Robotics with Lego EV3 + Scratch

cc BY NC SA

Andrea Sterbini – sterbini@di.uniroma1.it

Robotics: a very compelling problem setting

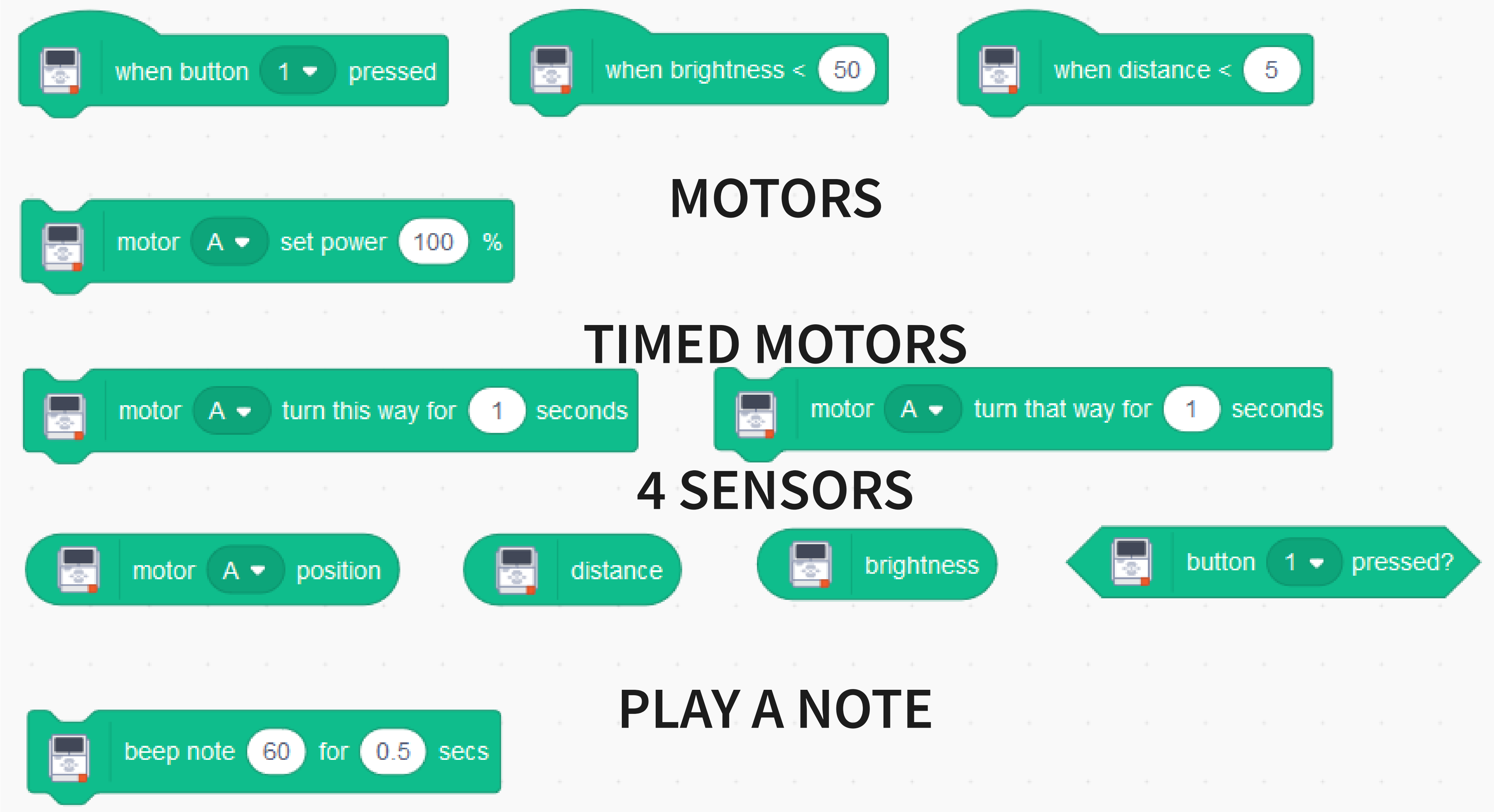
Using robots with kids allows you to:

- enhance motivation
- show concrete evidence of the kid programs’ actions
- tackle CONCURRENT problems
    - read sensors WHILE moving
    - coordinate the movement of many motors (drive/arms)

Scratch is VERY limited but, with Lego EV3, sufficient to build:

- a car moving in a labyrinth
- a robotic arm
- ...

Scratch Lego Mindstorms EV3 extension



3 NEW EVENTS

when button 1 pressed

when brightness < 50

when distance < 5

MOTORS

motor A set power 100 %

TIMED MOTORS

motor A turn this way for 1 seconds

motor A turn that way for 1 seconds

4 SENSORS

motor A position

distance

brightness

button 1 pressed?

PLAY A NOTE

beep note 60 for 0.5 secs

LIMITS of Lego EV3 Scratch extension: SENSORS

You can use ONLY ONE sensor for:

- DISTANCE (Ultrasound sensor)
    - DISTANCE IN INCHES???
- BRIGHTNESS (Light/colour sensor)
    - VERY LOW VALUES!!!

The BUTTON-PRESSED event is somewhat erratic

The DISTANCE-LESS-THAN event works better

The LIGHT-LESS-THAN event seems not to work!!! (!”$$”£!$!)

NOTICE:

the program runs in the browser and interacts with EV3 by BT

LIMITS of Lego EV3 Scratch extension: MOTORS

You can use UP TO 4

- motors on A, B, C, D ports
- touch sensors on ports 1, 2, 3, 4

BUT: CANNOT rotate one motor for a given angle (ONLY TIMED run)

- Calibration is very important (e.g. time vs distance)

MOTOR POSITION SEEMS NOT TO WORK WELL !$”$!”£|!$£|!

OTHER:

- BLUETOOTH IS “BLOODY TRICKY” … !|$!%£$!”%!”

CONCURRENCY and Robotics in Scratch

You must coordinate:

- many motors (at least 2 for differential movement)
- reading many sensors

You can define multiple threads for the same event/MESSAGE

- unfortunately, messages DO NOT carry arguments
    - (HACK: use global variables)

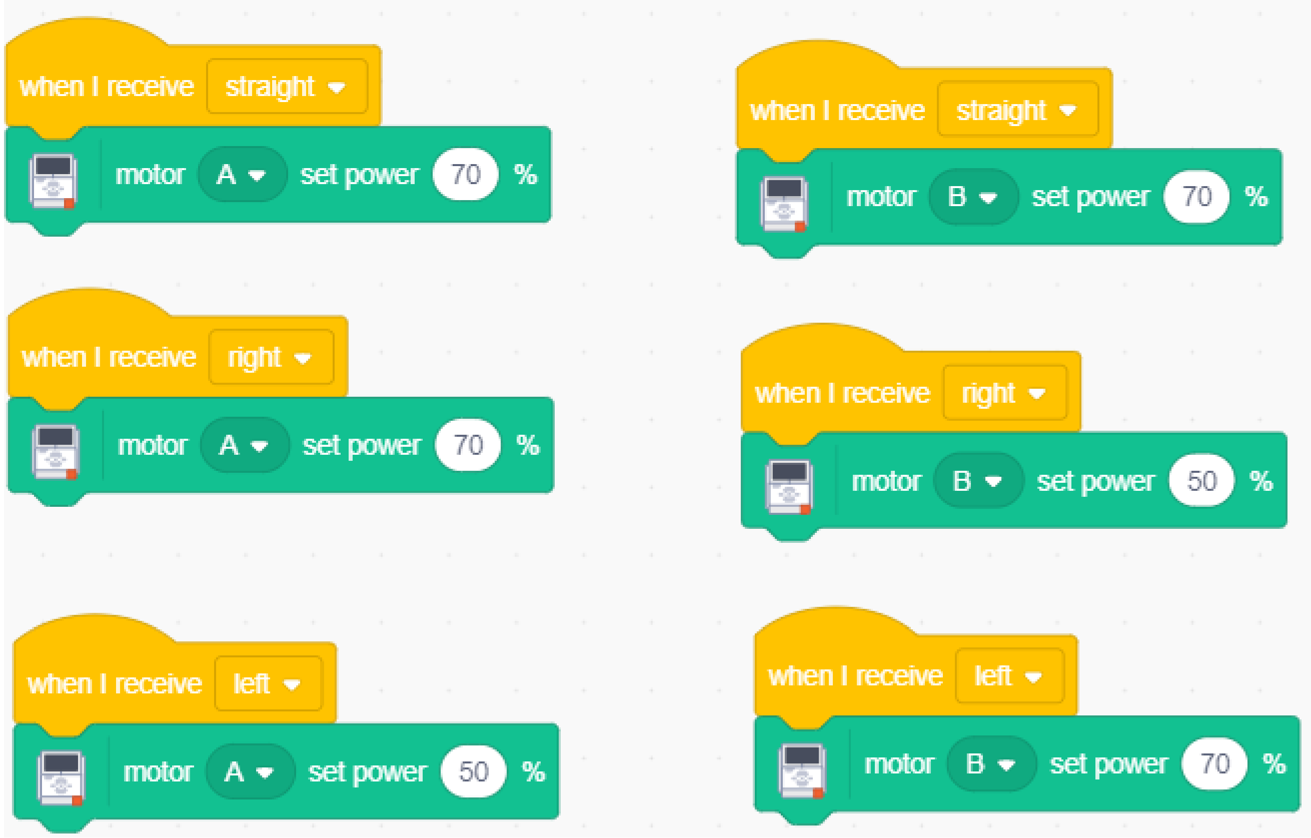
Choose a simple parametrization of single movements and JOIN

- time, speed, direction for each motor
- to get forward/backward, curves

Example: Line Follower with messages

Using messages to coordinate concurrent actions on both motors

MOTOR A MOTOR B

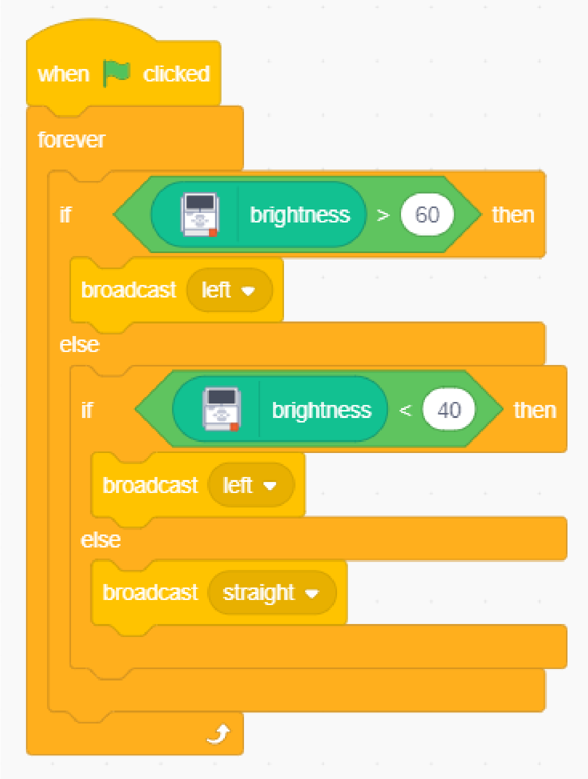


MOTOR A

```
when I receive [straight ▾]
motor (A ▾) set power (70) %

when I receive [right ▾]
motor (A ▾) set power (70) %

when I receive [left ▾]
motor (A ▾) set power (50) %
```

MOTOR B

```
when I receive [straight ▾]
motor (B ▾) set power (70) %

when I receive [right ▾]
motor (B ▾) set power (50) %

when I receive [left ▾]
motor (B ▾) set power (70) %
```



```
when 🏴 clicked
forever
    if <(brightness) > (60)> then
        broadcast [left ▾]
    else
        if <(brightness) < (40)> then
            broadcast [left ▾]
        else
            broadcast [straight ▾]
↻
```

Robots and Concurrency:
New factors to take into cosideration

You must calibrate movements and sensors values

Consider al possible compositions of concurrent actions
(and add semaphores to exclude incompatible actions)
(and remember to unlock semaphores)

Decide if an action should be Blocking/Non-blocking
i.e. if you must wait or not for message completion

DEMO