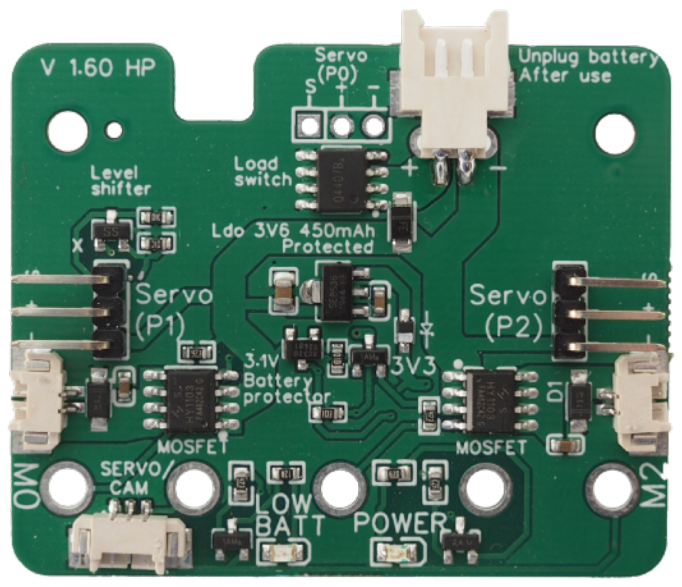
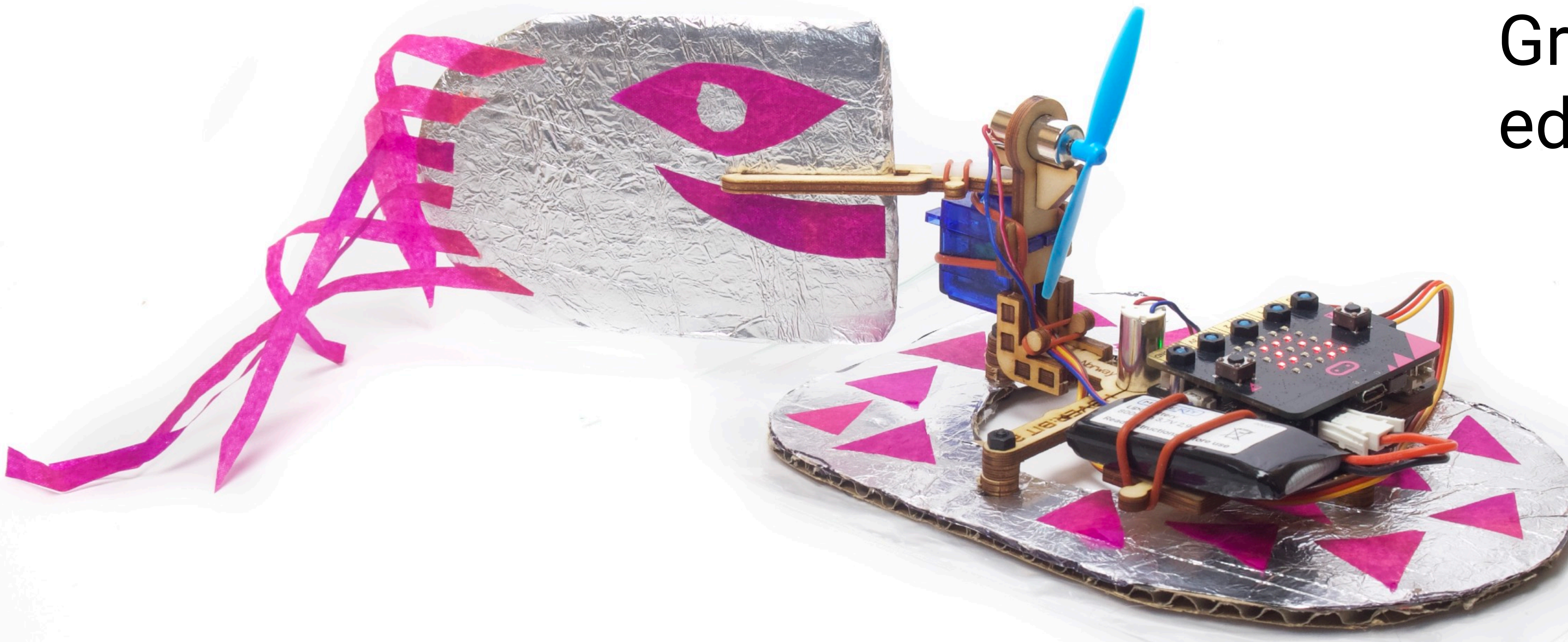
Hover:bit 2

Coding the micro:bit hovercraft



Green board edition



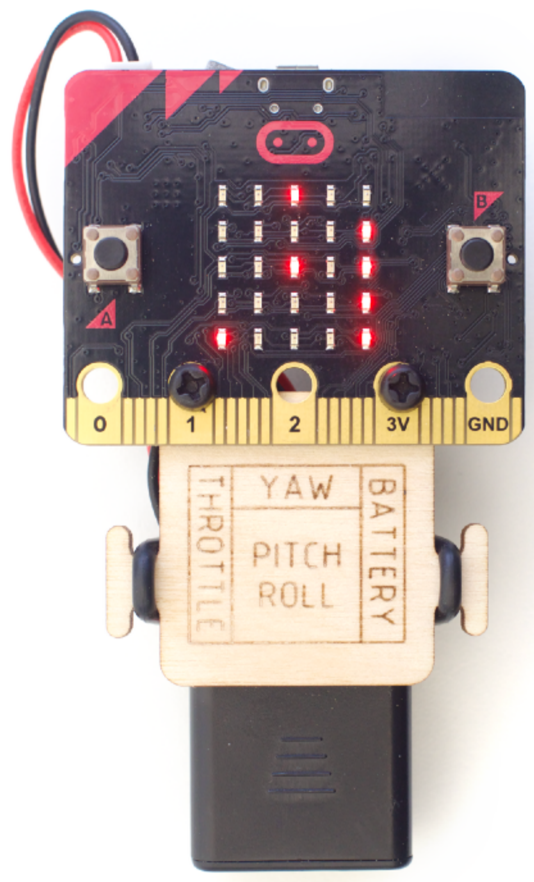
MAKEKIT

makekit.no

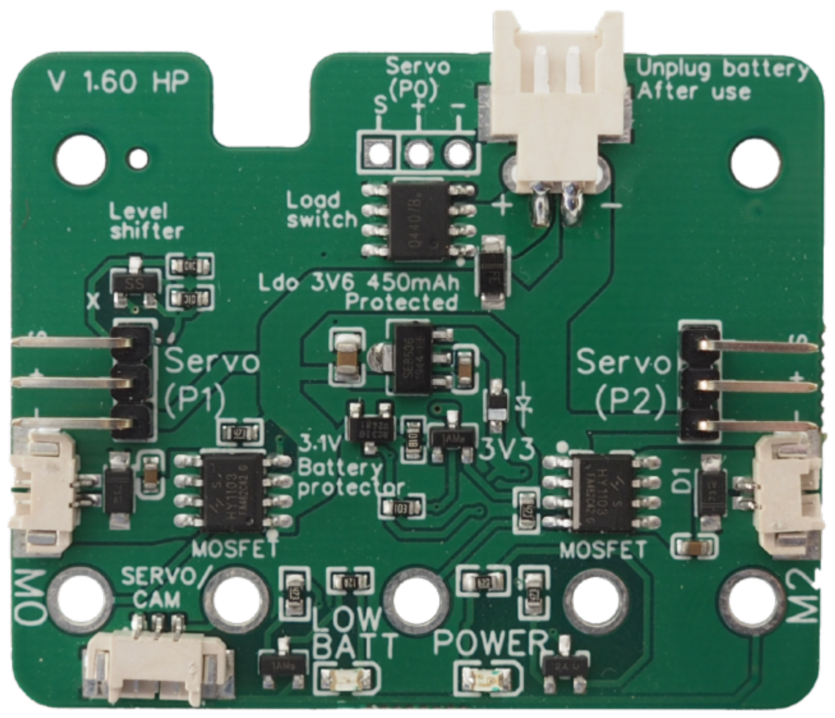
Hover:bit 2

Coding the micro:bit hovercraft

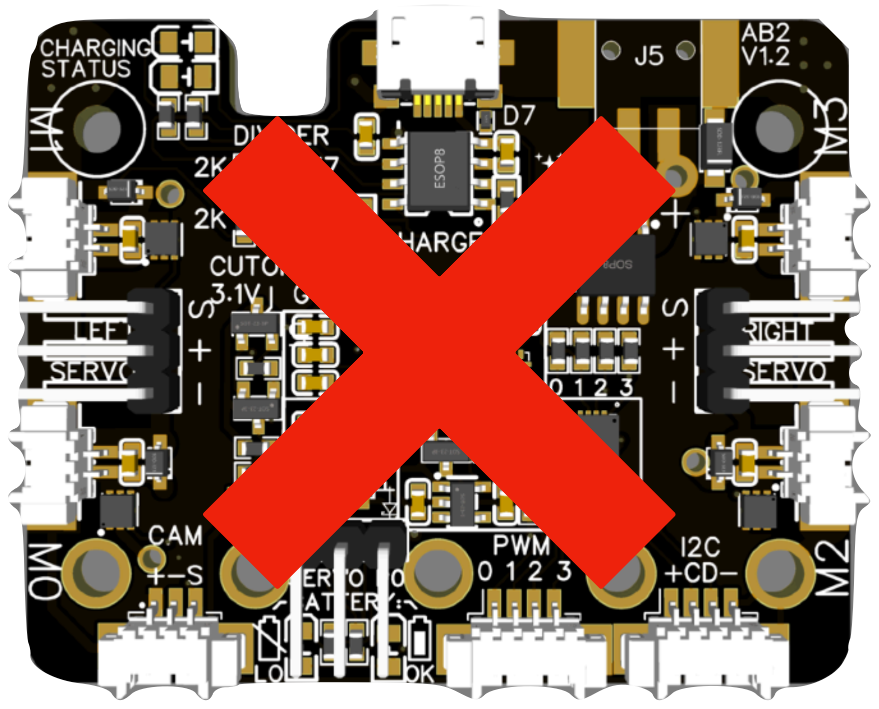
Tip! You can also download the readymade code from makekit.no/docs



Code the remote

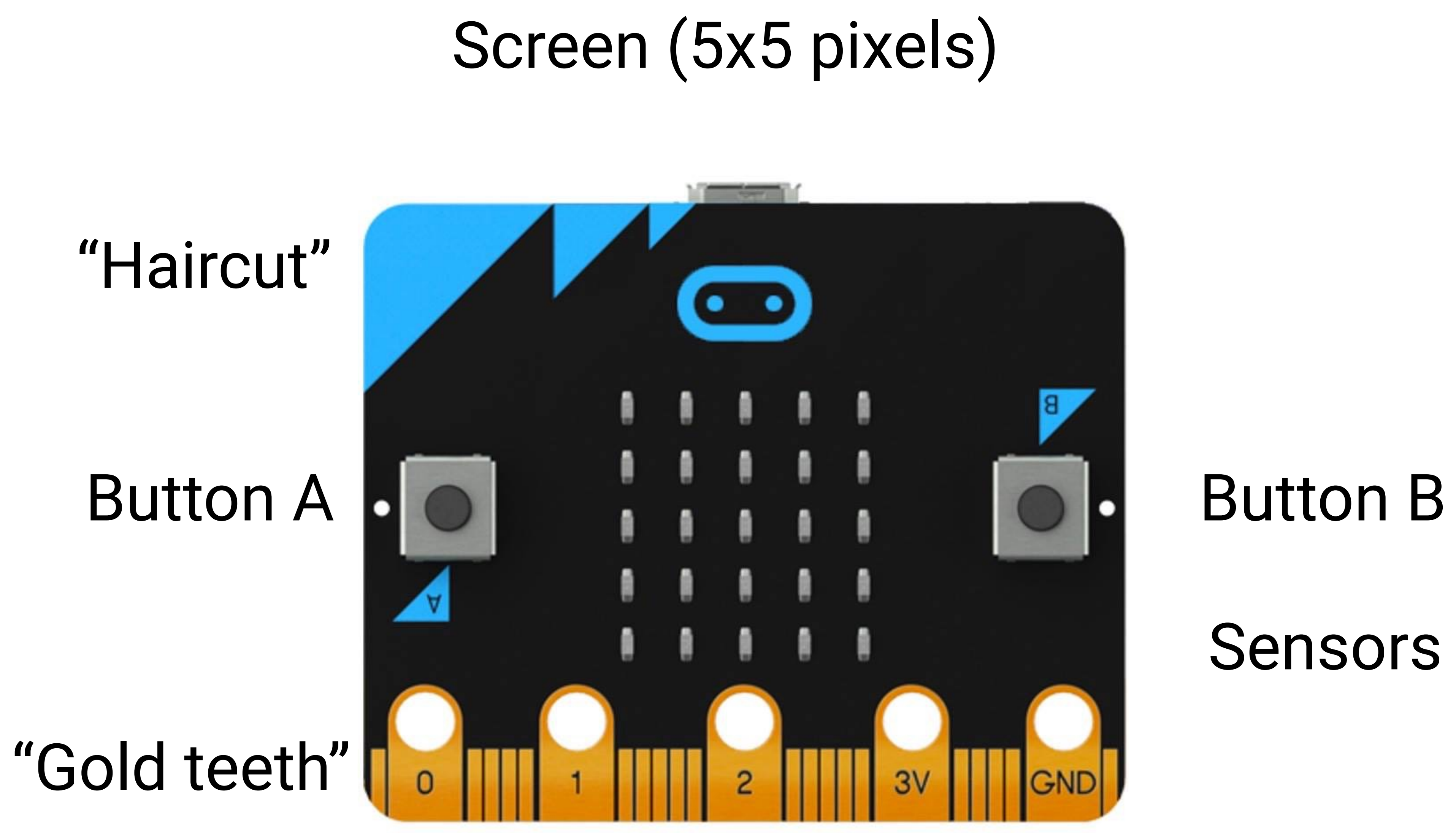


Code the green card

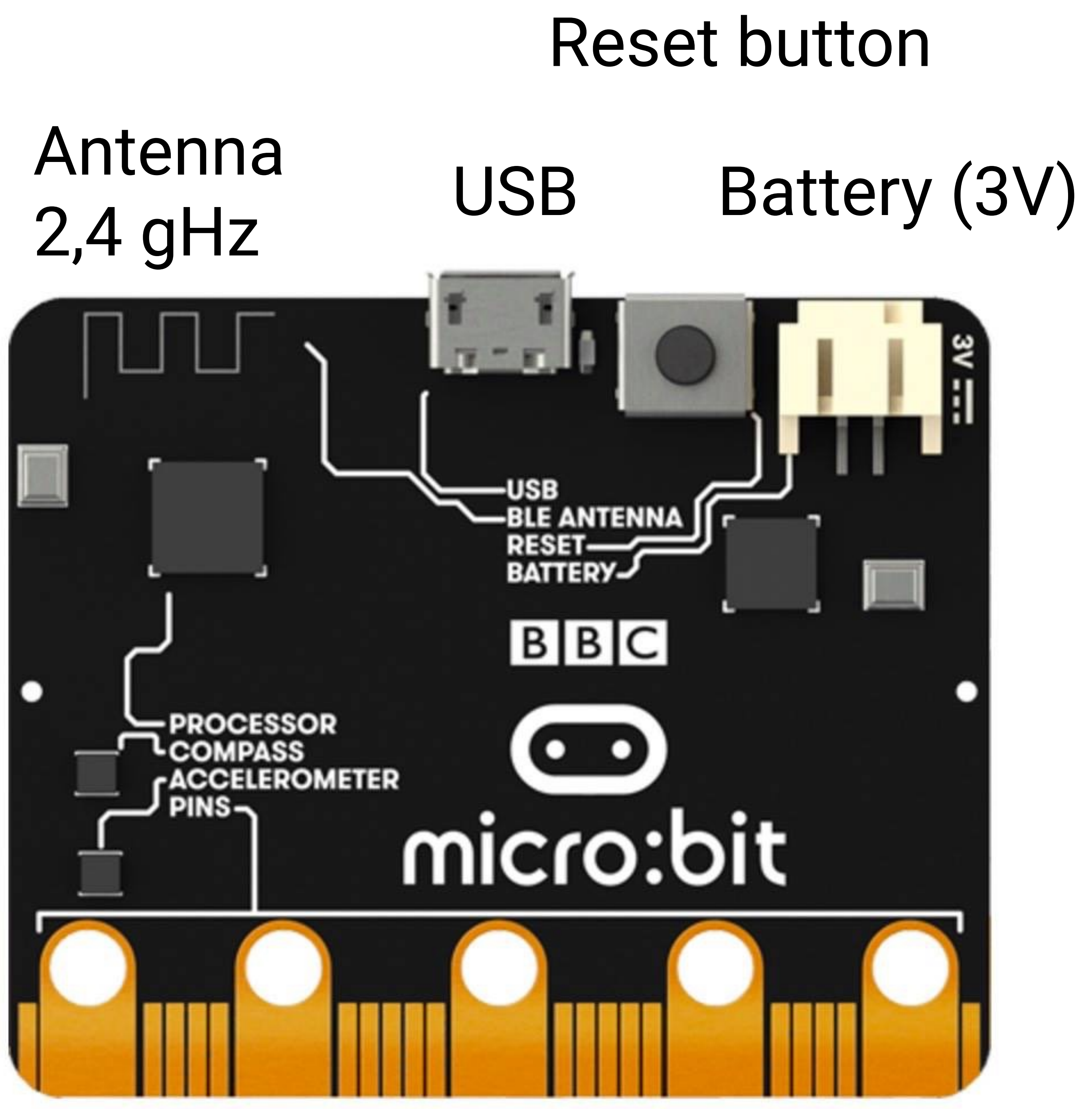


Code for black card is not yet available. Please download the finished code from docs.

Meet the micro:bit



Front



Back

micro:bit is a small computer with prosessor, sensors, display and radio. It has connection pins for external components like LEDs, speakers or various sensors.

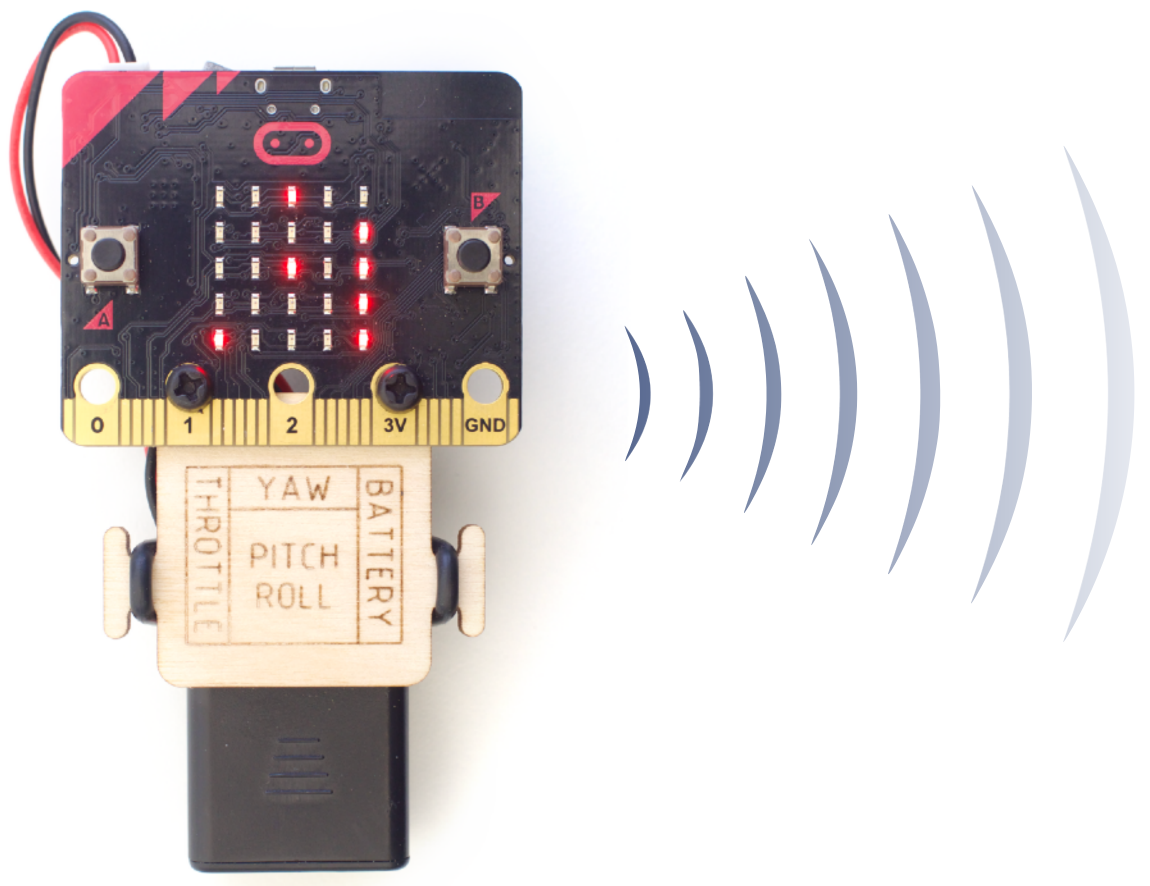
You can learn more at: https://tech.microbit.org/hardware/

The ART Rule

Three values that controls the hover:bit

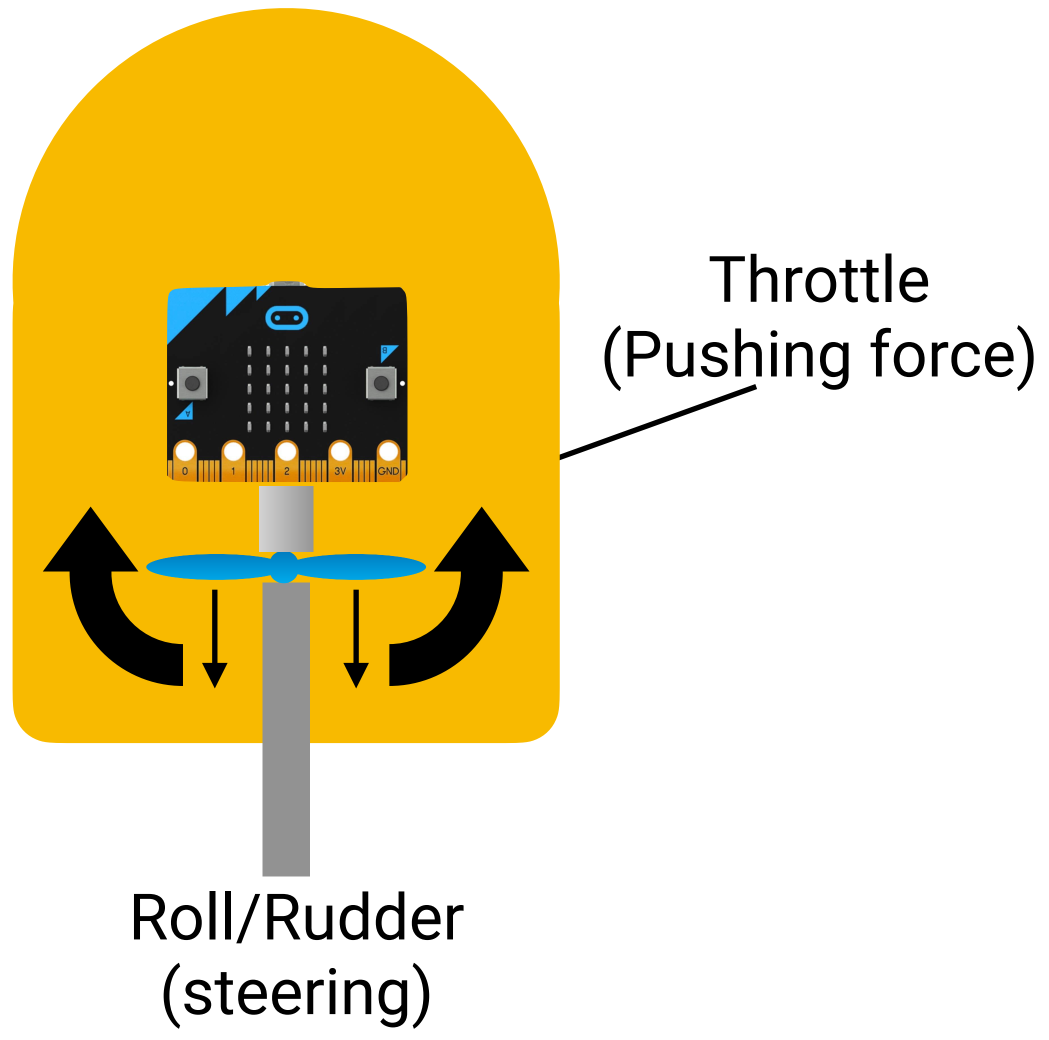
Controls

Arm starts and stops motors.
Roll controls the rudder and stearing
Throttle control the speed.
The values are being transmitter from remote control to the receiver (hovercraft)



ART

arm, roll, throttle



These 3 values control the hovercraft:

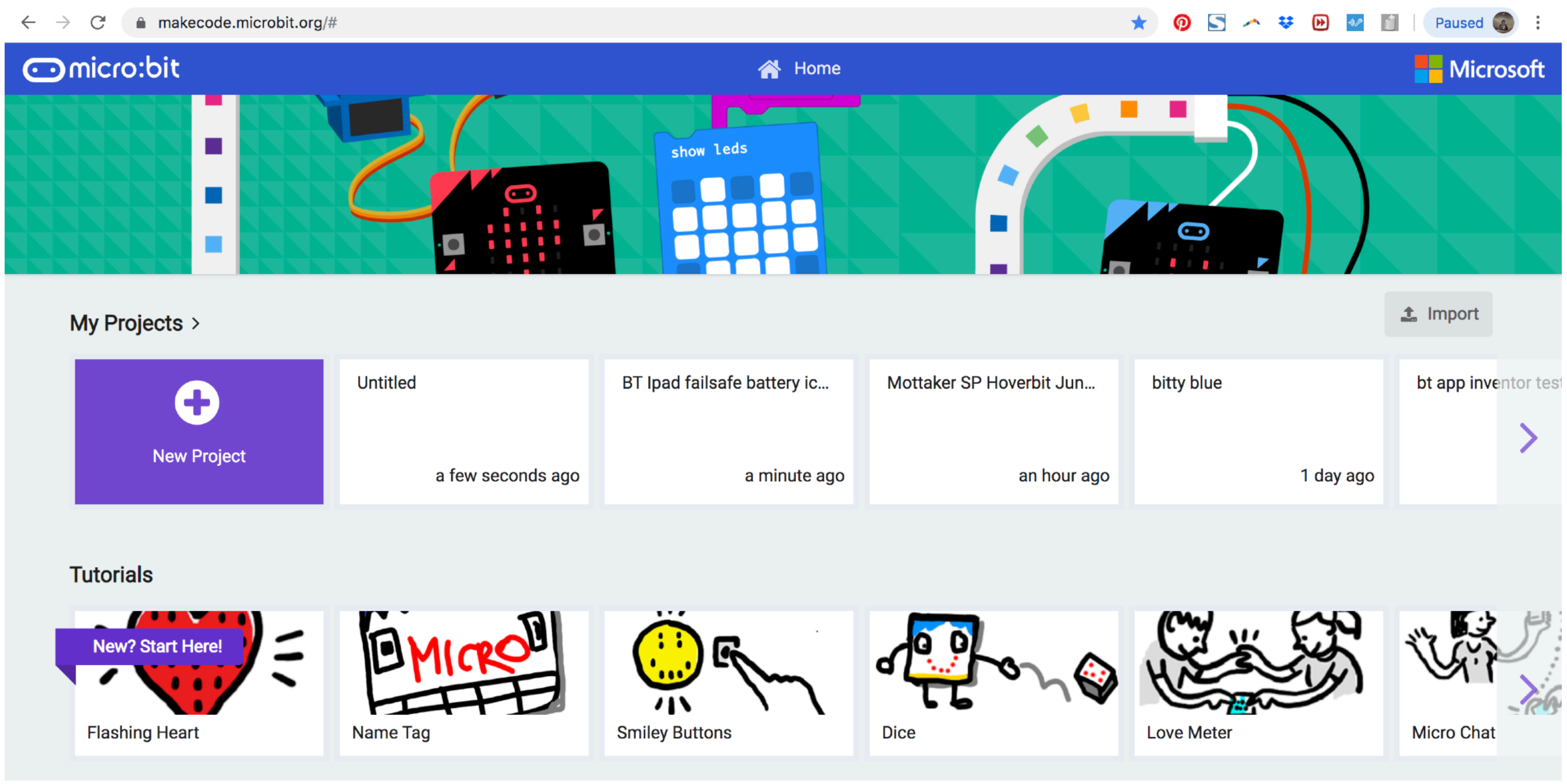
Variable:	Type	Minimum	Neutral	Maximum
Arm (start/stop)	Binary	0		1
Throttle (speed)	Percent	0	50	100
Roll (steering)	Degrees	-45	0	45

Let´s code!

```
on start
  set radioGroup ▾ to 13
  set batteryFactor ▾ to 4.692
  set flightMode ▾ to 1
  radio set group radioGroup ▾
  show number radioGroup ▾
  serial
  redirect to

forever
  set bat
  clear sc
  if
    call fu
  else
    call fu
```

Start at makecode.microbit.org

Chrome is recommended for better connection with the micro:bit



Select “New project”.
PS. If you’re new to micro:bit you should try one of the tutorials above first.

The editor

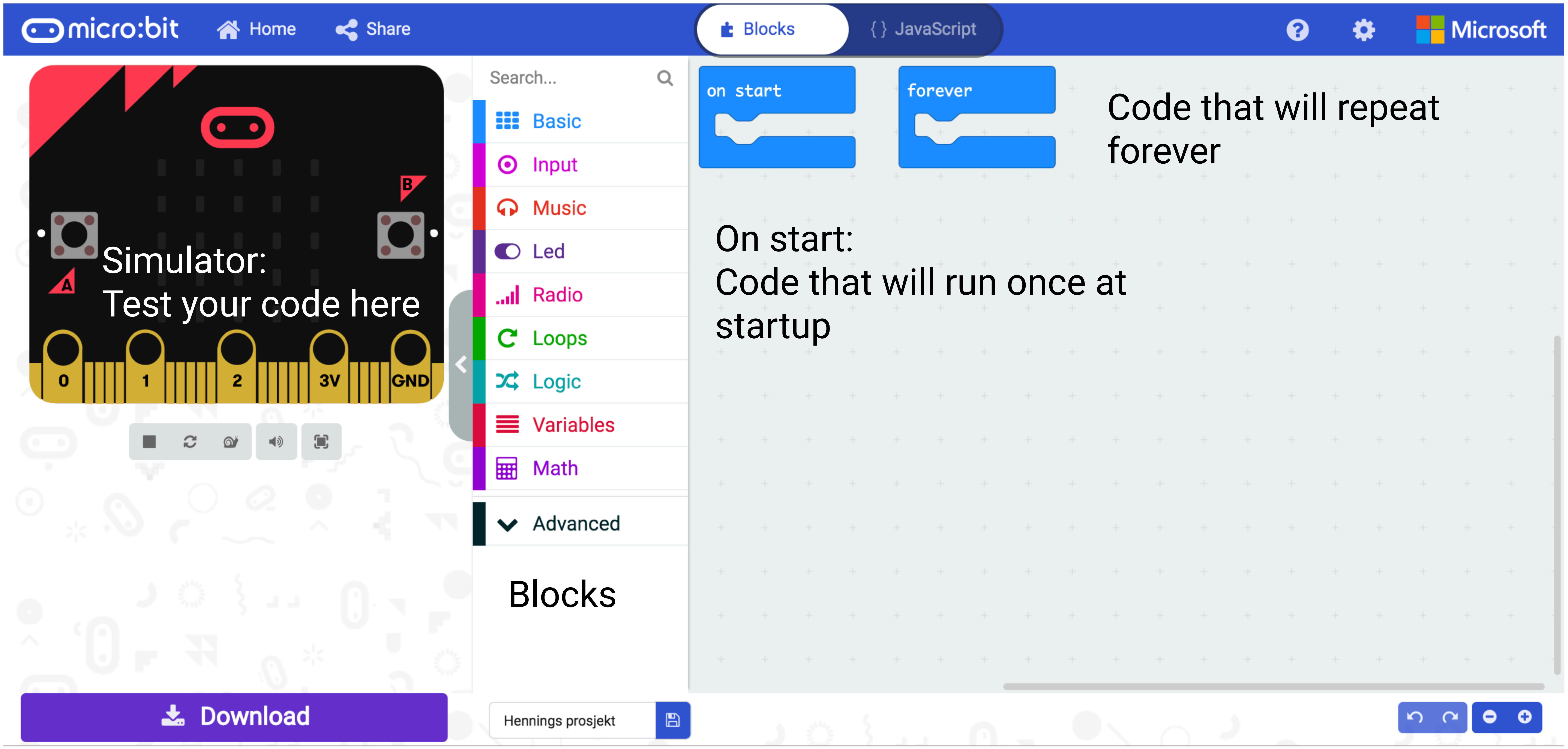
Main menu

Share your code

Block mode

Javascript mode

Settings



Code that will repeat forever

On start:
Code that will run once at startup

Simulator:
Test your code here

Blocks

Save a backup

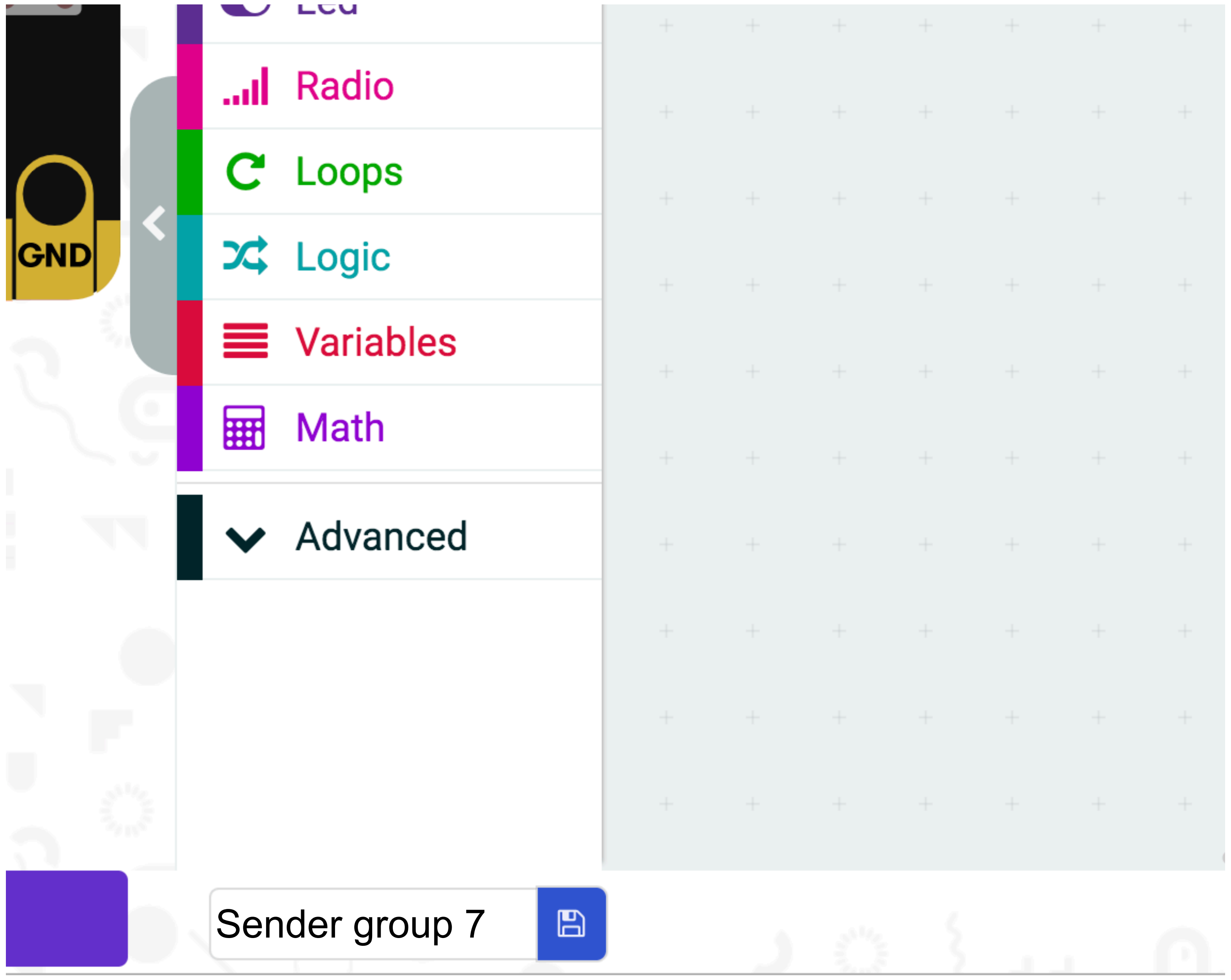
Undo zoom

Download to micro:bit

Name your project

Name your project

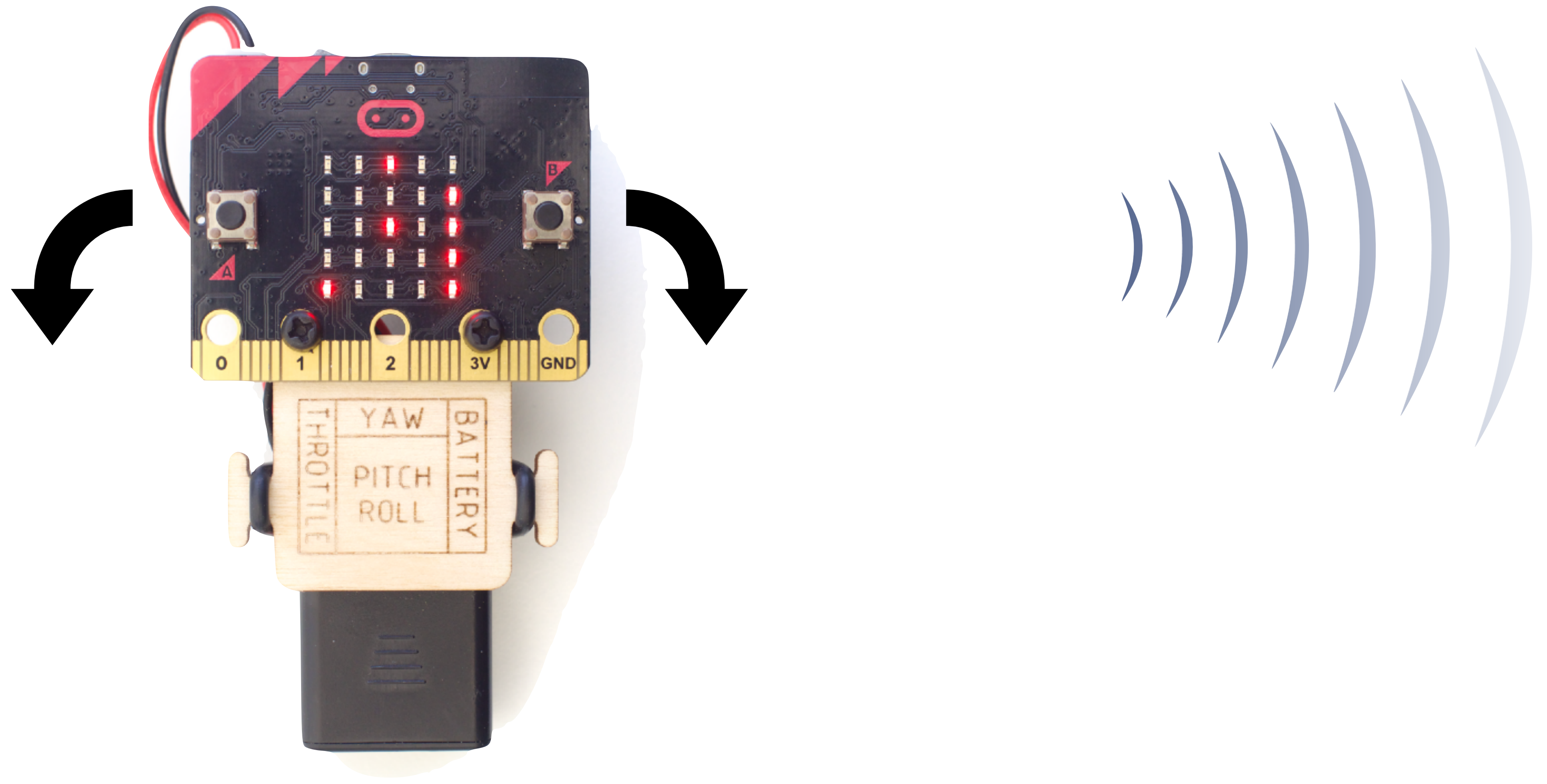
Start by giving your project a name like “Sender group x”. This is your unique radio channel. If you are alone you can use channel 7



Sender group 7

Code for transmitter

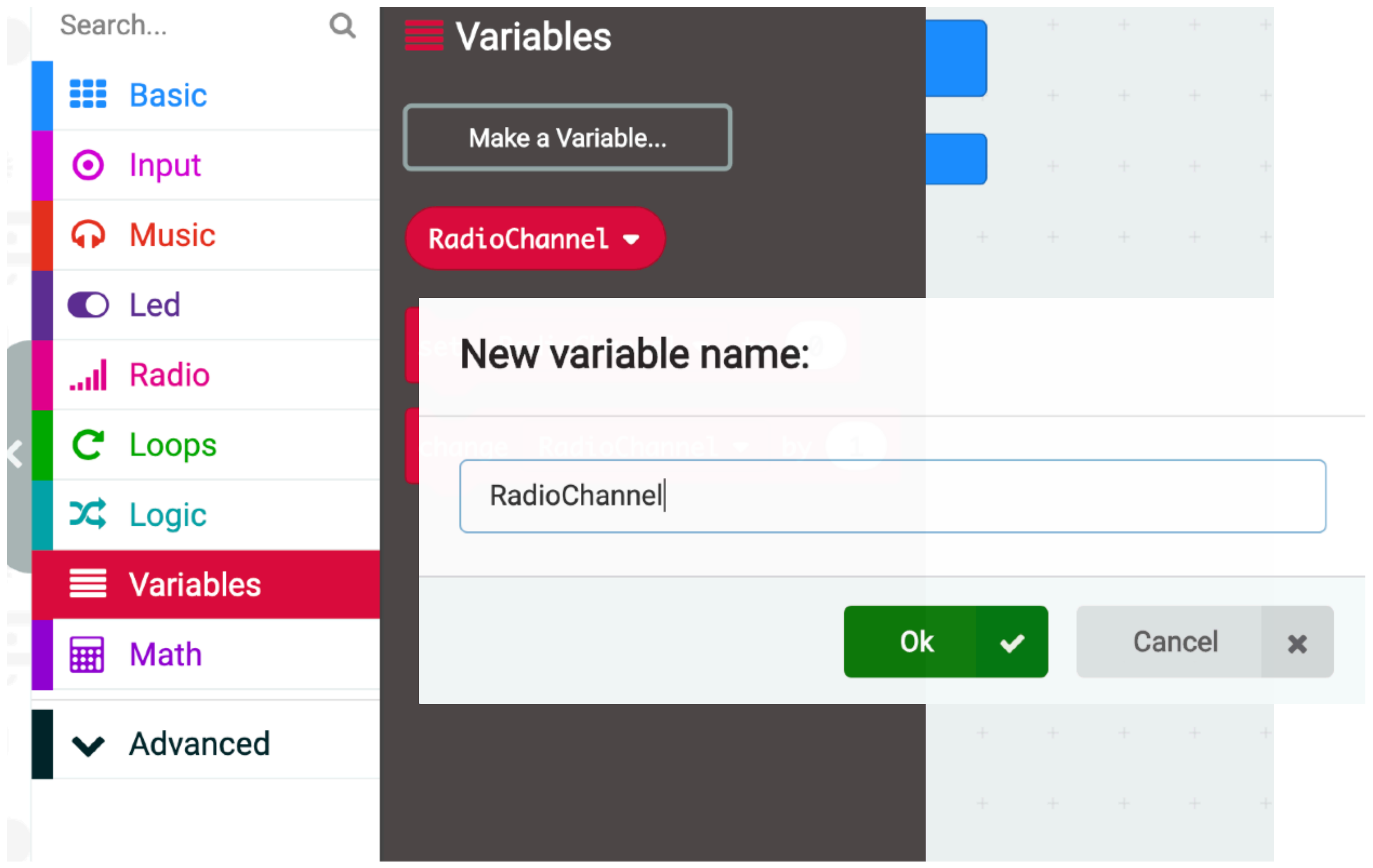
We will create a code that turns our transmitter into a remote control for the hovercraft.



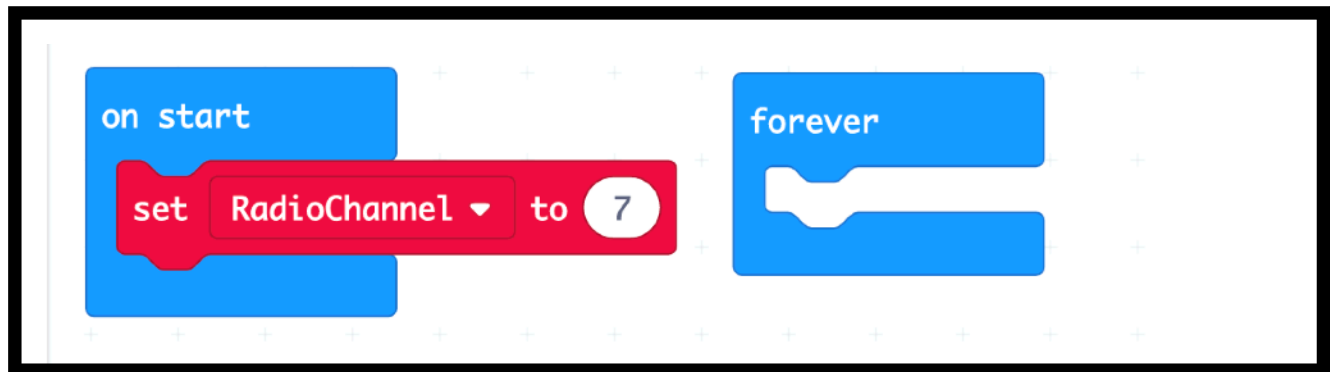
Radio

1.Make a variable called radioChannel
2.Set the radioChannel to 7 (or a number between 0 and 255). This number must also be used on the hovercraft later
3.Use Show Number (in the forever loop) to å verify that Arm is changing correctly
4.Use the “radio set group” to make the radio channel take effect
5.If you are in a classroom, each hover:bit maker need to choose their own channel.

1

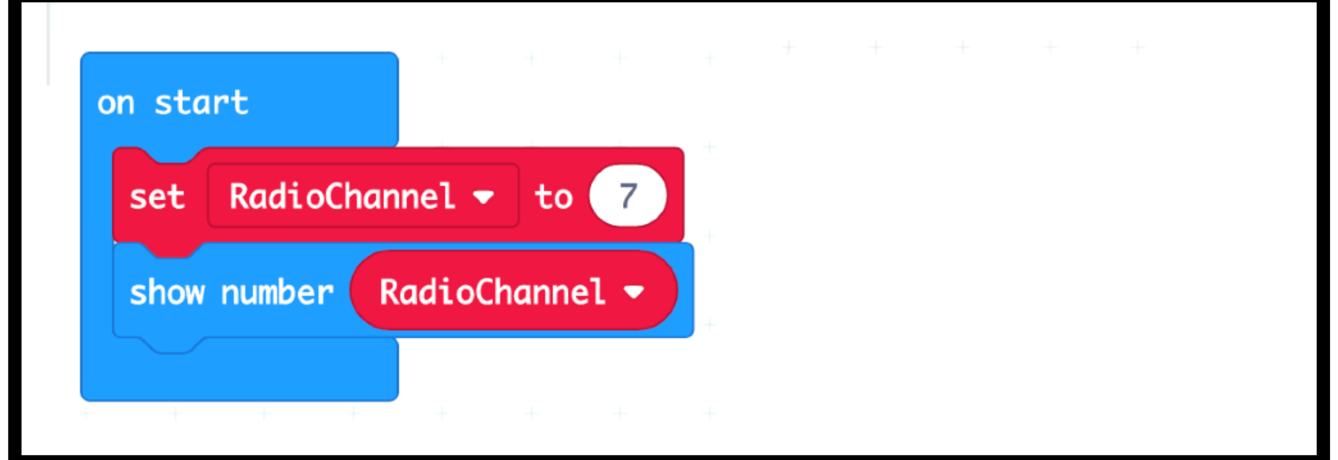


2



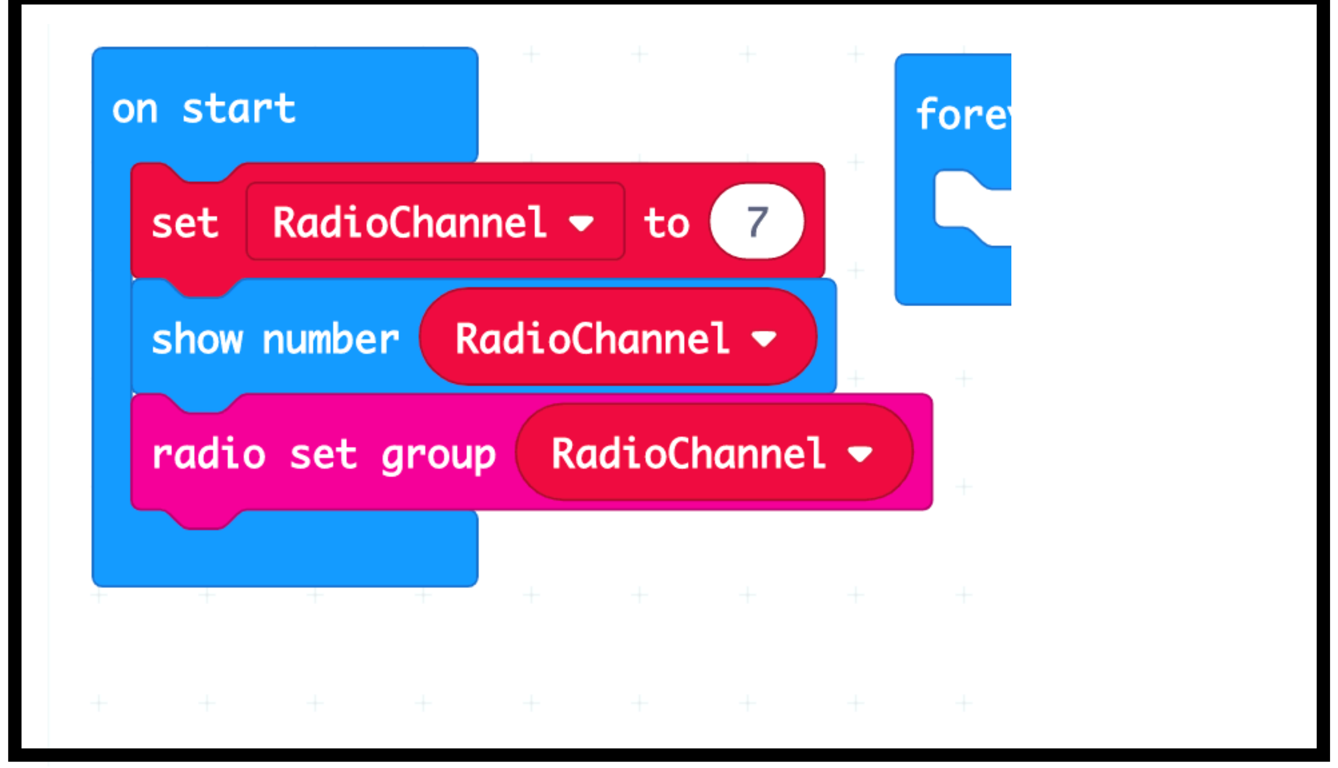
```
on start
    set RadioChannel to 7
forever
```

3



```
on start
    set RadioChannel to 7
    show number RadioChannel
```

4



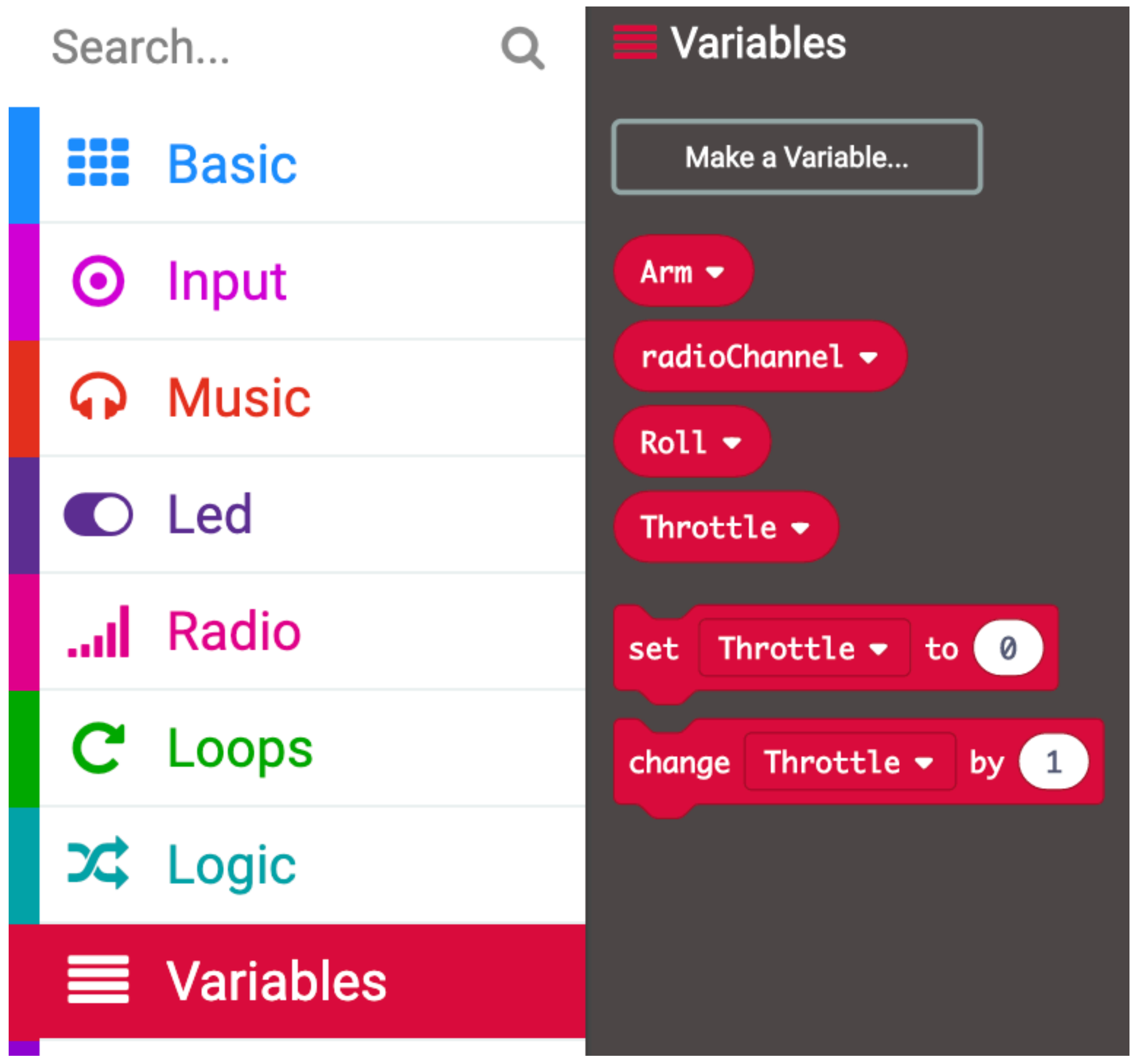
```
on start
    set RadioChannel to 7
    show number RadioChannel
    radio set group RadioChannel
fore
```

ART

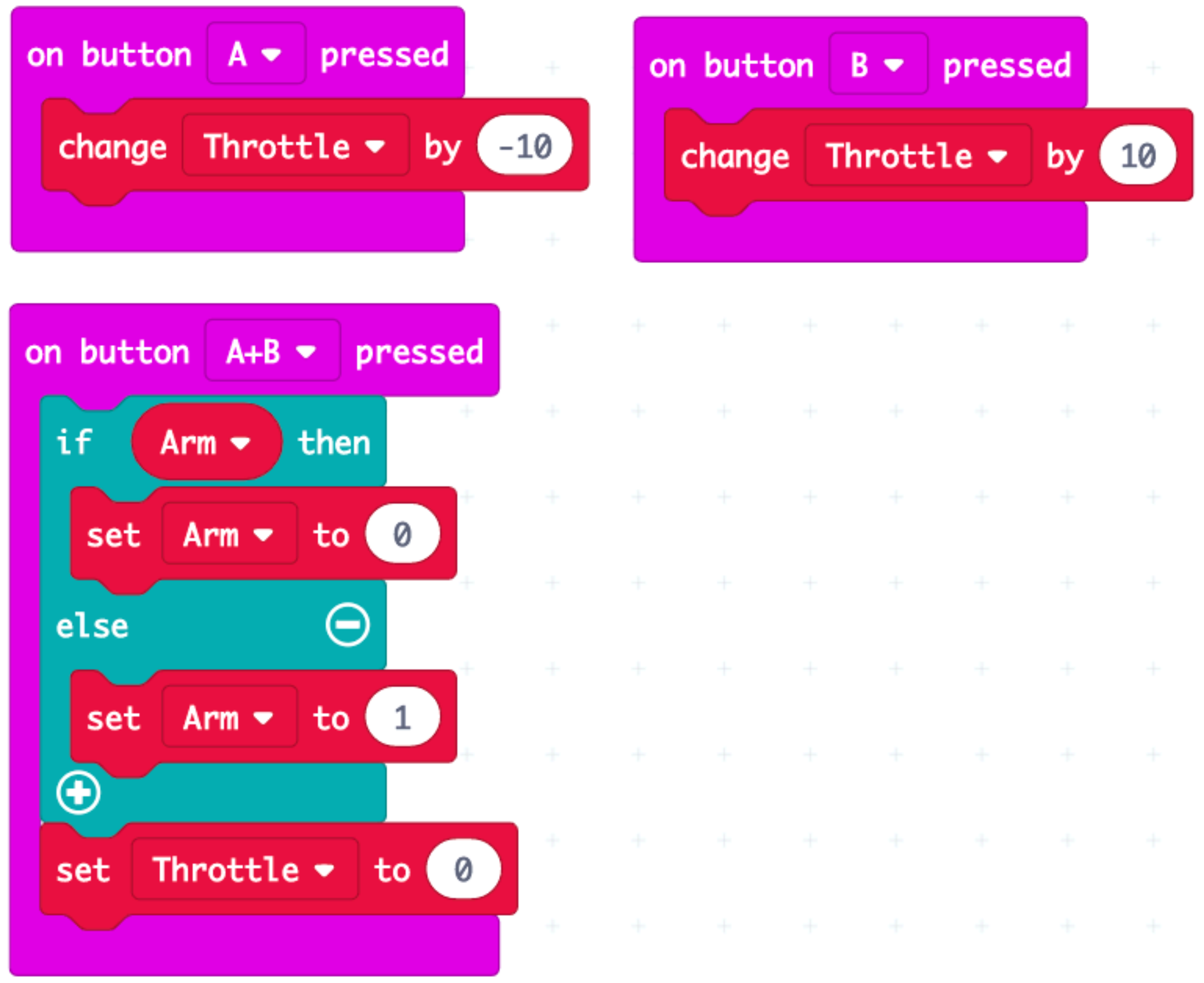
arm, roll, throttle

1. Make 3 variables called arm, roll and throttle
2. Use the button functions so button A makes throttle 10 (%) less, and button B makes it 10 more. Use “change”, not “set”
3. Use the buttons A + B (a combination) to change the Arm between 0 and 1 everytime A + B is pressed

1



2

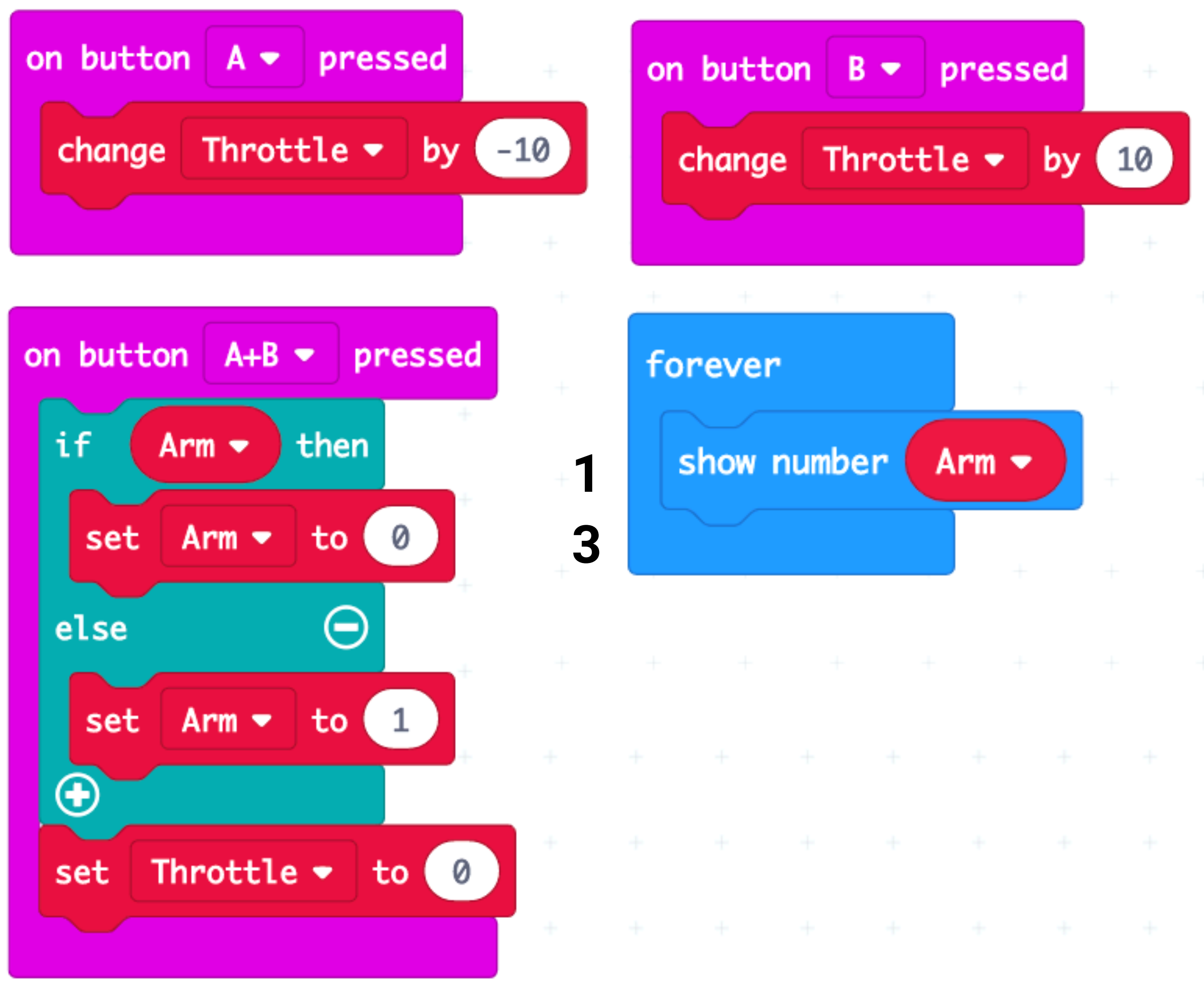


```
on button A pressed
    change Throttle by -10

on button B pressed
    change Throttle by 10

on button A+B pressed
    if Arm then
        set Arm to 0
    else
        set Arm to 1
    set Throttle to 0
```

Arm

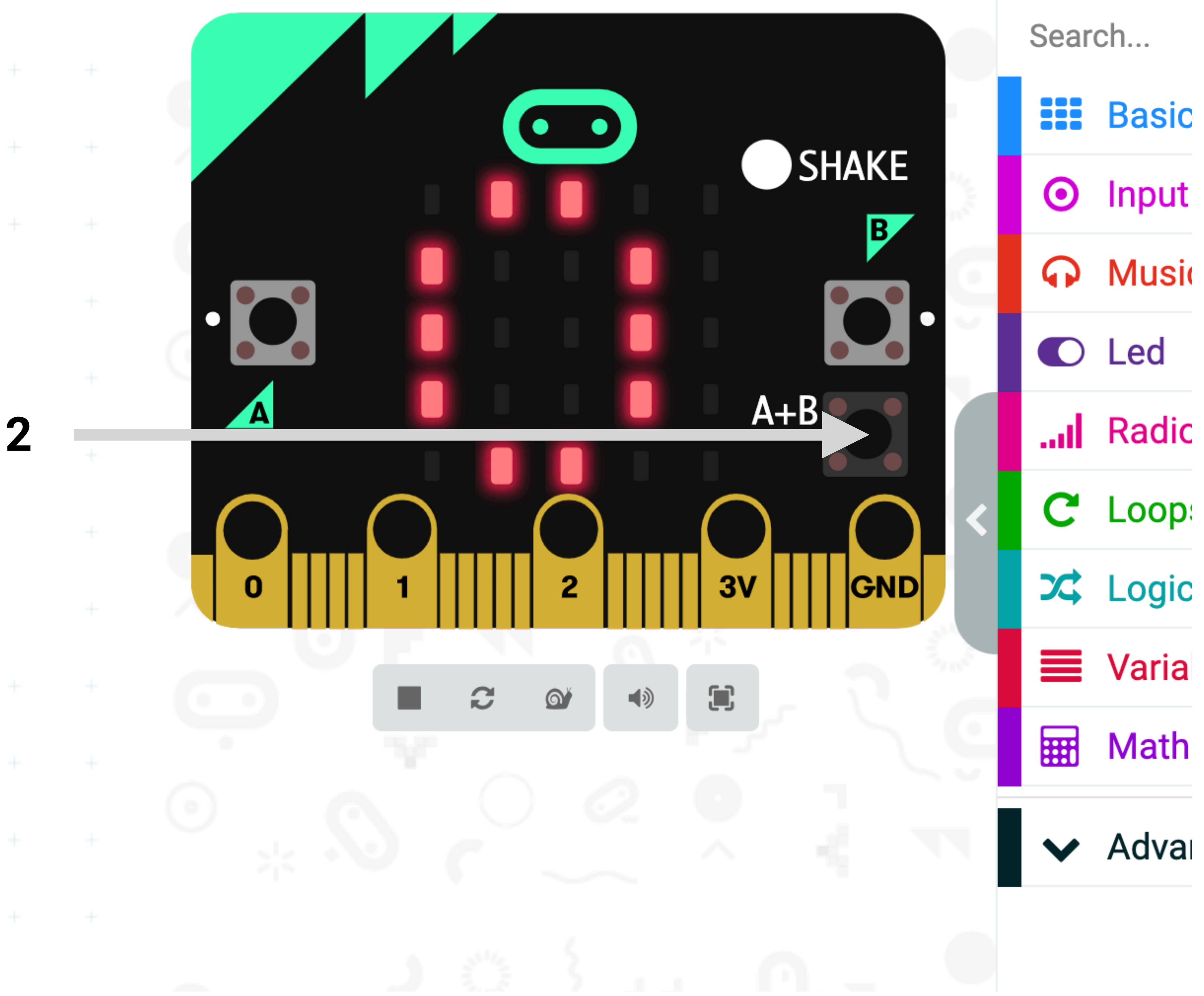
arm, roll, throttle

1. Put a Show Number (in the forever loop)
2. Use the simulator to test the A+B function (the number will switch between 0 and 1 and back)
3. Delete the same show number block when you have tested it. (use delete button or right click - delete)

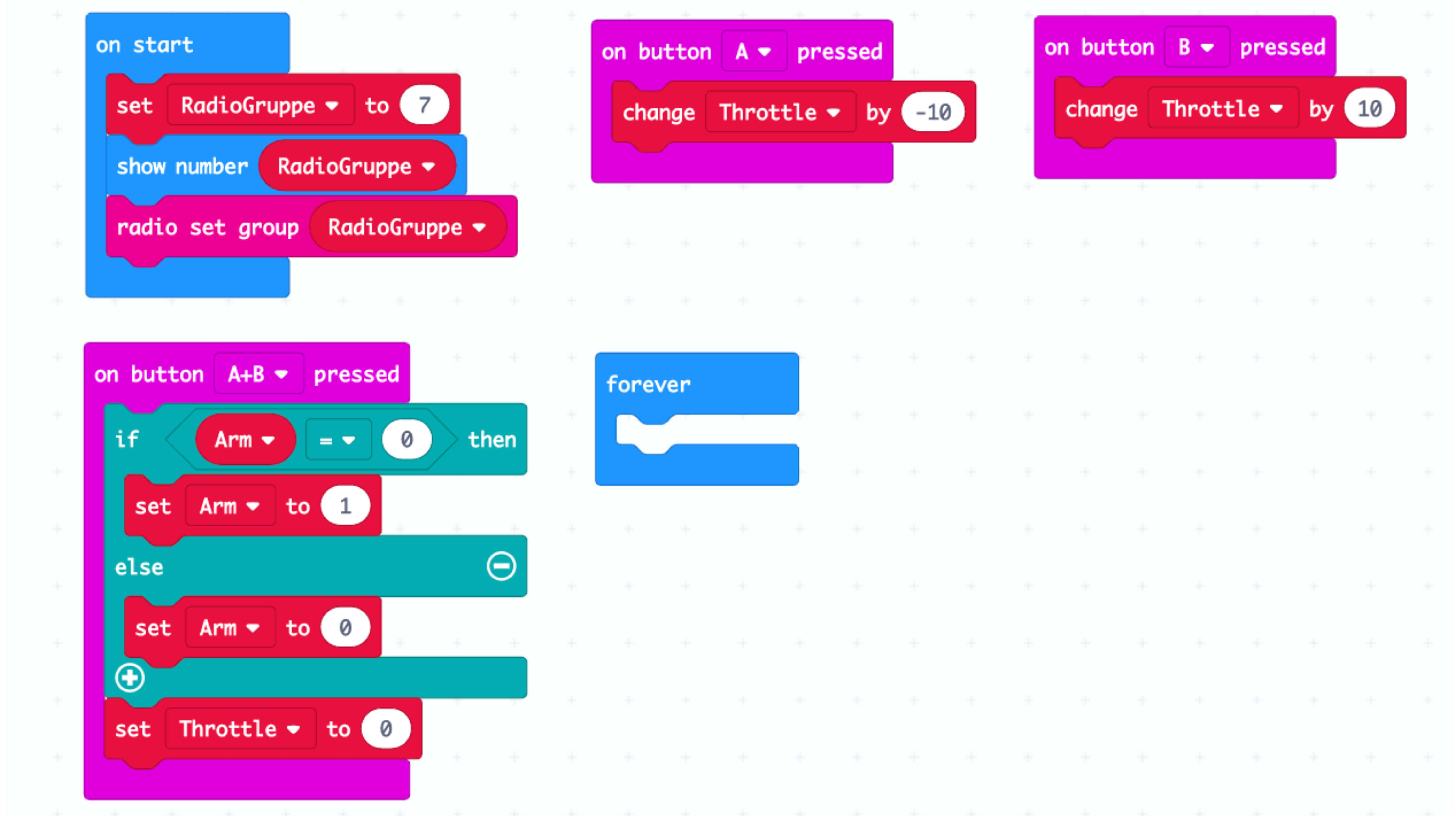


```
on button A pressed
    change Throttle by -10

on button B pressed
    change Throttle by 10

on button A+B pressed
    if Arm then
        set Arm to 0
    else
        set Arm to 1
    set Throttle to 0

forever
    show number Arm
```

Simulator



This is the code so far



```
on start
    set RadioGruppe to 7
    show number RadioGruppe
    radio set group RadioGruppe
```

```
on button A pressed
    change Throttle by -10
```

```
on button B pressed
    change Throttle by 10
```

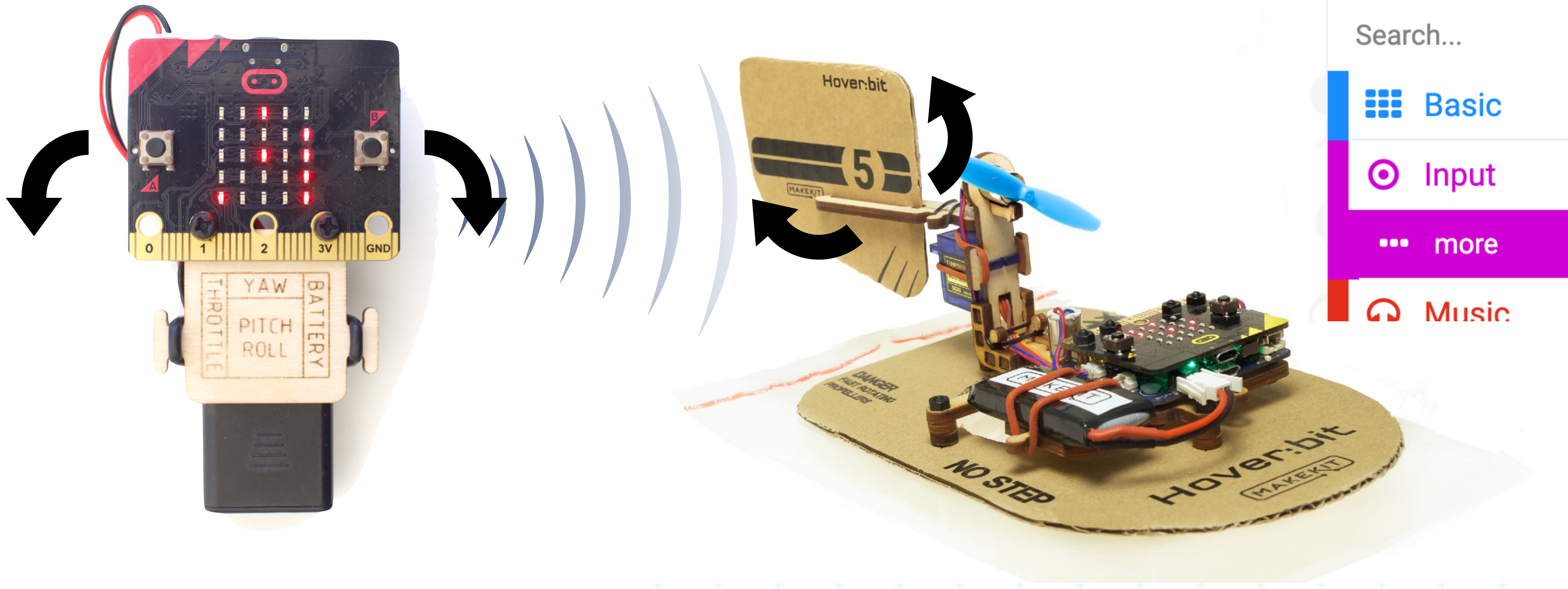
```
on button A+B pressed
    if Arm = 0 then
        set Arm to 1
    else
        set Arm to 0
    set Throttle to 0
```

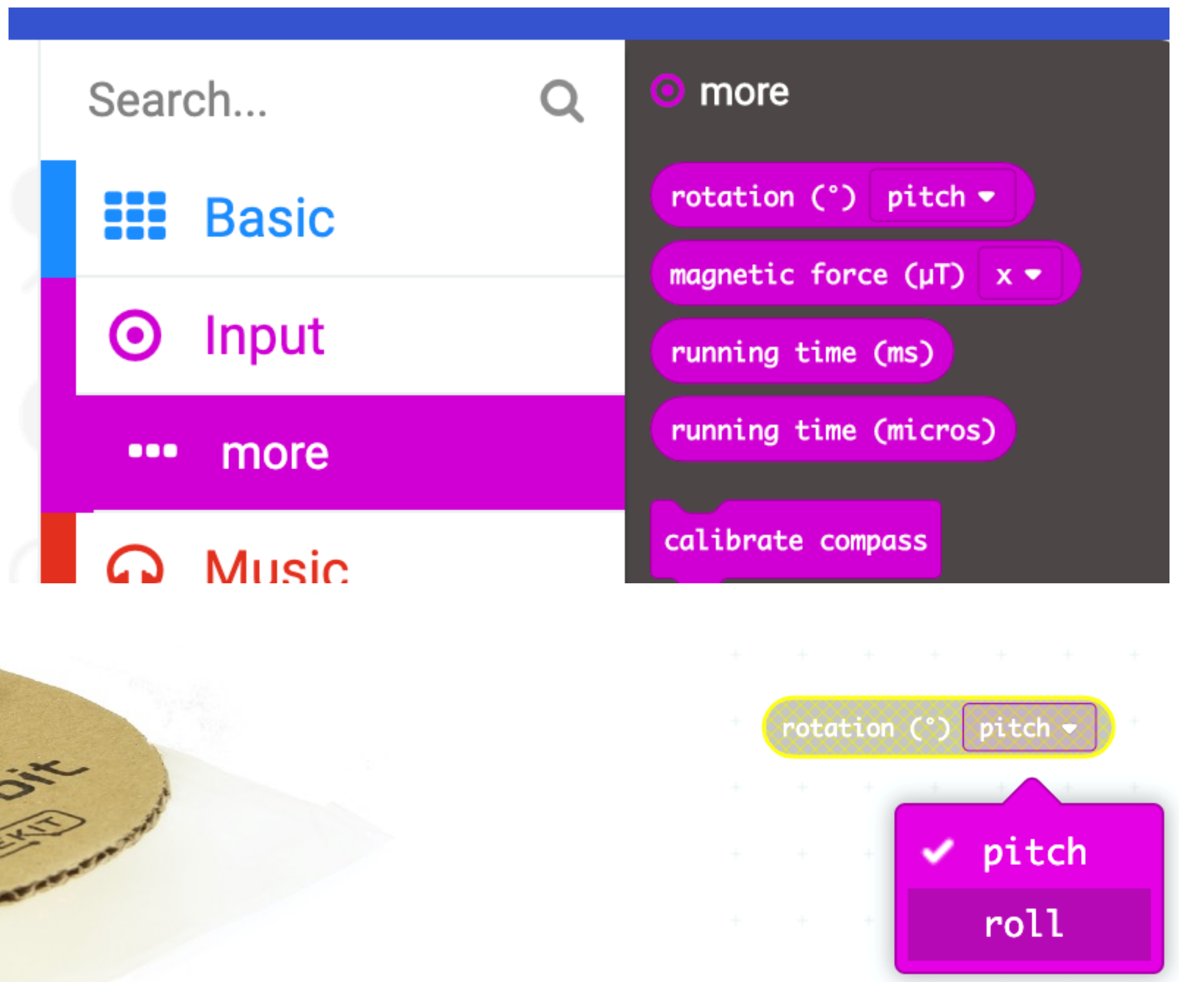
```
forever
```

Roll and rudder

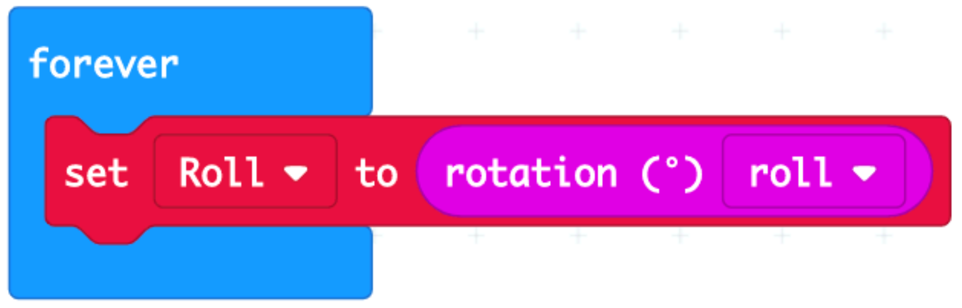
We want to control the hovercraft´s steering by using the orientation sensor on the micro:bit. This is called roll. This will control the rudder on the hovercraft.

Task: In the forever block, set the roll variable to the rotation roll. The block is called “rotation pitch”. Drag it out and change it to “roll” by clicking the small triangle at the right.





This is what you should end up with:

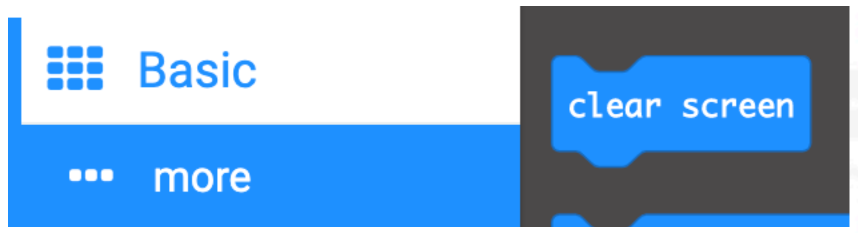


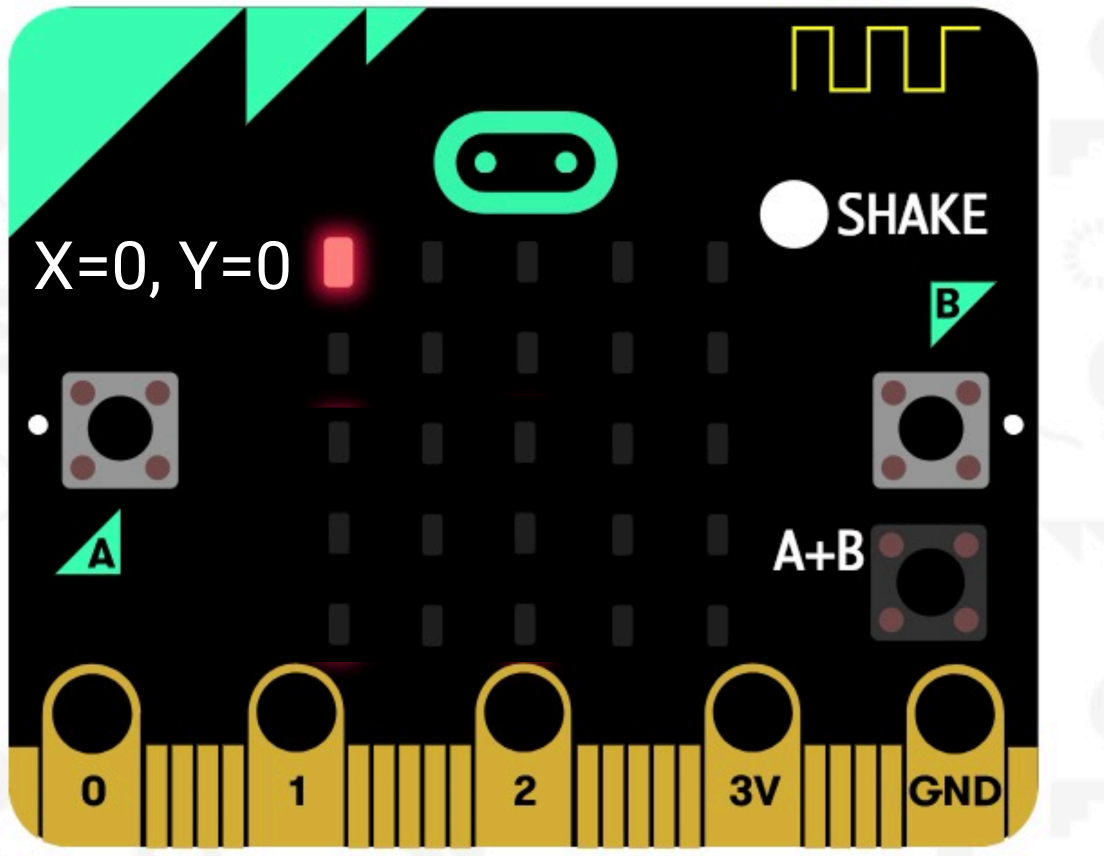
```
forever
  set Roll to rotation (°) roll
```

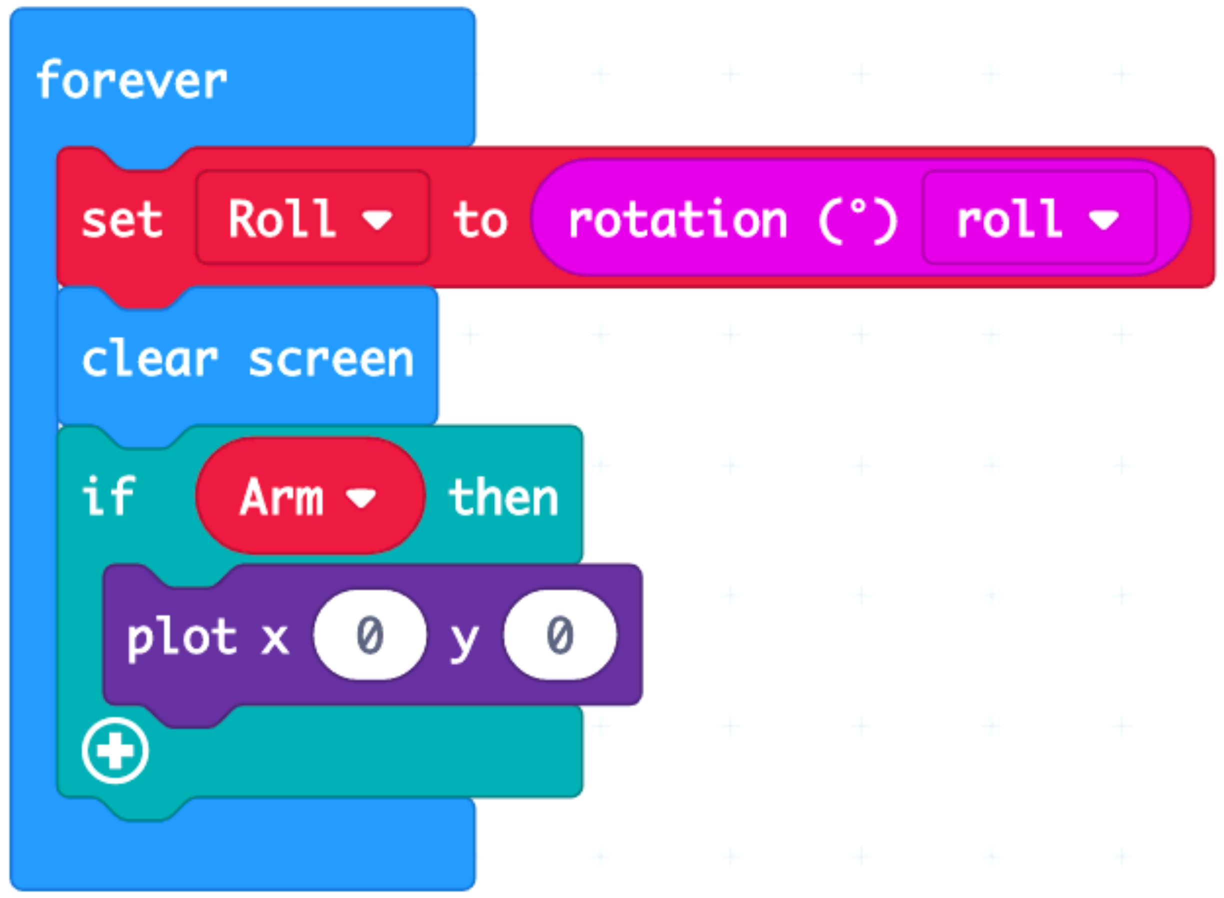
Display Arm

In the beginning of our code, our radio channel will be shown.
We want to use the display to also show our Arm, Roll and Throttle values as they change.
Before we plot, we use clear screen to prevent the screen to get filled with pixels.

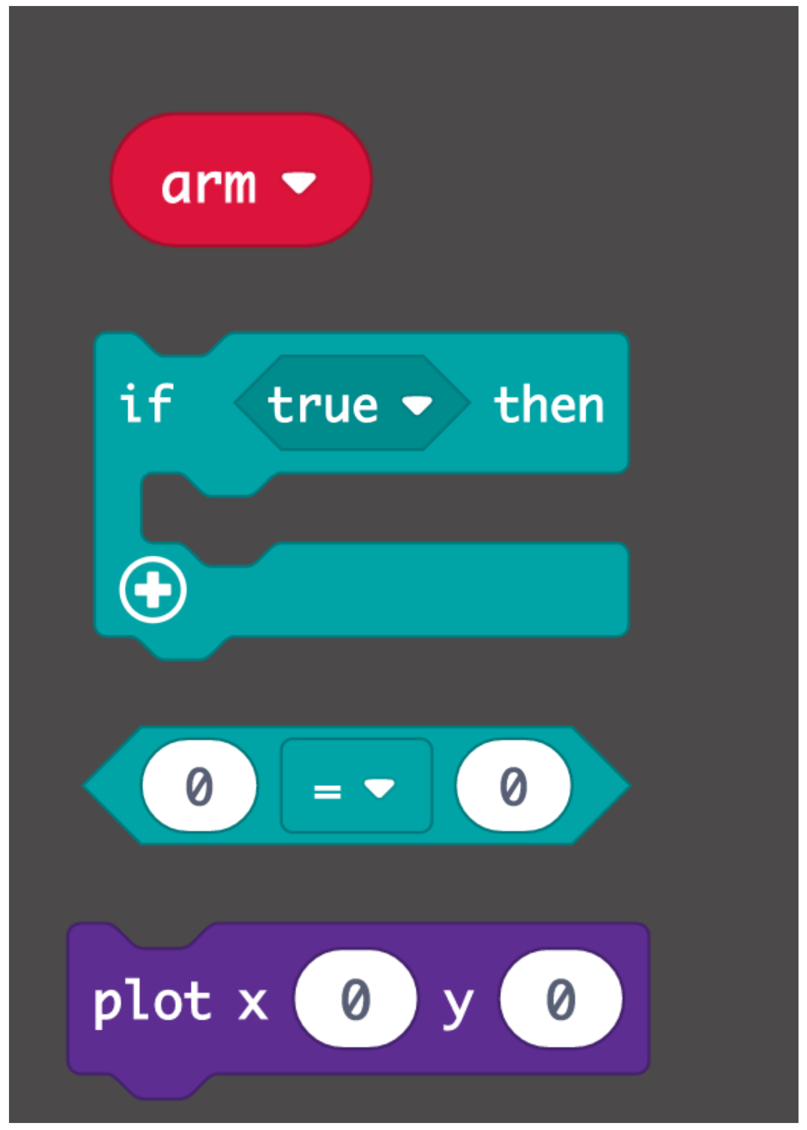
Show arming
Find the forever-block.
Insert a clear screen block under the roll block.
Use the if-block to check if arm is not zero (armed).
If armed plot at coordinate 0,0 (or another place of your choice)





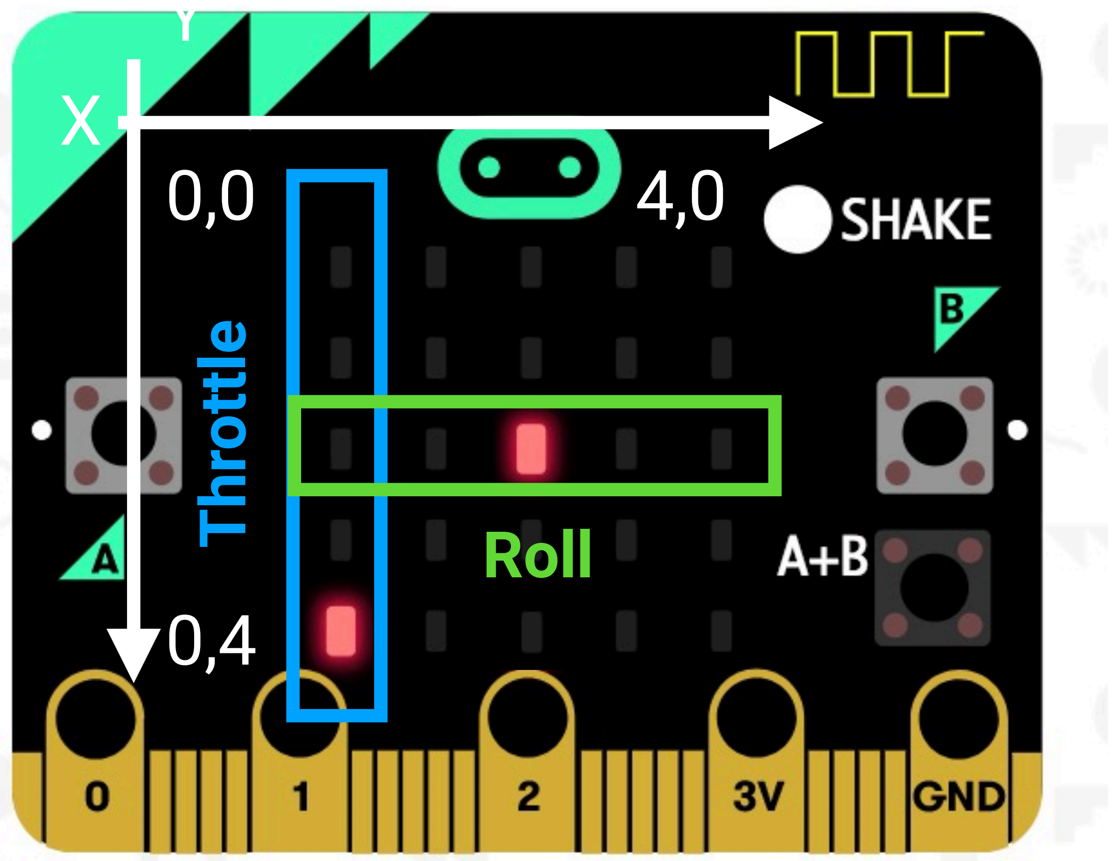


```
forever
  set Roll to rotation (°) roll
  clear screen
  if Arm then
    plot x 0 y 0
```



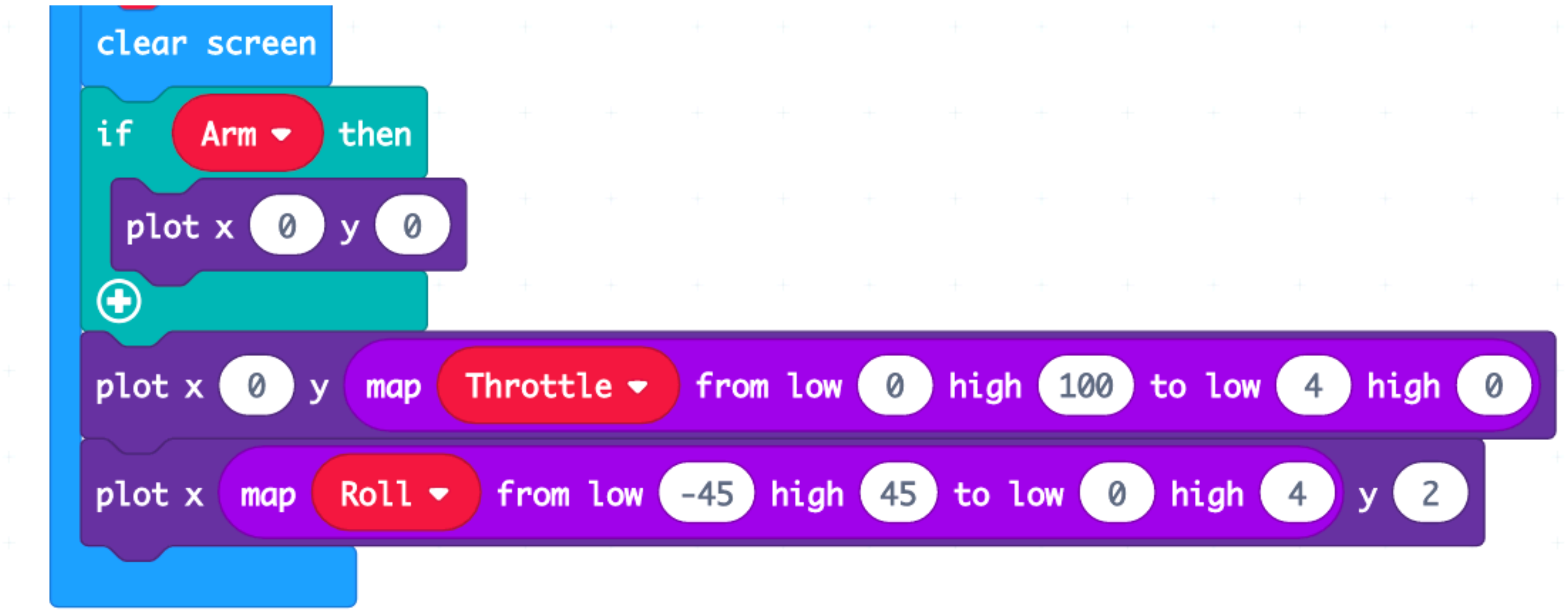
Optional: Display Roll and Throttle

We also want to show roll and throttle in the display. Use the blocks below to convert roll and throttle into values that can be plotted on the screen. We want the throttle pixel to move upwards, starting at coordinate 0,4, then climb towards 0,0
We want the roll pixel to slide across the screen, from 0,2 (middle left) to 4,2



Show Throttle
Continue with the forever-block.
Insert a plot-block. In the y-section, insert a map block.
In the map block, insert the Throttle variable, then 0,100,4,0.
This will take a number between 0 and 100, and squeeze it down to a number between 4 and zero.

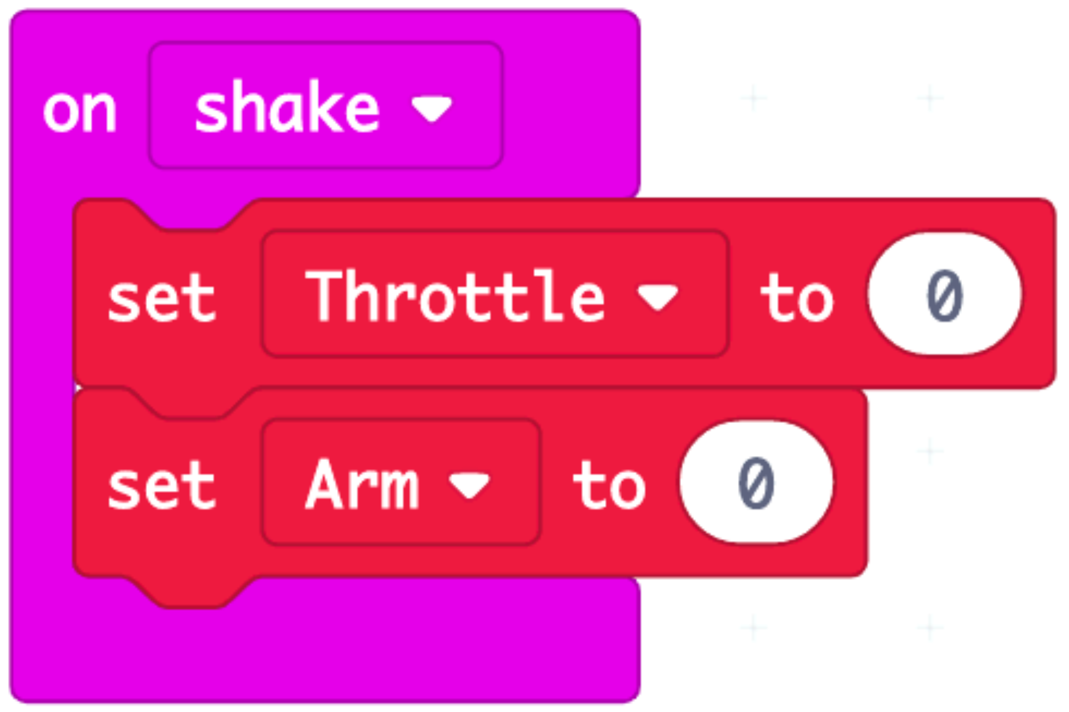
Show Roll
Continue with the forever-block.
Insert a plot-block. In the y-section, insert a map block.
In the map block, insert the Roll variable, then -45,45,0,4.
This will take a number between -45 and 45 (roll degrees), and squeeze it down to a number between zero and 4



```
clear screen
if Arm then
    plot x 0 y 0
plot x 0 y map Throttle from low 0 high 100 to low 4 high 0
plot x map Roll from low -45 high 45 to low 0 high 4 y 2
```

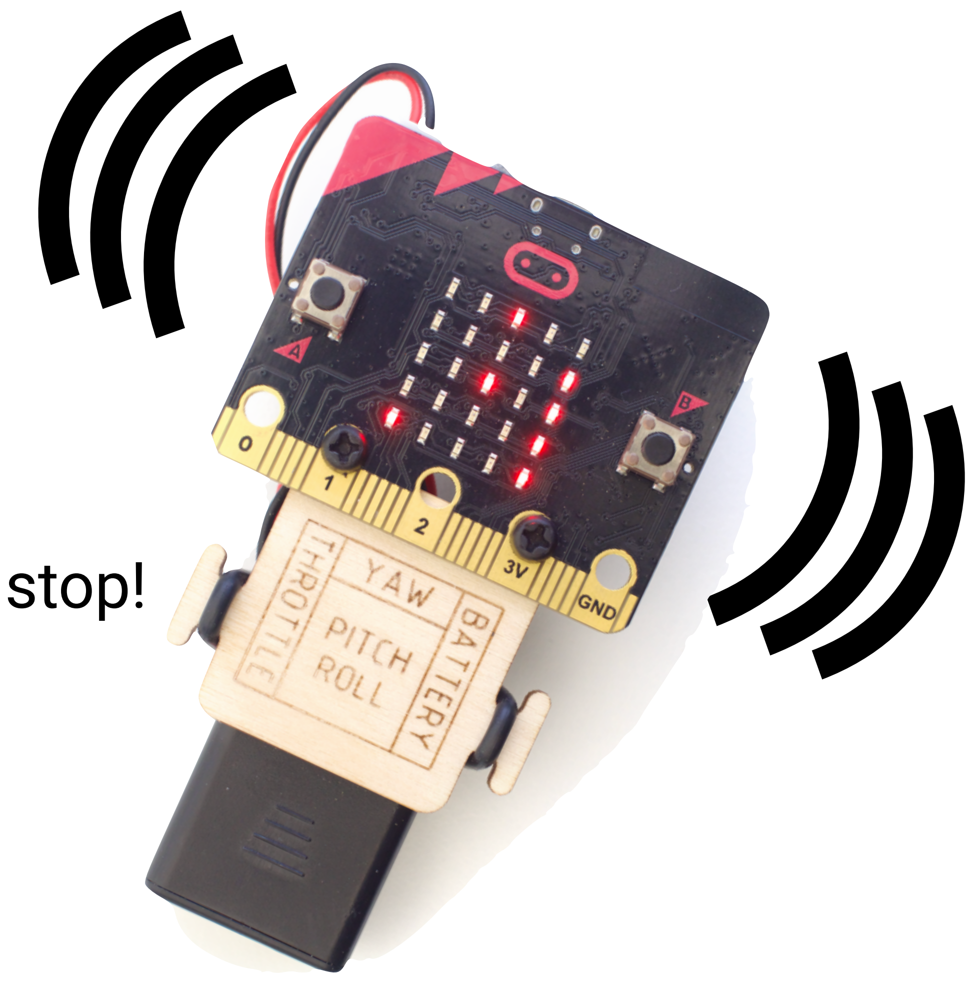
Emergency stop

Make this little code to create a quick and effective way to stop your hover:bit. All you need to do is to shake your controller and the motor will stop.



```
on shake
    set Throttle to 0
    set Arm to 0
```

Shake to stop!

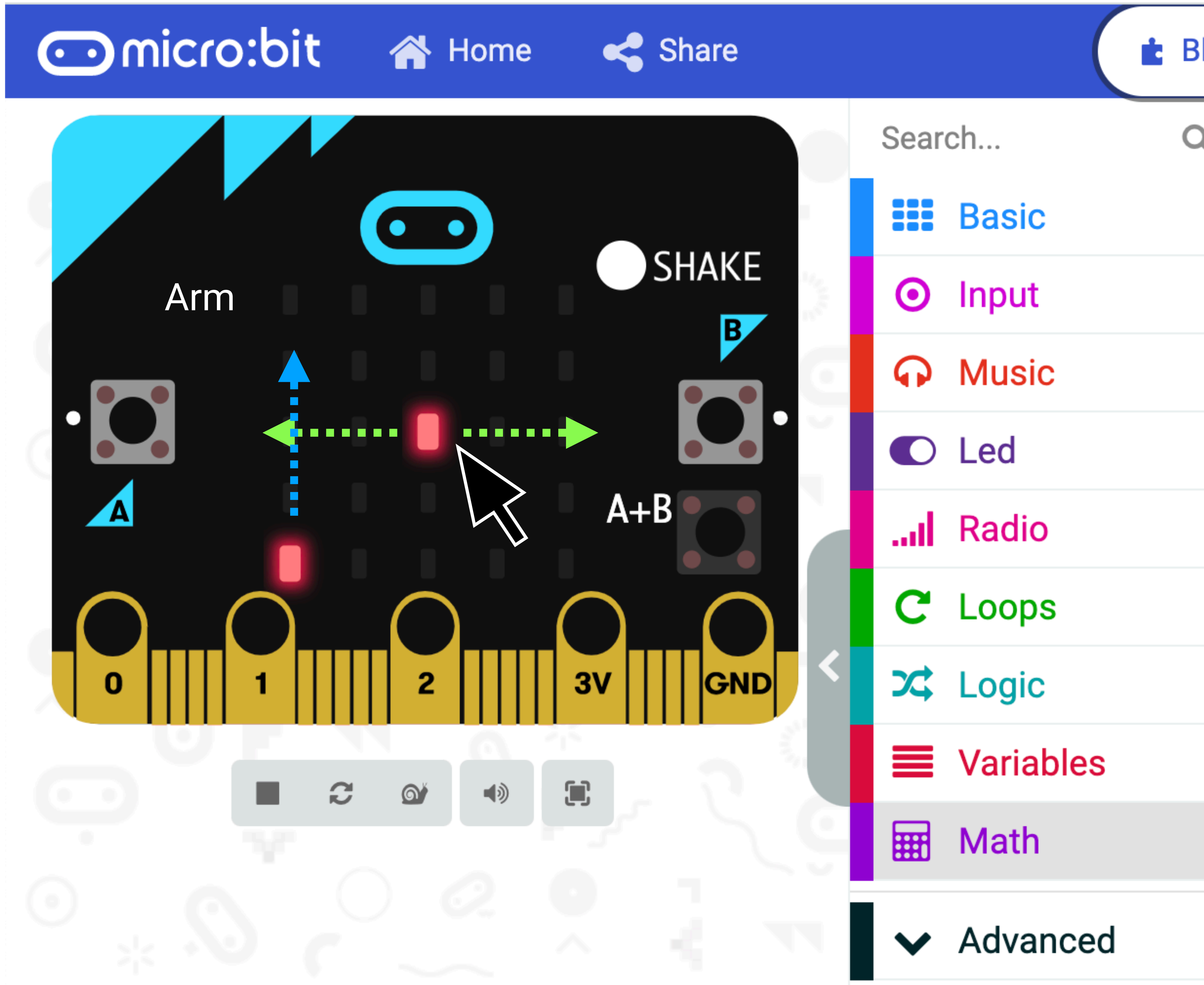


Test it!

Use the simulator (left side of make:code) to test your code.

- Press the A+B button to make the arming light to turn on (top left on your screen)
- Press “Shake” to simulate a shake that will turn off arm light.

- If you added the roll/throttle light, move the mouse cursor sideways over the micro:bit. Make sure the dot moves along your mouse arrow. (Green span)
- Press B button numerous times. Verify that the throttle is climbing upwards as in the blue span.

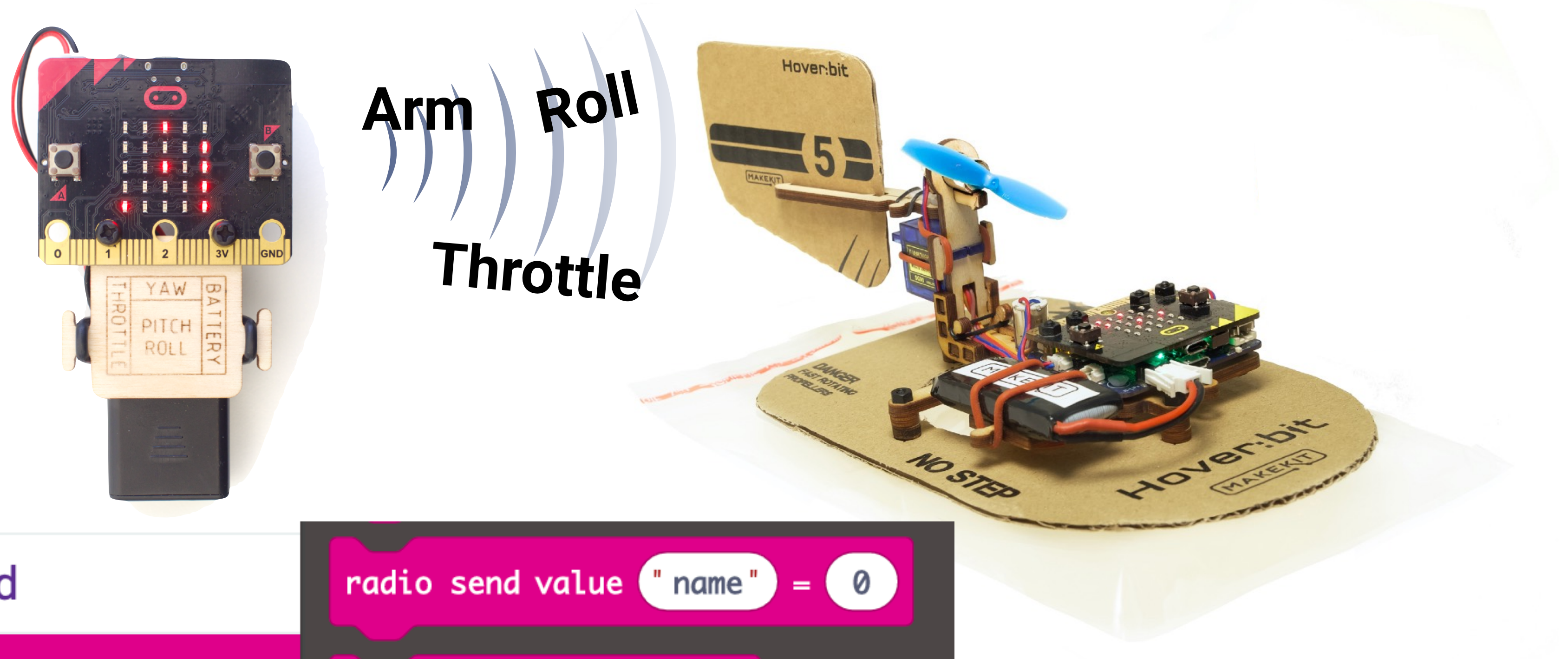


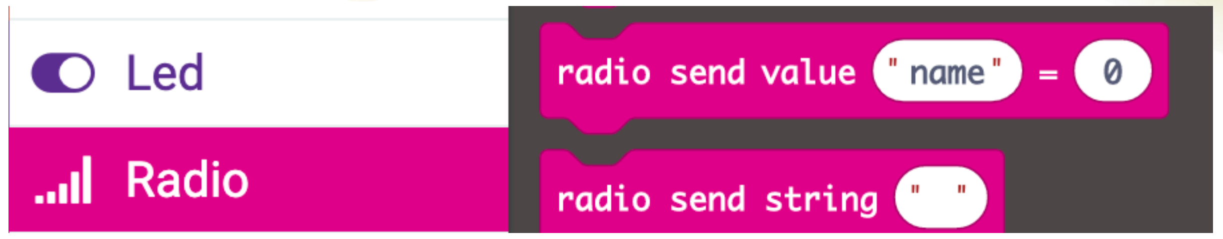
Send the values over radio

To make our remote control work wirelessly, we need to use the radio to transmit our ART-values. They will be sent as separate numbers, with a little name tag on them so the receiver can tell them apart.

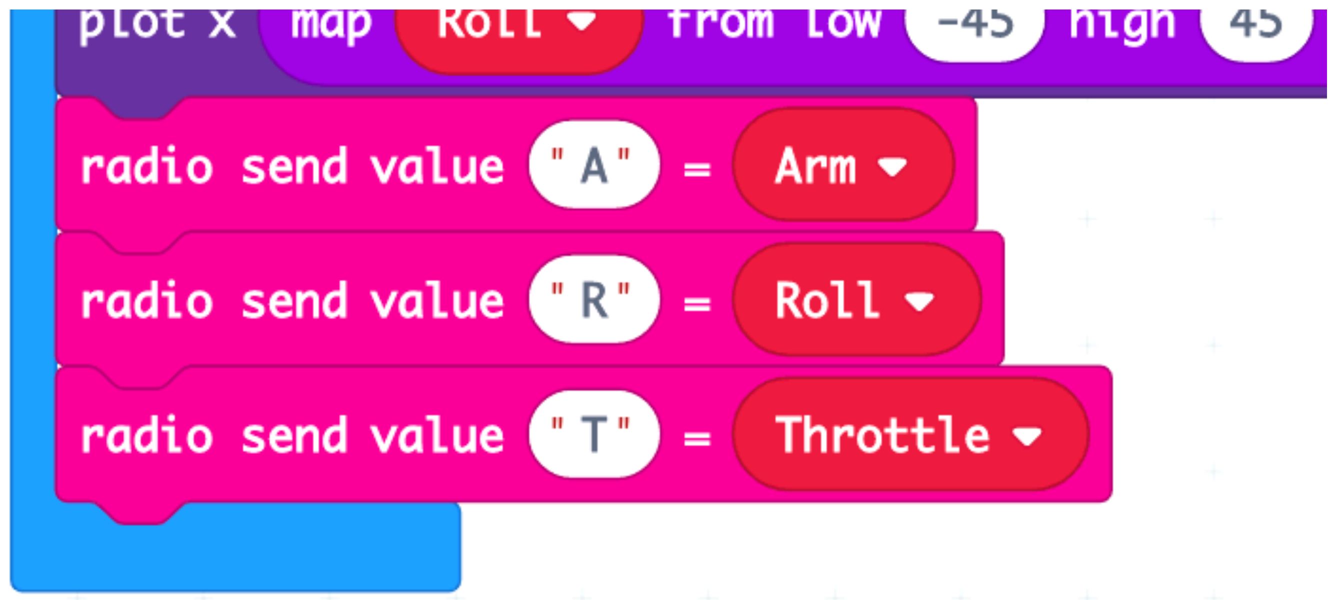
At the bottom of the Forever-loop:

- Use the radio send value = 0 block
- Make one block where you send the letter “A” (must be capital) together with the Arm value.
- Do the same with Roll (R) and throttle (T)





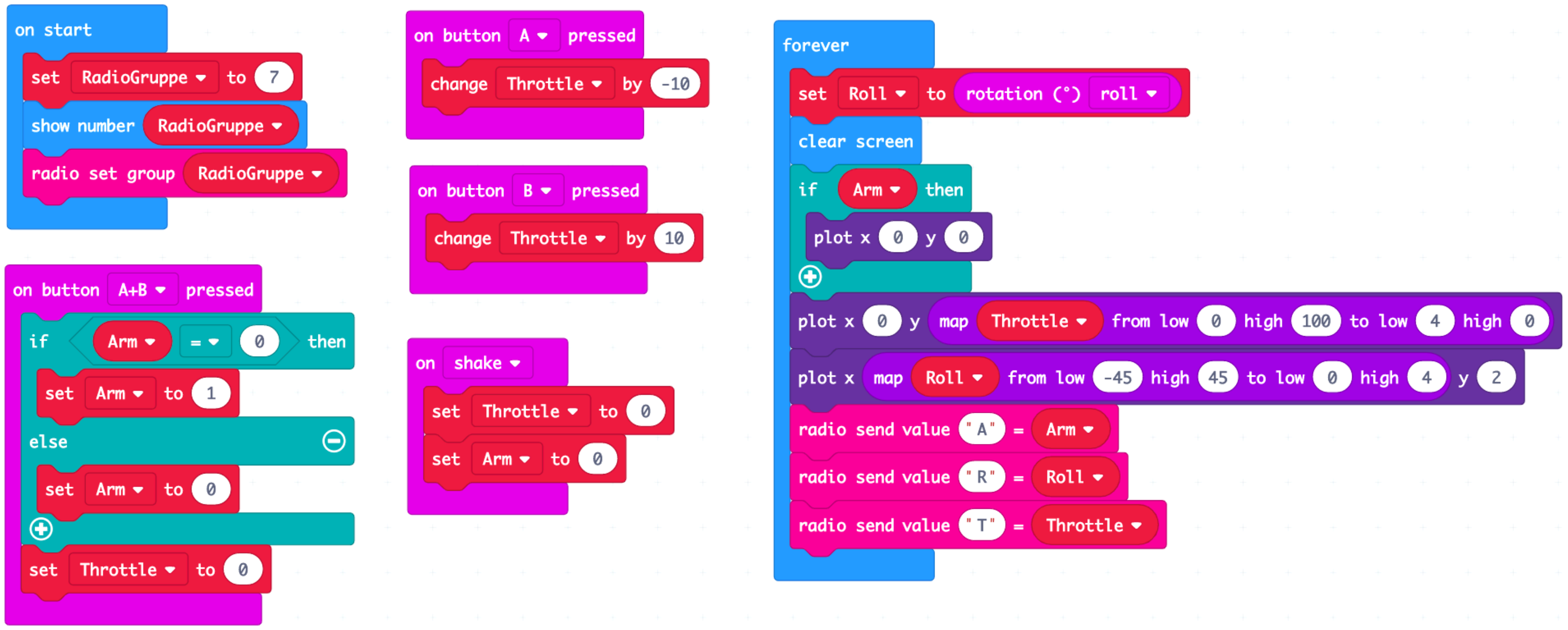
```
radio send value "name" = 0
radio send string " "
```



```
plot x map Roll from low -45 high 45
radio send value "A" = Arm
radio send value "R" = Roll
radio send value "T" = Throttle
```

Summary

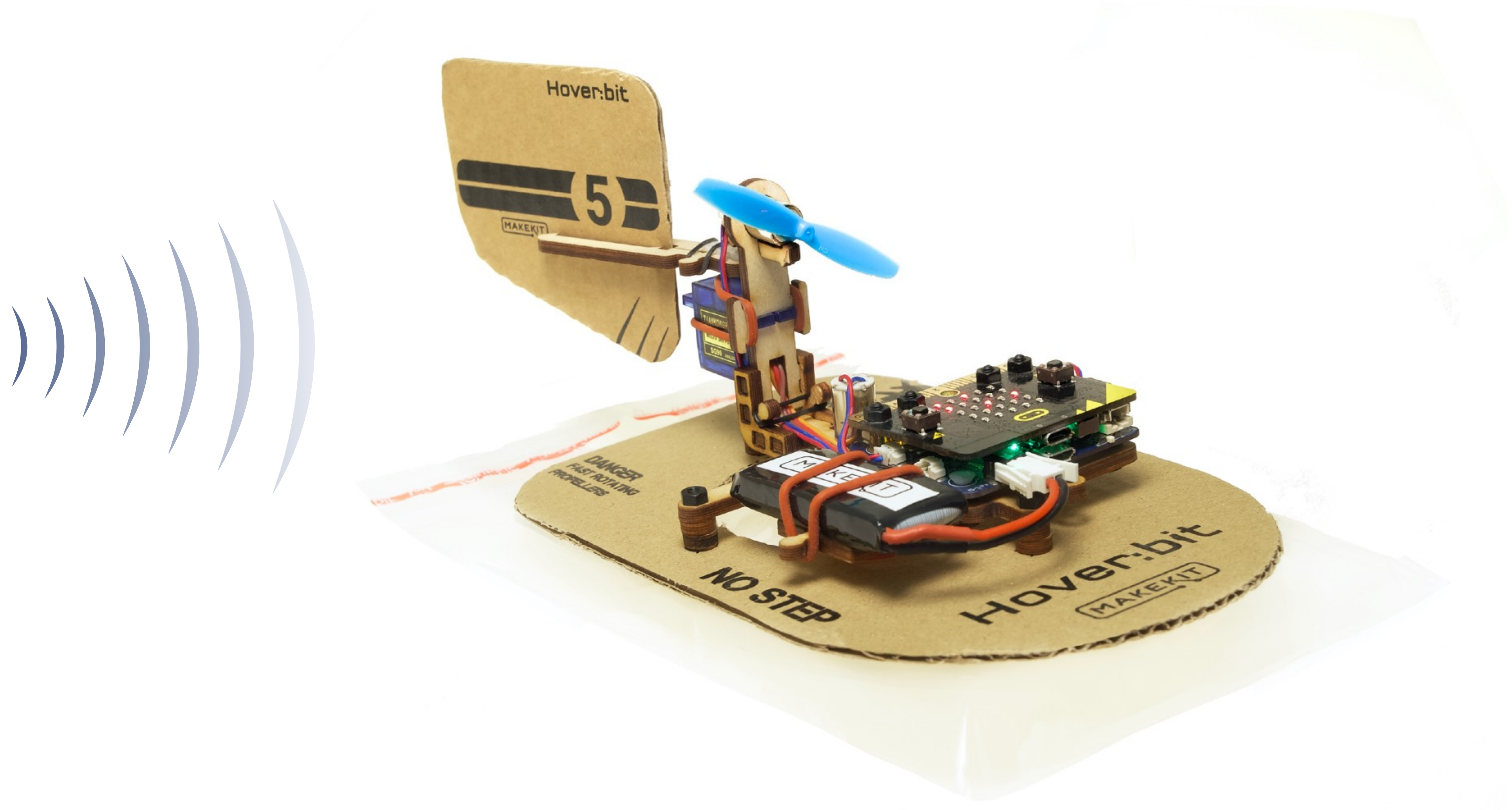
This is the full code



```
on start
    set RadioGruppe to 7
    show number RadioGruppe
    radio set group RadioGruppe

on button A+B pressed
    if Arm = 0 then
        set Arm to 1
    else
        set Arm to 0
    set Throttle to 0

on button A pressed
    change Throttle by -10

on button B pressed
    change Throttle by 10

on shake
    set Throttle to 0
    set Arm to 0

forever
    set Roll to rotation (°) roll
    clear screen
    if Arm then
        plot x 0 y 0
    plot x 0 y map Throttle from low 0 high 100 to low 4 high 0
    plot x map Roll from low -45 high 45 to low 0 high 4 y 2
    radio send value "A" = Arm
    radio send value "R" = Roll
    radio send value "T" = Throttle
```

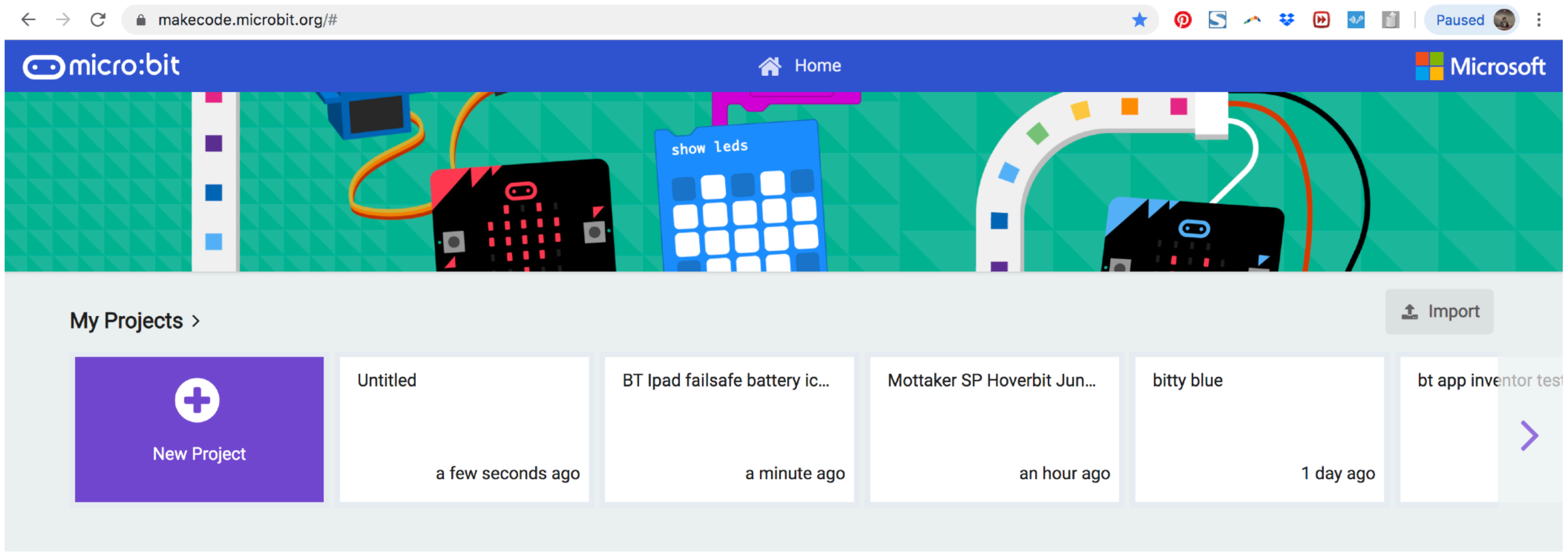
Receiver code

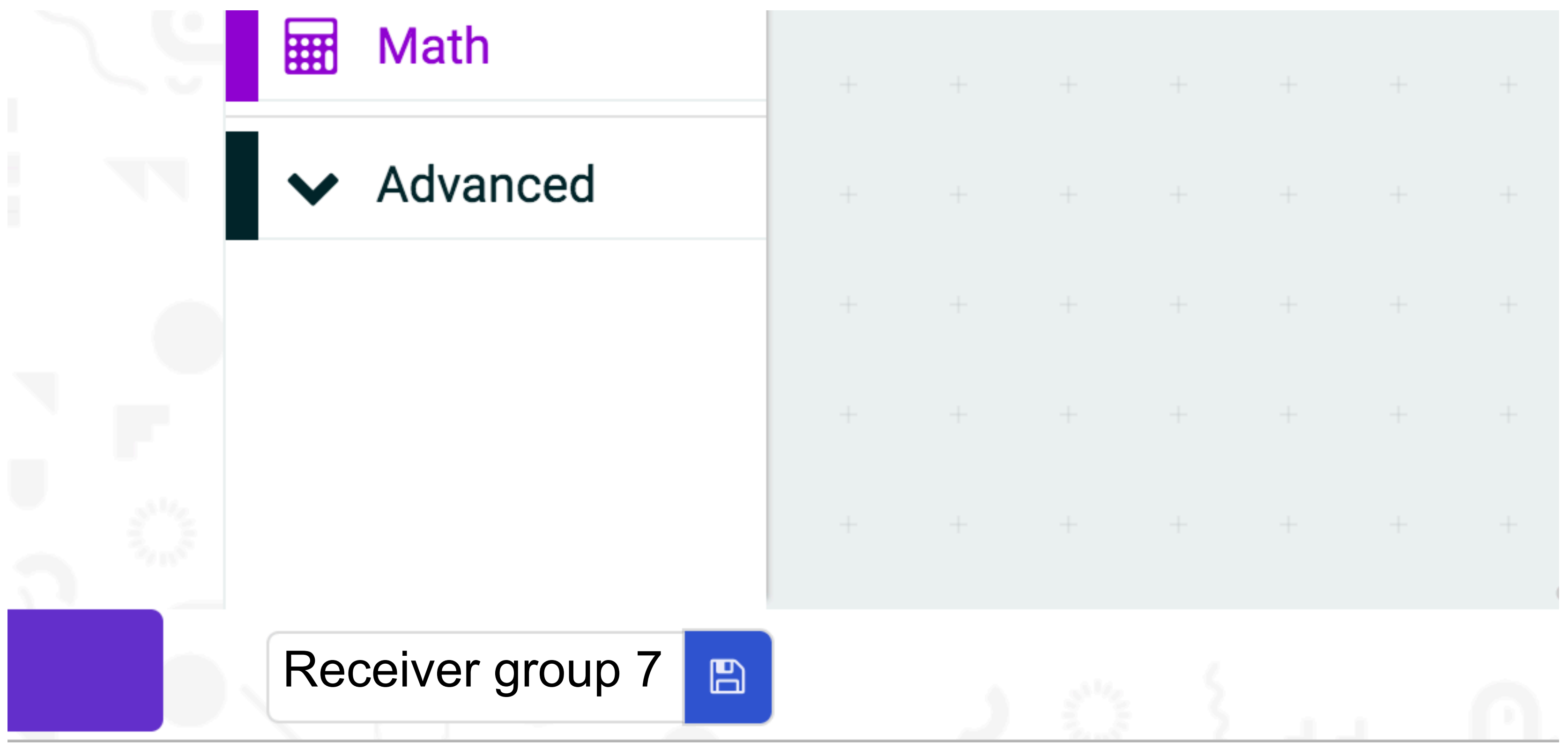
We create a code that will run onboard the micro:bit on the hover:craft



Create a new code

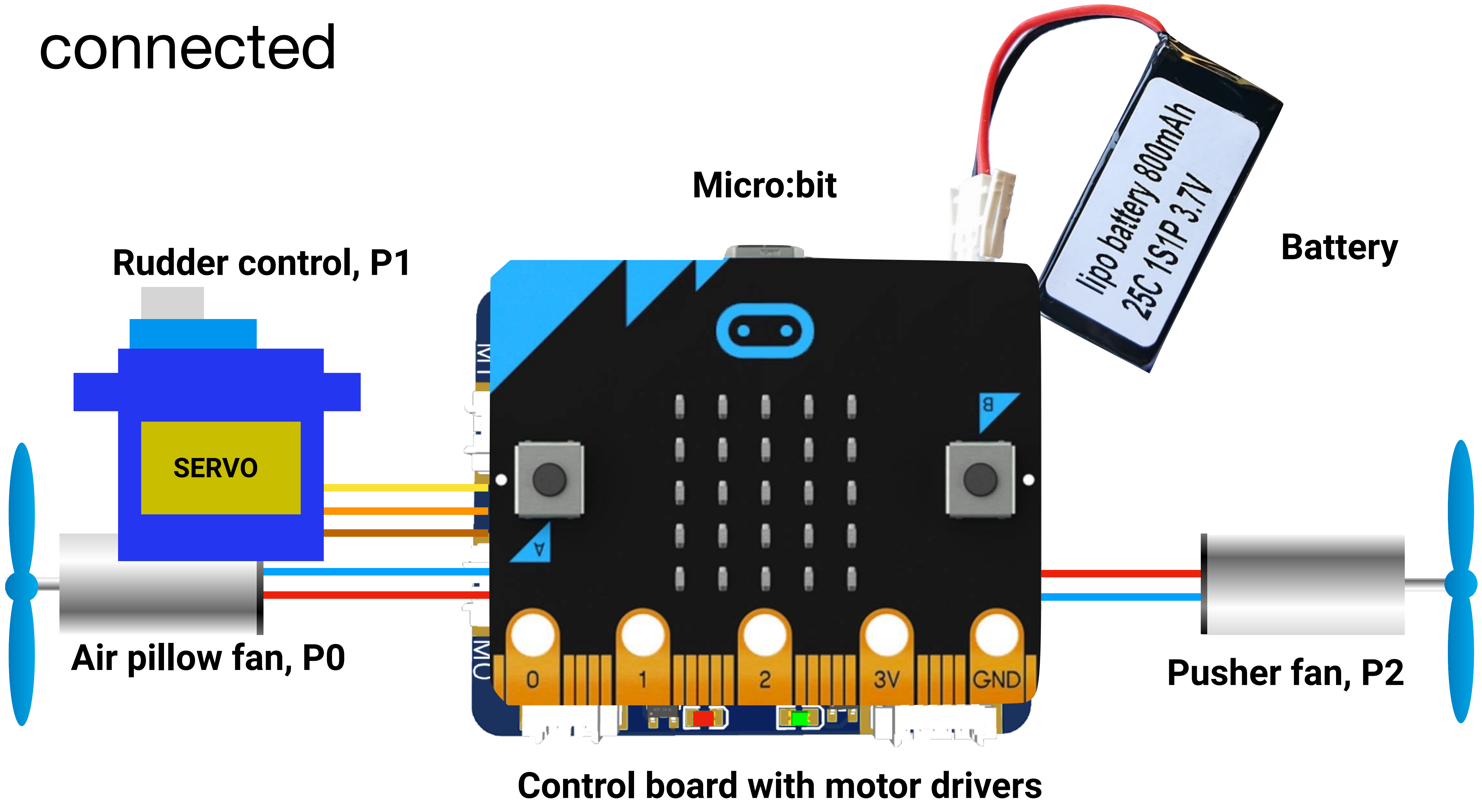
- Press the “micro:bit sign” top left to get back to the file menu.
- Enter “new project”
- Give your new project a name like “receiver group x”





Connections (reminder)

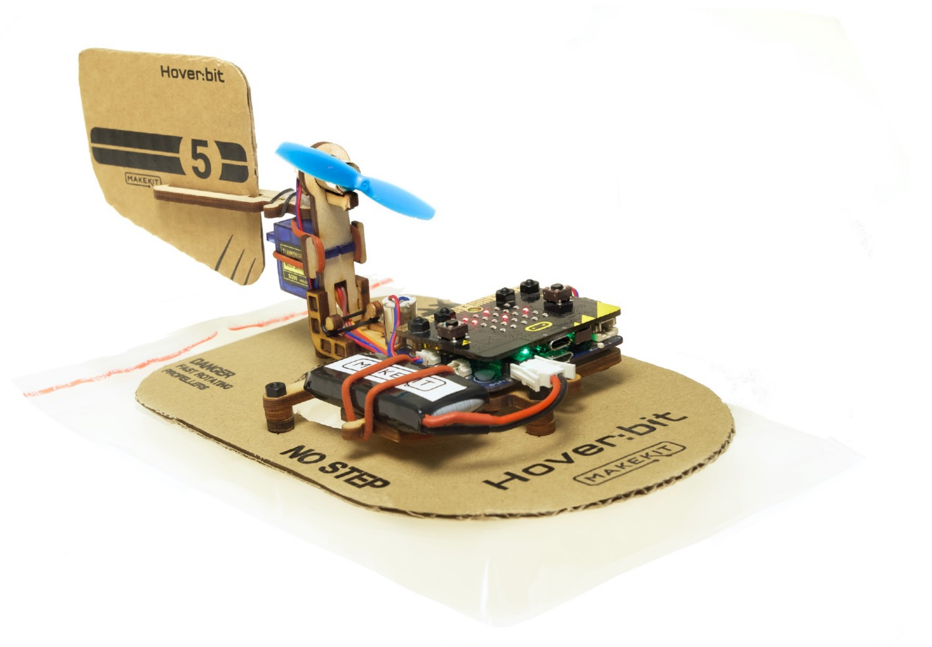
A reminder on how the motors and servo is connected

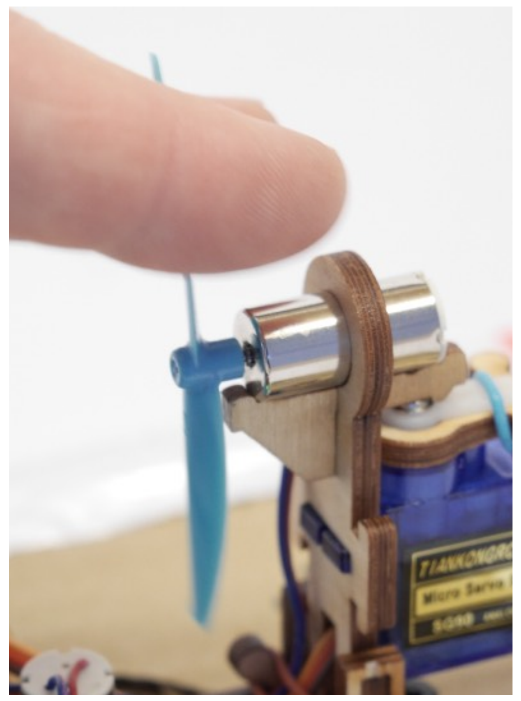
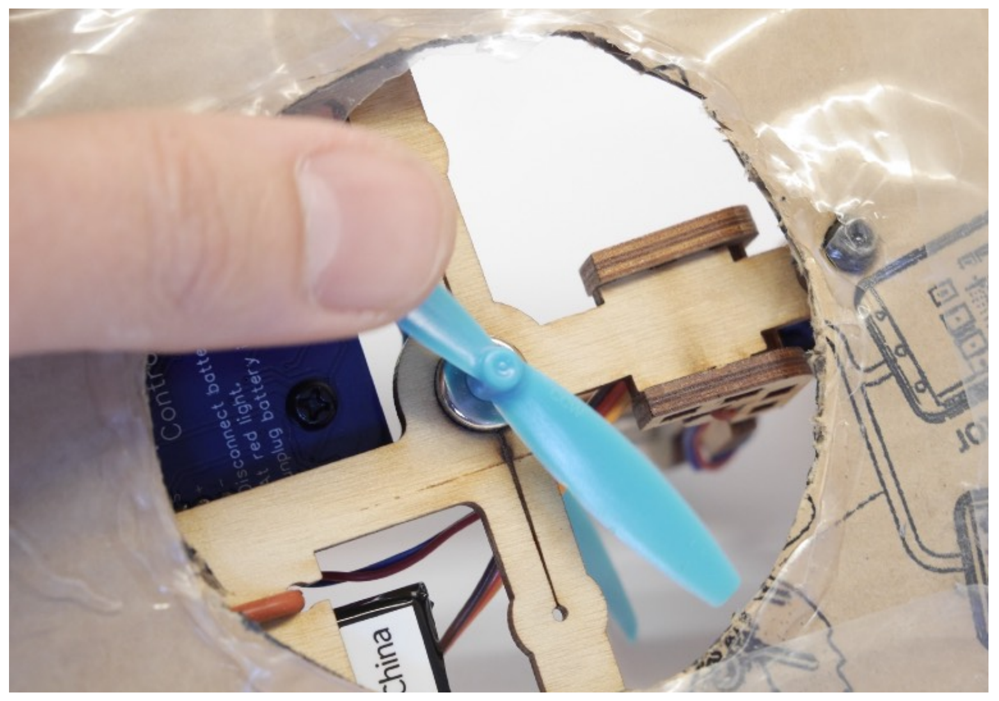


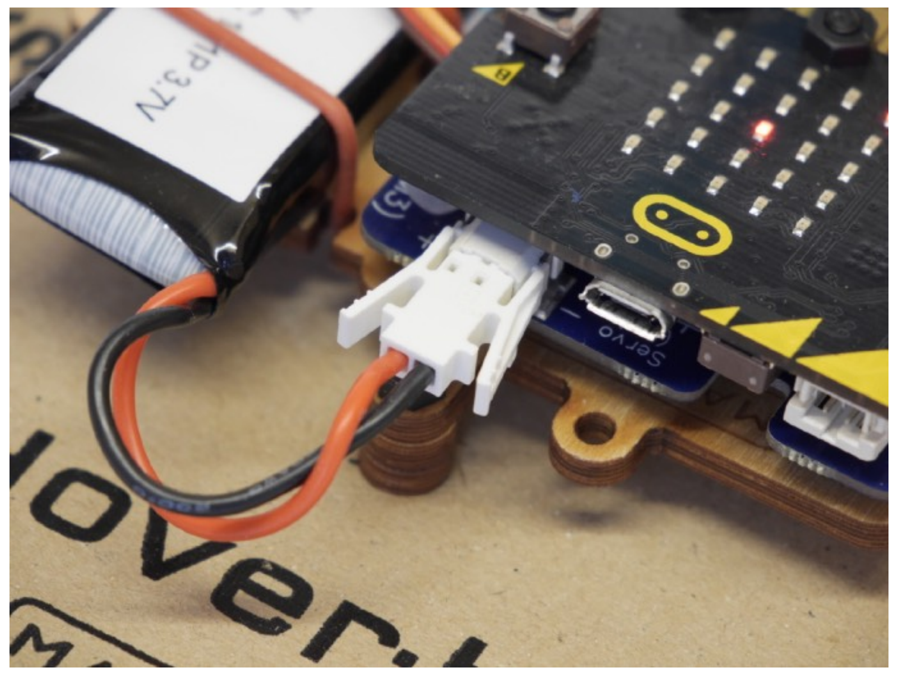
The micro:bit is receiving the signal, and controls the servo on pin P1, pillow fan on P0 and pusher fan from P2. The signals gets amplified with the control board and sent to the motors and servo.

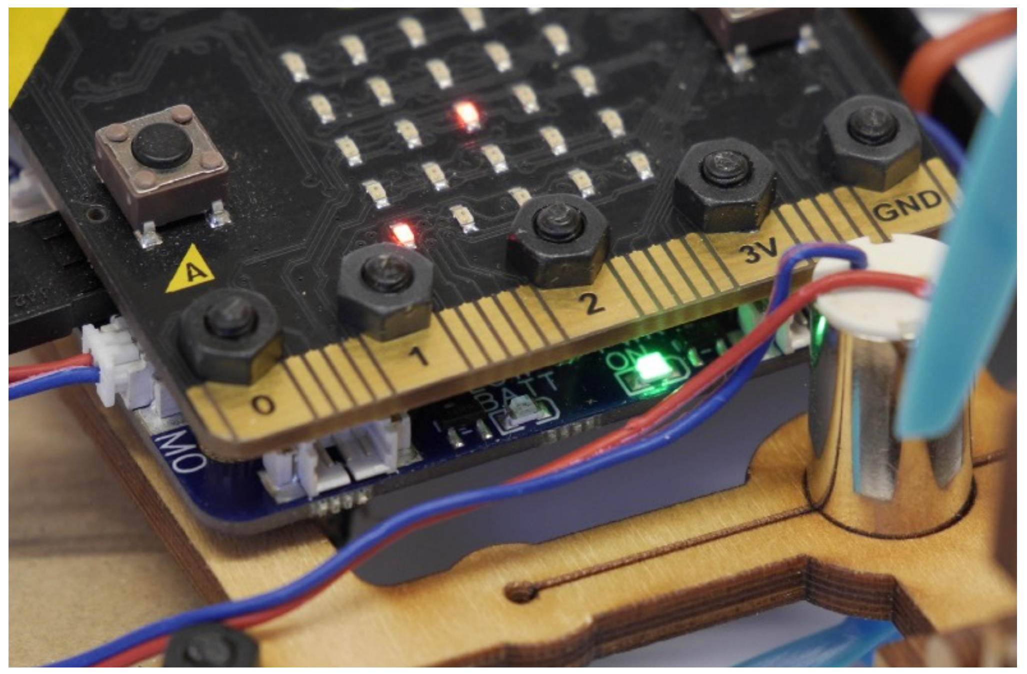
MAKEKIT

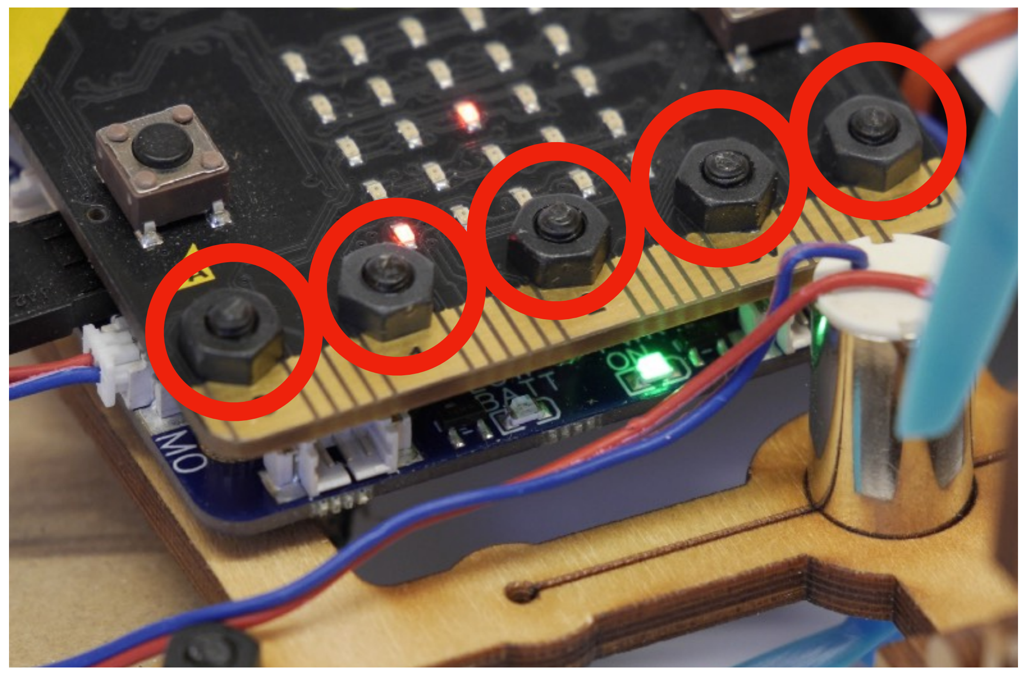
Prepare for testing


Find your assembled hovercraft.



Spin the motors and make sure no cable or cardboard can be hit by propellers. This is important, to prevent damage.


Connect the battery.

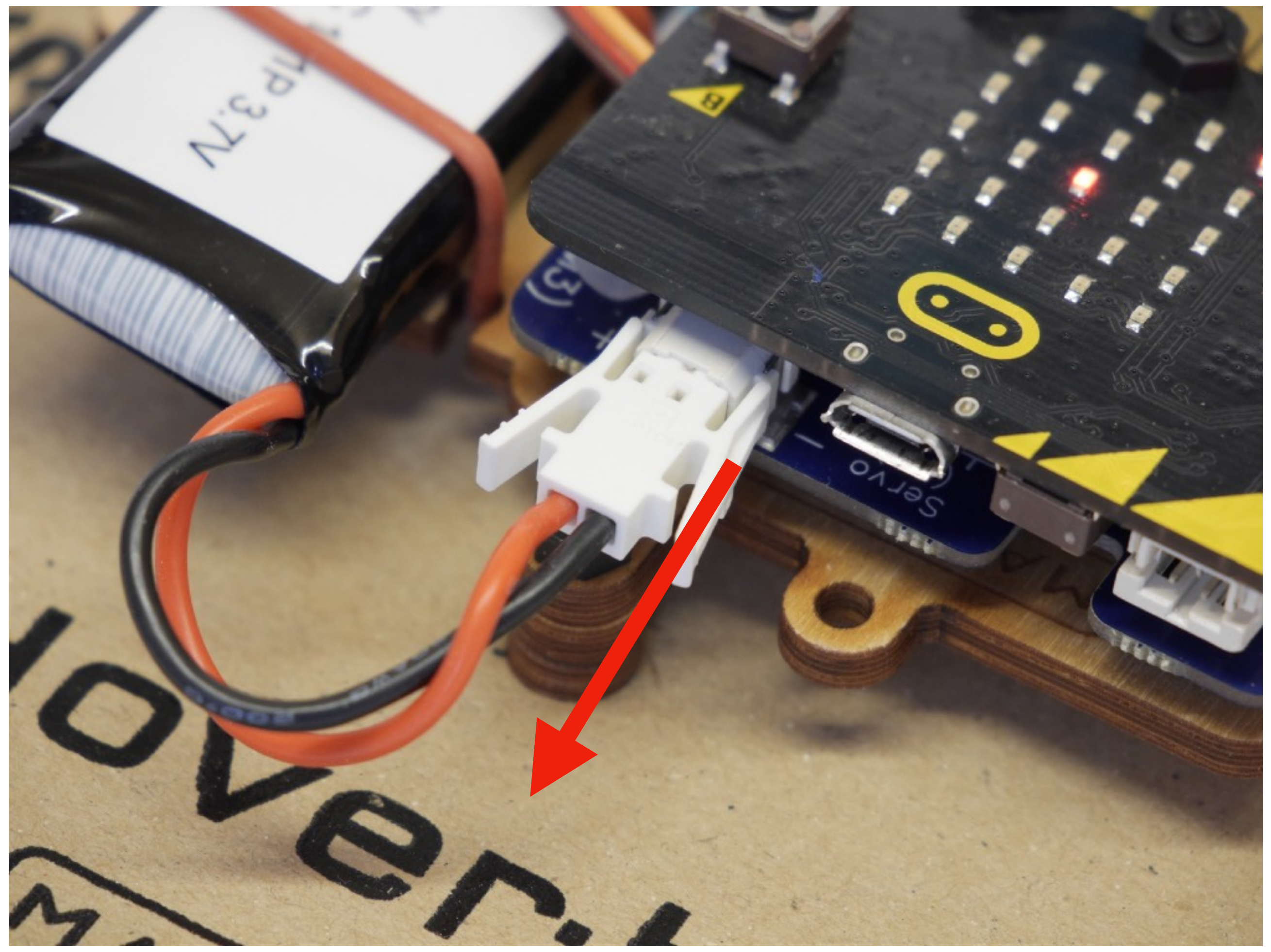

Watch for green light, indicating the battery is charged.
When a red light starts to blink, it is time to charge the battery.


Rember to thighten the brass nuts below the micro:bit, and the nuts on top. This is a common source of error (motors or servo will not work if they are not tight)

Warning:

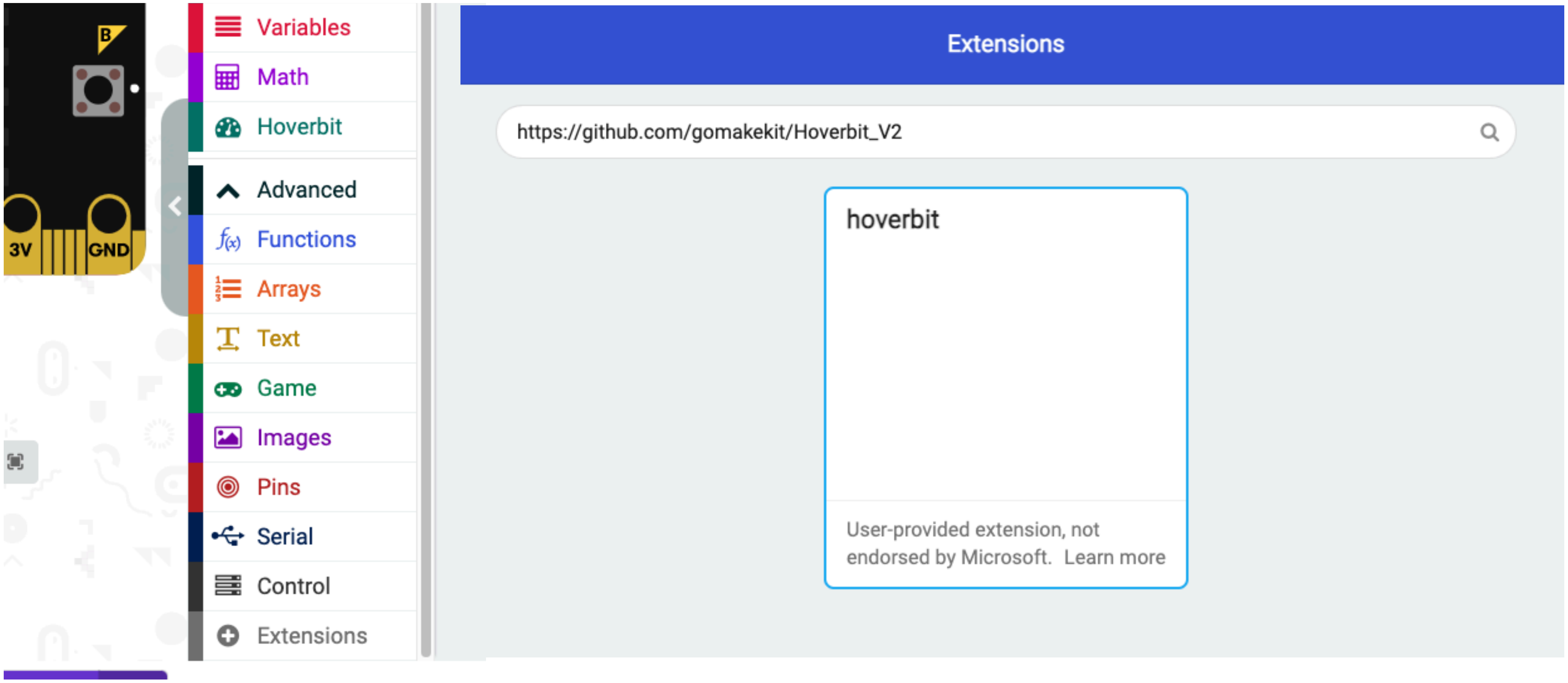
Warning! Keep hands off propellers while power is connected.

If something goes wrong, disconnect the battery immediately.



Install the library

- Under the Advanced tab, click Extensions
- Search for hoverbit and select the extension. If there is no search result, paste the following link into the text field: https://github.com/gomakekit/Hoverbit_V2
- Select the block that appears.



https://github.com/gomakekit/Hoverbit_V2

hoverbit

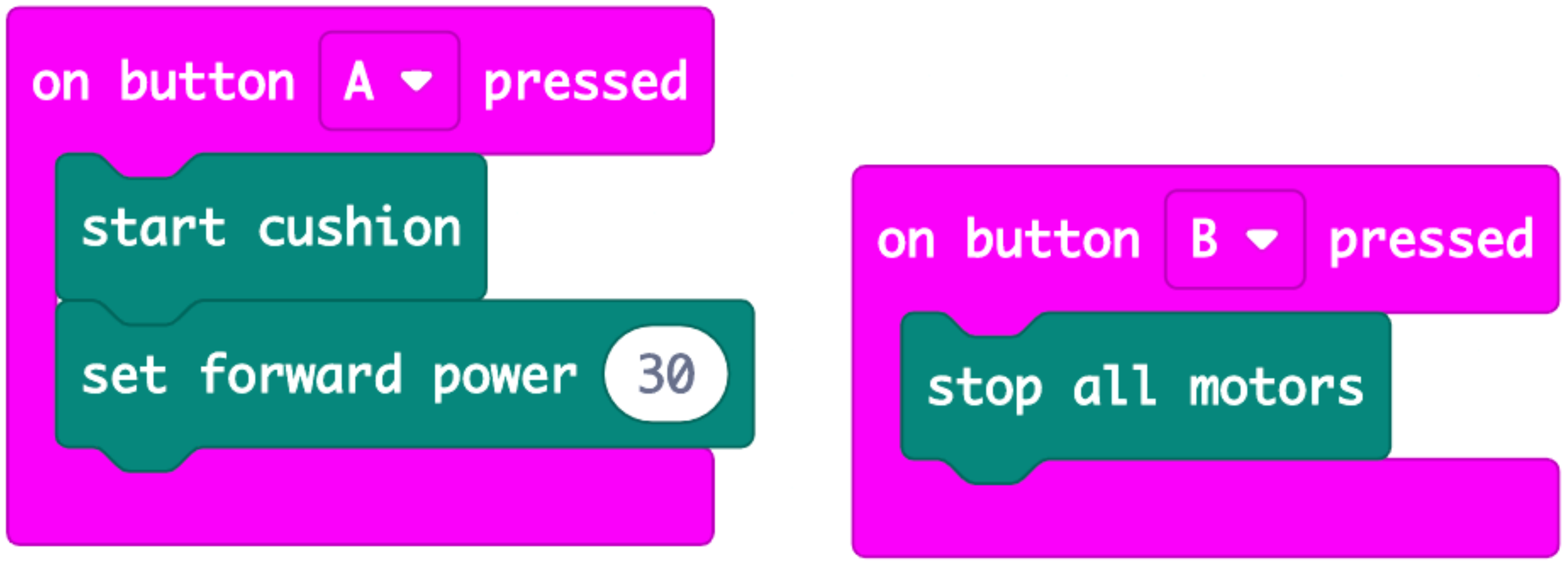
User-provided extension, not endorsed by Microsoft. Learn more

Test the motors

With the hover:bit blocks, write this test code and download it to the micro:bit on top of the hovercraft.

Press A to start motors. Both motors should now spin.

Press B to stop. Both motors should now stop.



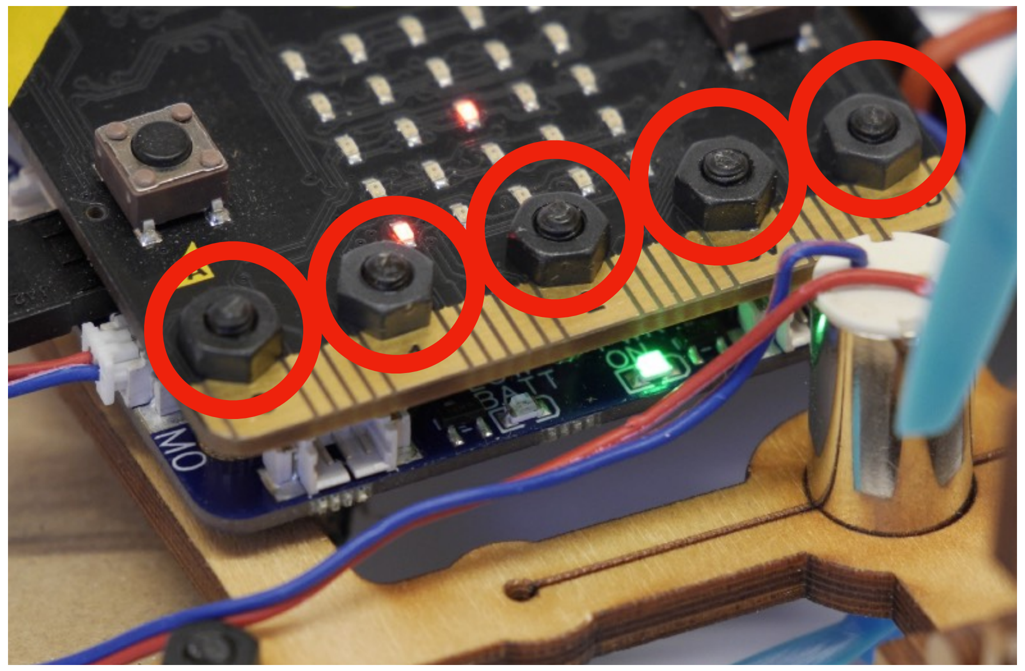
```
on button A pressed
    start cushion
    set forward power 30

on button B pressed
    stop all motors
```

Did it work?

If both motors did not spin, check the following:

Is the LiPo battery connected and is there a steady green light at the blue board, no red light when starting motors? Red light means you must charge the battery.

Are the five brass nuts under the micro:bit tight, and are the five nuts on top tight? Perhaps not!

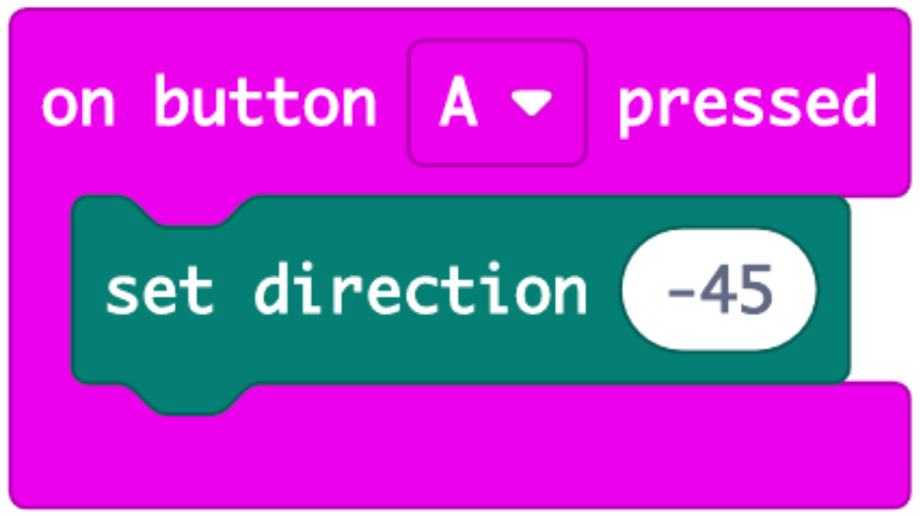


Are the motors connected properly on M0 and M2 outputs?

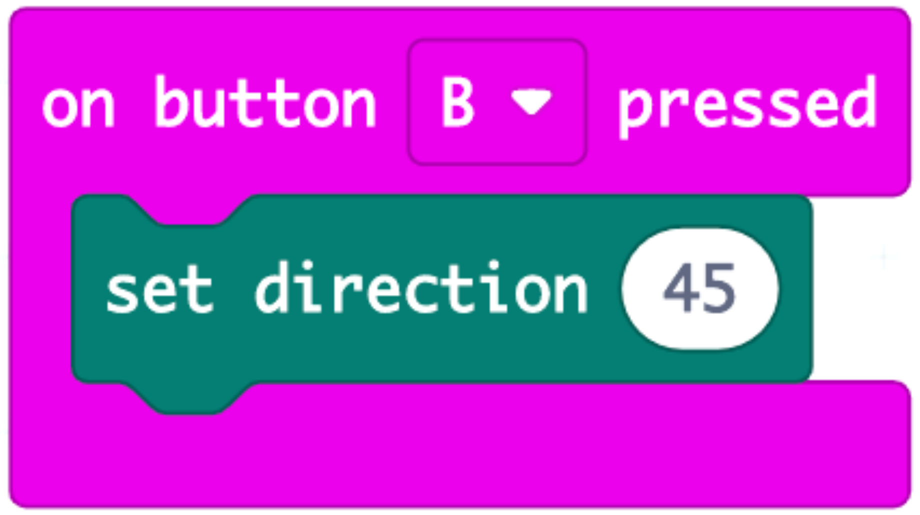
Did you really transfer the code to the micro:bit, or just downloaded it to your computer?

Test the servo

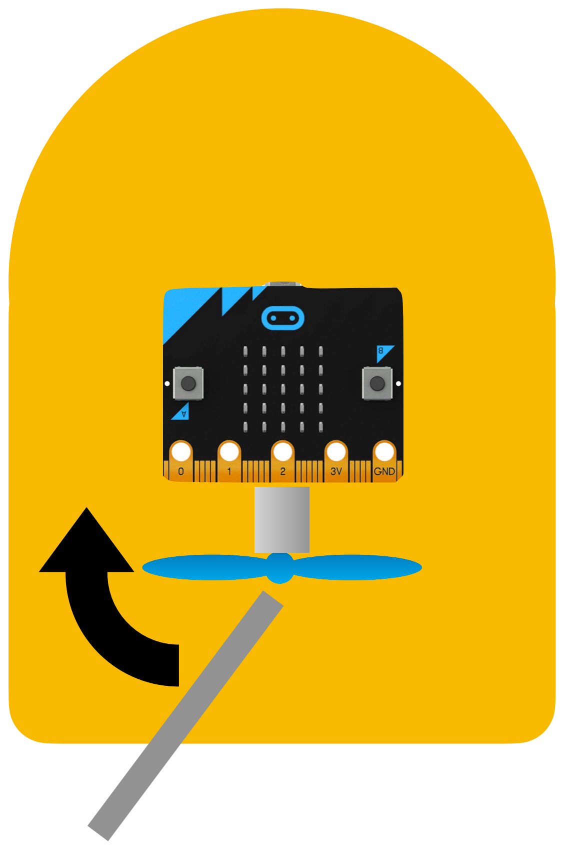
Replace the code with a this code (delete the old green blocks) and download it.
Press buttons A and B. Watch the movement of the rudder!

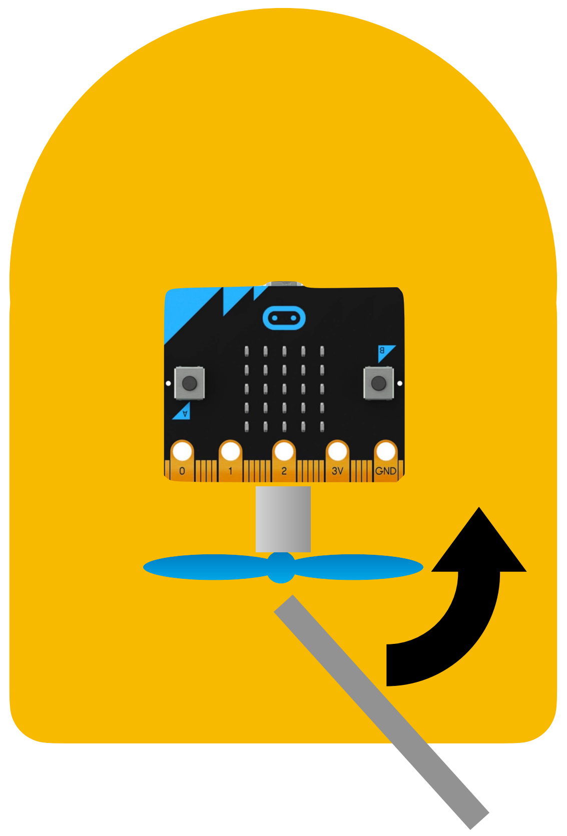


```
on button A ▾ pressed
  set direction -45
```



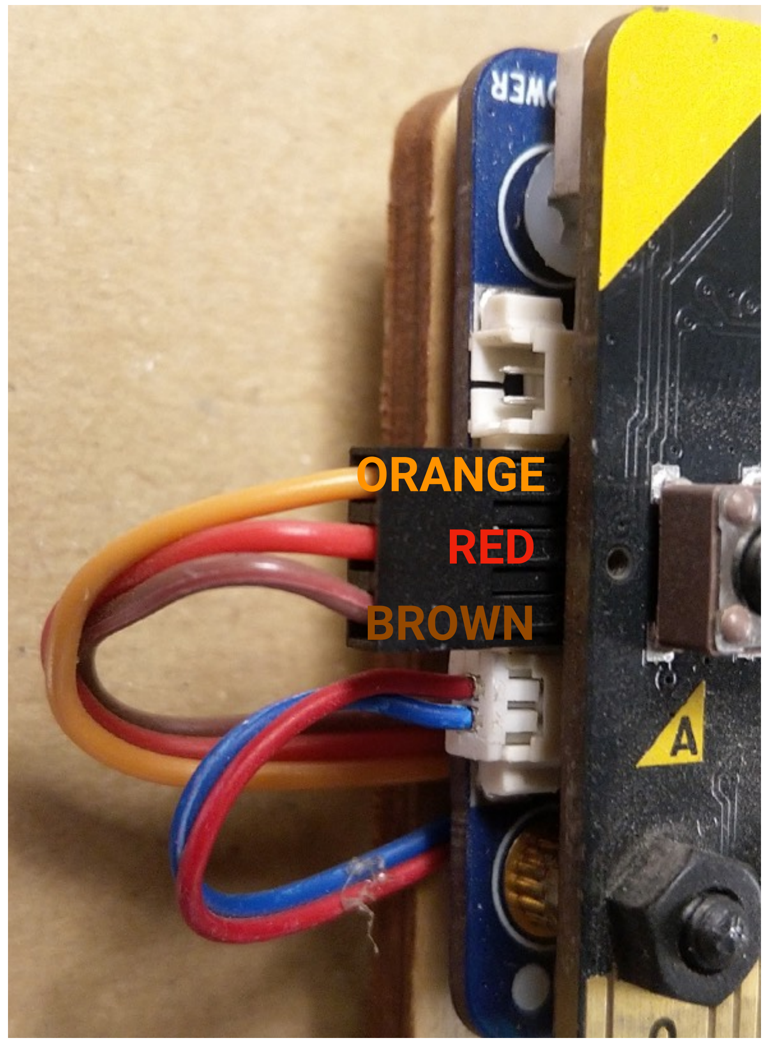
```
on button B ▾ pressed
  set direction 45
```





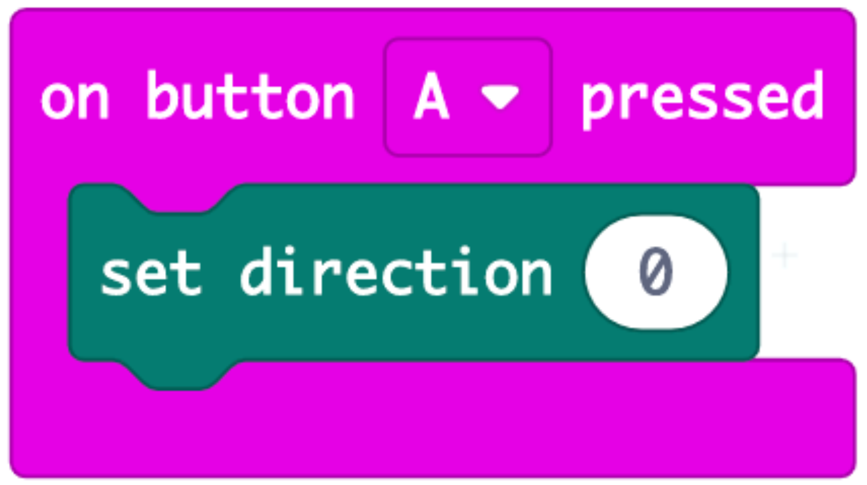
If the servo doesn´t move

- Is LiPo battery connected?
- Was the code transfered to micro:bit?
- Is the servo connected the right way?
- Are the brass nuts and the top nuts properly tightened?

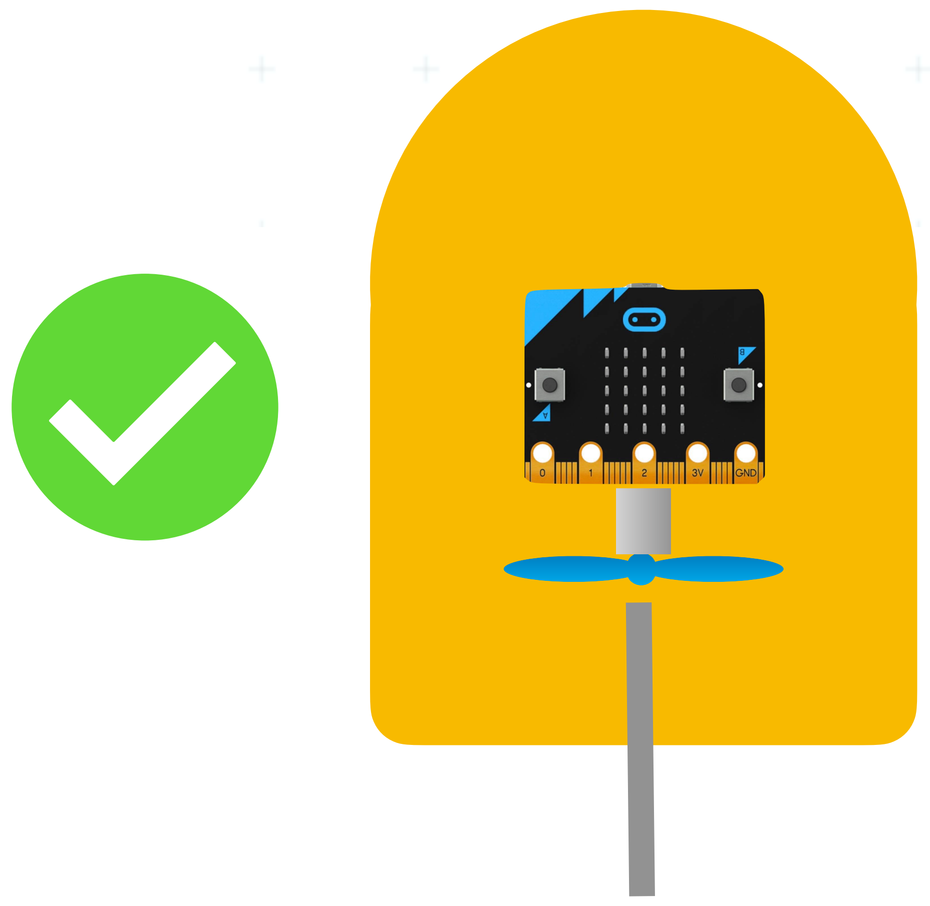


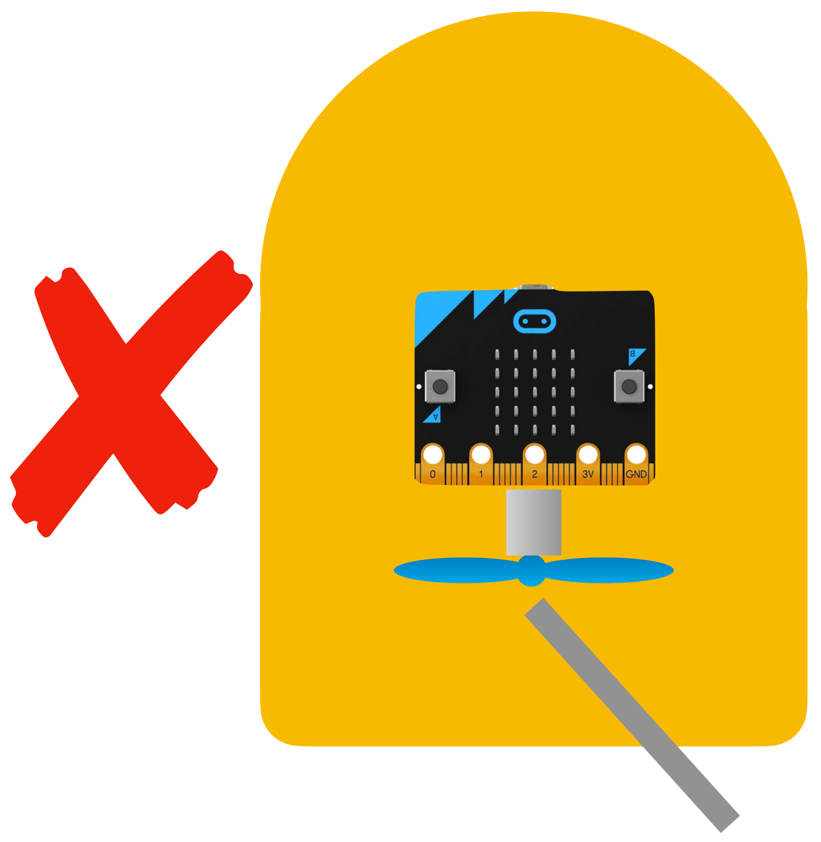
Calibrate the servo

Replace the code with this one and download it. Press A.
The servo should now go into neutral position. If not, we can fix it!



```
on button A ▾ pressed
    set direction 0
```

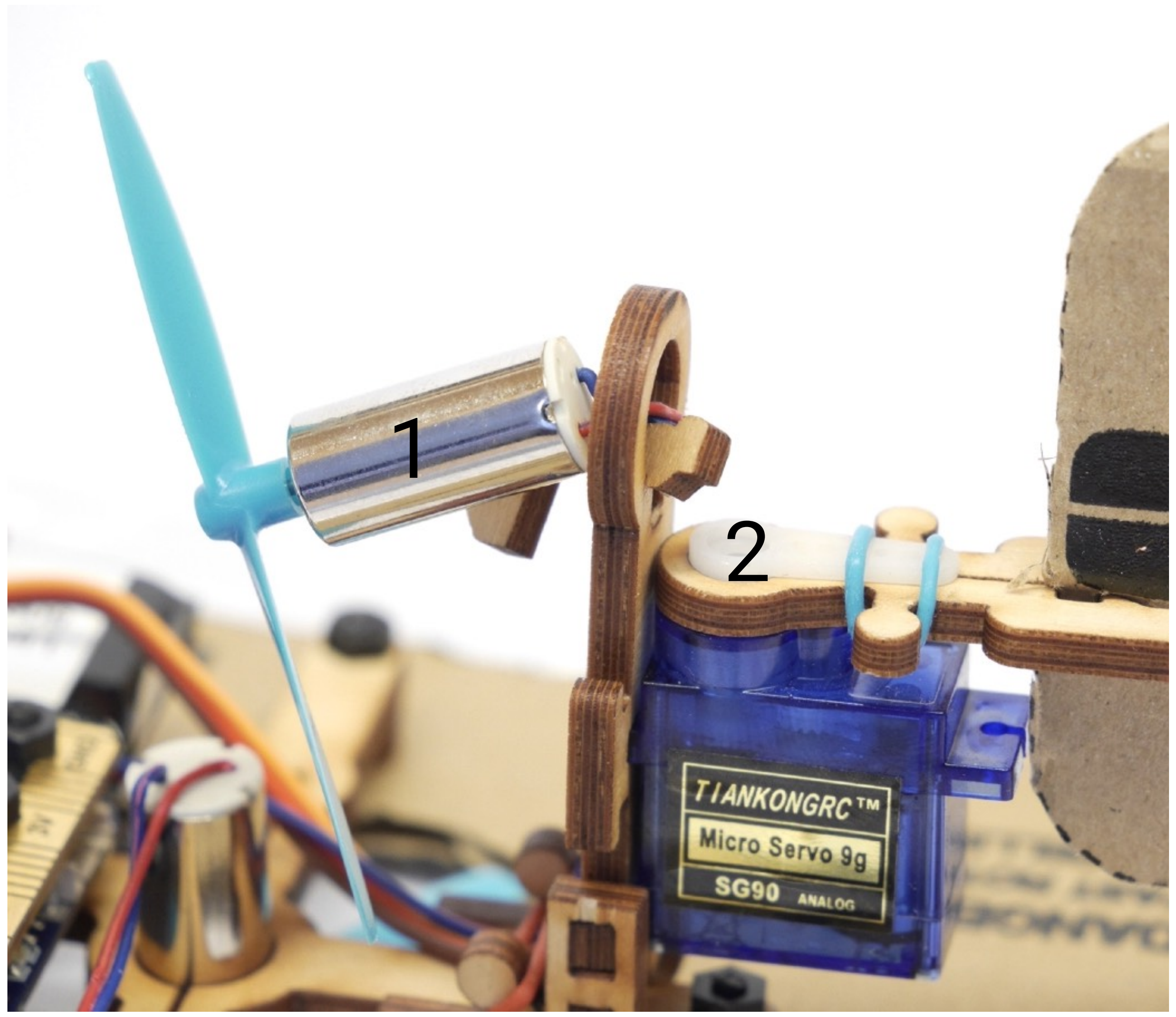




Fix the servo offset

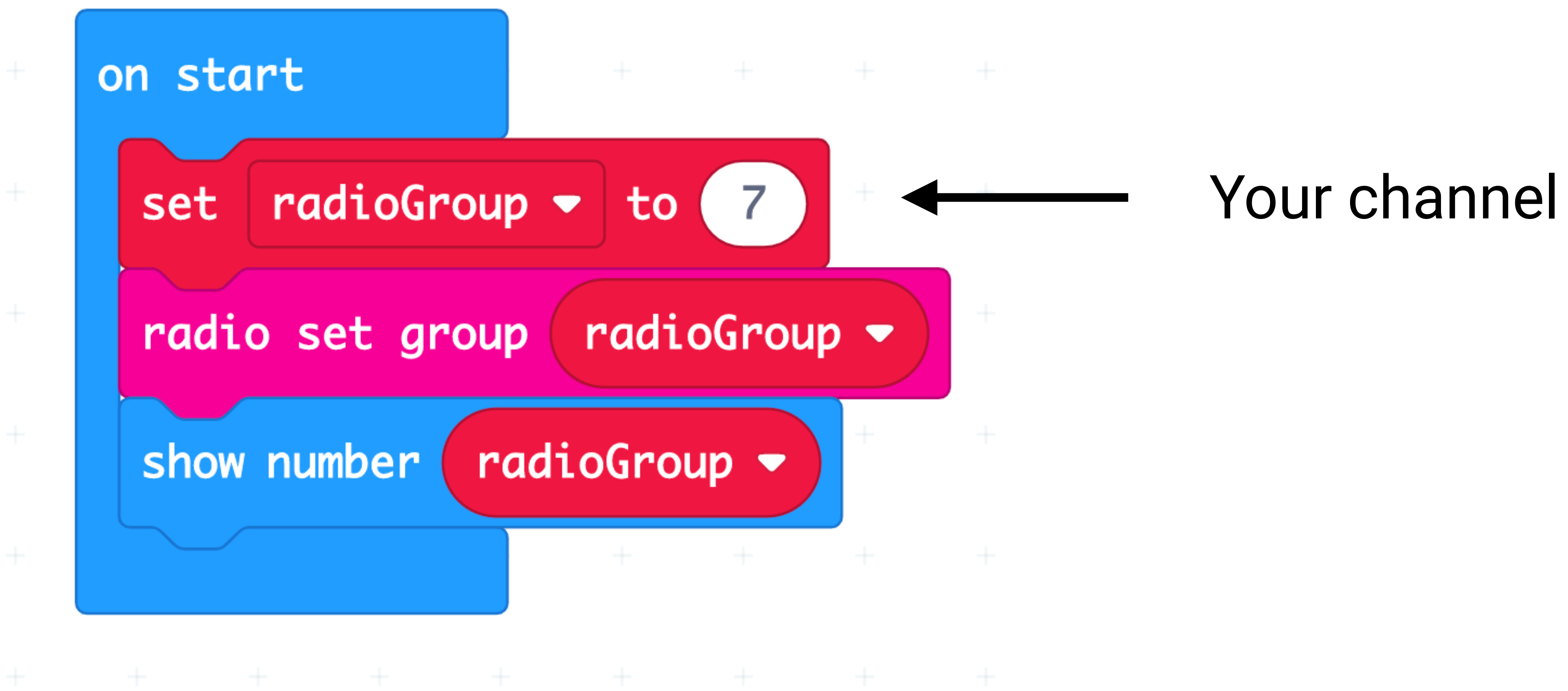
If the servo didn´t point straight backwards, to the following:

- Push the motor forward and release the wedge (1)
- Lift the white servo horn, straighten it, and push it back (2).
- Reinsert the wedge and motor. (Don´t push on the propeller, only the motor base)



Set the radio channel

In the on start block:
We need to set the same radio channel as in the receiver. These two numbers have to match! If you are in a classroom situation, each need to choose their own channel.

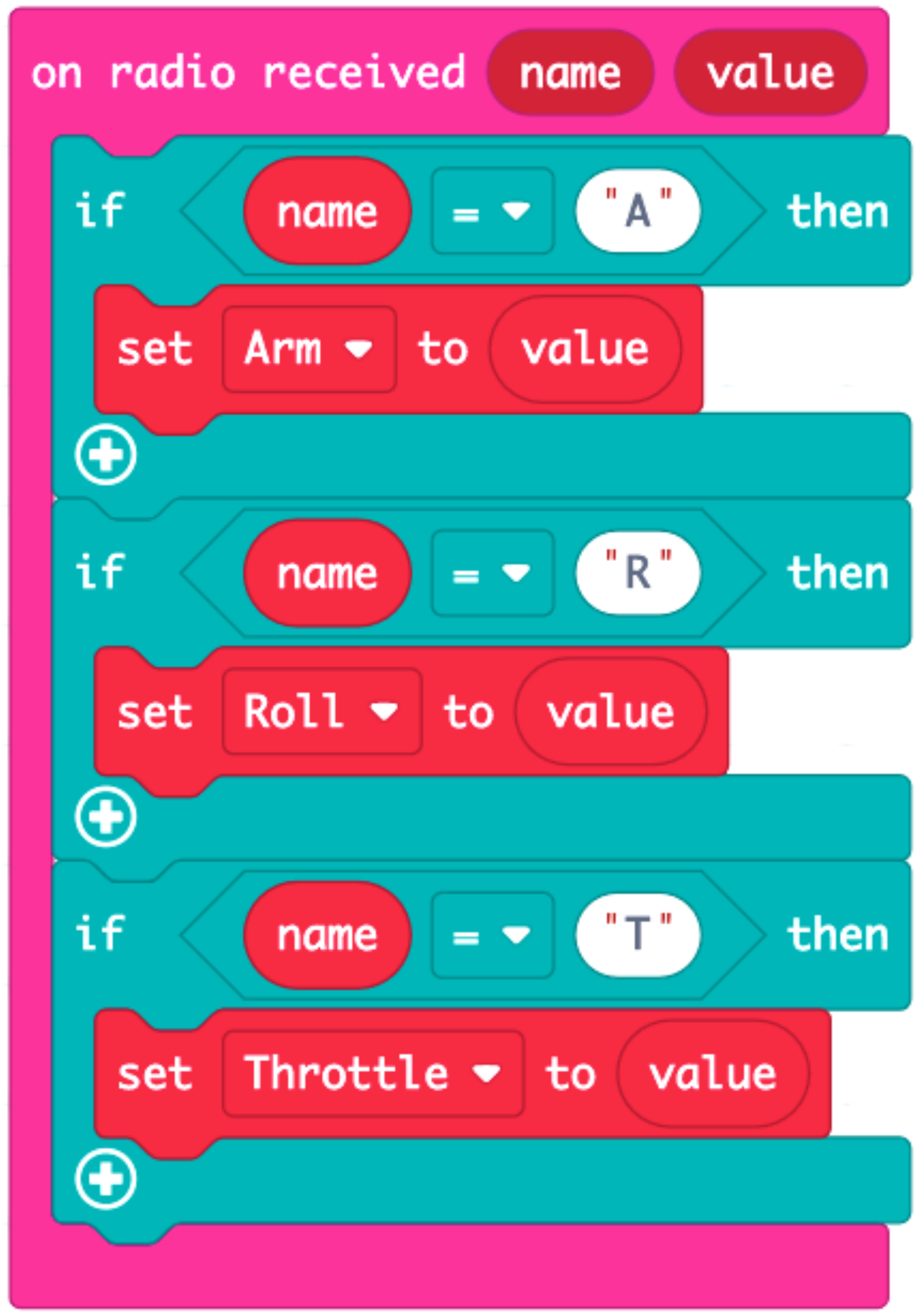


```
on start
    set radioGroup ▾ to 7
    radio set group radioGroup ▾
    show number radioGroup ▾
```

Receive the values over radio

Create the variables Arm, Roll and Throttle. Create the following code to receive and update the values each time they are received.

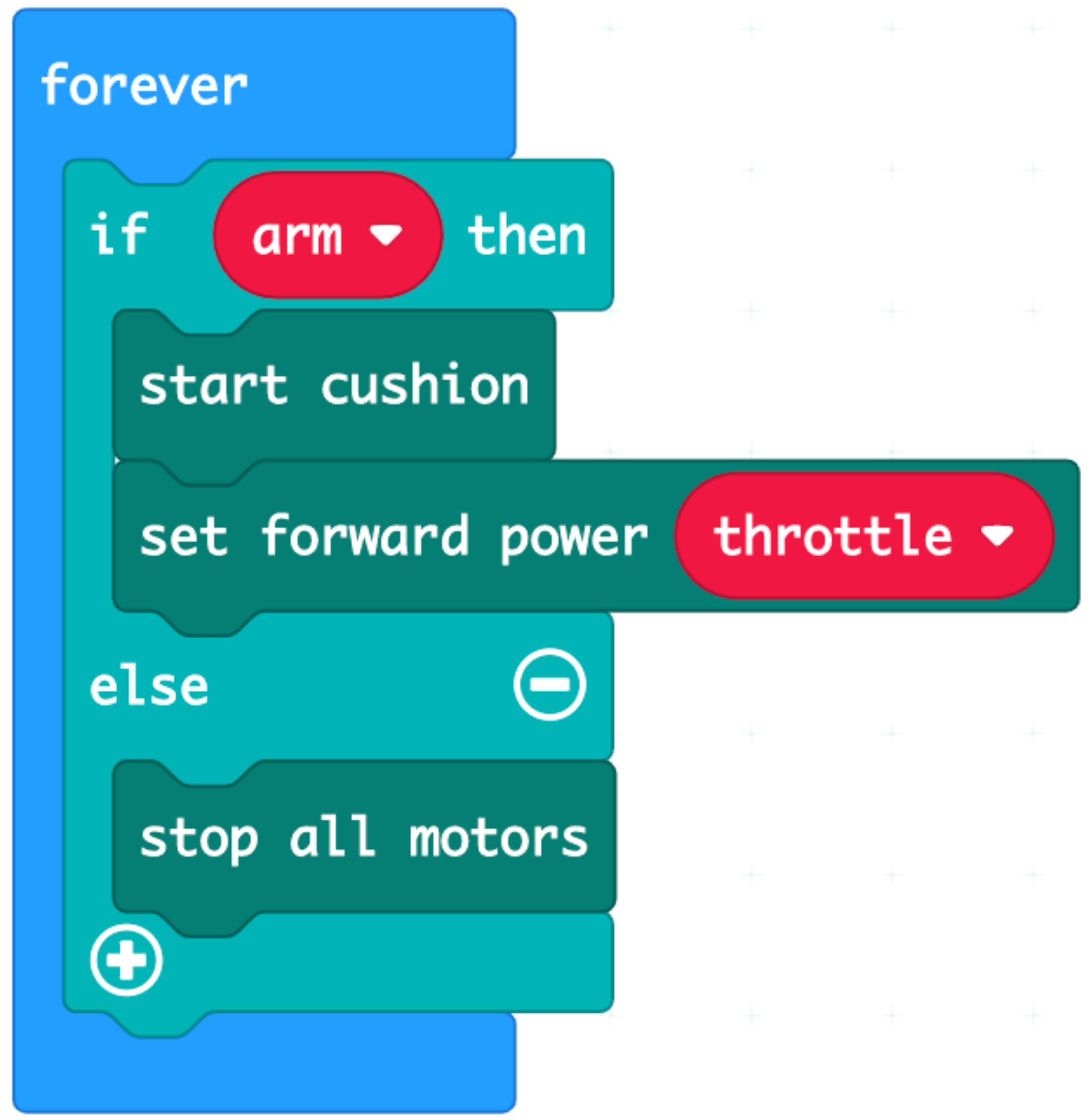
Remember to use capital letters, just as in our transmitter code.



```
on radio received name value
    if name = "A" then
        set Arm to value
    if name = "R" then
        set Roll to value
    if name = "T" then
        set Throttle to value
```

Control the speed

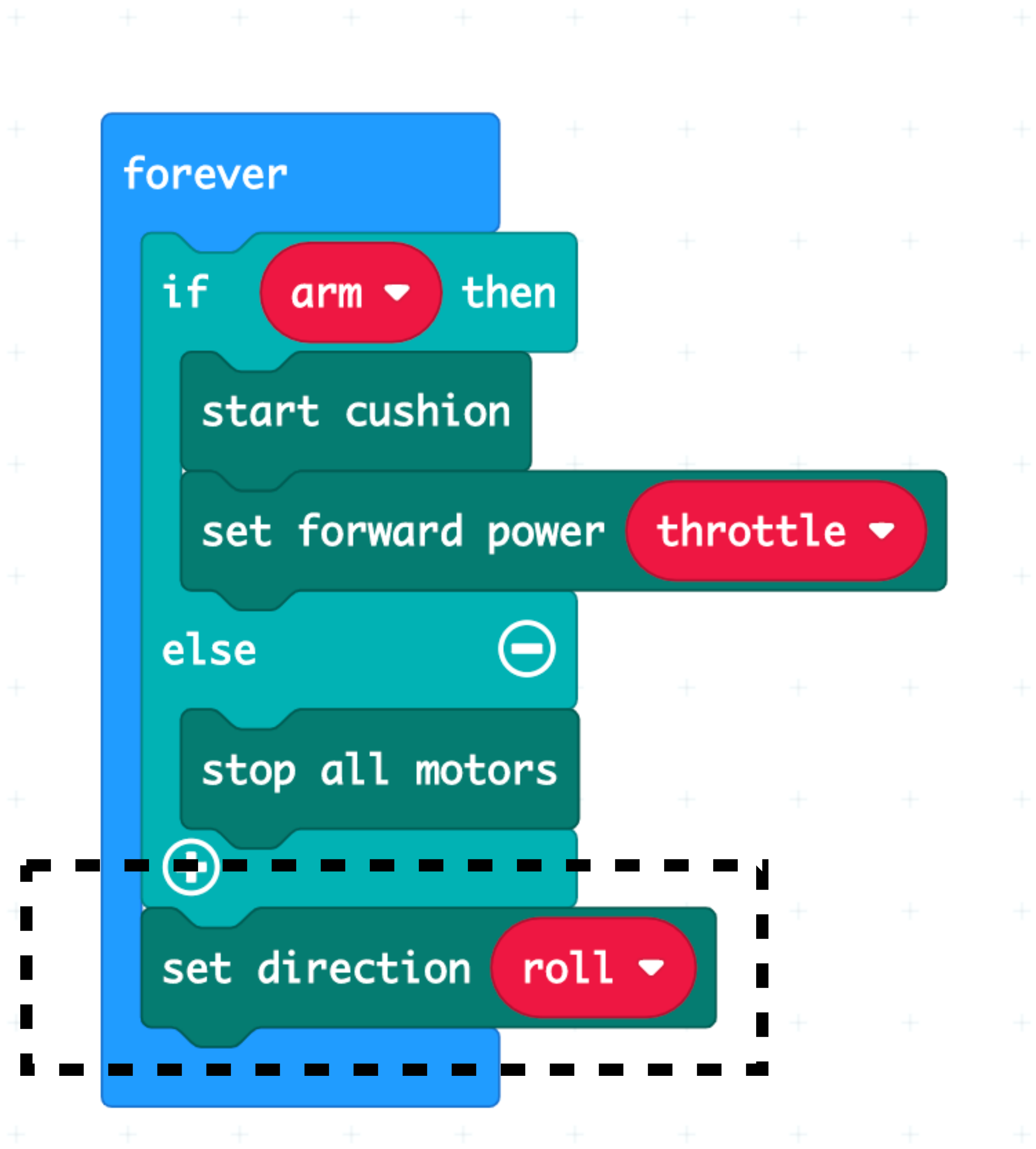
Make the following code using the hoverbit blocks:
In the forever loop, we check if arm is turned on. This should activate the vehicle. Start cushion will start the motor that blows air into the cushion. The forward power controls the pusher motor and is controlled by our throttle.
If we turn arm off (arm=0), then we stop all motors.



```
forever
    if arm ▾ then
        start cushion
        set forward power throttle ▾
    else
        stop all motors
```

Control the servo/rudder

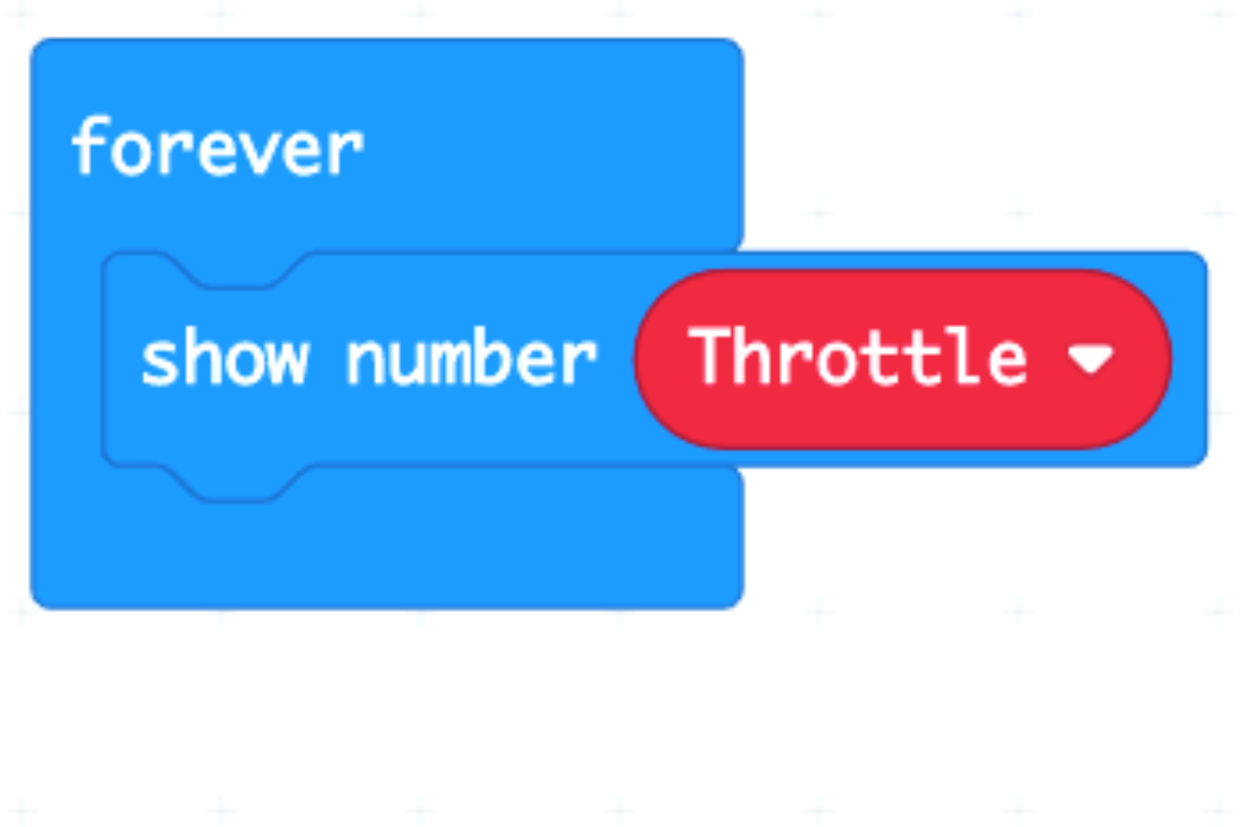
Add the bottom block, add the “set direction” hover:bit block.
This will control our rudder and it works independently of the arm value. As long as there is a battery connected, it should move.



```
forever
    if arm then
        start cushion
        set forward power throttle
    else
        stop all motors
    set direction roll
```

Display the throttle

Make a new forever-block and insert the following code.
This will display the throttle on the screen without interrupting other code.



```
forever
    show number Throttle ▾
```

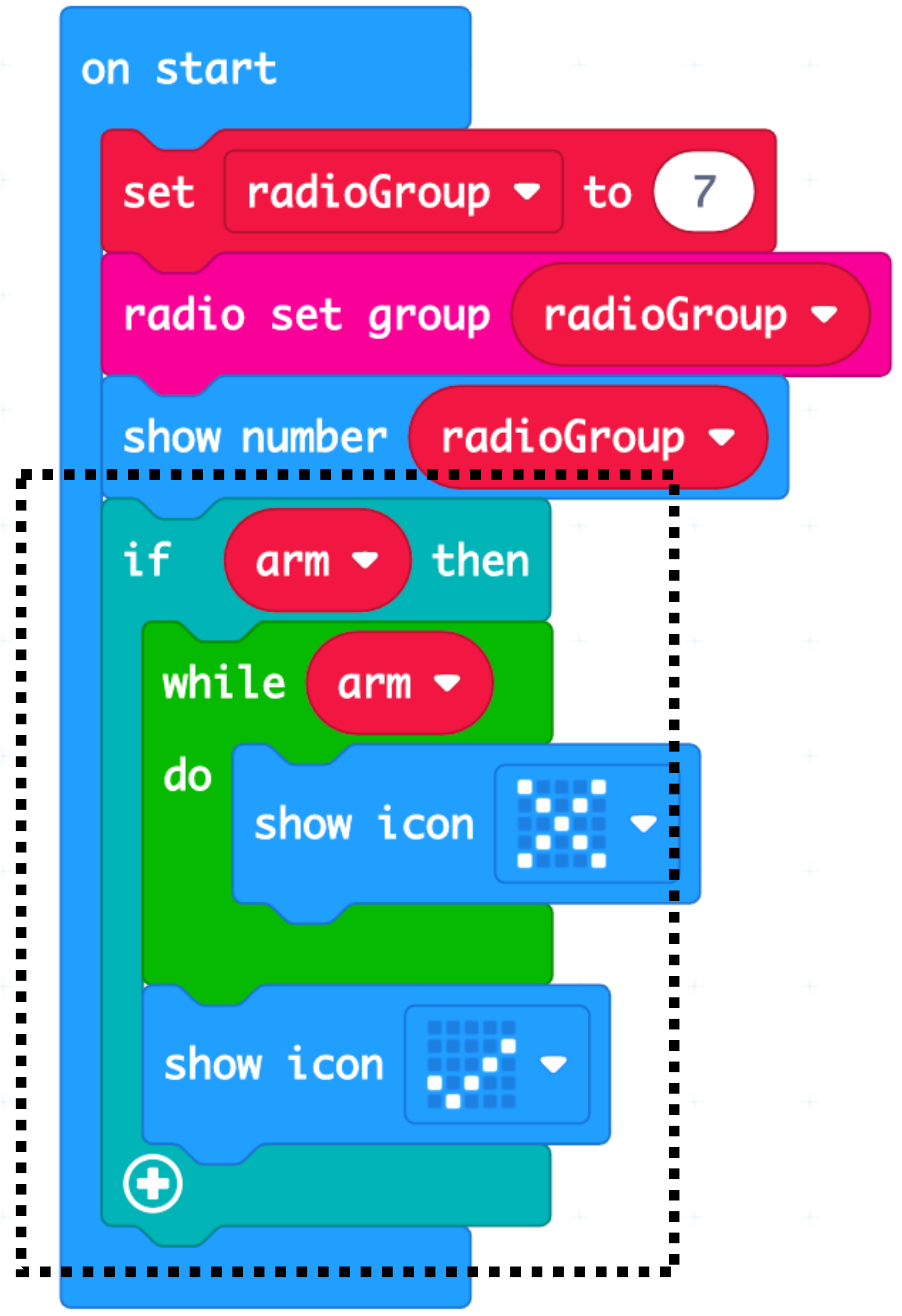
Optional: Start lock

Imagine if your receiver is set to arm and then you connect the battery to the hovercraft. The hovercrafts cushion motor would start immediately while you are holding it in your hands.

To avoid surprices, we can add a start lock. Now if arm is on, it will pause, showing a cross. You will need to turn arm off on the transmitter. A checkmark will be displayed.

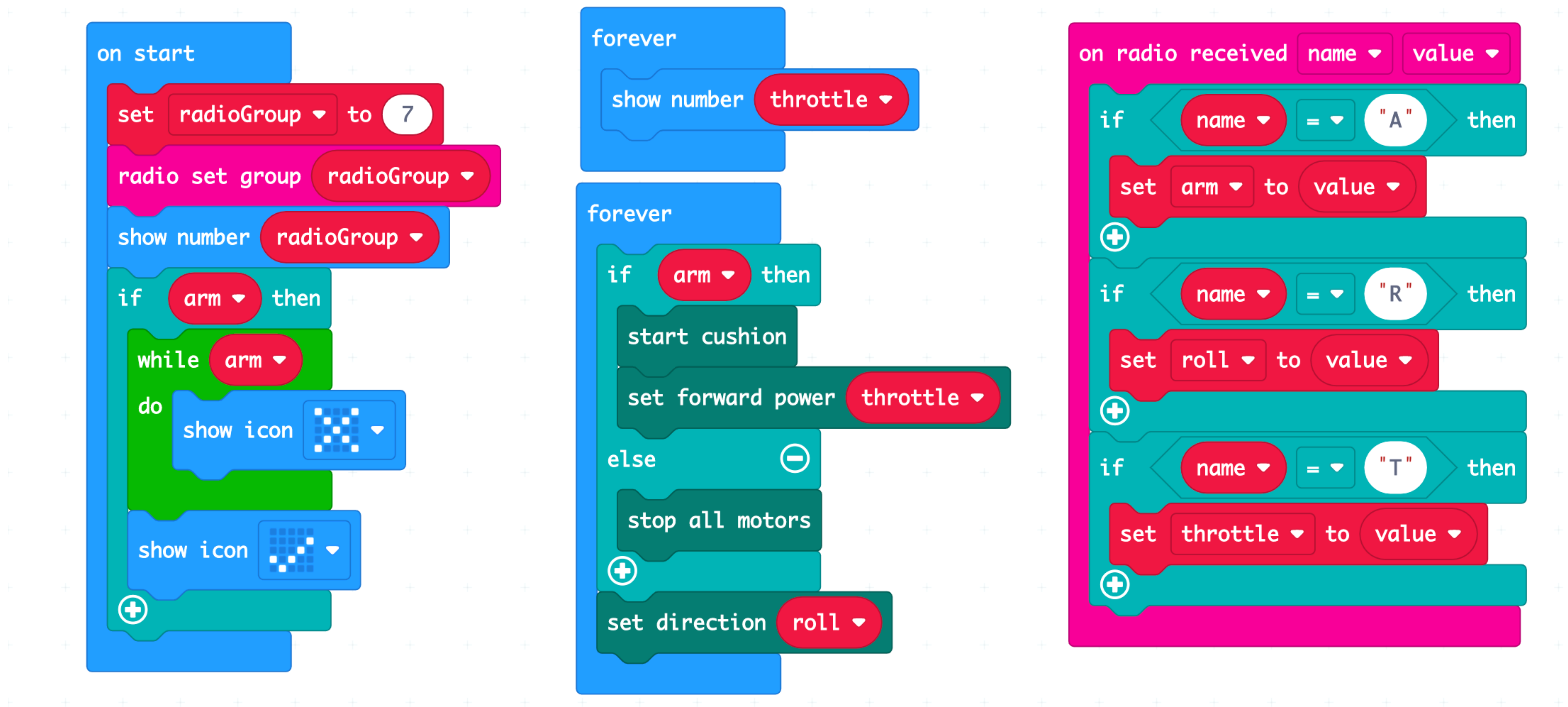
Then you can place the craft on the floor and set arm on again.

Download the full code to the hovercraft!



```
on start
    set radioGroup to 7
    radio set group radioGroup
    show number radioGroup
    if arm then
        while arm
        do
            show icon
        show icon
```

Full code



```
on start
    set radioGroup to 7
    radio set group radioGroup
    show number radioGroup
    if arm then
        while arm
        do
            show icon
        show icon
```

```
forever
    show number throttle
```

```
forever
    if arm then
        start cushion
        set forward power throttle
    else
        stop all motors
    set direction roll
```

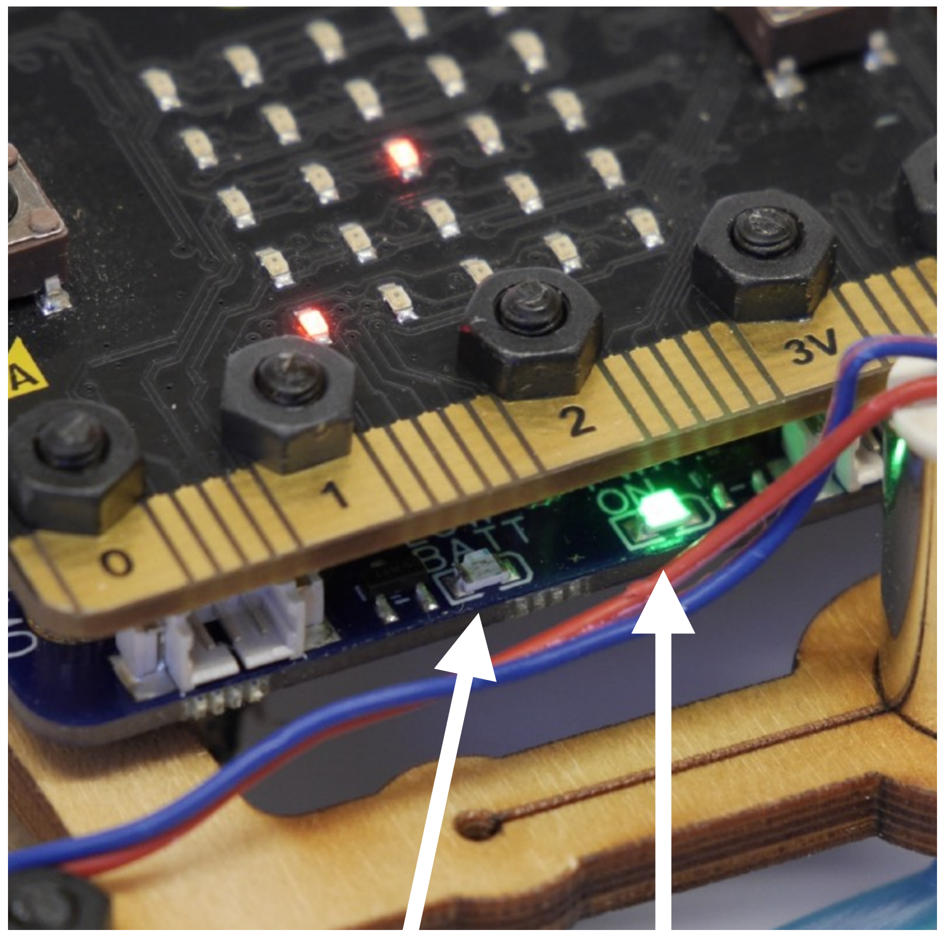
```
on radio received name value
    if name = "A" then
        set arm to value
    if name = "R" then
        set roll to value
    if name = "T" then
        set throttle to value
```

Test it

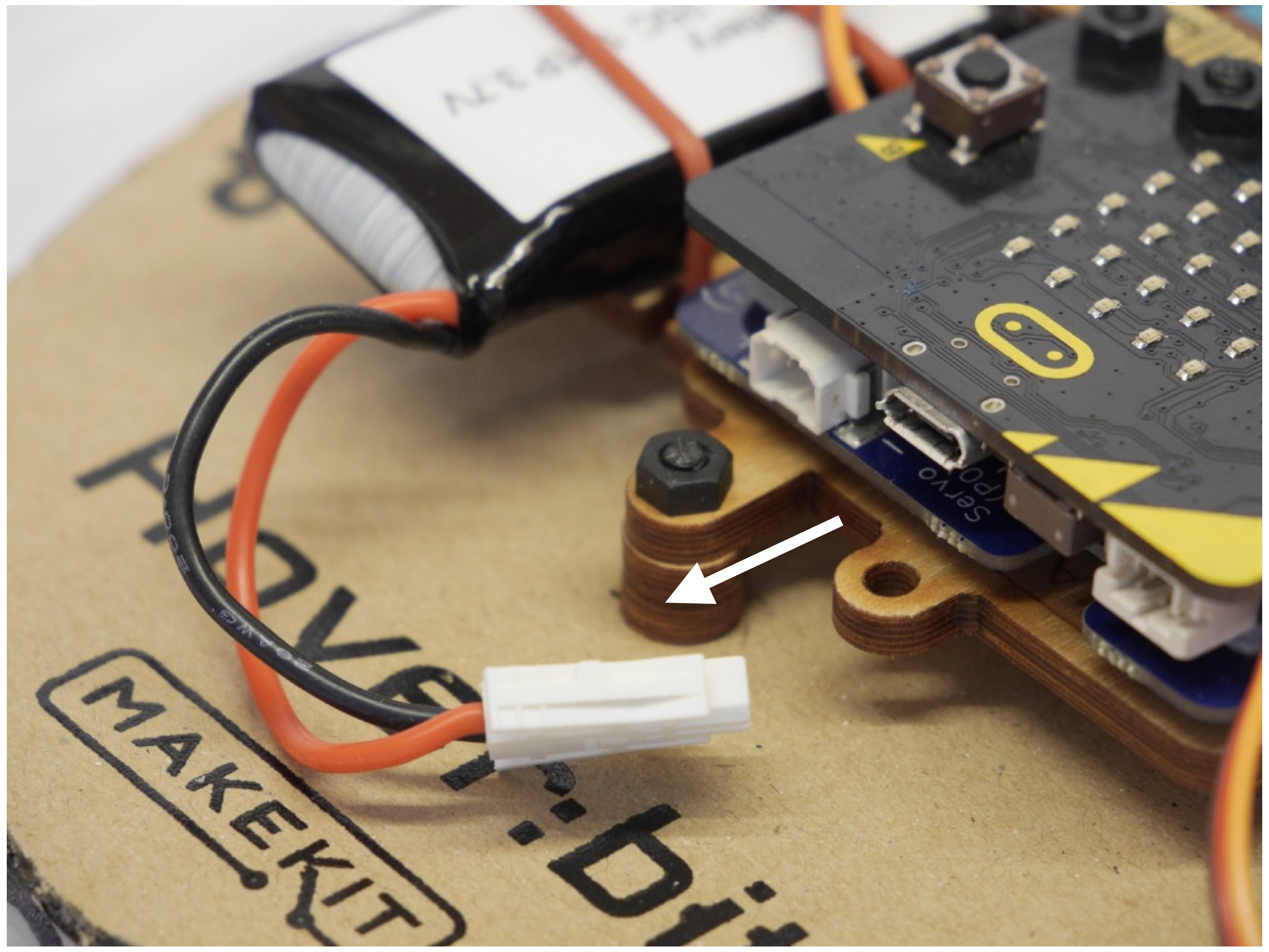
1. Make sure the complete transmitter code is downloaded to the transmitter, and the receiver code to the hovercraft.
2. Plug the power into your transmitter, and the LiPo battery to the hovercraft
3. Verify the number shown briefly on screen, the radio channel need to be the same.
4. Tilt the remote sideways to verify that the hover:bits rudder is moving
5. Press A+B to start engines
6. Increase speed with B button
7. Decrease speed with A button
8. Stop with A+B again or shake to stop quickly.
9. Have fun drifting your hover:bit! It can be used indoor and outdoor on dry clean surfaces like asphalt.

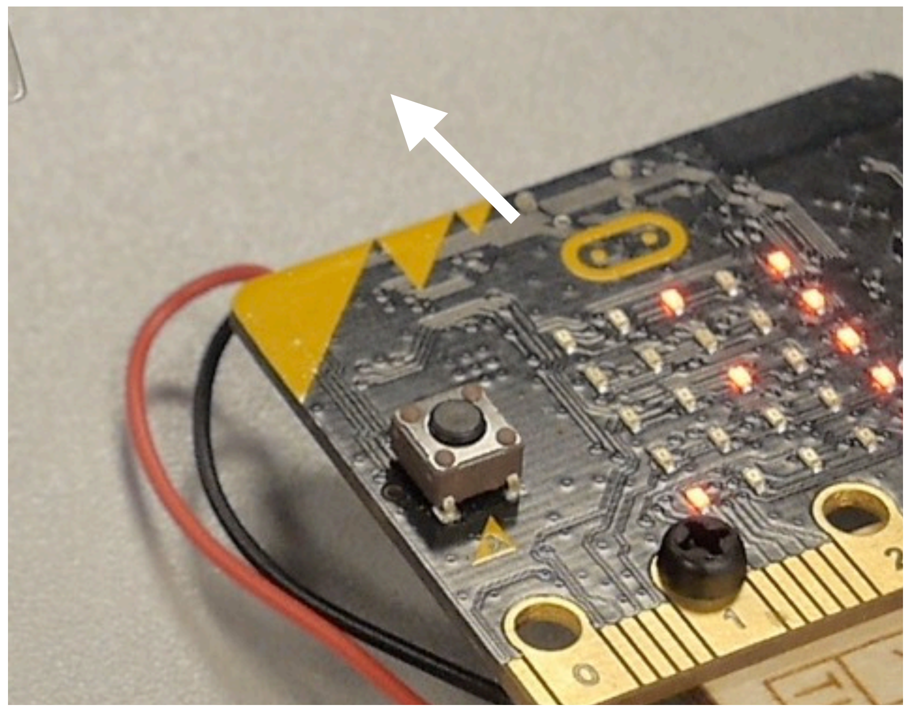
Remember:

If a red light starts to blink and the micro:bit loses power, it´s a sign of low battery. Please charge.
Always turn off transmitter and receiver after use. There is no auto power off.



Low batt Batt ok





Contact us:

Get help at our Facebook group:
www.facebook.com/groups/goairbit/

Do you have suggestions for improvements to the product or the guide?
Then I would like to hear from you!

You can contact me directly at henning@makekit.no

Henning Pedersen, Chief product developer

www.makekit.no support@makekit.no makekit gomakekit (also twitter)