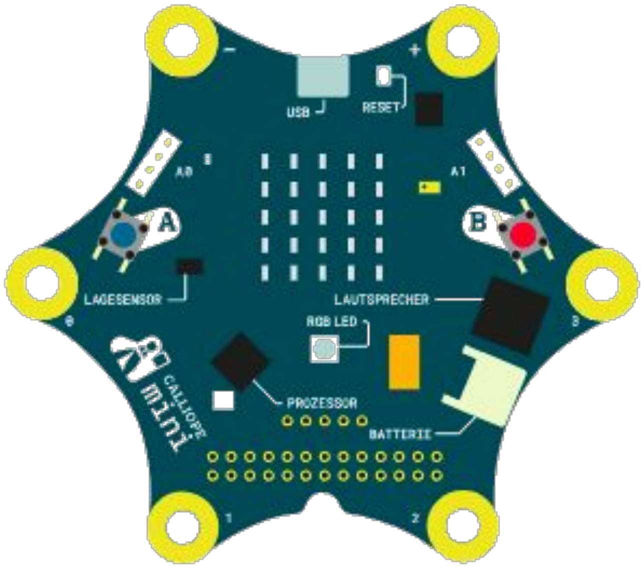
Translated from the German original into English by CALLIOPE

2.1 Nature: Temperature gauge

APP CAMPS



Note

The **temperature sensor** of the Calliope mini is not accurate and influenced by waste heat. Only an approximate value is given and it can happen that two calliopes in the same room show different values.

Info

Using variables, the Calliope mini can remember things. In our case, the Calliope mini remembers the temperature.
When the temperature changes, the value of the variable also changes.
The temperature is measured with the **temperature sensor**.

Task

We're developing a temperature gauge.

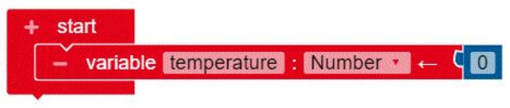
- First enter lab.open-roberta.org into your browser and then select Calliope (2017) as your system. → There you can program.

Step 1: Create a variable for the temperature.

You can create the variable by clicking on the small plus (+) next to **start**.

- Name your variable "temperature".
- Since we do not know the temperature yet, we simply enter the value 0 for now.

This is what your starting block should look like now:

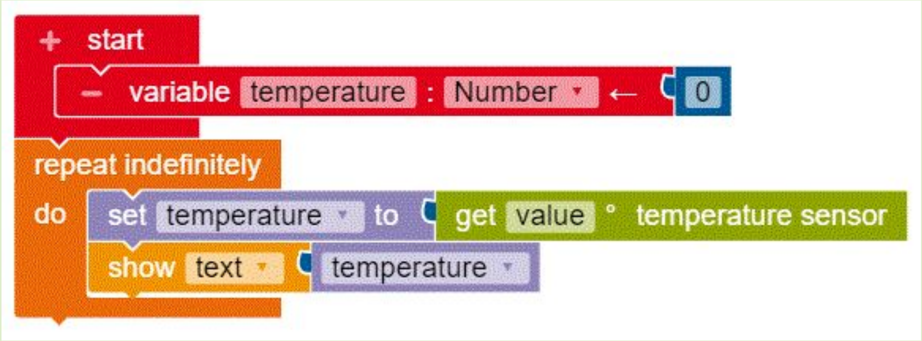


```
+ start
  − variable temperature : Number ← 0
```

Step 2: Check the temperature sensor

- Find the correct block to **measure the temperature**. Write the measured temperature in your **variable** and then display the value with a **show text** block.

If you did everything right, your result should look like this:

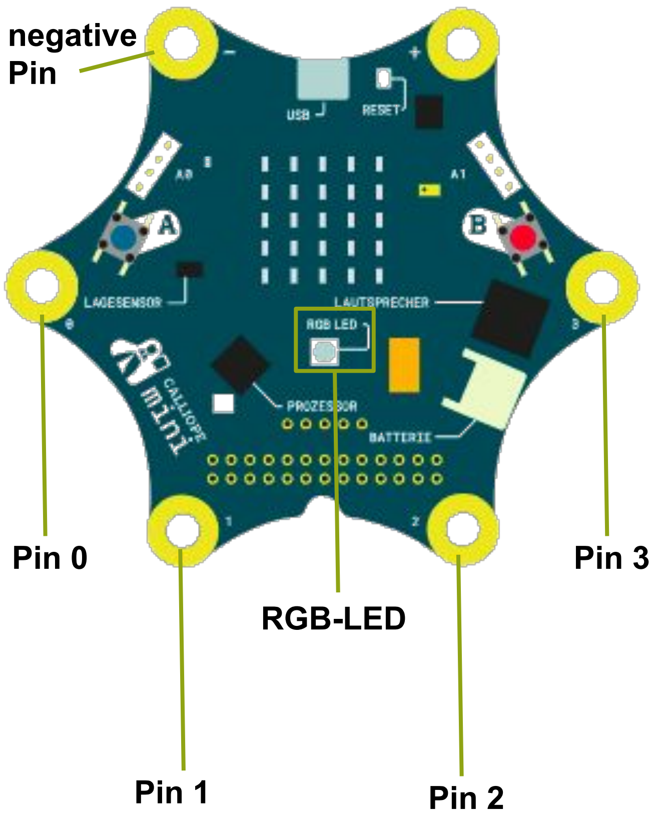


```
+ start
  − variable temperature : Number ← 0
repeat indefinitely
do  set temperature to get value ° temperature sensor
    show text temperature
```

cc BY NC SA

Translated from the German original into English by CALLIOPE

2.2 Nature: Color LEDs



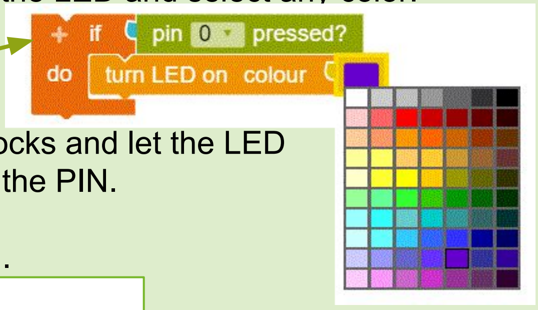
Info

The pins are not buttons like button A or button B, but can be used in a similar way. For example, to press or activate pin 0, you must simultaneously touch the negative pin with one finger and pin 0 with another finger.

Task

Now color comes into play: Depending on which PIN is pressed, the RGB LED (see left) should light up in a different color.

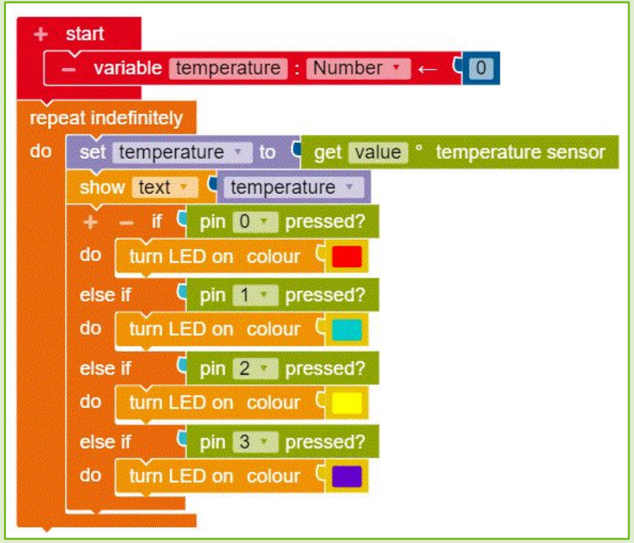
- Add a **if do** block under the last show text block.
- When **pin 0 is pressed**, turn on the LED and select any color.



```
+ if  pin 0 pressed?
do    turn LED on colour 🟪
```

- Use the **+** to add more **else if** blocks and let the LED light up differently depending on the PIN.

Here you see a possible solution.



```
+ start
  – variable temperature : Number ← 0
repeat indefinitely
do  set temperature to get value ° temperature sensor
    show text temperature
    + – if pin 0 pressed?
    do      turn LED on colour 🟥
    else if pin 1 pressed?
    do      turn LED on colour 🟦
    else if pin 2 pressed?
    do      turn LED on colour 🟨
    else if pin 3 pressed?
    do      turn LED on colour 🟪
```

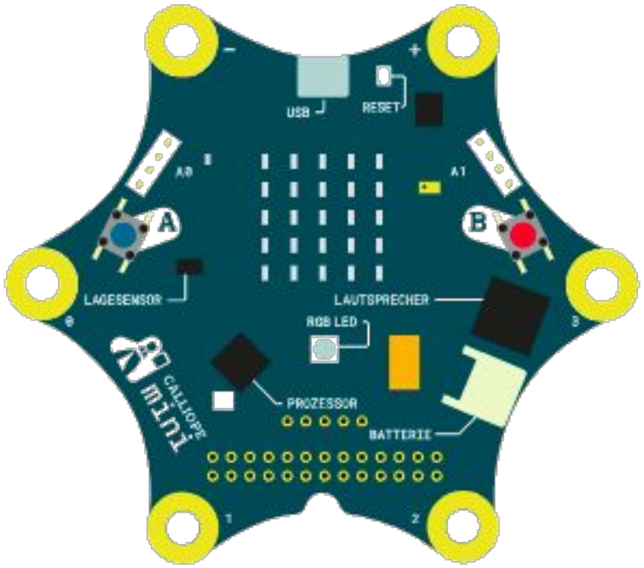
Next task

Now adjust your program so that the color of the LED is controlled by the current temperature and not by the PINs.

Try it yourself before you get your next flash card.

BY NC SA

2.3 Nature: Color temperature



Info

You can warm the Calliope mini with your hands to create small temperature differences.

Note

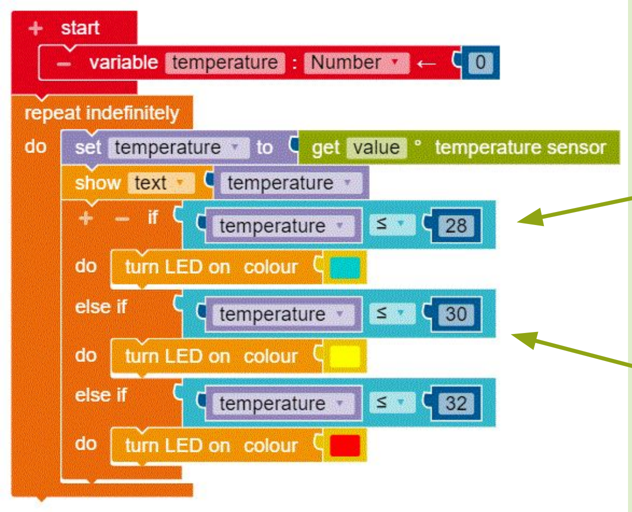
The order of the temperature ranges* is important! Think about why this is the case.

*In our example, up to 28 degrees, from 29 to 30 degrees, and from 31 to 32 degrees.

Task

We control the color of the LED via the temperature (and not via the PINs)

- Instead of pin blocks you now need logical comparisons in which you stake out temperature ranges.
  - If it is cold to warm, the LED should light up blue.
  - If it is very warm, the LED should light yellow.
  - When it is hot, the LED should light up red.
- We have created temperature ranges up to 28 degrees, 30 degrees and 32 degrees Celsius. Depending on how warmer it is at your place, you have to adjust the values accordingly. The ranges should be relatively close to each other, so that you can quickly detect temperature fluctuations.



```
+ start
  − variable temperature : Number ▾ ← 0
repeat indefinitely
do  set temperature ▾ to get value ° temperature sensor
    show text ▾ temperature ▾
    + − if temperature ▾ ≤ ▾ 28
    do turn LED on colour 🟦
    else if temperature ▾ ≤ ▾ 30
    do turn LED on colour 🟨
    else if temperature ▾ ≤ ▾ 32
    do turn LED on colour 🟥
```

If the temperature is less than or equal to 28 degrees, the LED should light blue.

else if the temperature is less than or equal to 30 degrees, the LED should light yellow etc.

Next task

Now we extend the program by a moisture meter, which can inform you e.g. if your plants have too little water. For this you need crocodile clips.

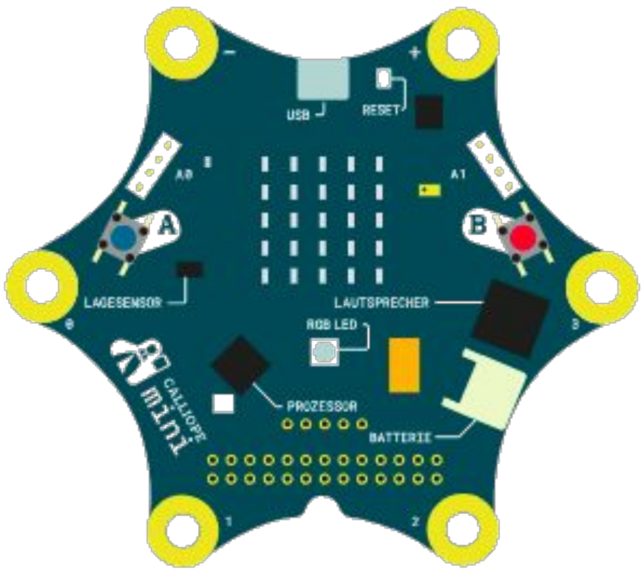
Try it yourself before you get your next flash card.

cc BY NC SA

Translated from the German original into English by CALLIOPE

2.4 Nature: Moisture meter

APP CAMPS



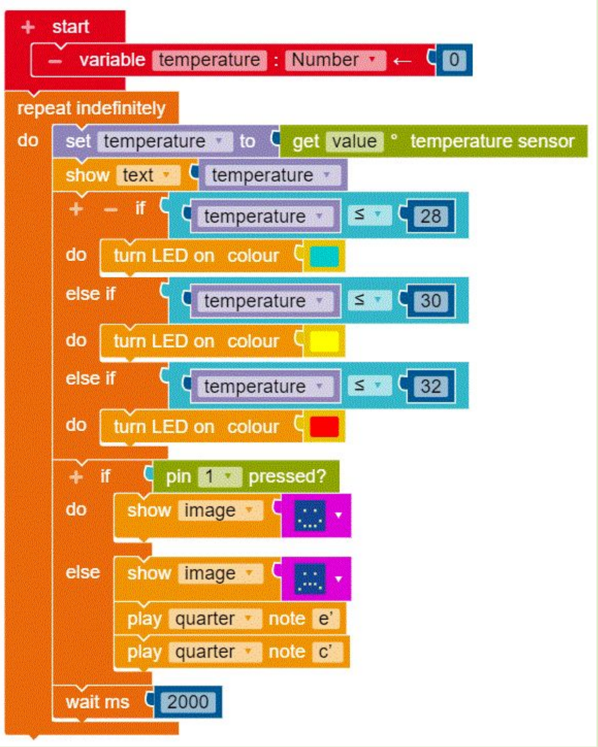
Info

So that the calliope warns you when your plant should be watered, you can, for example, put two spoons in a flower pot and then connect a crocodile clip with one spoon and the negative pin and another crocodile clip with pin 1 and the other spoon. If the soil is damp, it will conduct, otherwise not. If the soil no longer conducts, there is a danger that your plant will dry out. Your flower will contact you if it has too little water.

Task

We extend the program with a moisture meter that can tell you, for example, when your plant should be watered.

- You need a if else do block.
- If pin 1 is pressed, then display a laughing smiley, otherwise display a sad smiley and play two quarter notes.
- At the end insert a wait block with 2000 ms (2 sec).
- Test your program:
  - Connect the first crocodile clip to the negative pin and a conductive object (e.g. banana, apple or fork).
  - Connect the second crocodile clip to pin 1 and the same conductive object.



```
+ start
  − variable temperature : Number ← 0
repeat indefinitely
do  set temperature to get value ° temperature sensor
    show text temperature
    + − if temperature ≤ 28
    do  turn LED on colour 🟦
    else if temperature ≤ 30
    do  turn LED on colour 🟨
    else if temperature ≤ 32
    do  turn LED on colour 🟥
    + if pin 1 pressed?
    do  show image
    else show image
         play quarter note e'
         play quarter note c'
    wait ms 2000
```

Next task

Now we extend the program by a brightness meter, which can inform you e.g. if your plant gets too little light.

Do you have an idea for a possible solution? There are several possibilities.

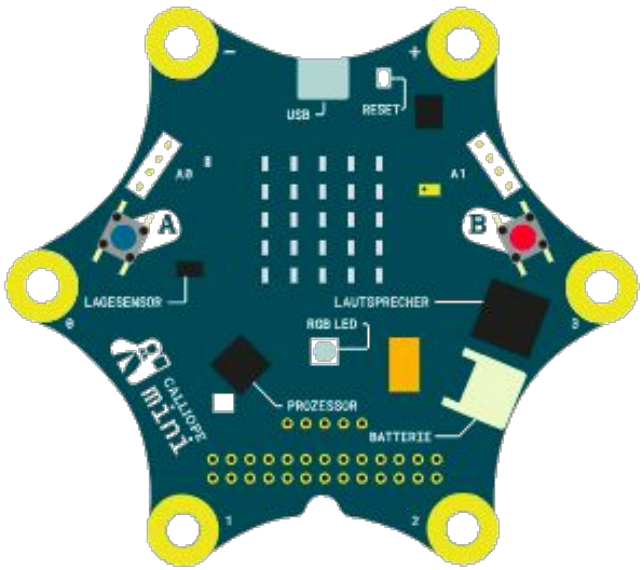
Think for yourself before you get the next flashcard.

cc BY NC SA

Translated from the German original into English by CALLIOPE

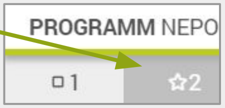
2.5 Nature: Brightness meter

APP CAMPS



Note

The area with the functions can be found in the extended editor (upper left).



PROGRAMM NEPO

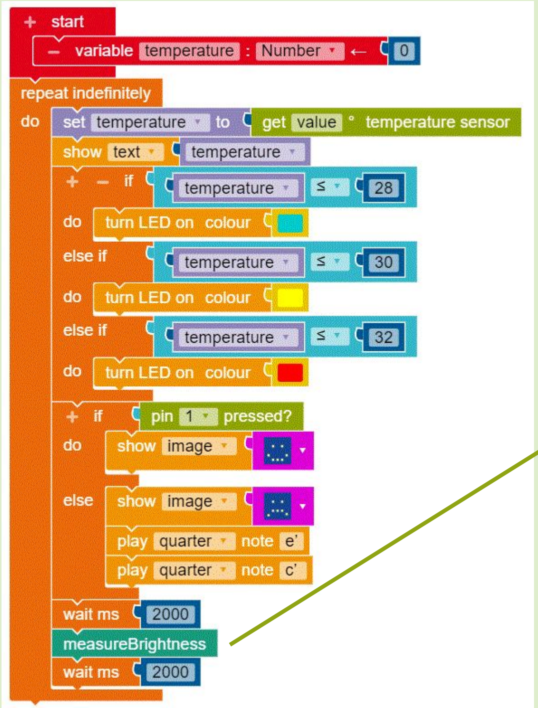
Info

Instead of the lines in the LED display you could of course display the brightness value differently.
To change the brightness, you can try to make shadows with your hands.

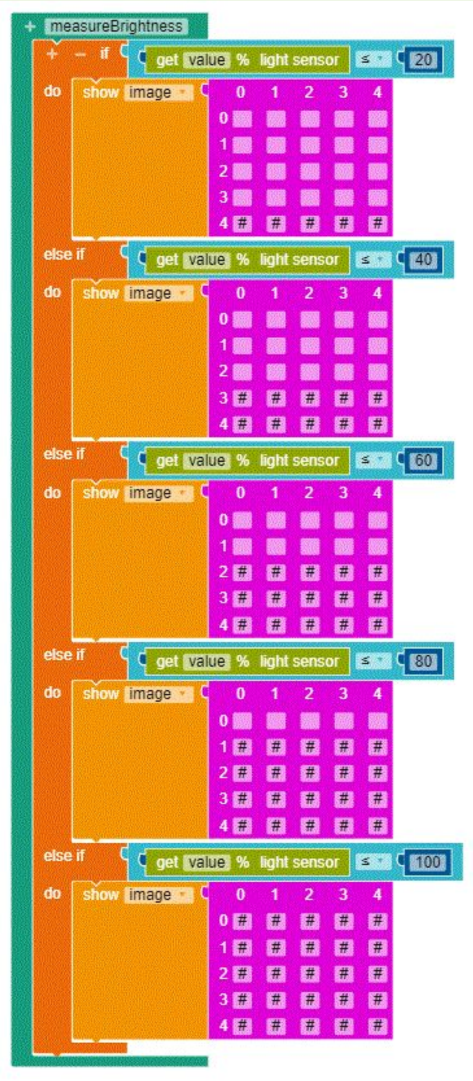
Task

We extend the program by a brightness meter. (Possible solution)

- Write your own **function** (extended editor) and name the function "measureBrightness".
- Depending on the brightness, the light sensor returns a value between 0 and 100.
- As with the temperature, we can now define a brightness range.
- Depending on the brightness we fill one to five lines of the LED display.
- Important: The function must be called!



```
start
  variable temperature : Number ← 0
repeat indefinitely
do  set temperature to get value ° temperature sensor
    show text temperature
    if temperature ≤ 28
    do   turn LED on colour 🟦
    else if temperature ≤ 30
    do   turn LED on colour 🟨
    else if temperature ≤ 32
    do   turn LED on colour 🟥
    if pin 1 pressed?
    do   show image
    else show image
         play quarter note e'
         play quarter note c'
    wait ms 2000
    measureBrightness
    wait ms 2000
```



```
measureBrightness
  if get value % light sensor ≤ 20
  do show image
       0 1 2 3 4
     0
     1
     2
     3
     4 # # # # #
  else if get value % light sensor ≤ 40
  do show image
       0 1 2 3 4
     0
     1
     2
     3 # # # # #
     4 # # # # #
  else if get value % light sensor ≤ 60
  do show image
       0 1 2 3 4
     0
     1
     2 # # # # #
     3 # # # # #
     4 # # # # #
  else if get value % light sensor ≤ 80
  do show image
       0 1 2 3 4
     0
     1 # # # # #
     2 # # # # #
     3 # # # # #
     4 # # # # #
  else if get value % light sensor ≤ 100
  do show image
       0 1 2 3 4
     0 # # # # #
     1 # # # # #
     2 # # # # #
     3 # # # # #
     4 # # # # #
```

CC BY NC SA

Bonus card

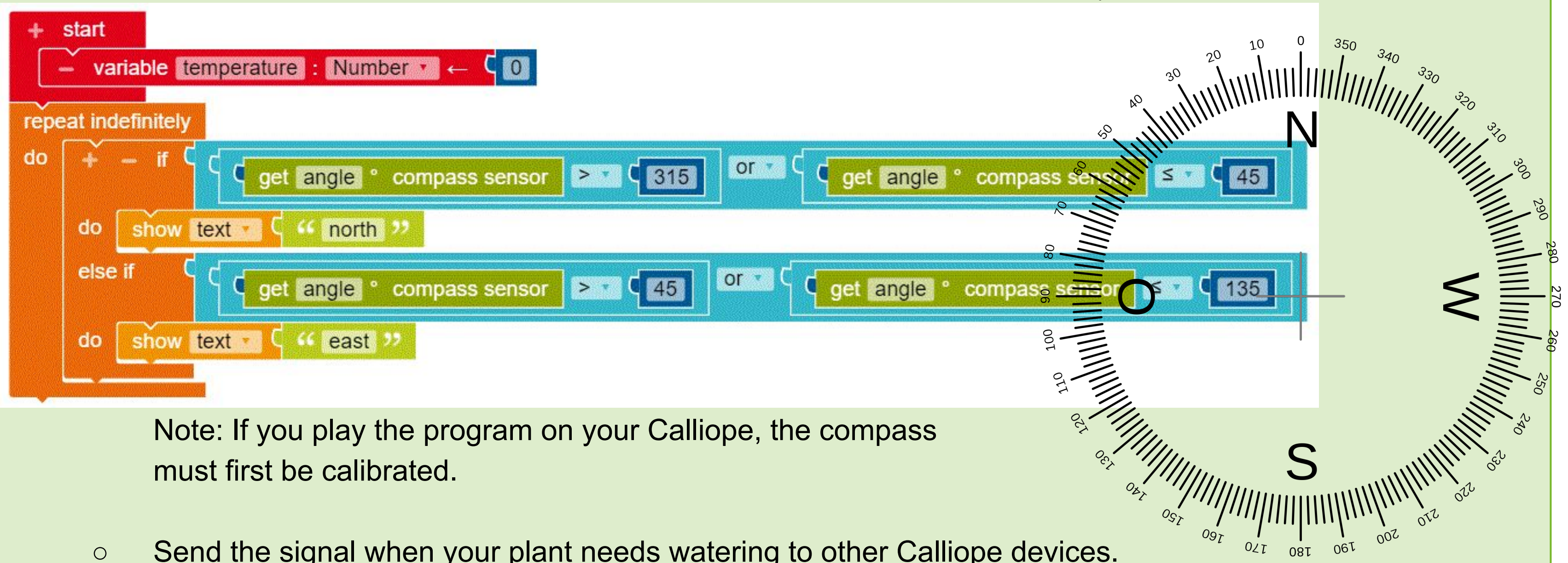
Everything done already?

You built your own little flower station today with the Calliope. You have learned how to use different sensors in a meaningful way. Here you can find a video with the setup: https://bit.ly/2L4nX5k

Now upgrade your flower care station.

At the end of the lesson you can then show what you have done to the others in your class.

- You could, for example:
  - ○ DeUse the compass to determine the correct location for your plant via the compass direction. (For example, it is best to place a cactus in the south, as it needs a lot of sun.) A small example of this....



```
+ start
  – variable temperature : Number ← 0
repeat indefinitely
do  + – if  get angle ° compass sensor > 315  or  get angle ° compass sensor ≤ 45
    do  show text “ north ”
    else if  get angle ° compass sensor > 45  or  get angle ° compass sensor ≤ 135
    do  show text “ east ”
```

Note: If you play the program on your Calliope, the compass must first be calibrated.

  - ○ Send the signal when your plant needs watering to other Calliope devices.
  - ○ Do you have any other ideas? Great, then you can try it out.

cc BY NC SA