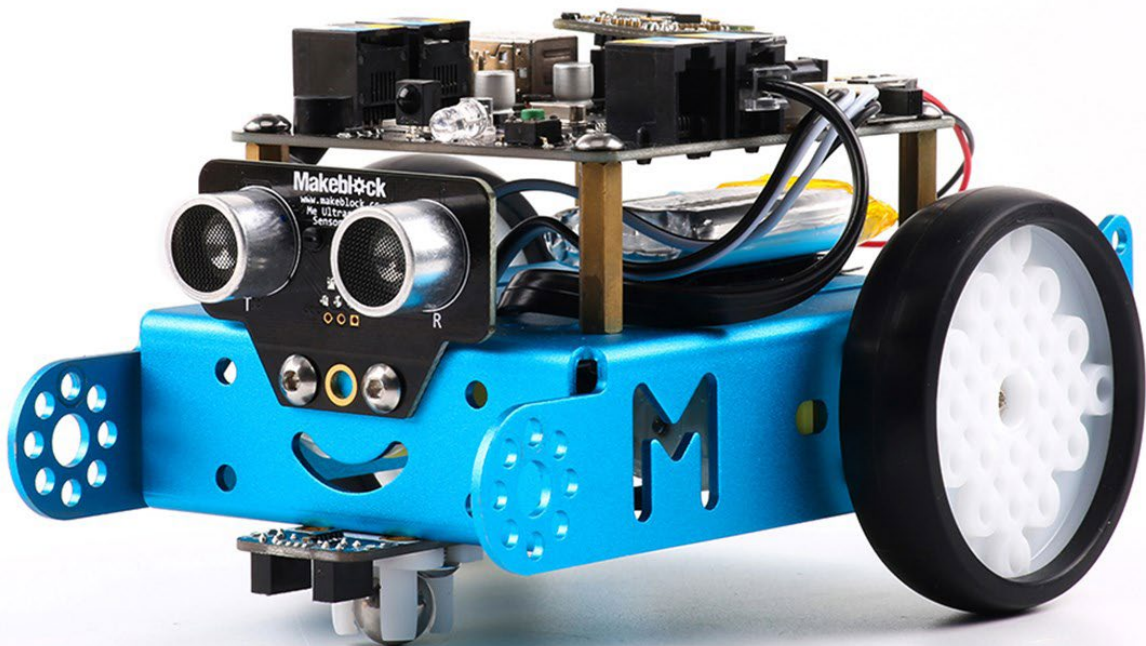


# mBlock & mBot

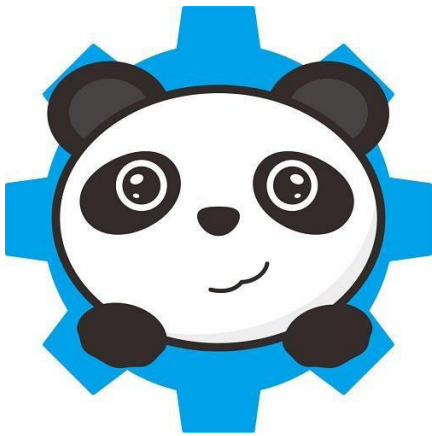
Roboter Programmierung  
auf der Basis von Scratch

Institut ICT & Medien  
[www.iim.phsg.ch](http://www.iim.phsg.ch)



# START

## mBot verbinden



➤ Icon der mBlock-Software

### mBlock mit mBot verbinden

Der mBot-Roboter kann mit der mBlock-Software programmiert werden. Dazu muss eine drahtlose Verbindung zwischen dem Roboter und dem Gerät auf welchem die mBlock-Software läuft hergestellt werden.

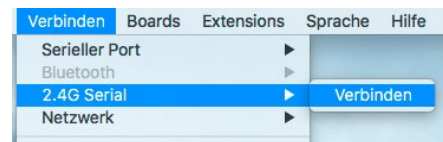
Am einfachsten geschieht dies mit dem «USB 2.4G Wireless Dongle». Jeder dieser Dongles baut eine 1:1 Verbindung zum entsprechenden mBot Roboter auf. So vermeiden wir, dass das Programm nicht versehentlich auf einem Roboter eines Kameraden landet.

Gehen sie für die Verbindung wie folgt vor:

- > Versichern sie, dass der entsprechende Wireless Dongle an der USB-Schnittstelle angeschlossen ist
- > Starten sie die «mBlock» Software
- > Schalten sie den mBot Roboter an
- > Wählen sie in der mBlock-Software den Menüpunkt Verbinden > 2.4G Serial > verbinden



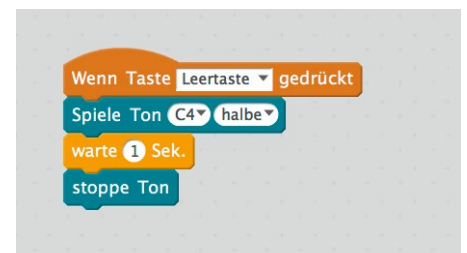
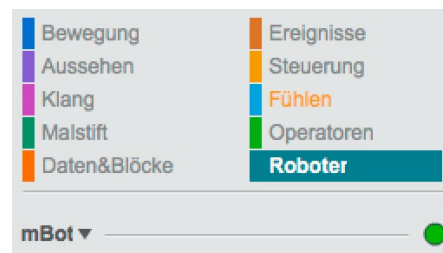
➤ Der USB Wireless Dongle verbindet sich mit dem Wirelessmodul auf dem Roboter



### Erster Test

Nun können wir mit einfachen Anweisungen die Verbindung testen. Zusätzlich zu den Standardbefehlsbereichen von Scratch (Bewegung, Aussehen, Klang etc.) kennt mBlock den Bereich «Roboter», wo die mBot spezifischen Anweisungen zu finden sind.

Mit den unten rechts abgebildeten mBlock Anweisungen veranlassen wir den mBot mit der Leertaste einen Ton abzuspielen.



# TASK 1

## LED programmieren

### Aufgabe 1-1

Der mBot verfügt selbst auch über eine Taste. Die Taste befindet sich auch der Vorderseite des mCore-Boards und ist mit Button angeschrieben.

Erstellen sie ein Programm welches folgende Anforderungen erfüllt:

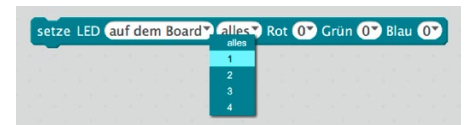
- > Wird auf der Tastatur die Leertaste gedrückt wird das Programm gestartet. Um diesen Schritt zu bestätigen spielt der mBot wie zuvor einen Ton während 1 Sekunde ab.
- > Danach wartet der mBot bis die Taste auf dem mBot selbst (grüner Button) gedrückt wird.
- > Nach dem Drücken der Taste leuchten die beiden LEDs zuerst 1 Sekunde Grün und danach 1 Sekunde Magenta (Mischung Rot und Blau) auf .
- > Zum Schluss wird wieder während einer Sekunde ein Ton abgespielt.

### Aufgabe 1-2

Die beiden LEDs des mBots können auch einzeln angesteuert werden. Bei dieser Aufgabe wollen wir, das abwechselnd das linke und rechte LED aufleuchtet.

Erstellen sie ein Programm welches folgende Anforderungen erfüllt:

- > Wird der grüne Button auf dem mBot betätigt so blinken die LEDs abwechselnd links rechts grün auf.
- > Dabei dauert ein Aufblinken eine halbe Sekunde (halbe Sekunde Links, halbe Sekunde Rechts, halbe Sekunde Links etc.)
- > Dies macht der mBot 5 Mal. Nutzen Sie dazu eine Schleife.
- > Zum Schluss spielt der mBot wieder einen Ton ab.



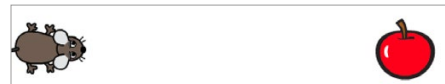
- Die LEDs des mBots können einzeln angesteuert werden. Position 1 und 2 sind die LEDs auf dem Board, 3 und 4 sind noch nicht beglegt.

# TASK 2

## mBot steuert Maus

### mBot Button als Trigger

Eine wesentliche Eigenschaft der Robotik ist, dass auf hardwareseitige Einflüsse (meistens durch Sensoren des Roboters) softwareseitig reagiert werden kann. Bei dieser Aufgabe geht es um ein normales Scratch Programm, welches durch den Button auf dem mBot initiiert wird.



Ausgangslage des Programms



Szenario nach Betätigung des mBot Buttons

Erstellen sie ein Programm welches folgende Anforderungen erfüllt:

- > Erstellen sie 2 Figuren (Sprites): Eine Maus und ein Apfel. Beide finden sie in der Bibliothek. Wie sie es von Scratch wahrscheinlich kennen, können sie jede Figur mit eigenen Scratchanweisungen versehen.
- > Beide Objekte liegen wie abgebildet auf der gegenüberliegenden Seite der Bühne
- > Mit dem Betätigen des Buttons auf dem mBot gleitet die Maus zum Apfel.

#### Anweisungen für die Maus

- > Wird die Leertaste betätigt verschiebt sich die Maus an ihre Anfangsposition  $X=-187$   $Y=-5$ .
- > Danach wird für eine halbe Sekunde ein Ton abgespielt.
- > Wird nun der grüne Button auf dem mBot betätigt, so gleitet die Maus an die Position  $X=150$   $Y=-5$ .

#### Anweisungen für den Apfel

- > Der Anweisungsblock wird auch mit der Leertaste betätigt
- > Nun wartet das Programm bis sich die Maus und der Apfel berühren
- > Danach werden je für eine halbe Sekunde zwei unterschiedliche Töne abgespielt

Bewegung	Ereignisse
Aussehen	Steuerung
Klang	<b>Fühlen</b>
Malstift	Operatoren
Daten&Blöcke	Roboter

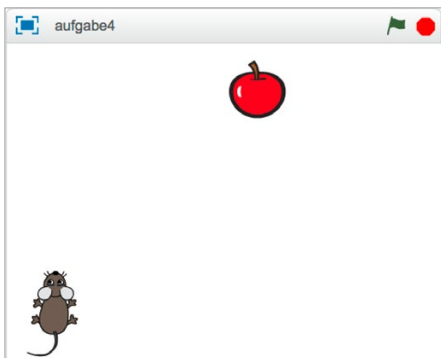
➔ Anweisungen zu Kollisionen / Berührungen finden sie im Anweisungsbereich «Fühlen»

# TASK 3

## Ultraschallsensor

### Spielsteuerung mit der Hand

Nun entwickeln wir ein kleines Spiel! Bei dieser Aufgabe lassen wir Äpfel an einer beliebigen X-Position in der Y-Achse fallen. Das Ziel ist es, dass die Maus die Äpfel auffängt, indem er sich in der X-Achse hin und her bewegt. Der Spieler «steuert» die Maus jedoch nicht mit den Pfeiltasten, sondern mit dem Distanzsensoren (Ultraschallsensoren) des mBots.



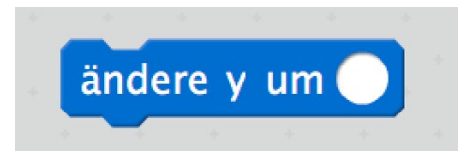
*Die Äpfel fallen an beliebiger Stelle nacheinander hinunter. Der «Spieler» steuert die X-Position der Maus mit dem Distanzsensoren des mBots.*

#### Anweisungen für die Figur Apfel

- > Mit einem Klick auf die grüne Flagge wird das Programm gestartet.
- > An einer beliebigen Stelle in der X-Achse (Bereich -200 bis +200) «fällt» ein Apfel herunter bis er den unteren Rand berührt.
- > Danach fällt wiederum ein Apfel an einer anderen Stelle herunter. Diese Prozedur wiederholt sich fortlaufend.

#### Anweisungen für die Figur Maus

- > Mit einem Klick auf die grüne Flagge wird das Programm gestartet
- > Die X-Position der Maus wird durch den Ultraschallsensoren (Distanz Hand – mBot) bestimmt. Das Ermitteln des Distanzwertes und die Neupositionierung der Maus erfolgt dabei fortlaufend. So kann mit der Hand die Maus gesteuert werden. Nutzen Sie dabei ein Distanzspektrum zwischen 5cm und 25cm (5cm = Maus ganz links, 25cm = Maus ganz rechts)
- > Berührt die Maus den Apfel, so wird für eine halbe Sekunde ein Ton abgespielt.
- > Nach fünf solchen Berührungen endet das Spiel.



- Das Herunterfallen der Äpfel erreichen wir durch die Bewegungsanweisung «Ändere Y um ...»

- **TIPP 1:** Evt. sind zwei Unterprogramme für die Maus notwendig um die Ziele zu erreichen. mBlock ist in der Lage mehrere Programme (Blöcke) gleichzeitig zu behandeln.

- **TIPP 2:** Der Ultraschallsensoren gibt einen Wert in cm aus, die Bühne ist jedoch in Pixeln bemessen. Der Distanzwert muss also entsprechend umgerechnet werden

# TASK 4

## Fernsteuerung

### mBot fährt nach Tastenbefehlen

Jetzt lassen wir den mBot fahren. Die Anweisungen sind simpel.

Erstellen sie ein Programm welche folgendes Anforderungen erfüllt:

- > Mit einem Klick auf die grüne Flagge wird das Program initiiert und mBot gibt einen Ton aus.
- > Wird die Pfeil-Oben Taste gedrückt, so fährt der Roboter mit der Geschwindigkeit 100 geradeaus.
- > Wird die Pfeil-Oben Taste losgelassen, so stoppt der Roboter seine Fahrt.
- > Wird die Pfeil-Rechts Taste gedrückt, so dreht sich der Roboter (Geschwindigkeit 100) nach rechts bis die Taste losgelassen wird. Das selbe geschieht mit der Pfeil-Links Taste.
- > Zusätzlich wird während dem die T-Taste gedrückt wird, der Turbomodus aktiviert und sämtliche Geschwindigkeiten werden auf 255 erhöht. Wird die T-Taste wieder losgelassen, so gilt wieder Tempo 100.
- > Speichern sie die aktuell geltende Geschwindigkeit in einer Variable und greifen sie bei den Tastenereignissen auf diese zu.



➤ Variablen werden im Anweisungsbereich «Daten & Blöcke» definiert.

### Zusatz

- > Wie sie wahrscheinlich bemerkt haben funktionieren die Tastenbefehle nur in sequentieller Abfolge (fahren-drehen-fahren) – es ist beispielsweise nicht möglich zu fahren und gleichzeitig abzdrehen. Entwickeln sie diese Funktion. Ist diese überhaupt möglich?

# TASK 5

## Selbstfahrer

### Hindernisse vermeiden

In der vorherigen Aufgabe wurde mBot ferngesteuert. Jetzt nimmt mBot aufgrund realer Gegebenheiten selbst Einfluss auf die Steuerung.

Erstellen sie ein Programm welches folgende Anforderungen erfüllt:

- > Mit einem Klick auf die Leertaste wird das Program initiiert.
- > Laufend wird der Wert des Ultraschallsensors in die Variable «distanz» geschrieben.
- > Wird die Leertaste gedrückt, so fährt mBot mit Tempo 100 gerade aus. Wird die Leertaste losgelassen stoppt mBot.
- > Fährt mBot auf ein Hindernis zu (z.B. Distanz < 10cm) so dreht er links ab.

### Zusatz 1

- > Vor dem Abdrehen fährt der mBot jeweils eine Sekunde rückwärts, so werden Hindernisse besser umfahren.
- > Der mBot dreht zufällig nach rechts oder links ab. Setzen Sie dazu eine neue Variable ein, welche die Richtung beinhaltet.

### Zusatz 2

- > mBot erhält einen visuellen Hindernisassistenten: Bei einer Distanz unter 50cm zu einem Hindernis beginnen die LEDs rot aufzuleuchten.
- > Je näher mBot an das Hindernis heranfährt, desto heller leuchten die LEDs.

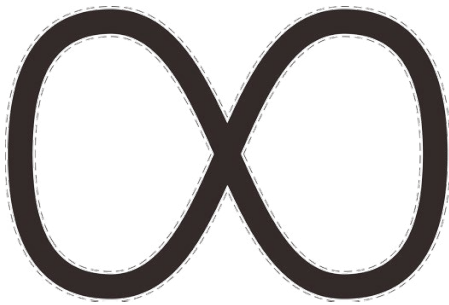
➤ **HINWEIS:** Damit mBot eine Fahrt aufnehmen kann, benötigt er aufgrund der Haftreibung eine Geschwindigkeit (Leistung) von 60 - 100. Er wird wohl bis auf 40 abbremsen können, benötigt jedoch einen Wert von 100 um wieder anzufahren.

# TASK 6

## Line follower

### mBot folgt einer Linie

Bei diesem Task wird mBot einer breiten schwarzen Linie folgen. «Droht» ein Verlassen der Linie, so korrigiert mBot den Kurs selbst.



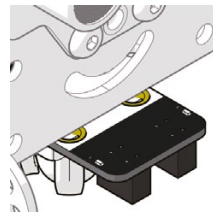
➤ Mögliche Testroute in A2 Format

Erstellen sie ein Programm welches folgende Anforderungen erfüllt:

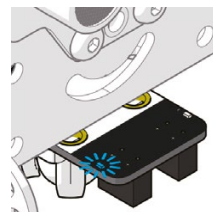
- > Wird die Leertaste gedrückt, so fährt mBot. Wird die Leertaste losgelassen stoppt mBot.
- > Befindet sich mBot ganz in einem weissen Bereich, so fährt mBot rückwärts bis er wieder auf eine schwarze Linie trifft.
- > Nutzen sie den Linie-Follower-Sensor um mBot einer Linie entlang fahren zu lassen. Der Line-Follower-Sensor (genau genommen sind es 2 Sensoren) gibt die Werte 0, 1, 2, 3 zurück.

➤ **TIPP:** Entwickeln sie zuerst eine Strategie wie die Problemstellung gelöst werden kann.

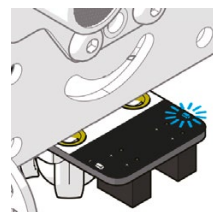
### Rückgabewerte des Line-Follower-Sensors:



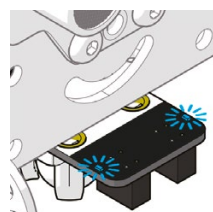
- Beide Sensoren befinden sich über einem schwarzen Untergrund  
**Rückgabewert: 0**



- Nur der linke Sensor befindet sich über einem schwarzen Untergrund  
**Rückgabewert: 1**



- Nur der rechte Sensor befindet sich über einem schwarzen Untergrund  
**Rückgabewert: 2**



- Keine der Sensoren befinden sich über einem schwarzen Untergrund  
**Rückgabewert: 3**