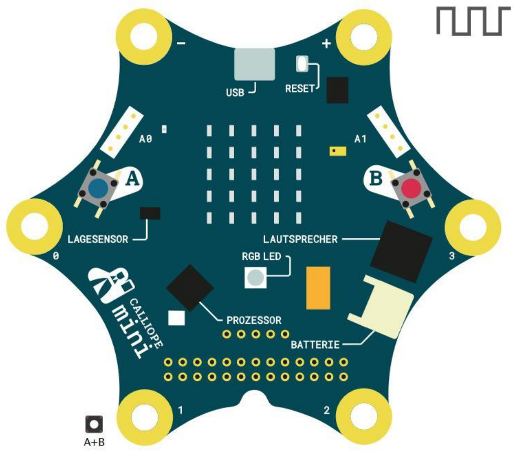
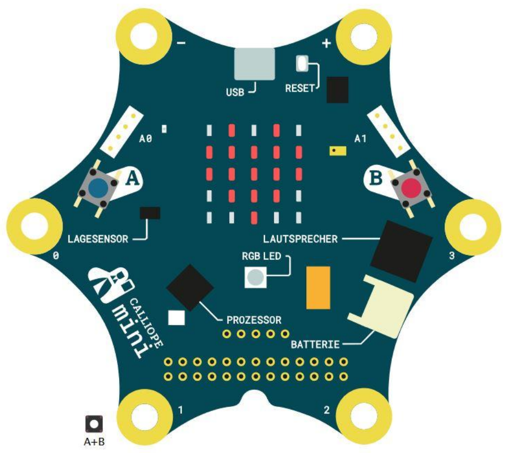
Translated from the German original into English by CALLIOPE

4.1 Radio: Send hearts

APP CAMPS





Tip

As soon as you send something in the preview, a second Calliope mini appears.

- **Test** (in the preview)
- **Download**
- **Transfer**
- Press **reset button**

Start

- Type **makecode.calliope.cc** into a browser.
- Start a new project and call it *radio*.

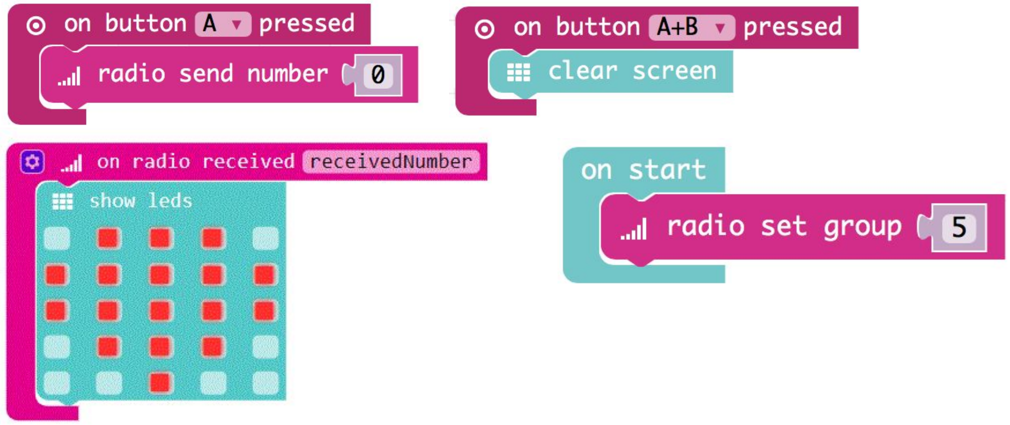
Task

Two Calliope devices should send data back and forth.

1. Send: **If button A is pressed**, a number should be sent. To do this, use **send number** from **radio**. Radio
2. Receiving: Display a heart when you receive something. For this you need

   on radio received receivedNumber

3. Delete the heart (use **clear screen**) when **button A+B is pressed**.

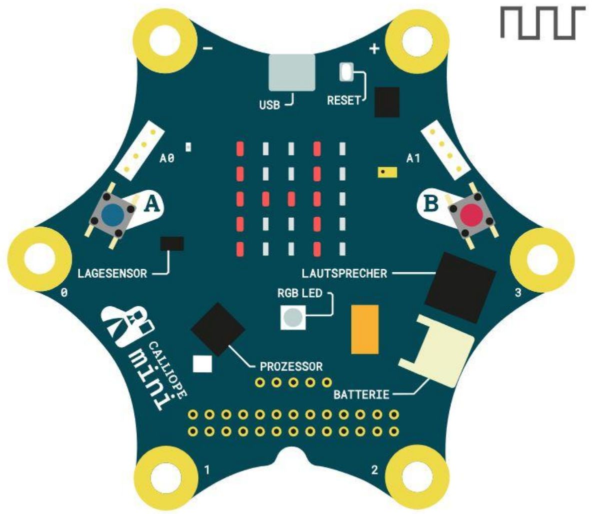


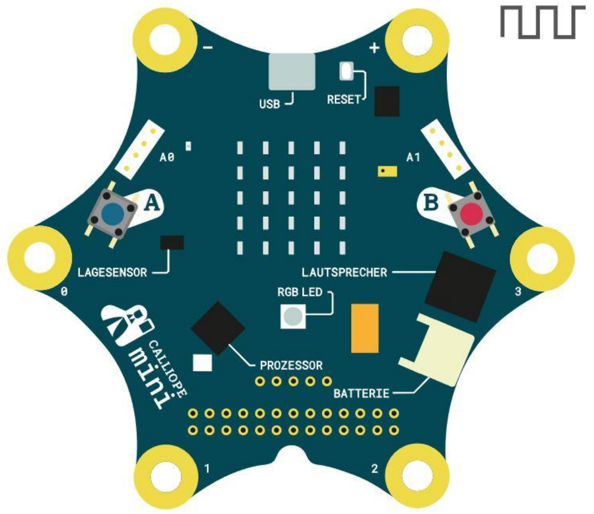
```
on button A pressed
    radio send number 0

on button A+B pressed
    clear screen

on radio received receivedNumber
    show leds
        . # # # .
        # # # # #
        # # # # #
        . # # # .
        . . # . .

on start
    radio set group 5
```

Try it out!

Find another group that has already worked on this card and try your program.
Note: In order for your Calliope to be able to communicate with each other, you must agree on a group number. You can find the block **radio set group** at **radio** at ••• More

4.2 Radio: Receive messages





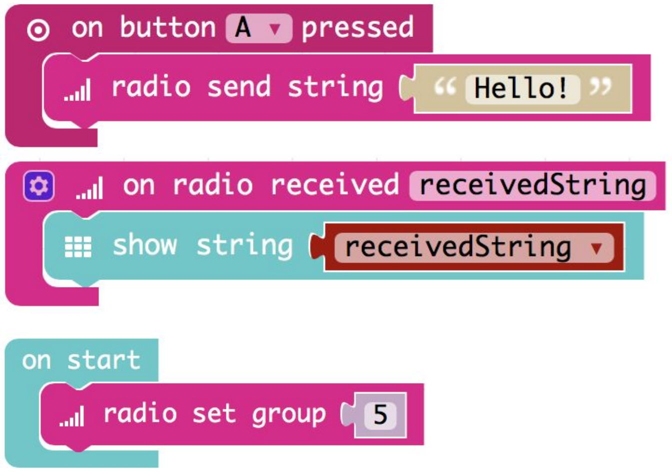
Info

The radio module of the Calliope mini uses **Bluetooth**.

Task

Now we want to send text messages.

1. Note: **Delete** the blocks of the previous flashcard. (otherwise it won't work)
2. Send: **Send string** "Hello!" **if button A is pressed**.
3. Receiving: Add **on radio received receivedString**.
4. To view the received message use **show string** from **basics** and receivedString ▾ from variables.
5. Set a group number at the start.

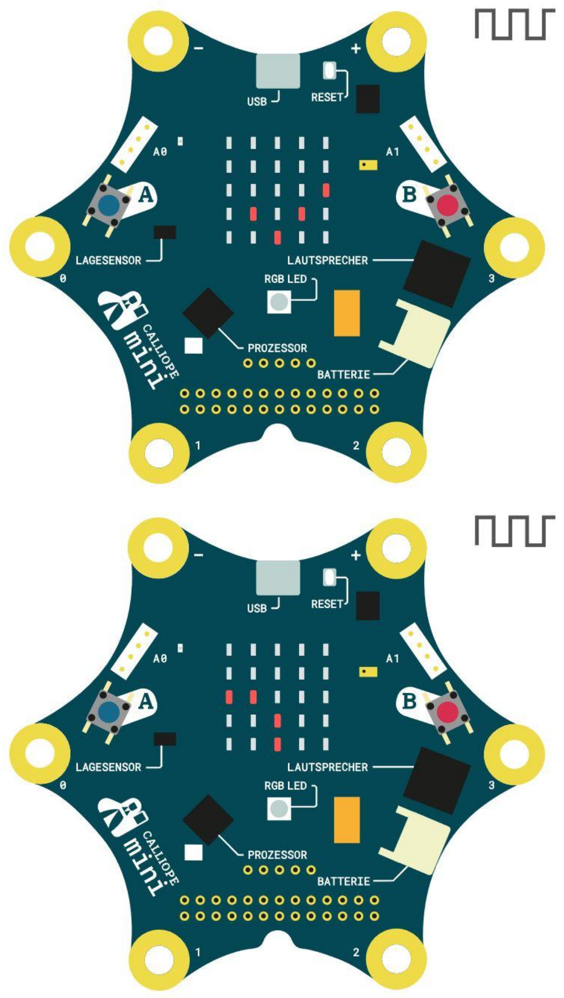


```
on button A ▾ pressed
    radio send string "Hello!"

on radio received receivedString
    show string receivedString ▾

on start
    radio set group 5
```

Try it out!

Find another group that has already worked on this card and try your program. After that, get the next flashcard.

4.3 Radio: Extend transmission



Info

We program **feedback** on this card.
With this you can make invisible processes visible.

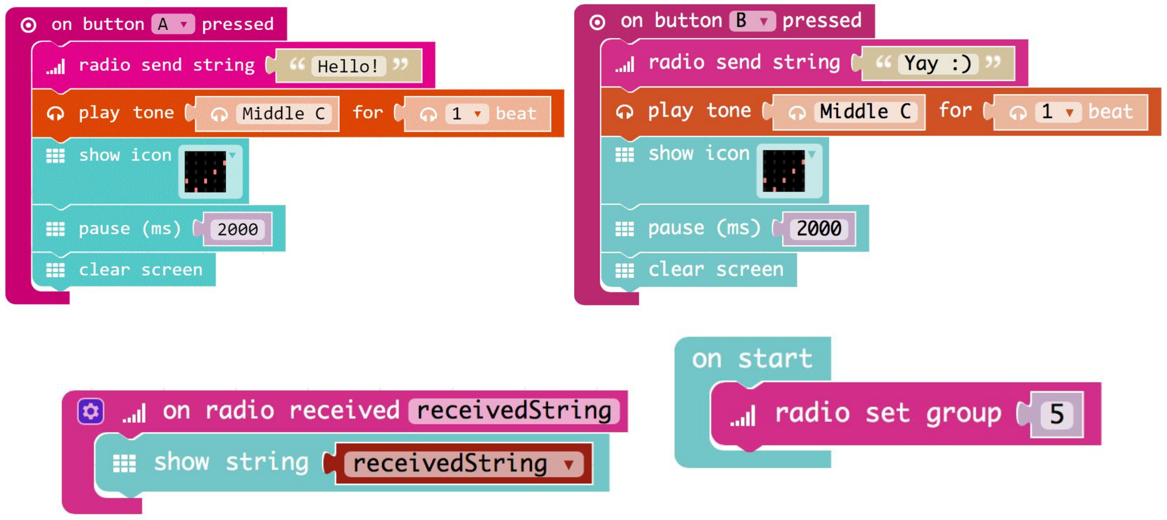
Task

If you send something, it arrives on the other Calliope. But you don't get any feedback from your own Calliope saying that it worked. We'll change that now.

1. Add **play tone**, **show icon**, **pause** and **clear screen** after sending a string.

Test your program in the preview. Is it working?

2. Think of another string to send when **button B is pressed**.

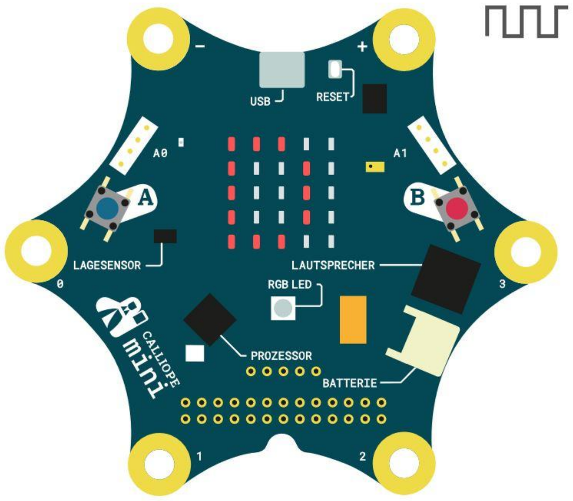


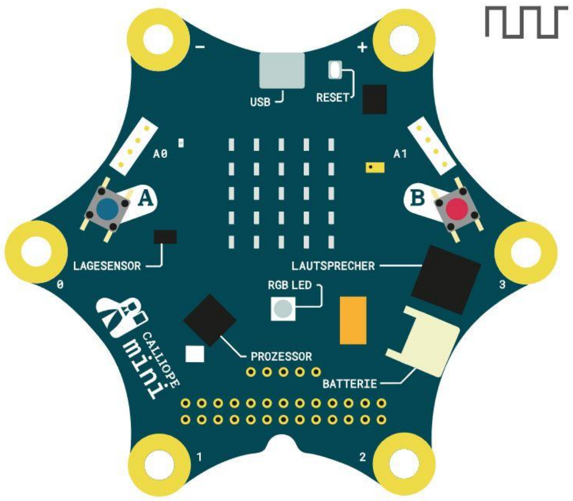
```
on button A pressed
    radio send string "Hello!"
    play tone Middle C for 1 beat
    show icon
    pause (ms) 2000
    clear screen

on button B pressed
    radio send string "Yay :)"
    play tone Middle C for 1 beat
    show icon
    pause (ms) 2000
    clear screen

on radio received receivedString
    show string receivedString

on start
    radio set group 5
```

Try it out!

Try different strings. Test your program again with another group.

4.4 Radio: Send confirmation back





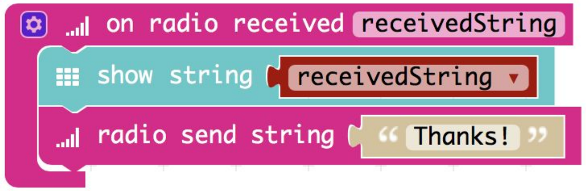
Info

Endless loops are usually errors. The computer does the same thing forever. Only RESET helps.

Task

Now we want to automatically send back a text when we have received something.

1. Add **send string** “Thanks!" after the received string is displayed. Can you imagine what will happen?
   Try it in the preview.

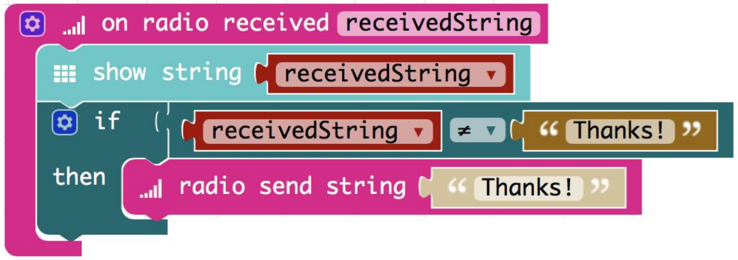


```
on radio received receivedString
    show string receivedString
    radio send string "Thanks!"
```

Task

We want to return "Thanks!" **only once**, so we don't have an endless loop (see info box on the left).

2. Send "Thanks!" only if the received message is NOT "Thanks!".
   a. Use if-then and = from Logic and change = to ≠.
   b. Compare the variable receivedString with the text "Thanks!". Text can be found at Advanced



```
on radio received receivedString
    show string receivedString
    if receivedString ≠ "Thanks!"
    then
        radio send string "Thanks!"
```

Try it out!

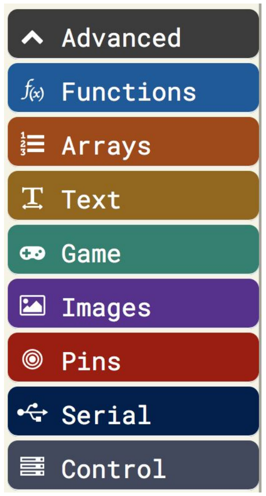
See how far you can move the Calliope devices apart and still get the messages.

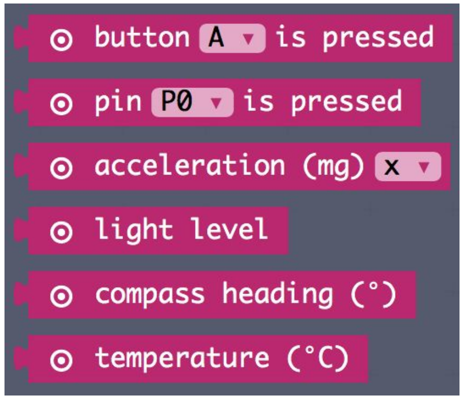
Bonus card: Radio

Everything done already?

You've experienced a more complicated application with two Calliope mini today and have noticed that you have to work very accurately.
Now try for yourself what is possible. At the end of the lesson you can show it to the others in your class.
You could, for example:

- Program the rock–paper–scissors game of the second session again and let the Calliope automatically count the score by communicating with each other.
- Build a thermometer and send the temperature to the other Calliope.
- Try all the other blocks that didn't yet appear in the flashcards.
- ...





You can try these blocks.