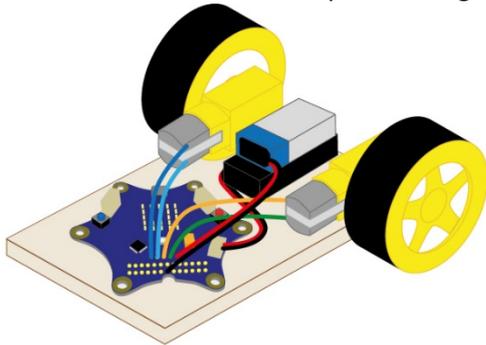


## Mit einem zweiten Controller das Calliope-Mobil fernsteuern

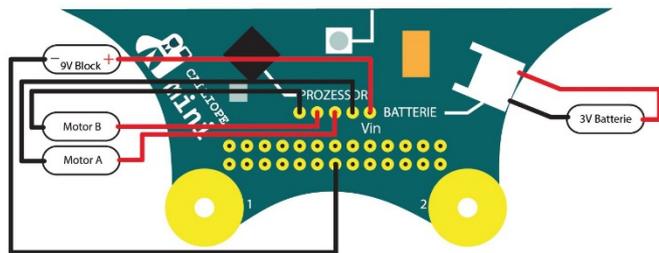
Da Calliope minis über Bluetooth miteinander kommunizieren können, kann ein Controller die Motorsteuerung übernehmen und ein zweiter die Steuersignale übermitteln.

### Aufbau

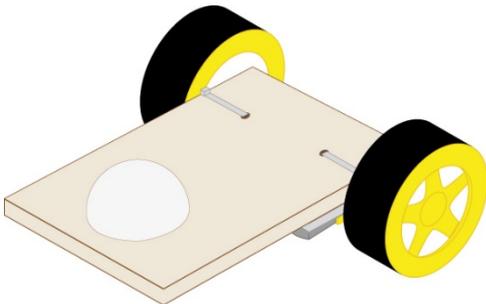
Die Abbildungen links zeigt das Fahrzeug von der Ober- und Unterseite. Die Motoren sind mit Kabelbindern auf einem 9 x 14 cm großen Sperrholzbrettchen befestigt und mit Heißkleber gegen seitliches Verdrehen gesichert. Ein halber Tischtennisball hält das Fahrzeug in der Waagerechten. Die Detailansicht des Calliope mini zeigt, wie Motoren und Batterien anzuschließen sind.



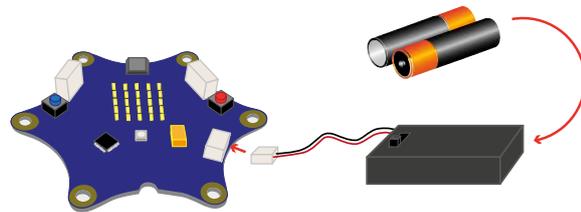
Oberseite des Fahrzeugs



Anschlussplan (Pluspol rot, Minuspol schwarz)



Unterseite



Calliope mini, der als Fernbedienung dienen soll

Um die Verbindungen zu den Motoren und dem 9V Block jederzeit lösen zu können, sind in die auf den Calliope vorhandenen Ösen Stifte gelötet worden, zu denen es passende Jumperkabel mit Buchsen gibt. Diese Kabel werden mit einem Ende an die Motoren gelötet und über die Buchsen am anderen mit den Kontakten unterhalb des Aufdrucks „Prozessor“ verbunden. An dem verbleibenden Stift wird der Pluspol des 9V Blocks angeschlossen. Der Minuspol des 9V Blocks wird mit einem der mittleren Kontakte in den darunterliegenden Reihen verbunden.

Für die 3V Batterie gibt es am rechten Rand des Calliope mini eine genormte Steckverbindung, die so beschaffen ist, dass Plus- und Minuspol nicht vertauscht werden können.

### Übernahme des Programmcodes

Um die Übernahme einfach zu machen, werden die einzelnen Programme in Java-Script und als Blockcode vorgehalten. Java-Script kann aus einem Textdokument über Copy & Paste auf den Makecode-Editor übertragen und so in einfacher Weise zur Programmierung der Mikrocontroller genutzt werden. Aus einem PDF-Dokument ist das nicht ohne weiteres möglich. Das Text-Dokument kann bei [g.stein@mint-unt.de](mailto:g.stein@mint-unt.de) angefordert werden.

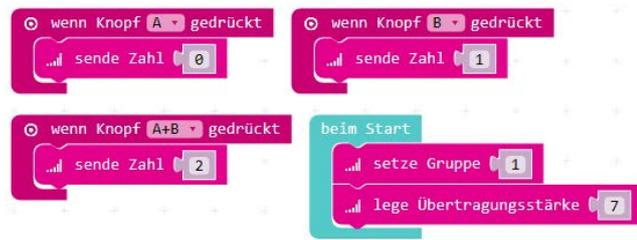
### Programmierung

Lösung I:

Wenn zwei Calliope minis mit Sendefunktionen der gleichen Gruppe angehören, können sie miteinander kommunizieren und so Steuerimpulse übertragen werden. Im einfachsten Fall sind das

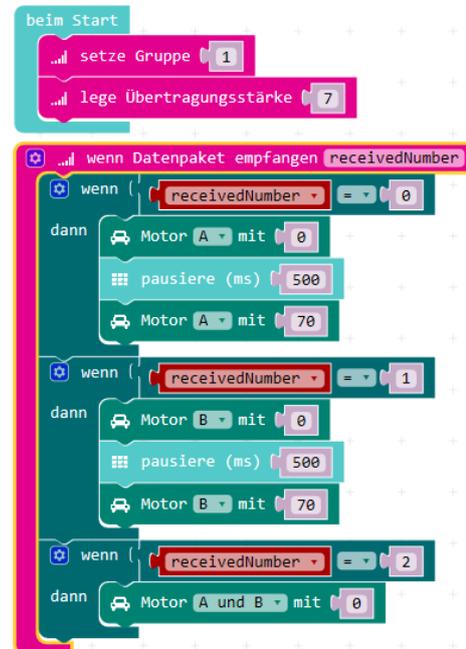
Zahlen, die auf Knopfdruck übermittelt werden: Knopf „A“ beispielsweise eine „0“, Knopf „B“ eine „1“ und beide Knöpfe („A+B“) eine „2“.

```
input.onButtonPressed(Button.A, () => {
  radio.sendNumber(0)
})
input.onButtonPressed(Button.B, () => {
  radio.sendNumber(1)
})
input.onButtonPressed(Button.AB, () => {
  radio.sendNumber(2)
})
radio.setGroup(1)
radio.setTransmitPower(7)
```



Dem Calliope, der als „Empfänger“ dient, muss dann „gesagt“ werden, was er machen soll, wenn diese Zahlen bei ihm eintreffen, hier also die Motoren starten und stoppen.

```
radio.onDataPacketReceived( ({ receivedNumber }) => {
  if (receivedNumber == 0) {
    motors.dualMotorPower(Motor.A, 0)
    basic.pause(500)
    motors.dualMotorPower(Motor.A, 70)
  }
  if (receivedNumber == 1) {
    motors.dualMotorPower(Motor.B, 0)
    basic.pause(500)
    motors.dualMotorPower(Motor.B, 70)
  }
  if (receivedNumber == 2) {
    motors.dualMotorPower(Motor.AB, 0)
  }
})
radio.setGroup(1)
radio.setTransmitPower(7)
```



Der Algorithmus ist folgender: Mit Knopf „A“ wird der eine und mit Knopf „B“ der andere Motor gestartet. Ein erneutes Betätigen der Knöpfe bringt den jeweiligen Motor kurzzeitig zum Stillstand. Werden beide Knöpfe gleichzeitig gedrückt, stoppen die Motoren. So kann das Fahrzeug gestartet, nach links und rechts gesteuert und angehalten werden.

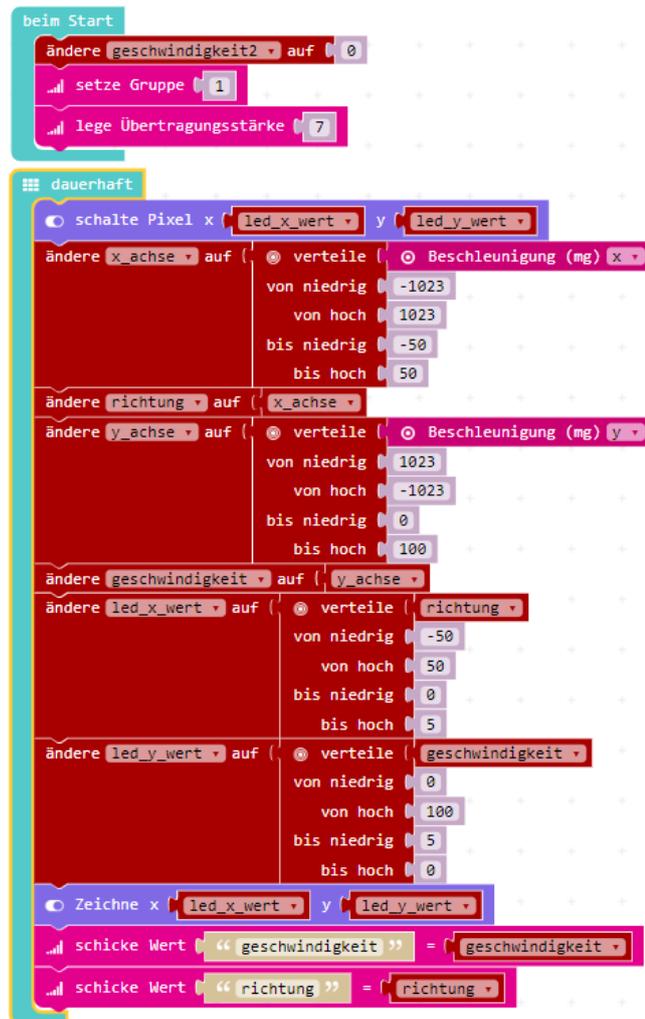
#### Lösung II:

Jeder Calliope mini verfügt über einen Bewegungssensor, über den der Neigungsgrad der Platine ermittelt werden kann. Entsprechend programmiert kann so mit einer Drehung um die x-Achse (Calliope nach links oder rechts kippen) die Lenkung, mit einer Neigung um die y-Achse (Calliope nach vorn oder hinten kippen) die Geschwindigkeit beeinflusst werden. Um das Fahrzeug zu beschleunigen, muss der Mini nach vorn, um es abzubremesen nach hinten geneigt werden. Wird er nach links oder rechts gedreht, reagiert das Fahrzeug entsprechend. Hier die Programmierung des Senders:

```

let geschwindigkeit = 0
let y_achse = 0
let x_achse = 0
let led_y_wert = 0
let led_x_wert = 0
let richtung = 0
let geschwindigkeit = 0
radio.setGroup(1)
radio.setTransmitPower(7)
basic.forever(() => {
  led.unplot(led_x_wert, led_y_wert)
  x_achse = pins.map(
    input.acceleration(Dimension.X),
    -1023,
    1023,
    -50,
    50
  )
  richtung = x_achse
  y_achse = pins.map(
    input.acceleration(Dimension.Y),
    1023,
    -1023,
    0,
    100
  )
  geschwindigkeit = y_achse
  led_x_wert = pins.map(
    richtung,
    -50,
    50,
    0,
    5
  )
  led_y_wert = pins.map(
    geschwindigkeit,
    0,
    100,
    5,
    0
  )
  led.plot(led_x_wert, led_y_wert)
  radio.sendValue("geschwindigkeit", geschwindigkeit)
  radio.sendValue("richtung", richtung)
})

```



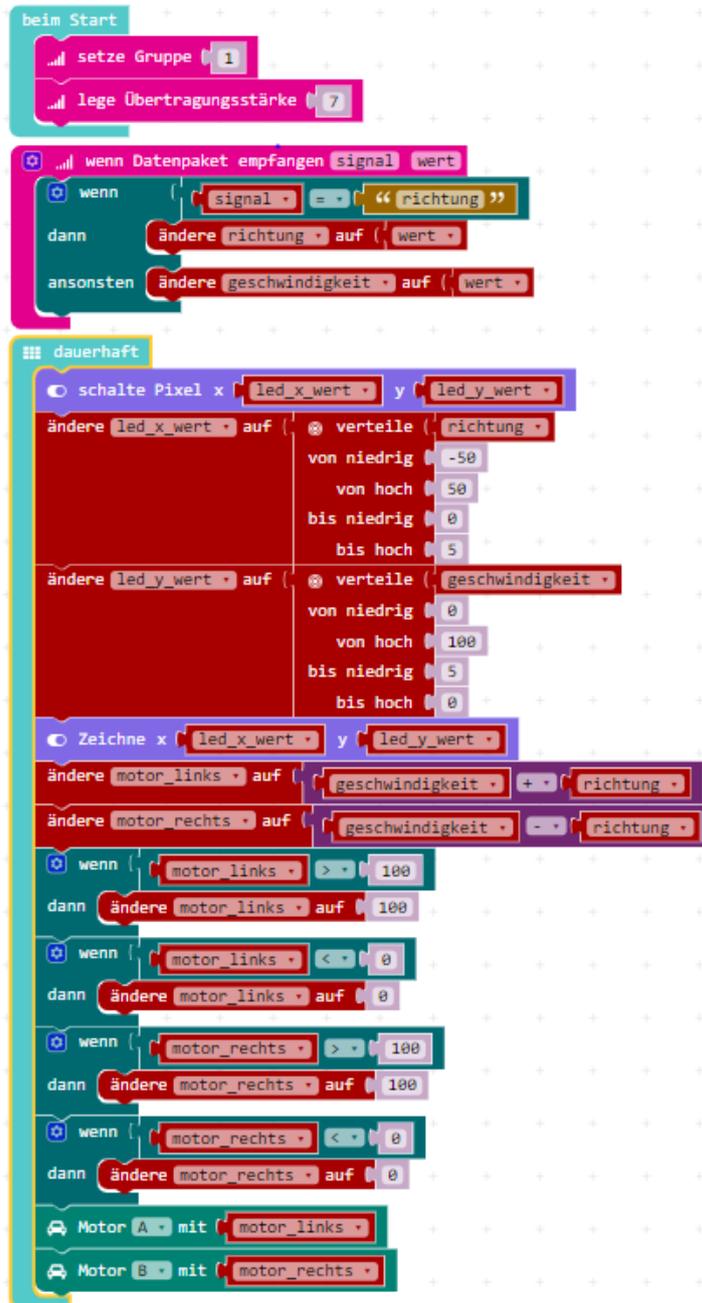
Ist der Code auf den Calliope mini übertragen, zeigt sich die Funktionstüchtigkeit des Programms, wenn beim Kippen des Boards eine einzelne LED an unterschiedlichen Stellen auf der Matrix aufleuchtet und so das Licht der Bewegung des Boards folgt.

Die beiden Calliope mini verbinden sich selbsttätig, sodass Steuerbefehle von der Fernbedienung direkt zum Controller auf dem Fahrzeug gelangen, der nach diesen Vorgaben die Geschwindigkeiten der Motoren regelt. Das Lauflicht ist auch auf dem zweiten Calliope realisiert. Hinzu kommt dort die Steuerung der Motoren:

```

let motor_rechts = 0
let motor_links = 0
let led_y_wert = 0
let led_x_wert = 0
let geschwindigkeit = 0
let richtung = 0
radio.onDataPacketReceived( ({
  receivedString: signal,
  receivedNumber: wert }) => {
  if (signal == "richtung") {
    richtung = wert
  } else {
    geschwindigkeit = wert
  }
})
radio.setGroup(1)
radio.setTransmitPower(7)
basic.forever(() => {
  led.unplot(led_x_wert,
  led_y_wert)
  led_x_wert = pins.map(
  richtung,
  -50,
  50,
  0,
  5
  )
  led_y_wert = pins.map(
  geschwindigkeit,
  0,
  100,
  5,
  0
  )
  led.plot(led_x_wert,
  led_y_wert)
  motor_links = geschwindigkeit +
  richtung
  motor_rechts = geschwindigkeit
  - richtung
  if (motor_links > 100) {
    motor_links = 100
  }
  if (motor_links < 0) {
    motor_links = 0
  }
  if (motor_rechts > 100) {
    motor_rechts = 100
  }
  if (motor_rechts < 0) {
    motor_rechts = 0
  }
  motors.dualMotorPower(Motor.A, motor_links)
  motors.dualMotorPower(Motor.B, motor_rechts)
})

```



Wenn alles geklappt hat, wird zunächst der Calliope mini auf dem Fahrzeug eingeschaltet und danach der als Fernbedienung fungierende. Nach einem kurzen Moment sollte sich das Fahrzeug in Bewegung setzen und auf Lageveränderungen der Fernbedienung reagieren.