Air:bit

Glove

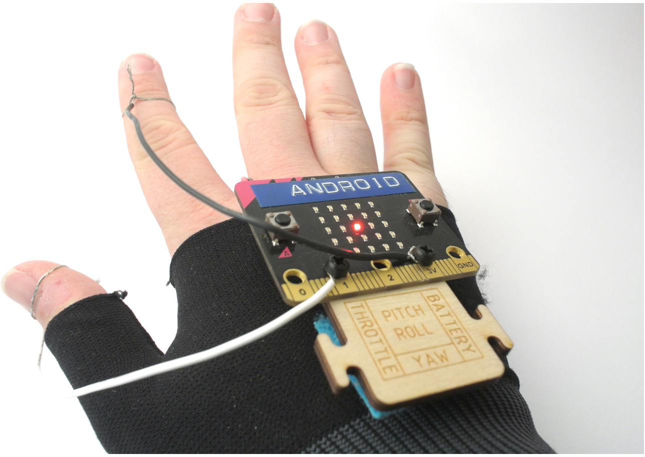
MAKEKIT

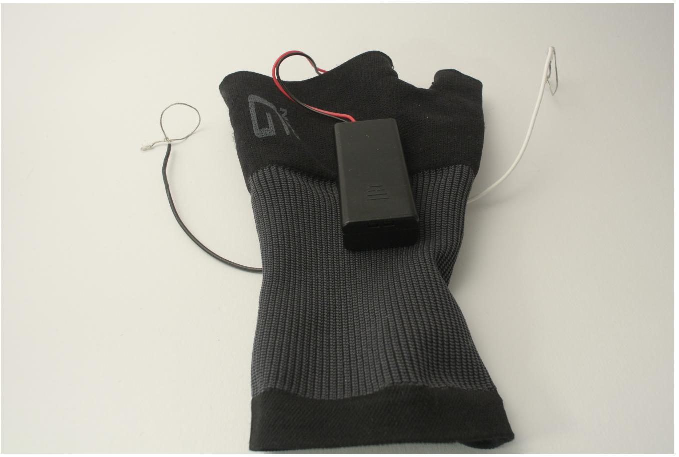
REPAIRWARE

Finished glove

Here you see the finished smart glove!

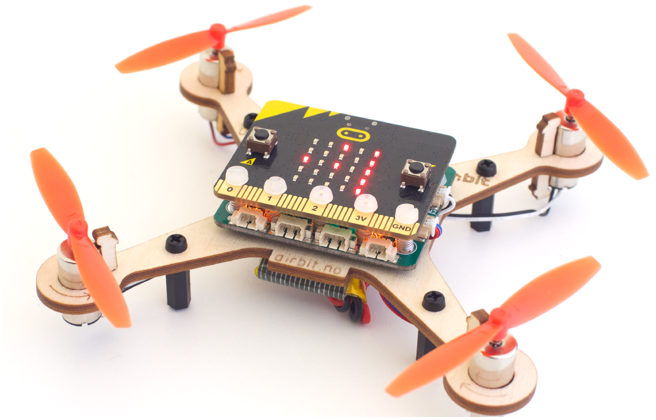
With the help of this glove, you can control the Air:bit, by using hand movement. You still use A and B for more or less throttle, and by using the finger sensors, you can also give a boost for more throttle, by squeezing the fingers together. When you no longer squeeze the cables together, the throttle will go back to the level you had before you squeezed your fingers together.

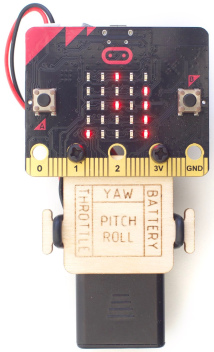


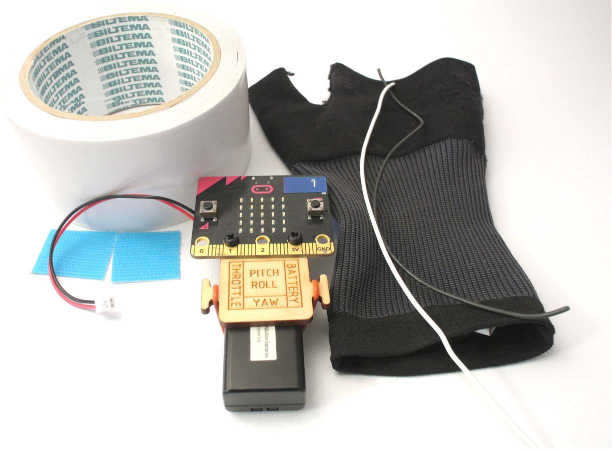


Parts

- 1x pre-made Air:bit
- 1x remote controller
- Velcro/double-sided tape
- A glove (that you can cut the finger tips off)
- Two cables of approx. 20. cm.



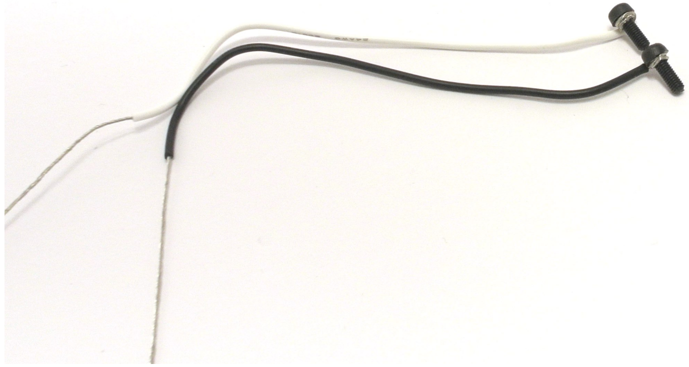




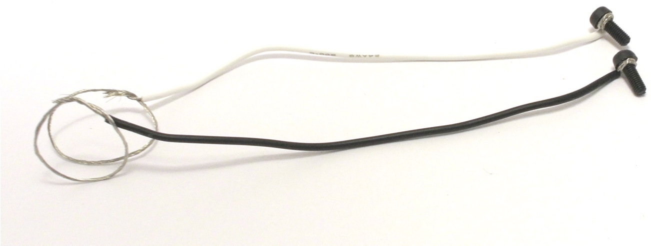
Assembly

- Glove (which you can cut the finger tips off), the cables must be stripped, about 6-8 cm at one end to go around your fingers, and about 2-3 cm to be screwed on with the screw

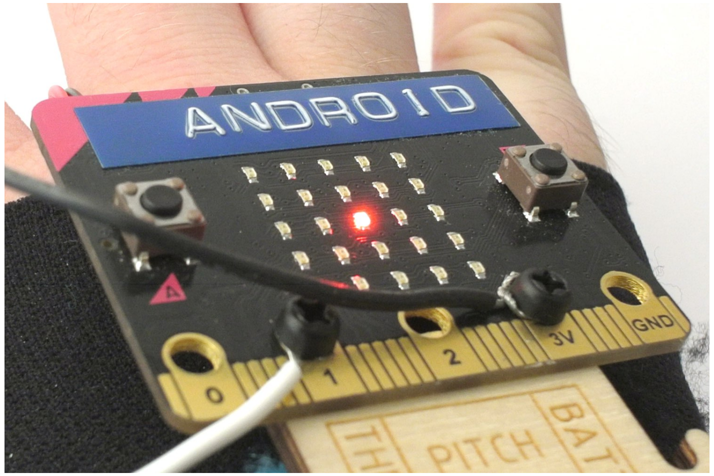
- Twist the cables around the screw so that they are firmly attached



- Then twist the cables around your thumb and index-finger so that the cables have a round end

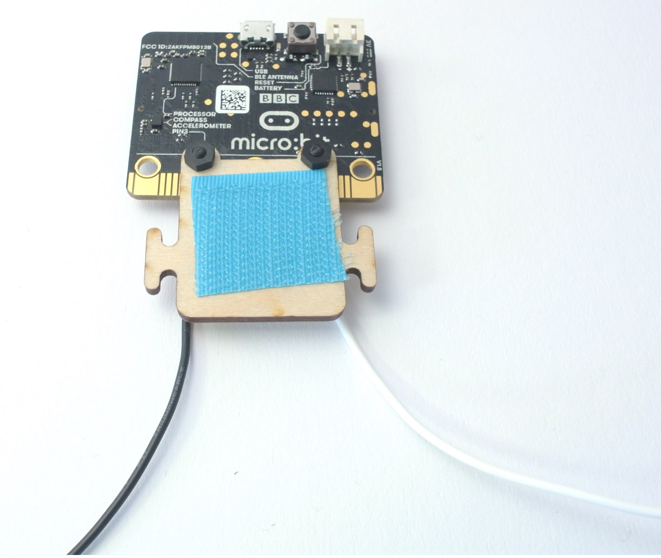


- The cables must then be screwed to Pins 1 and 3V

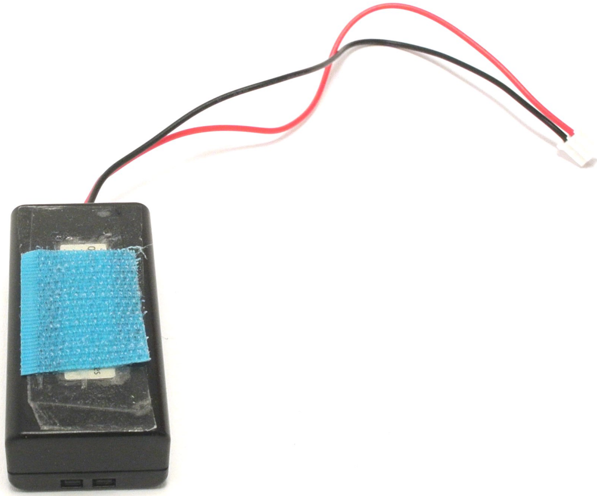


Assembly

- Attach a Velcro piece to the back of the battery holder bracket

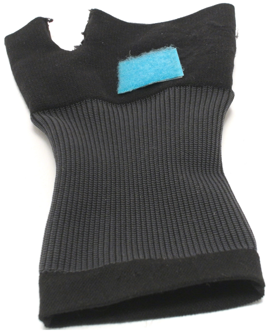


- Attach a Velcro piece to the back of the battery holder

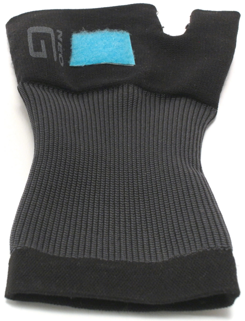


Assembly

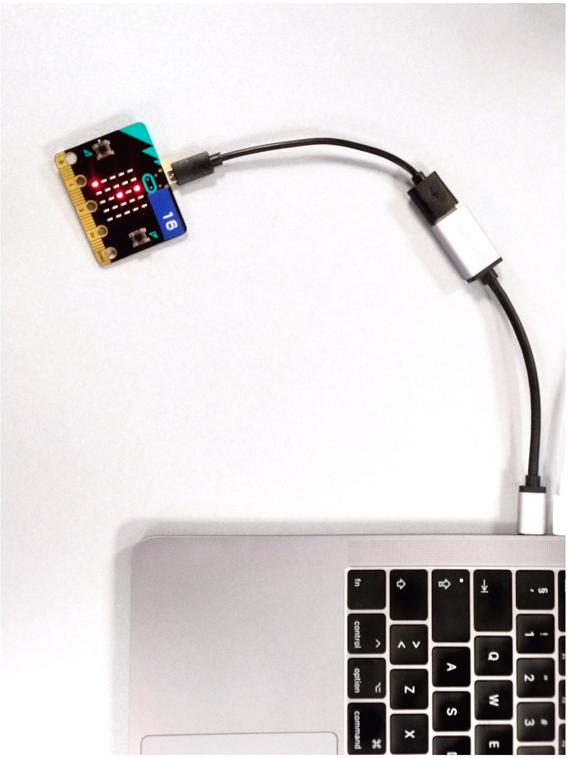
- Attach the counterpart to the velcro on the front and back of the glove



The micro:bit remote controller, should be on top of the glove, the battery box should be on the bottom of the glove



Connect The Micro:bit to Mac/PC

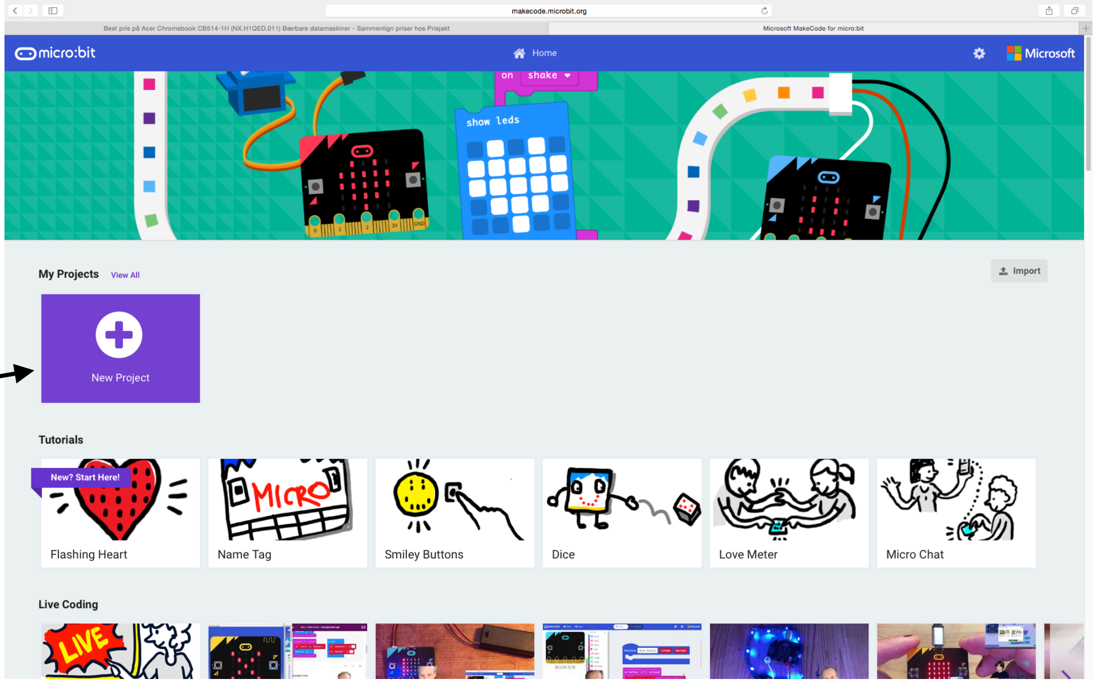


MakeCode

- This is the website we are going to use for testing the finger sensors, but also the coding

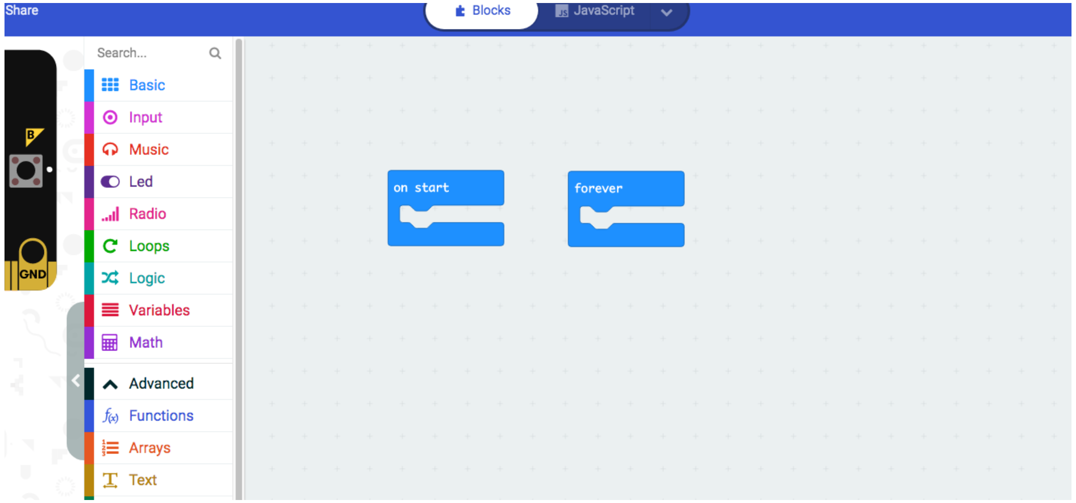
  https://makecode.microbit.org

- Before each new project, we must click on "New Project"

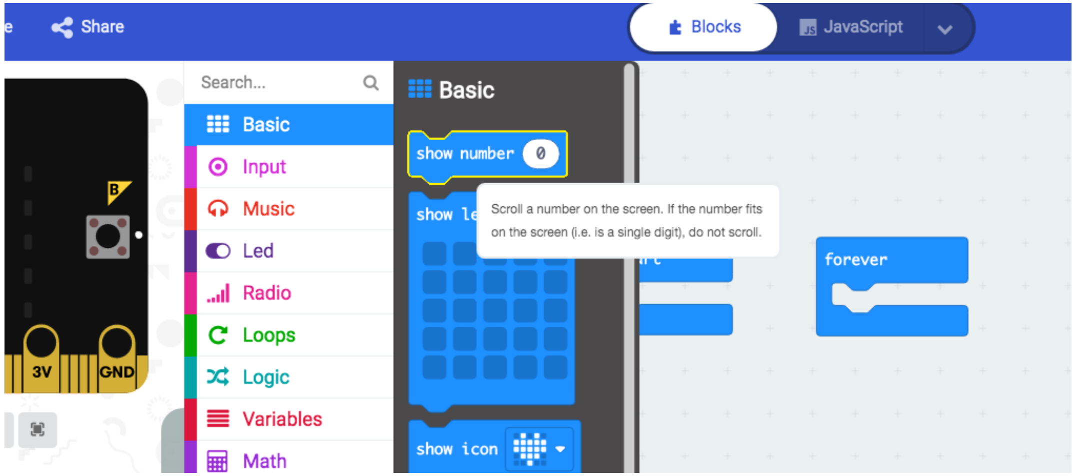


Testing The Finger Sensors

- First we need to find what the voltage between the cables is, when they are not pinched together, and when they are pinched together.
  To find out the voltage between the cables, we need some code blocks.
  First look for “show number 0”, in the Basic block



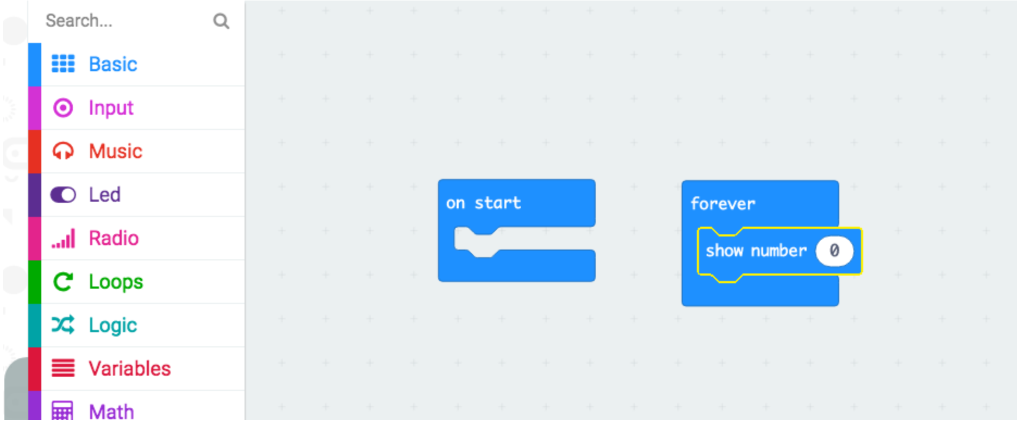
We measure the voltage, to find out the lowest, and highest value, so that we can enter the correct value in the code. If you enter a value that is incorrect, the fingerprint sensor will not work.



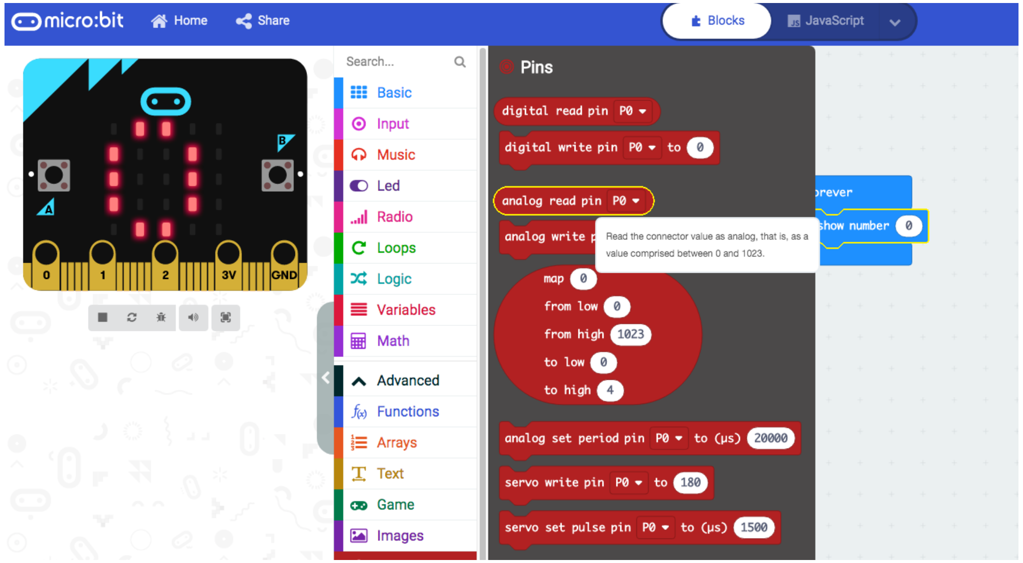
Scroll a number on the screen. If the number fits on the screen (i.e. is a single digit), do not scroll.

Testing The Finger Sensors

- Place the “show number” block, in the forever block.



- Now we need a block from Pins, look for the “analog read pin P0”



Read the connector value as analog, that is, as a value comprised between 0 and 1023.

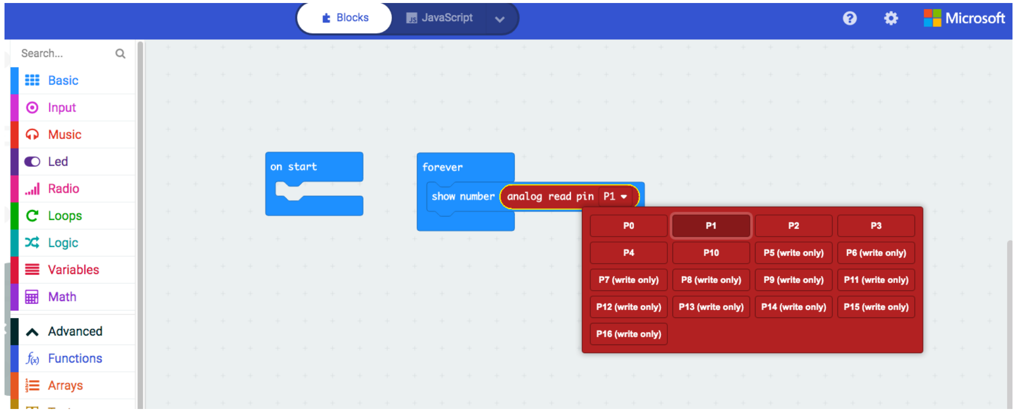
Testing The Finger Sensors

- Place “analog read pin P0” in the “show number” block
- Change P0 to P1

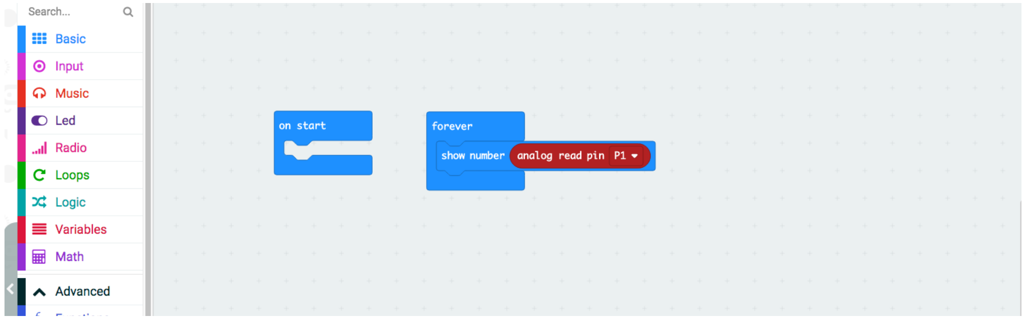
This is because we want to find out what the voltage between pin 1 and 3V is.

Press download, so that the code is uploaded to the Micro:bit remote controller, and we are ready for testing

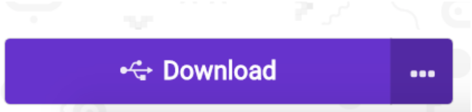
Note the difference between when you press and do not press the cables.
One squeeze will usually give a value above 1020, and when you don`t squeeze the finger sensor together, the value is usually below 1000



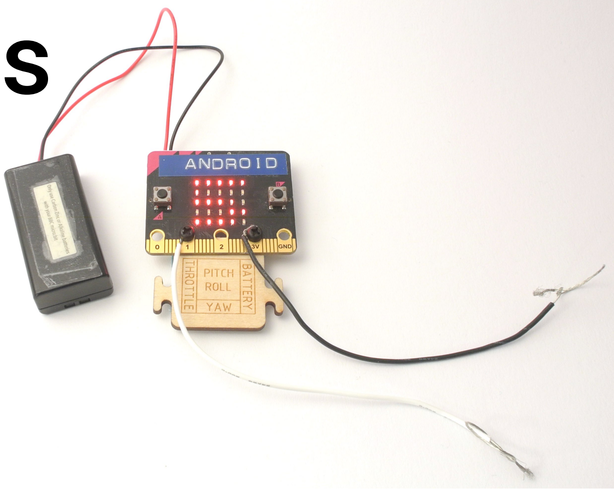
```
on start

forever
  show number analog read pin P1
```



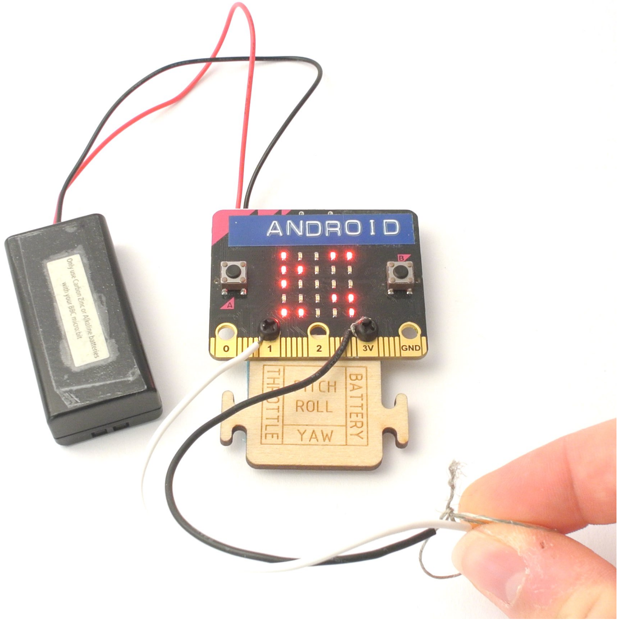
```
on start

forever
  show number analog read pin P1
```



Testing The Finger Sensors



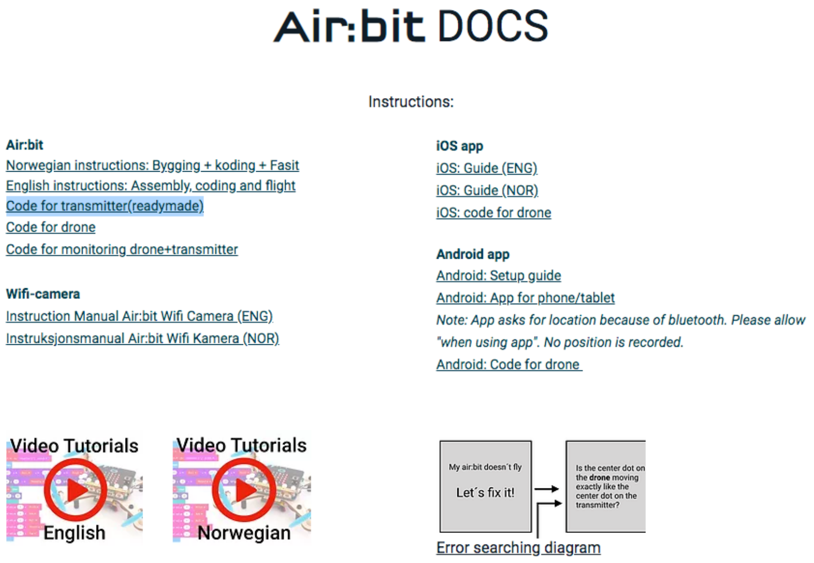
- In this picture you see that some numbers go across the screen, this whole number, ex. 995, is the value (voltage) when the cables do not touch each other.



- In the last picture you also see numbers that go across the screen, this whole number, ex. 1023, is the value (voltage) when the cables touch each other. We must choose a value higher than 995 but lower than 1023. We will try the value 1010.

Coding

- Go to this webpage, **https://www.makekit.no/airbit-docs**
- There you need to download the code for Air:bit's remote controller, called "Code for transmitter (readymade)"
- Once you have downloaded the code, you are ready for the next step



Air:bit DOCS

Instructions:

Air:bit
Norwegian instructions: Bygging + koding + Fasit
English instructions: Assembly, coding and flight
Code for transmitter(readymade)
Code for drone
Code for monitoring drone+transmitter

Wifi-camera
Instruction Manual Air:bit Wifi Camera (ENG)
Instruksjonsmanual Air:bit Wifi Kamera (NOR)

iOS app
iOS: Guide (ENG)
iOS: Guide (NOR)
iOS: code for drone

Android app
Android: Setup guide
Android: App for phone/tablet
Note: App asks for location because of bluetooth. Please allow "when using app". No position is recorded.
Android: Code for drone

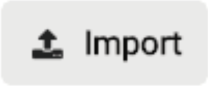
Video Tutorials English

Video Tutorials Norwegian

My air:bit doesn´t fly
Let´s fix it!
Is the center dot on the **drone** moving exactly like the center dot on the transmitter?

Error searching diagram

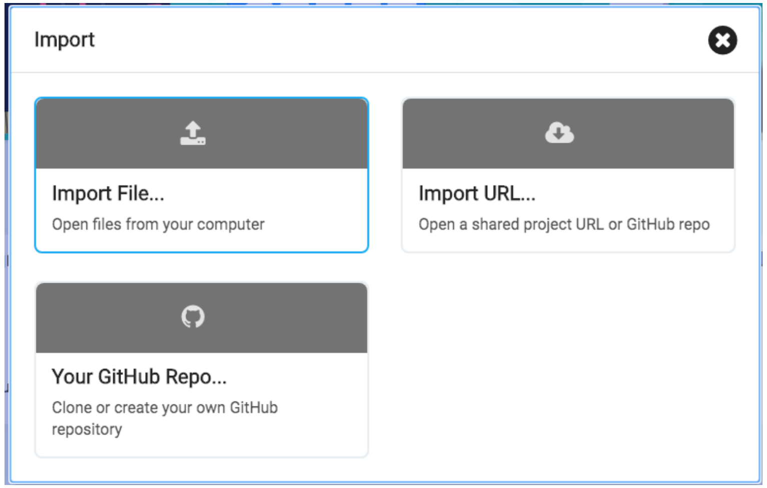
Coding



- Use the code in MakeCode, on this webpage **https://makecode.microbit.org**

- Now we are going to load the code into the program, so that we can modify it

- To do this, you either have the MakeCode home page up, find the folder where the file was downloaded to (usually downloads), hold left-click on the file you want to use, and then drag the mouse over to the MakeCode home page, then let go of left-click. Then the code will be loaded by itself. Or by pressing the "import" button on the right of the page, click on "import file…", click on "choose file", find the folder where the code was downloaded to, double click, and the page will load the code



Import

Import File...

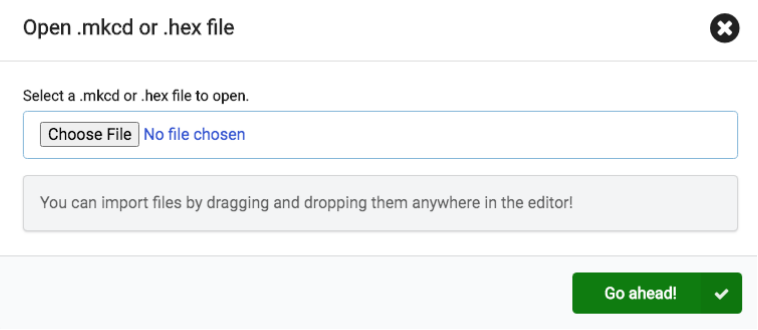
Open files from your computer

Import URL...

Open a shared project URL or GitHub repo

Your GitHub Repo...

Clone or create your own GitHub repository

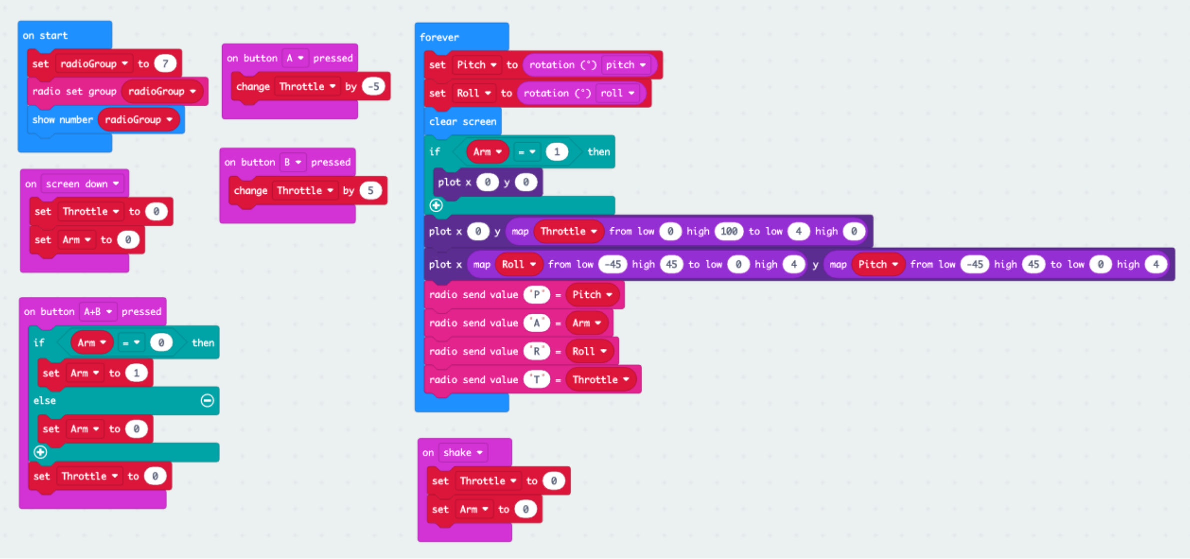


Open .mkcd or .hex file

Select a .mkcd or .hex file to open.

You can import files by dragging and dropping them anywhere in the editor!

Coding

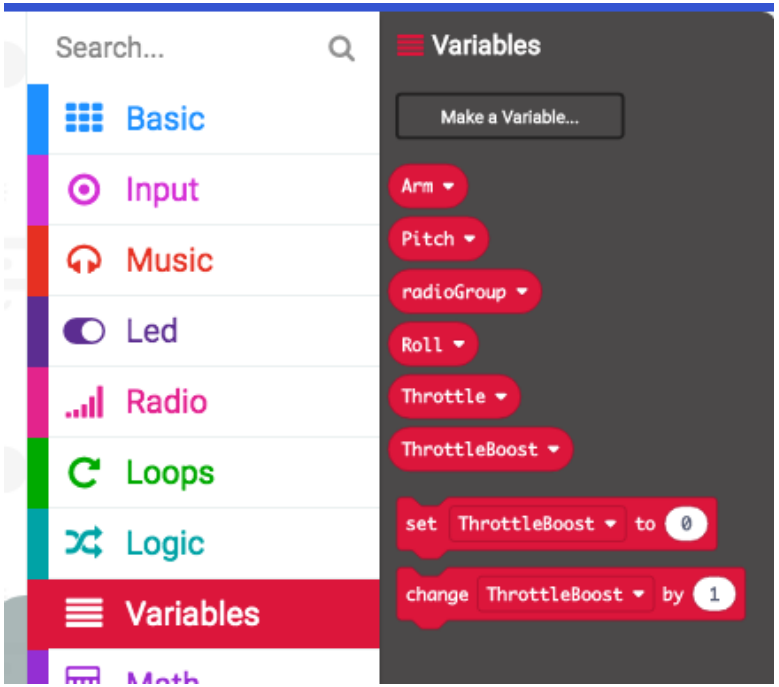


```
on start
    set radioGroup to 7
    radio set group radioGroup
    show number radioGroup

on screen down
    set Throttle to 0
    set Arm to 0

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 -5

on button B pressed
    change Throttle by 5

forever
    set Pitch to rotation (°) pitch
    set Roll to rotation (°) roll
    clear screen
    if Arm = 1 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 map Pitch from low -45 high 45 to low 0 high 4
    radio send value "P" = Pitch
    radio send value "A" = Arm
    radio send value "R" = Roll
    radio send value "T" = Throttle

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

- This is what the finished code for the Air:bit Drone looks like, and we will modify this code, in the Forever block, and the code can be downloaded from https://www.makekit.no/airbit-docs

Variables

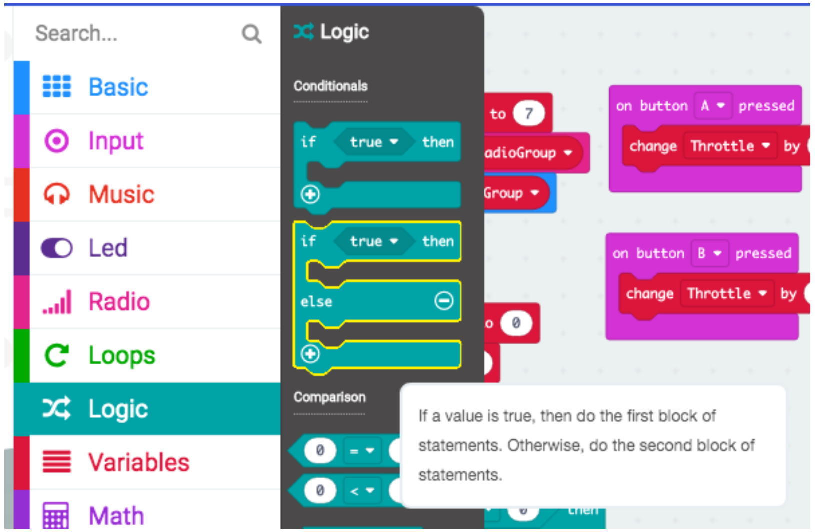
- Here you see the variables that have been entered before. We only need one extra: “ThrottleBoost”
- You do this by pressing Make a Variable, in Variables, then enter the name of the variable, in this case "ThrottleBoost", and press ok



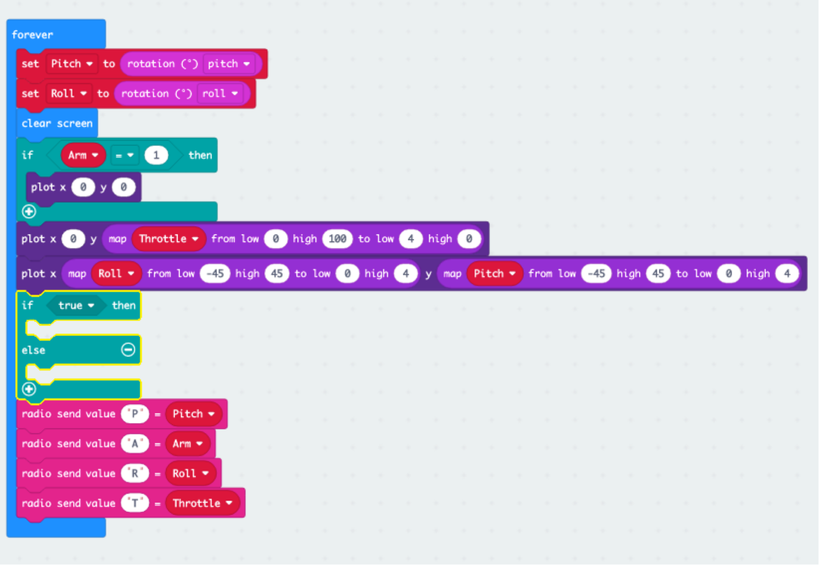
Logic

- Then we need logic, you will find it in the Logic block, and the block we are looking for is the "If true then else" block

- Place the “If true then else” Logic, in the Forever loop, under plot, and over the blocks with “radio send value”



If a value is true, then do the first block of statements. Otherwise, do the second block of statements.

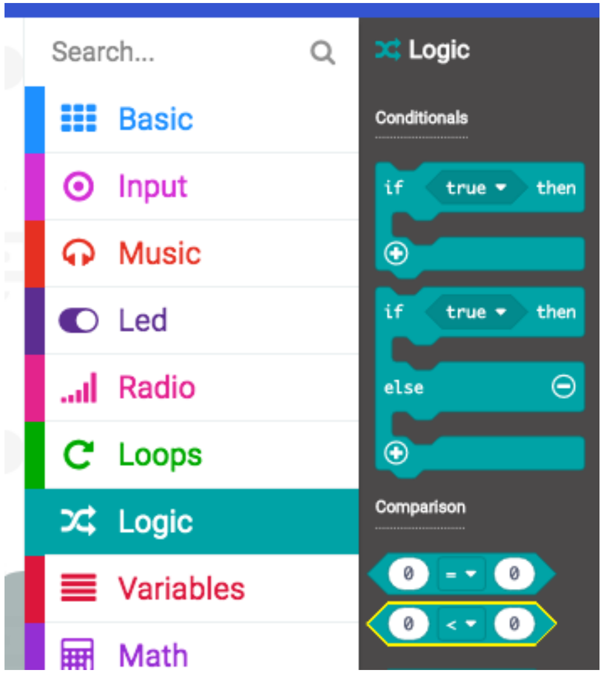


```
forever
  set Pitch to rotation (°) pitch
  set Roll to rotation (°) roll
  clear screen
  if Arm = 1 then
    plot x 0 y 0
  plot x 0 y map Throttle from low 0 high 100 to low 4 high 0
  plot x map Roll from low -45 high 45 to low 0 high 4 y map Pitch from low -45 high 45 to low 0 high 4
  if true then
  else
  radio send value "P" = Pitch
  radio send value "A" = Arm
  radio send value "R" = Roll
  radio send value "T" = Throttle
```

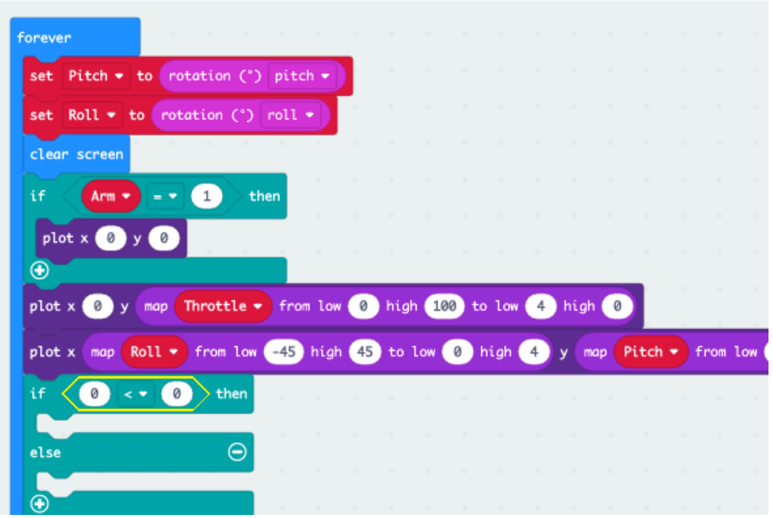
Logic

- Så trenger vi en “comparison”, den ligger også under Logic, det er blokken opplyst i gult vi trenger

- Then we need a "comparison", it`s also under the Logic Blocks, it is the block illuminated in yellow we need



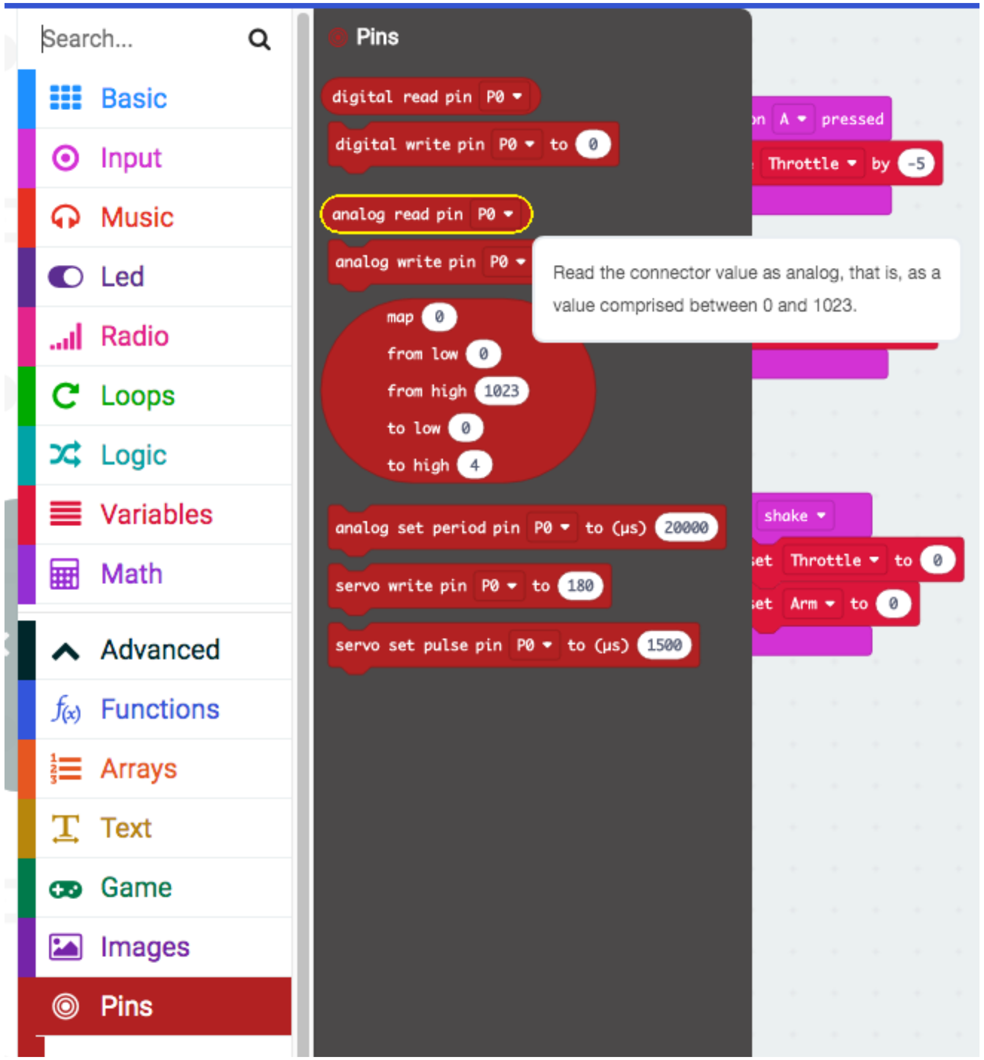
- Place "comparison" where it says true, then you will see that “true” disappears, and comparison is there instead



```
forever
  set Pitch to rotation (°) pitch
  set Roll to rotation (°) roll
  clear screen
  if Arm = 1 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 map Pitch from low
  if 0 < 0 then
  else
```

Logic and Pins

- Now we need Pins
  We are looking for the "Analog read pin P0" block



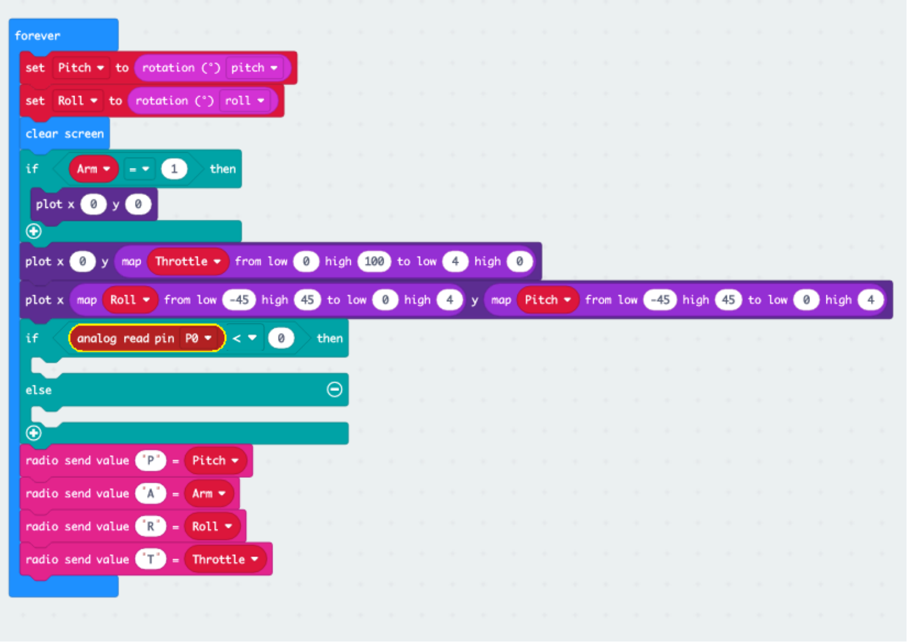
```
digital read pin P0
digital write pin P0 to 0
analog read pin P0
analog write pin P0
map 0
from low 0
from high 1023
to low 0
to high 4
analog set period pin P0 to (µs) 20000
servo write pin P0 to 180
servo set pulse pin P0 to (µs) 1500
```

Read the connector value as analog, that is, as a value comprised between 0 and 1023.

```
on A pressed
Throttle by -5
shake
et Throttle to 0
et Arm to 0
```

Pins

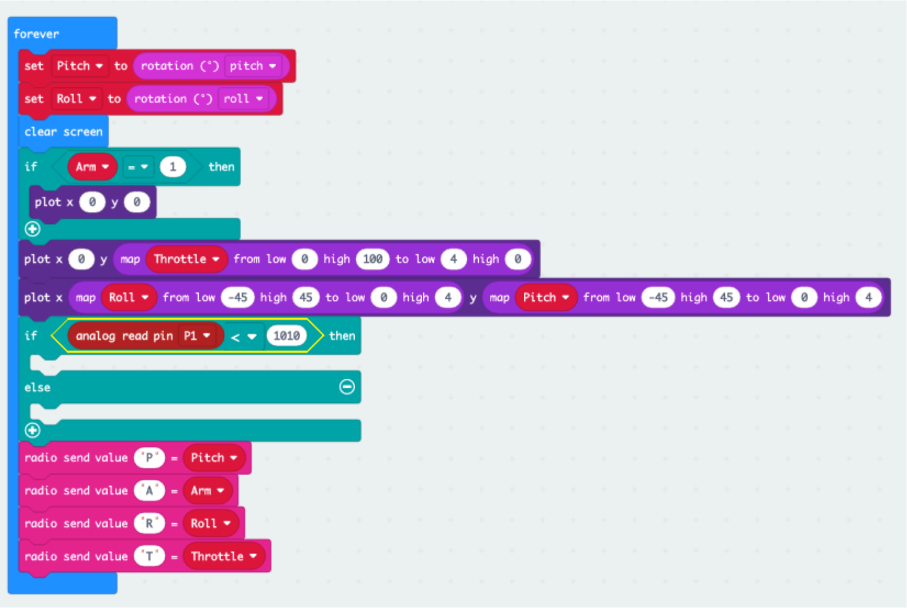
- Place the “analog read pin P0” block in the comparison block, on the right



```
forever
  set Pitch ▾ to rotation (°) pitch ▾
  set Roll ▾ to rotation (°) roll ▾
  clear screen
  if Arm ▾ = ▾ 1 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 map Pitch ▾ from low -45 high 45 to low 0 high 4
  if analog read pin P0 ▾ < ▾ 0 then
  else ⊖
  ⊕
  radio send value "P" = Pitch ▾
  radio send value "A" = Arm ▾
  radio send value "R" = Roll ▾
  radio send value "T" = Throttle ▾
```

- Change so that it says P1, and not P0
- Enter the value 1010 on the left

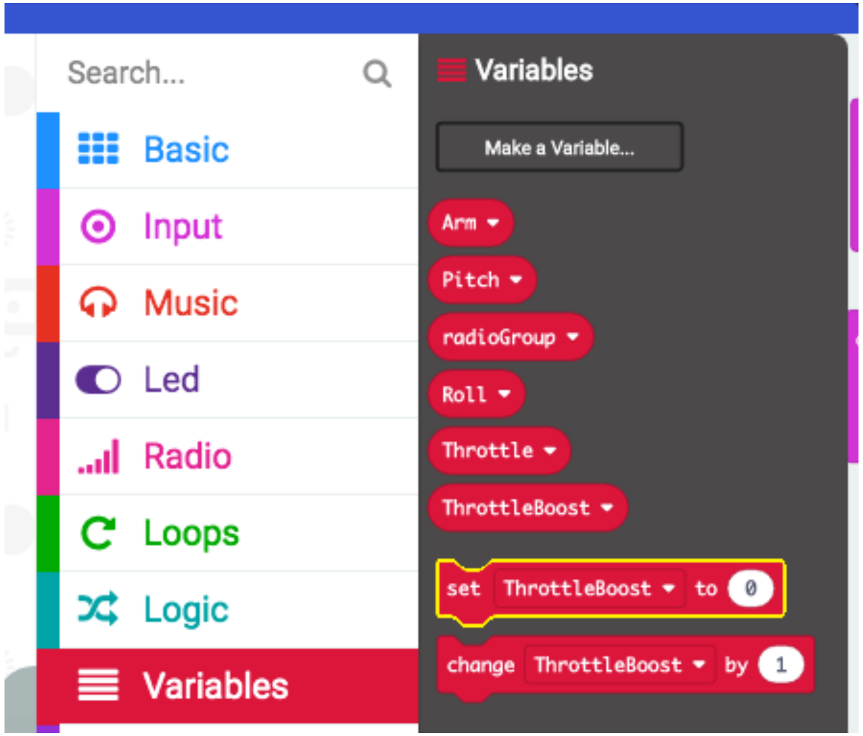
This is the value we found when we tested the finger sensor. The value 1010 will vary depending on how sweaty you are in your hand, so be aware of that



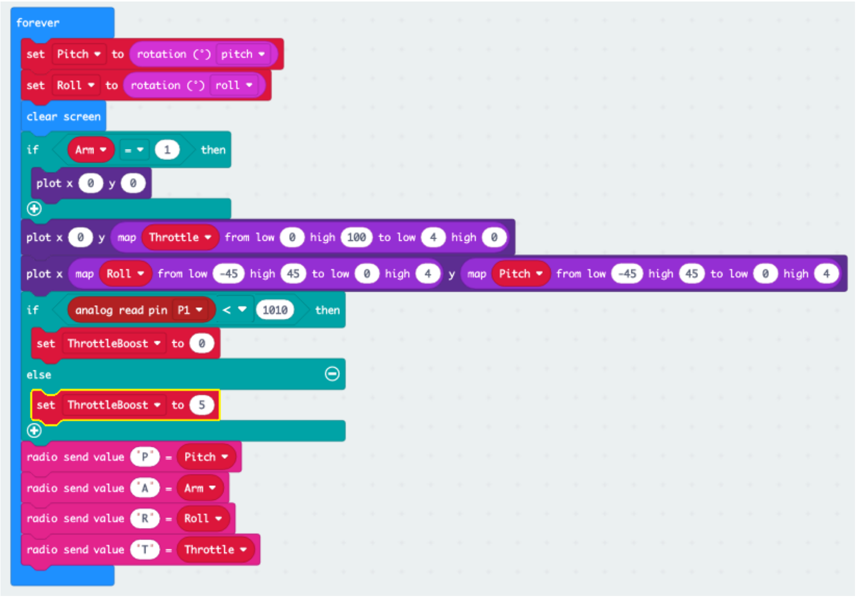
```
forever
  set Pitch ▾ to rotation (°) pitch ▾
  set Roll ▾ to rotation (°) roll ▾
  clear screen
  if Arm ▾ = ▾ 1 then
    plot x 0 y 0
  ⊕
  plot x 0 y map Throttle ▾ from low 0 high 100 to low 4 high 0
  plot x map Roll ▾ from low -45 high 45 to low 0 high 4 y map Pitch ▾ from low -45 high 45 to low 0 high 4
  if analog read pin P1 ▾ < ▾ 1010 then
  else ⊖
  ⊕
  radio send value "P" = Pitch ▾
  radio send value "A" = Arm ▾
  radio send value "R" = Roll ▾
  radio send value "T" = Throttle ▾
```

Variables

- Click on variables, and find the "ThrottleBoost" variable



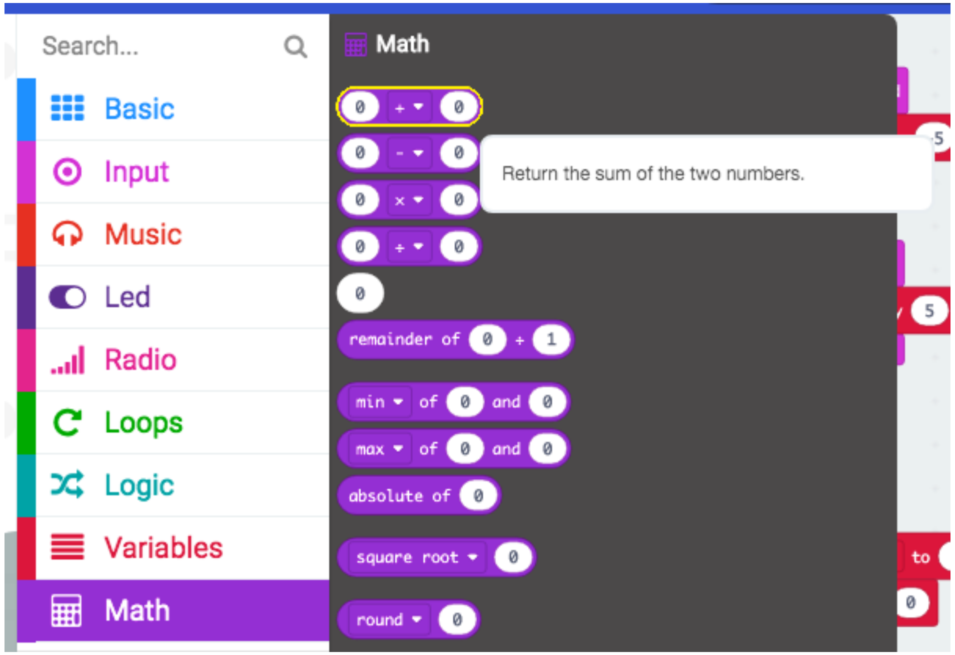
- Place the set “ThrottleBoost to 0” directly below the analog read pin P1
- Place the set "ThrottleBoost to 0", just below the “else”
- Change the value of the bottom variable to 5



```
forever
  set Pitch to rotation (°) pitch
  set Roll to rotation (°) roll
  clear screen
  if Arm = 1 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 map Pitch from low -45 high 45 to low 0 high 4
  if analog read pin P1 < 1010 then
    set ThrottleBoost to 0
  else
    set ThrottleBoost to 5
  radio send value "P" = Pitch
  radio send value "A" = Arm
  radio send value "R" = Roll
  radio send value "T" = Throttle
```

Math

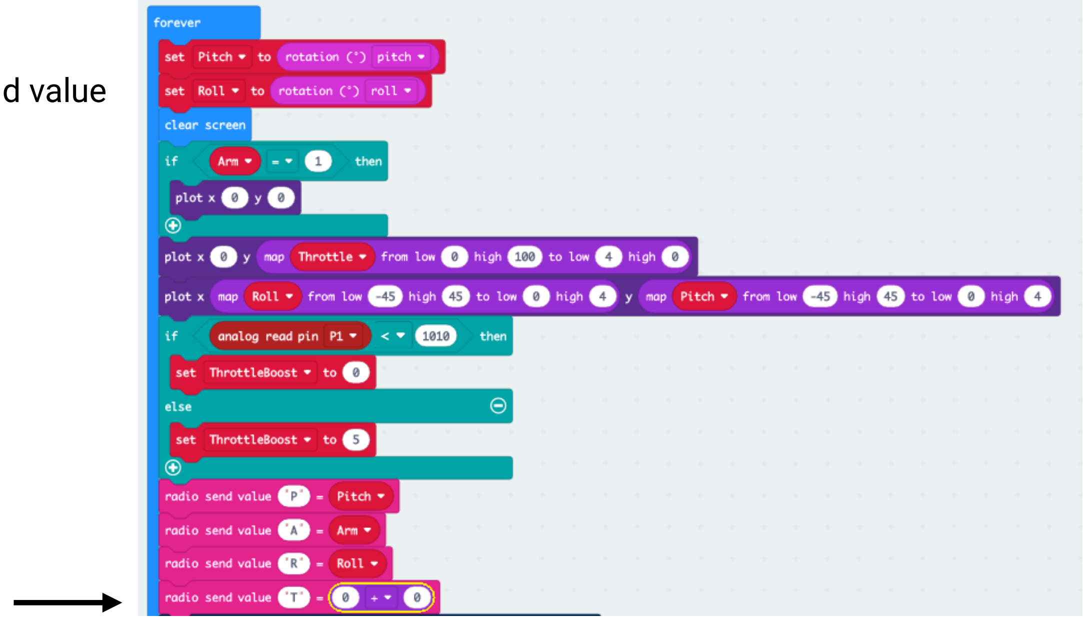
- Now we need a math block with a + sign



```
0 + 0
0 - 0
0 × 0
0 ÷ 0
0
remainder of 0 ÷ 1
min of 0 and 0
max of 0 and 0
absolute of 0
square root 0
round 0
```

Return the sum of the two numbers.

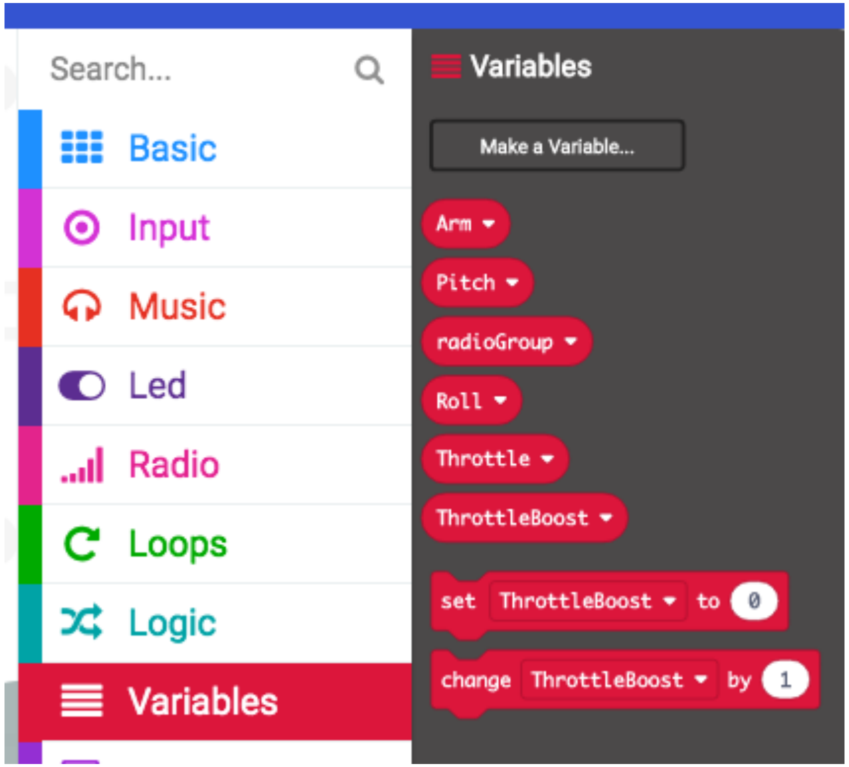
- Place it where the number 0 is displayed, in the radio send value “T” - block



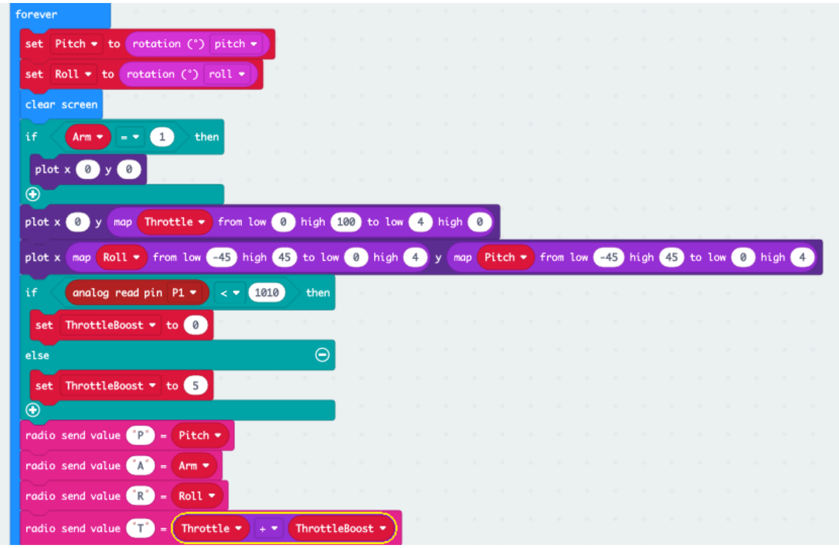
```
forever
  set Pitch to rotation (°) pitch
  set Roll to rotation (°) roll
  clear screen
  if Arm = 1 then
    plot x 0 y 0
  ⊕
  plot x 0 y map Throttle from low 0 high 100 to low 4 high 0
  plot x map Roll from low -45 high 45 to low 0 high 4 y map Pitch from low -45 high 45 to low 0 high 4
  if analog read pin P1 < 1010 then
    set ThrottleBoost to 0
  else ⊖
    set ThrottleBoost to 5
  ⊕
  radio send value "P" = Pitch
  radio send value "A" = Arm
  radio send value "R" = Roll
  radio send value "T" = 0 + 0
```

Variables

- Now we need two variables, a "Throttle", and a "ThrottleBoost"



- Place “Throttle” in the Math variable we just entered, in the first 0 box, in the radio send value “T” block
- Place "ThrottleBoost" in the last section of the Math block

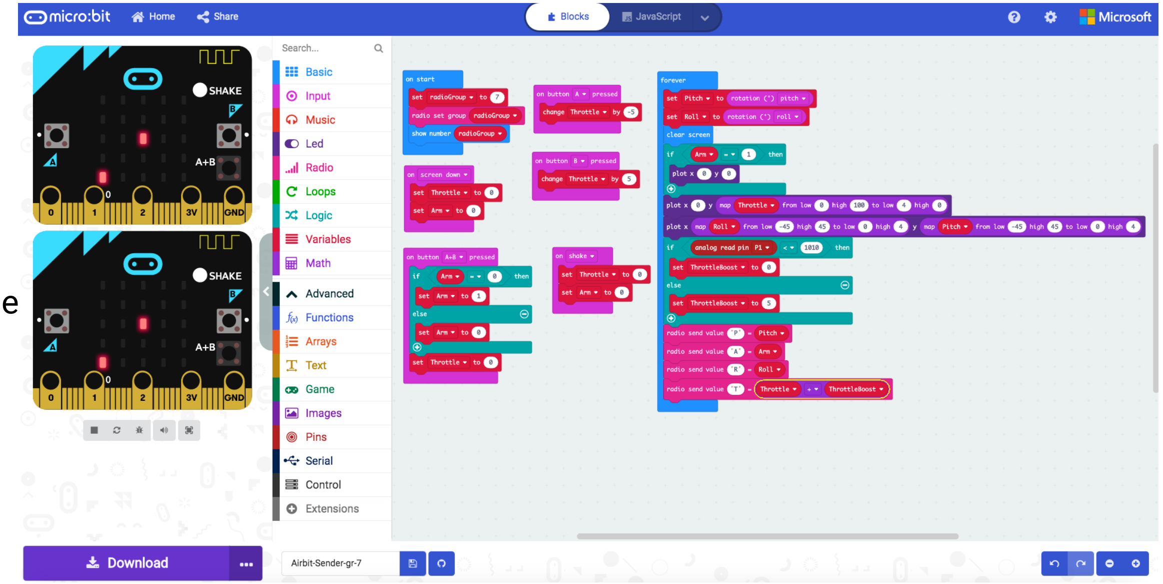


```
forever
  set Pitch to rotation (°) pitch
  set Roll to rotation (°) roll
  clear screen
  if Arm = 1 then
    plot x 0 y 0
  plot x 0 y map Throttle from low 0 high 100 to low 4 high 0
  plot x map Roll from low -45 high 45 to low 0 high 4 y map Pitch from low -45 high 45 to low 0 high 4
  if analog read pin P1 < 1010 then
    set ThrottleBoost to 0
  else
    set ThrottleBoost to 5
  radio send value "P" = Pitch
  radio send value "A" = Arm
  radio send value "R" = Roll
  radio send value "T" = Throttle + ThrottleBoost
```

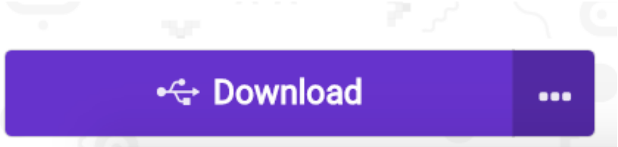
Variables - and Download

- Now the code is complete and we are ready to connect the Micro:bit to the Mac/PC and download the code.

If the finger sensor does not respond, it may be because the variable entered in "analog read pin P1" is faulty. Then you have to go back to testing the finger sensor, to find the correct value.

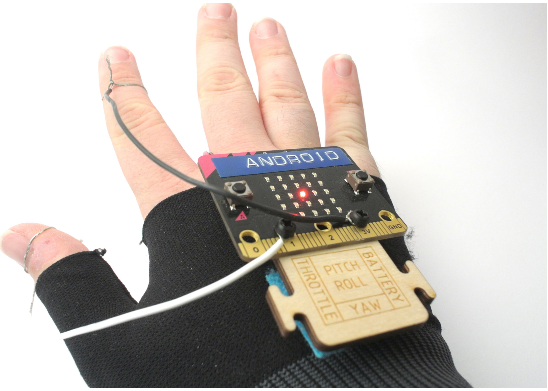


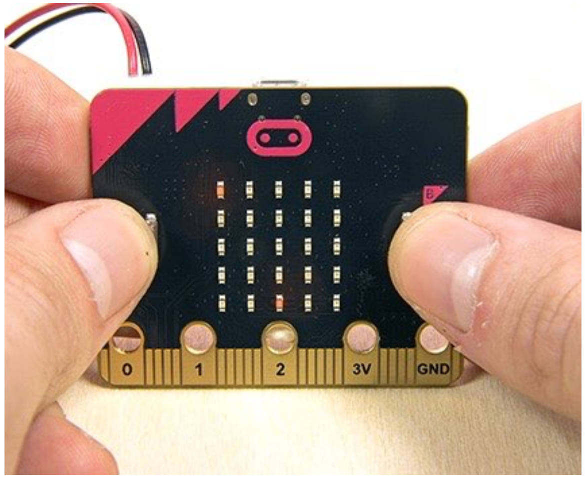
```
on start
    set radioGroup to 7
    radio set group radioGroup
    show number radioGroup

on button A pressed
    change Throttle by -5

on button B pressed
    change Throttle by 5

on screen down
    set Throttle to 0
    set Arm to 0

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

on shake
    set Throttle to 0
    set Arm to 0

forever
    set Pitch to rotation (°) pitch
    set Roll to rotation (°) roll
    clear screen
    if Arm = 1 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 map Pitch from low -45 high 45 to low 0 high 4
    if analog read pin P1 < 1010 then
        set ThrottleBoost to 0
    else
        set ThrottleBoost to 5
    radio send value "P" = Pitch
    radio send value "A" = Arm
    radio send value "R" = Roll
    radio send value "T" = Throttle + ThrottleBoost
```



How to use the Glove

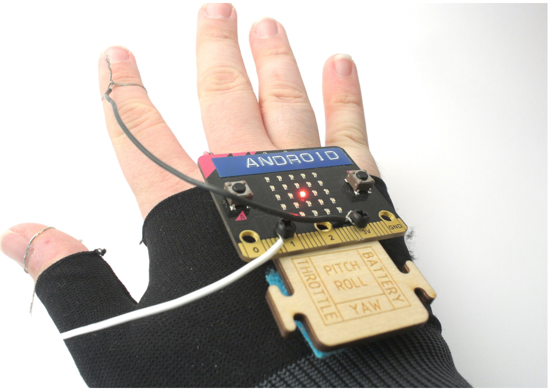
- Turn on the Micro:bit by connecting the cable from the battery holder to the Micro:bit
- Place the drone in the middle of an open area and turn on the power
- Turn on the drone, as usual
- Press A and B at the same time to start the engines
- To gain more Throttle, press B, to get less Throttle, press A
- Increase the throttle until the drone begins to rise and reach a suitable height (1-2 meters)
- Adjust the Throttle to the height as it drops
- Use ThrottleBoost, by squeezing the finger sensors together, the drone will then climb. When the drone has climbed enough, you release ThrottleBoost (let go of the cables), and the Throttle will then return to the level it was, before giving a ThrottleBoost



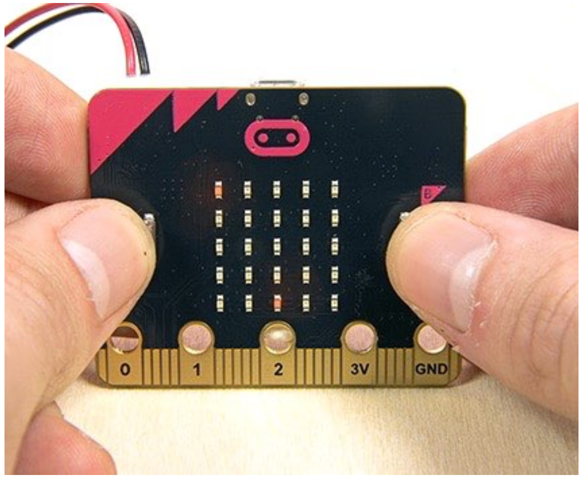


How to use the Glove

- You control which direction you want the Air:bit to go, by giving a hand movement, where you tilt your hand, in the direction you want the Air bit to fly
  - Ex. a hand movement straight ahead will cause the Air:bit to fly straight ahead
  - Ex. a hand movement to the left will cause the Air:bit to fly to the left

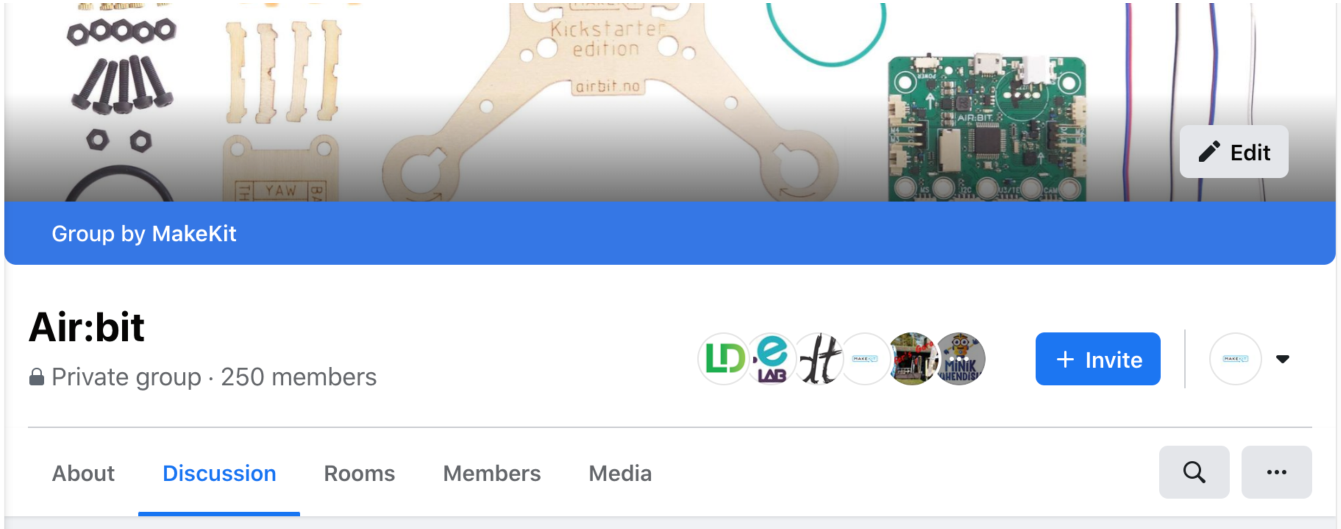


- You turn off the Air:bit as before, by pressing A and B at the same time, or by activating the panic function by shaking the remote control



Contact us:

Get advice and hints in our facebook group
www.facebook.com/groups/goairbit/



Group by MakeKit

Air:bit

Private group · 250 members

⌂ www.makekit.no ✉ support@makekit.no f makekit gomakekit (also twitter)