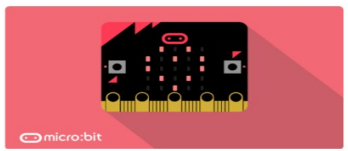
Robotics with Lego EV3 + MS Makecode

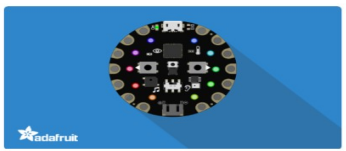
Andrea Sterbini – sterbini@di.uniroma1.it

Microsoft Makecode.com

Many development systems supported (embedded/robotics/game)

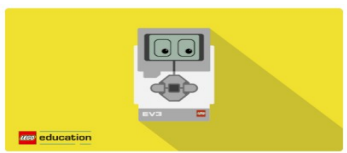


micro:bit

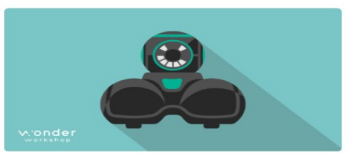


Adafruit

Minecraft

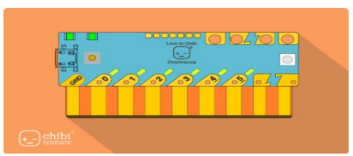


Lego EV3



Cue

Arcade



Chibi chip

Blockly-based visual programming

More systems in https://makecode.com/labs

MS Makecode: EV3 robotics

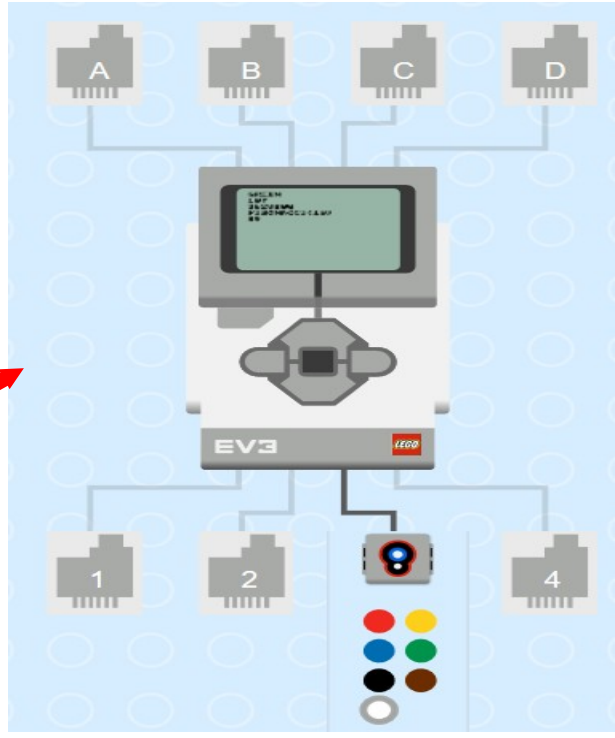
https://makecode.mindstorms.com

SET-UP
- just upgrade the EV3 firmware to 1.10E
- the IDE runs in the browser

DEPLOY THE CODE
- EV3 is seen as a disk when connected by USB
- Just download the generated file to the EV3

EXECUTION
- the program RUNS STRAIGHT ON THE BRICK

DEBUG
- browser-based minimal simulator
  (with AUTO-CONFIGURED connections)



Makecode standard block features

Based on TypeScript (typed Javascript)

Types!!!	integers, strings, floats, lists	
Lists of any?	YES	
Functions?	YES	
Function args?	NO	(YES in TypeScript mode)
Return?	NO	(YES in TypeScript mode)
Variables?	GLOBAL	(LOCAL in TypeScript mode)
Messages?	YES	
Message params.?	YES	
Static TypeScript?	YES	(NEW!!!!)

Makecode EV3-specific blocks

Brick buttons:
- on button XXX pressed event
- pause until ...
- is button … ?

Brick LCD screen:
- clear, show image, show text
  show number, show port

Touch sensor:
- on touch XXX event
- pause until ...
- is touch … ?

NO BLUETOOTH

Color sensor:
- on color XXX detected event
- on color sensor X dark/light
- pause until …
- color

Ultrasonic sensor:
- on US X object detected
- pause until …
- distance

Gyroscope sensor:
- rate, angle, reset

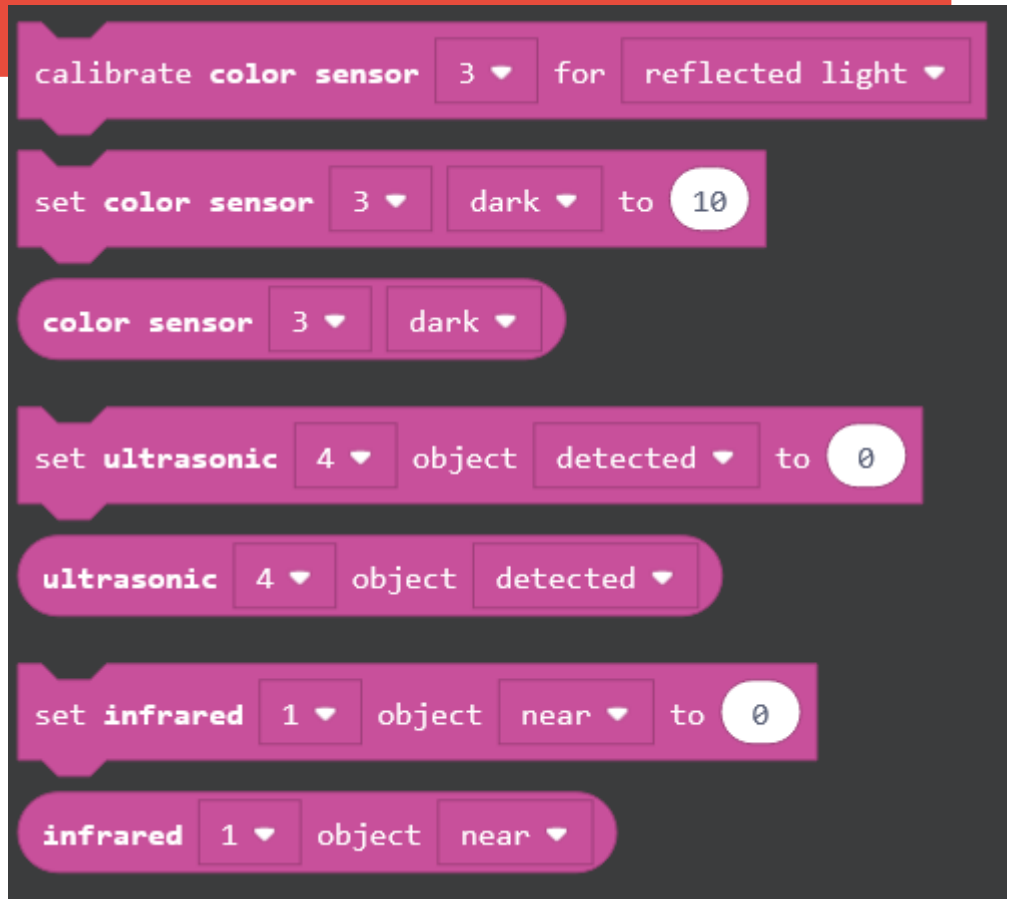
EV3 Sensor Calibration blocks

Calibrate color sensor XXX for reflected/ambient light

Set color sensor XXX dark/bright to THRESHOLD

Set ultrasonic sensor XXX object detected/near to THRESHOLD

Set infrared sensor XXX object detected/near to THRESHOLD



EV3 Motors (with coordinated differential control)

Run motor X/XY at V speed for N rotations/degrees/seconds/msec

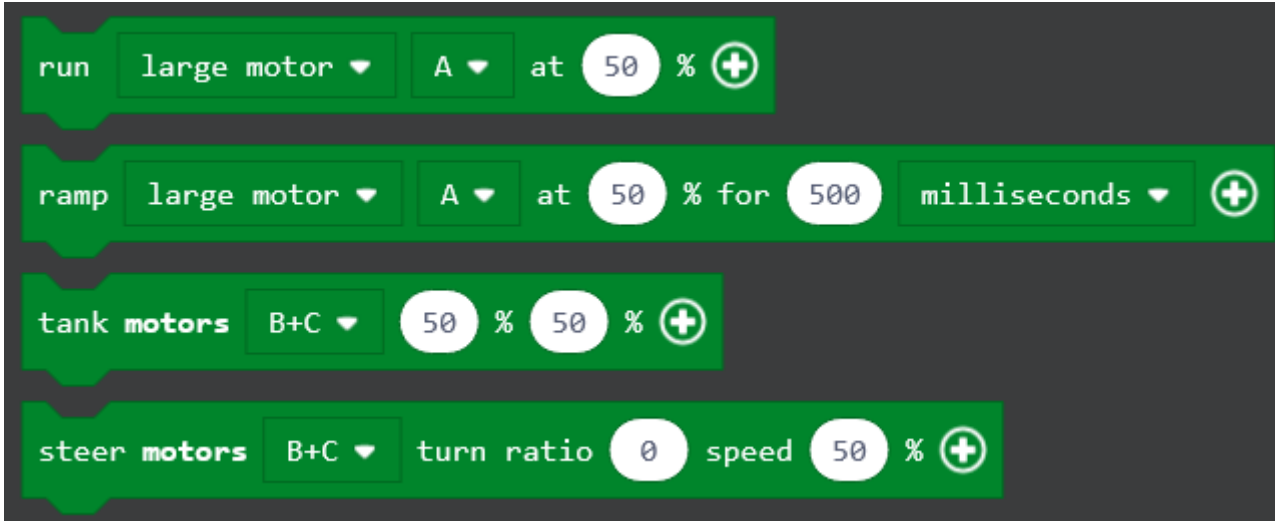
Drive motors XY at V1,V2 speeds for N rot/deg/sec/msec

Steer motors XY at Y ratio V speed for N rot/deg/sec/msec

Pause until motor X/XY ready

Read Motor X speed/angle

Set motor X brake/pause/
inverted/regulated ON/OFF

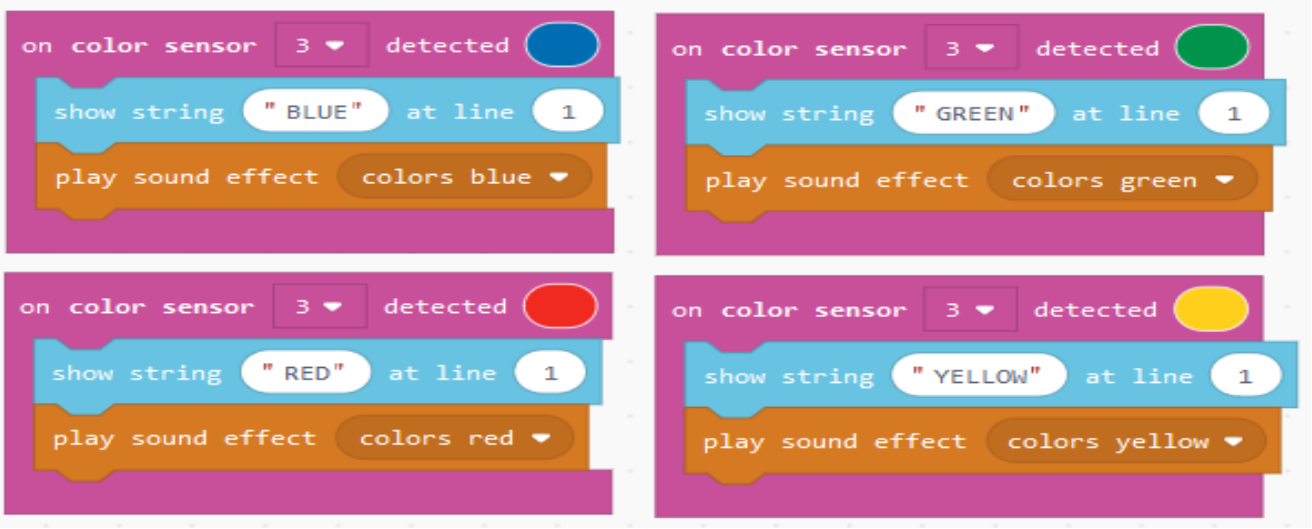


Control flow (blocks)

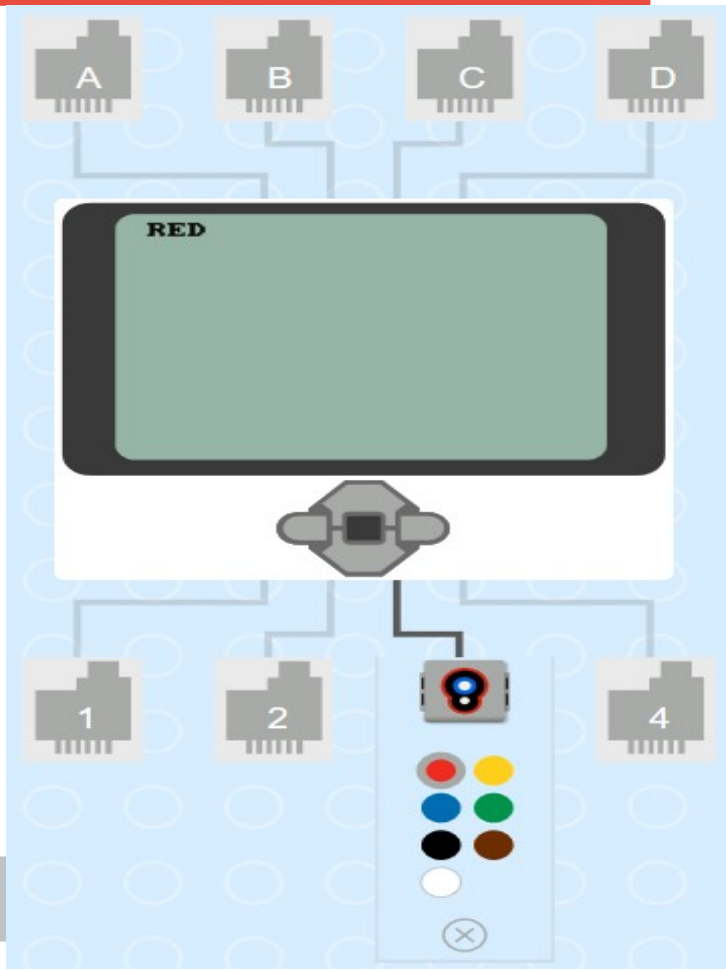
One main thread	NO MULTI
One forever loop	NO MULTI
Sensor events	ONE EACH
Counted loops?	YES
Foreach?	YES
Do-while?	NO
While-do?	YES

Parallel threads? (“run in parallel” block)	EXPLICIT
Wait for all threads?	YES
New (numeric) events?	YES
Parametric events?	YES
Wait for event?	YES
Timers?	YES
Messages? (with the “Broadcast” extension)	YES

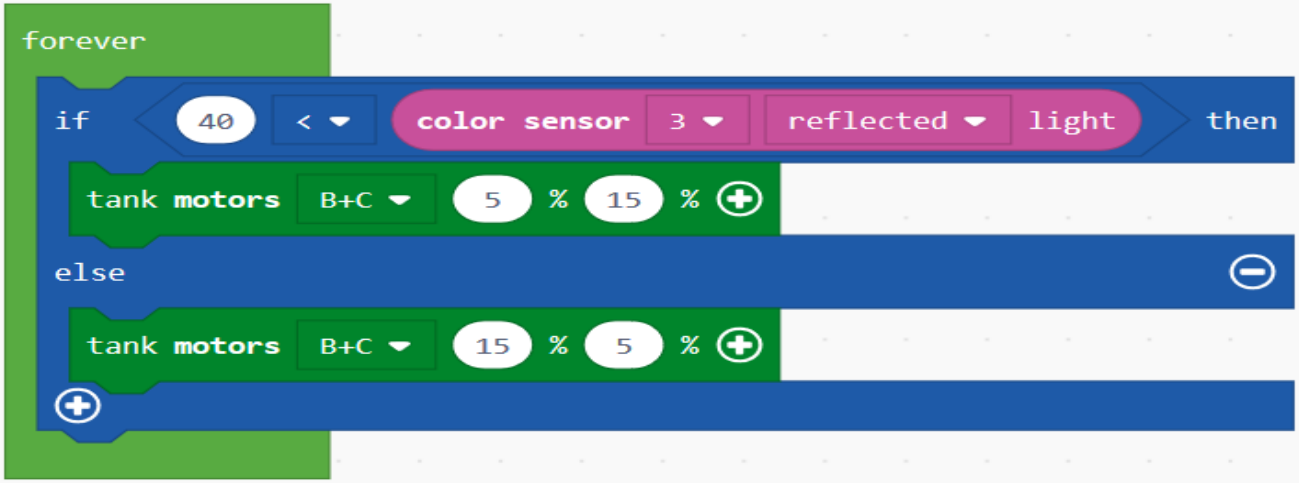
Color recognizer example



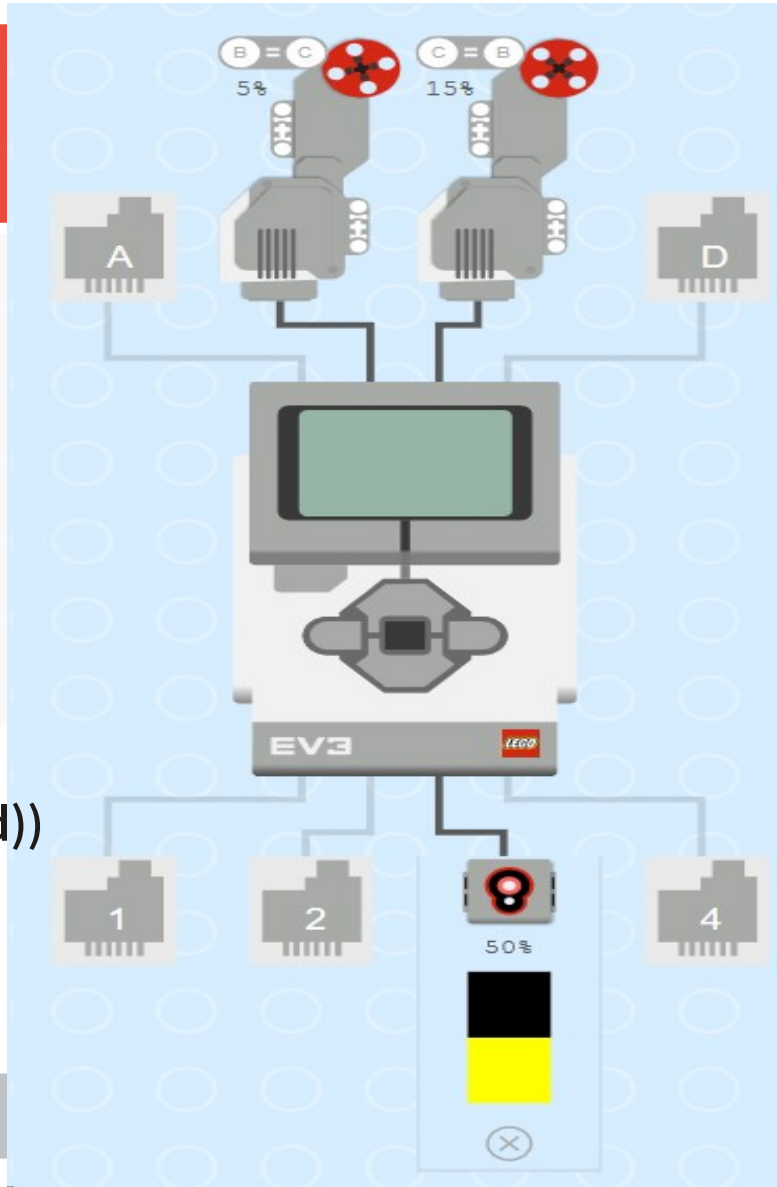

```
sensors.color3.onColorDetected(ColorSensorColor.Blue, function (){
    brick.showString("RED", 1);
    music.playSoundEffect(sounds.colorsBlue);
})
```



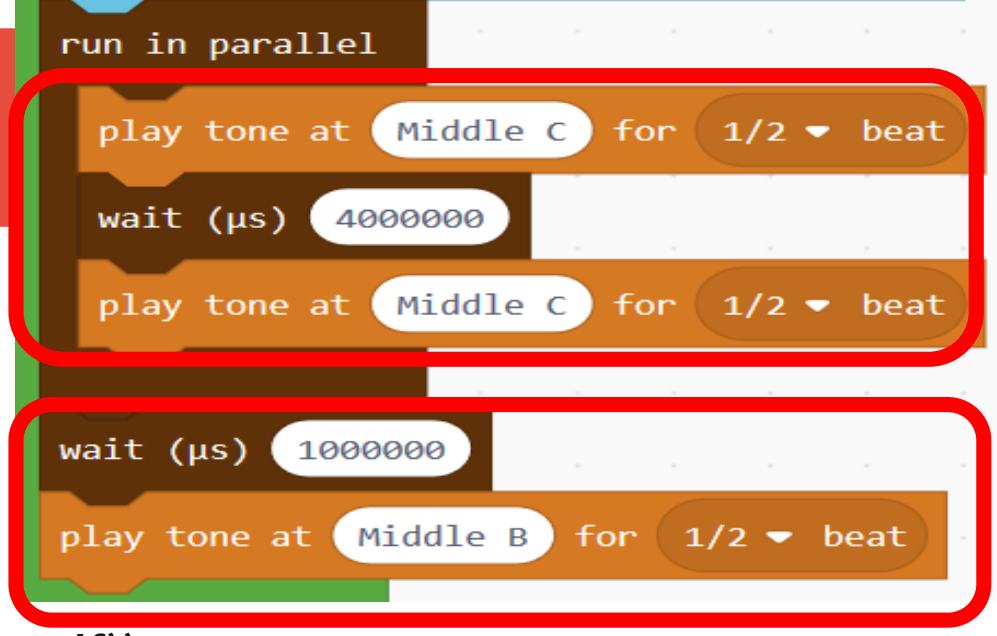
Line follower example




```
forever(function () {
    if (40 < sensors.color3.light(LightIntensityMode.Reflected))
                {               motors.largeBC.tank(5, 15)    }
else{       motors.largeBC.tank(15, 5)      }})
```



Parallel thread example



“run in a parallel/different thread”

<u>In parallel do</u>:

- beep, wait then beep (other thread)
- wait then beep

```
control.runInParallel(function () {
    music.playTone(262, music.beat(BeatFraction.Half))
    control.waitMicros(4000000)
    music.playTone(262, music.beat(BeatFraction.Half))
})
control.waitMicros(1000000)
music.playTone(494, music.beat(BeatFraction.Half))
```

TypeScript mode

Editor with colour highlight, autocompletion and documentation

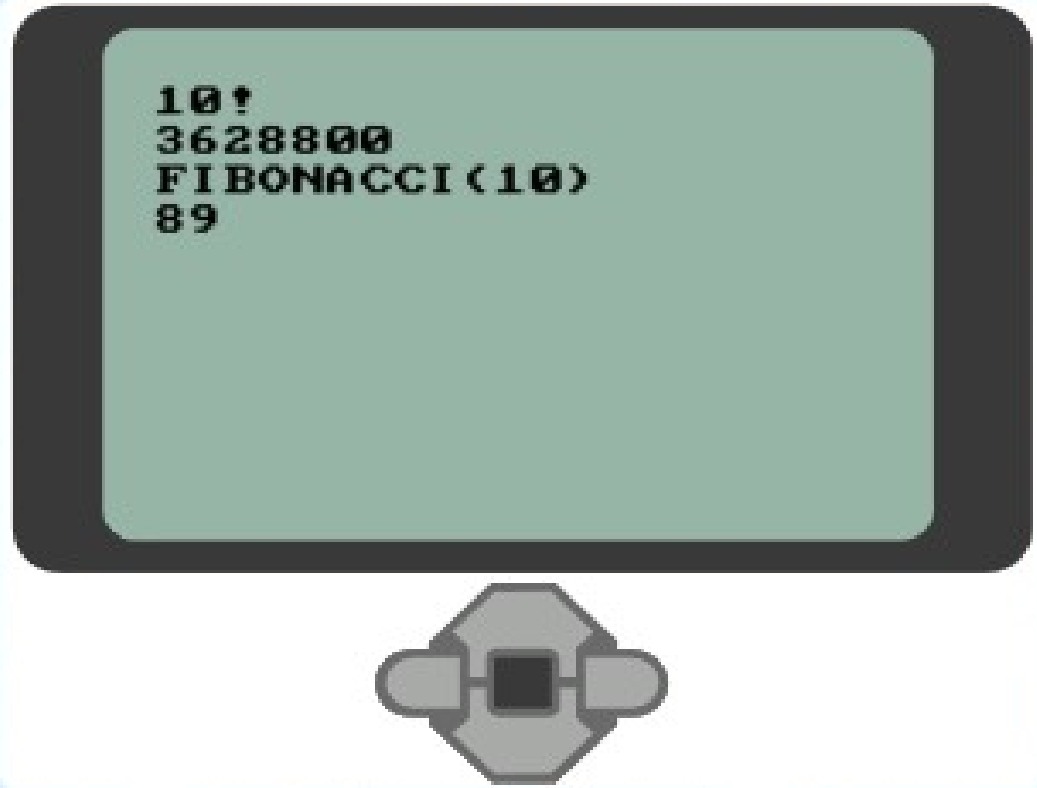
Static TypeScript (Typed JavaScript)

Object-oriented! (to be investigated)

A sequence of statements and declarations
FOLLOWED by an infinite loop

Static Python in a near future?

Recursion example with Typescript functions




```
on start
function factorial(N: number): number {
    if (N < 2)  return 1
    else        return N * factorial(N - 1)
}
function fibonacci(N: number): number {
    if (N < 2)  return 1
    else        return fibonacci(N - 2) + fibonacci(N - 1)
}
show string "10!" at line 2
show number factorial(10) at line 3
show string "FIBONACCI(10)" at line 4
show number fibonacci(10) at line 5
```

Extensions can be loaded in the editor

MESSAGES! (“Broadcast” extension)

- onMessage XXX Received EVENT
- sendMessage XXX
- sendMessage XXX andPause

STORAGE! (read/save files on USB stick)

- permanent / temporary
- TXT or CSV files

BUT: they are NAMED Messages without value
(You could emulate Message passing with GLOBAL vars)

AUTOMATION!
- use a PID (Proportional Integral Derivative controller) to control a robot
- behavior-based control (unfortunately no documentation or examples are available)

Demo

https://makecode.mindstorms.com

DEMO