InO-Bot

Teacher Guide - Blockly

www.tts-shopping.com

Product Code: EL00483

InO-Bot

InO-Bot - full of inputs and outputs

Contents

Previous Experiences
Prior to using InO-Bot, children should have some experience of using Scratch and ideally experience of other robots such as Bee-Bot and Blue-Bot.

Progression
The activities outlined below are in a suggested order of progression. There is no specific amount of time to be spent on each as this will vary from one situation to another. It may also be necessary to break some of the activities down further to suit children's needs.

National Curriculum
The National Curriculum for Computing references listed below are indicative of some aspects of the curriculum the activities cover. They are not an exhaustive list nor do they indicate that one activity fully covers that curriculum area. The activities support children in learning computing skills and applying computational thinking.

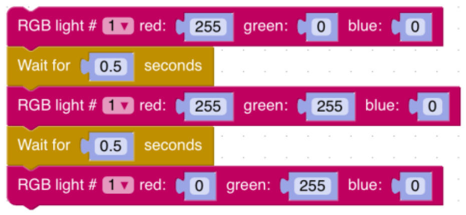
Attainment Targets

Key Stage 2
- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

Key Focus 1 – Getting started with InO-Bot

InO-Bot is full of inputs and outputs but it is best to start as simply as possible. Begin with outputs and get children to explore what can be done with the 8 RGB LED lights on top.

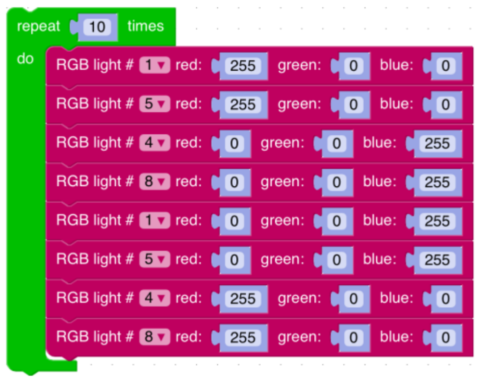
Use the ‘RGB light’ blocks and ‘wait for’ blocks to light one LED different colours. Investigate what happens if the ‘wait for’ time is changed or removed.



```
RGB light # 1 red: 255 green: 0 blue: 0
Wait for 0.5 seconds
RGB light # 1 red: 255 green: 255 blue: 0
Wait for 0.5 seconds
RGB light # 1 red: 0 green: 255 blue: 0
```

The LEDs are numbered 1 to 8 starting at the back right (InO-Bot facing away from you). Children could explore changing the LED number and work out which LED is which.

Introducing the concept of repetition will allow this activity to be developed. Children can explore lighting the LEDs in different orders to create alternative patterns. The example (right) could be seen as an emergency vehicle light pattern.

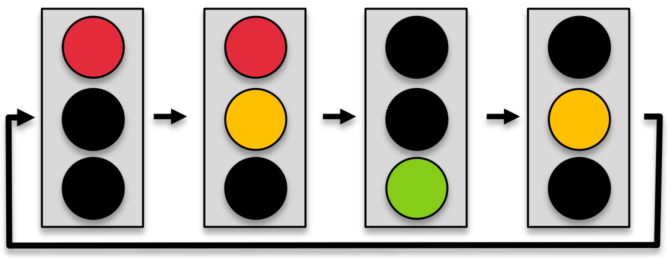


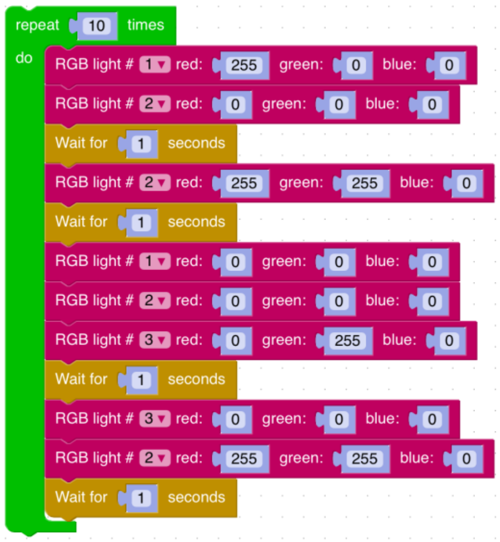
```
repeat 10 times
do  RGB light # 1 red: 255 green: 0 blue: 0
    RGB light # 5 red: 255 green: 0 blue: 0
    RGB light # 4 red: 0 green: 0 blue: 255
    RGB light # 8 red: 0 green: 0 blue: 255
    RGB light # 1 red: 0 green: 0 blue: 255
    RGB light # 5 red: 0 green: 0 blue: 255
    RGB light # 4 red: 255 green: 0 blue: 0
    RGB light # 8 red: 255 green: 0 blue: 0
```

Having explored different patterns, the next step could be to mimic something real world. Traffic lights are a great option. Ask children to write a program that lights LEDs 1, 2 and 3 as if they were traffic lights.

It helps to have an image of the sequence traffic lights cycle through. This will allow children to plan their program before they start work in Blockly.

.





```
repeat 10 times
do  RGB light # 1 red: 255 green: 0 blue: 0
    RGB light # 2 red: 0 green: 0 blue: 0
    Wait for 1 seconds
    RGB light # 2 red: 255 green: 255 blue: 0
    Wait for 1 seconds
    RGB light # 1 red: 0 green: 0 blue: 0
    RGB light # 2 red: 0 green: 0 blue: 0
    RGB light # 3 red: 0 green: 255 blue: 0
    Wait for 1 seconds
    RGB light # 3 red: 0 green: 0 blue: 0
    RGB light # 2 red: 255 green: 255 blue: 0
    Wait for 1 seconds
```

An example program for traffic lights is shown above.
Note: There will be a number of ways to achieve the same result.

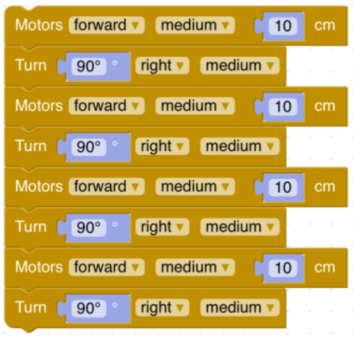
Extension Activity
To extend this activity, ask children to create a second set of lights in a counter sequence. These could be lights on cross-roads or where a road narrows to one lane with lights at either end. This can be achieved using the two sides of InO-Bot, i.e. LEDs 1, 2 and 3 against 8, 7 and 6.

Key Focus 2 – Getting into shape

InO-Bot has a lift and lower pen carrying mechanism. This feature is really useful to see where InO-Bot has moved. Drawing shapes is a good activity to develop programming skills using repetition and also maths knowledge.

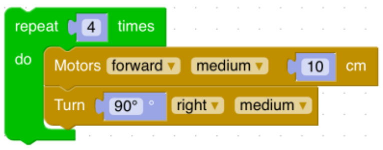
The first challenge is to make InO-Bot move in a square shape.

InO-Bot has large wheels and these may account for some small errors. Although not mathematically accurate, turns may need to be increased or decreased by a degree or two. The accuracy of movement may also be affected by the type of surface InO-Bot moves over.



```
Motors forward medium 10 cm
Turn 90° right medium
Motors forward medium 10 cm
Turn 90° right medium
Motors forward medium 10 cm
Turn 90° right medium
Motors forward medium 10 cm
Turn 90° right medium
```

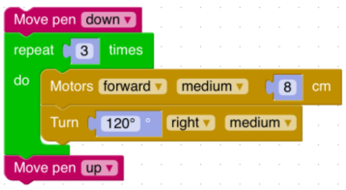
The example program (shown above) works but isn't very efficient. Using repetition can help reduce the number of instructions required. "Forwards 10 cm then spin right 90 degrees" is repeated four times. The example (right) shows what this would look like.



```
repeat 4 times
do  Motors forward medium 10 cm
    Turn 90° right medium
```

Now use InO-Bot's pen holder to see what shape has been drawn. What other regular polygons can the children create?

Try drawing an equilateral triangle. This is actually quite challenging and can provoke useful discussions about internal and external angles.



```
Move pen down
repeat 3 times
do  Motors forward medium 8 cm
    Turn 120° right medium
Move pen up
```

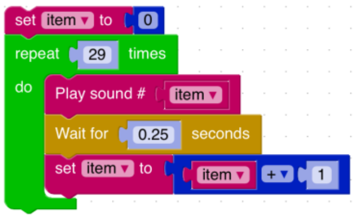
You may also wish to talk about the relationship between the number of sides and the angles, e.g.

A square - 4 sides, external angle 90°
A pentagon - 5 sides, external angle 72°
An octagon - 8 sides, external angle 45°
Tip: For each of the above multiply the number of sides by the angle.

Key Focus 3 – Name that tune

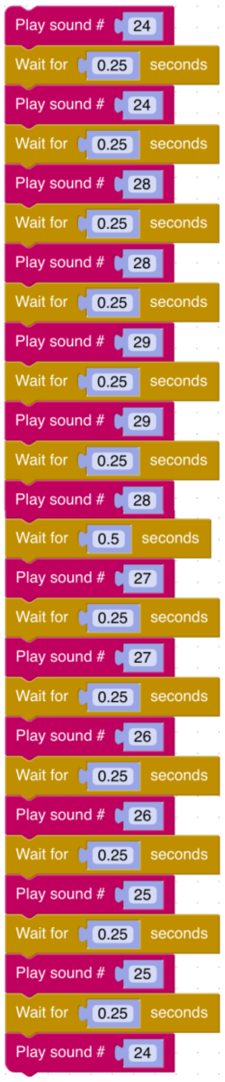
InO-Bot has 29 built-in sounds. Full details of these are listed in the user guide.

Begin by creating a simple loop to enable stepping through the sounds. The example (right) makes use of a variable. In the 'Variables' section is a default called 'item'. The program (shown right) sets 'item' to 0. It then plays that sound number. 'Item' is then incremented by 1 i.e. set to itself plus 1. The loop then plays the next sound etc.



```
set item to 0
repeat 29 times
do  Play sound # item
    Wait for 0.25 seconds
    set item to item + 1
```

Having listened to all the different sounds, a program could be written to play a tune. Below is a list of the musical notes InO-Bot can play. To the right is an example of a known tune. (Can you find out what it is?)

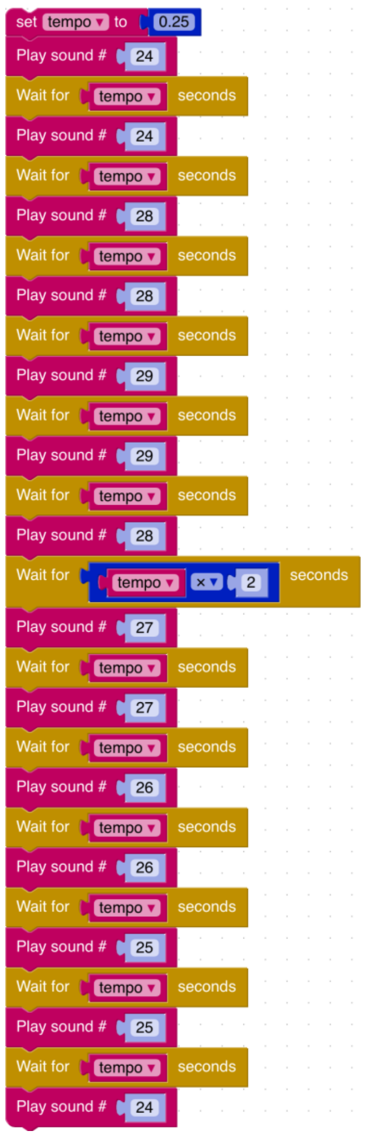


```
Play sound # 24
Wait for 0.25 seconds
Play sound # 24
Wait for 0.25 seconds
Play sound # 28
Wait for 0.25 seconds
Play sound # 28
Wait for 0.25 seconds
Play sound # 29
Wait for 0.25 seconds
Play sound # 29
Wait for 0.25 seconds
Play sound # 28
Wait for 0.5 seconds
Play sound # 27
Wait for 0.25 seconds
Play sound # 27
Wait for 0.25 seconds
Play sound # 26
Wait for 0.25 seconds
Play sound # 26
Wait for 0.25 seconds
Play sound # 25
Wait for 0.25 seconds
Play sound # 25
Wait for 0.25 seconds
Play sound # 24
```

Sound	Musical Note
17	1c
18	1d
19	1e
20	1f
21	1g
22	2a
23	2b
24	2c
25	2d
26	2e
27	2f
28	2g
29	3a

It is important to have the 'wait' blocks in place as these set the pauses between notes. These pauses could actually be created as a variable. This would mean the pause duration could be altered in one place rather than many times through the program. See next page.

This is an example of using a variable to set the pauses throughout the tune. One change at the top of the program will change the pauses throughout. Note: where a longer pause is needed the tempo variable is doubled (* 2). It could also be shortened by halving it (/ 2).

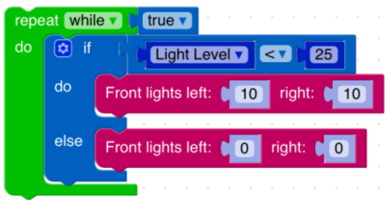


```
set tempo to 0.25
Play sound # 24
Wait for tempo seconds
Play sound # 24
Wait for tempo seconds
Play sound # 28
Wait for tempo seconds
Play sound # 28
Wait for tempo seconds
Play sound # 29
Wait for tempo seconds
Play sound # 29
Wait for tempo seconds
Play sound # 28
Wait for tempo × 2 seconds
Play sound # 27
Wait for tempo seconds
Play sound # 27
Wait for tempo seconds
Play sound # 26
Wait for tempo seconds
Play sound # 26
Wait for tempo seconds
Play sound # 25
Wait for tempo seconds
Play sound # 25
Wait for tempo seconds
Play sound # 24
```

Key Focus 4 – Automatic headlights

InO-Bot has a range of sensors which can be used as input triggers. These triggers can then activate outputs; for example to switch on headlights when dark.

The example on the right shows a conditional statement ('If…do…else' block) in use. 'If' the value from the light sensor is less the 25 then 'do' turn the front lights on. 'If' it's not less than 25 (25 or more) then 'else' turn them off. The check needs to be run continually (not just once) therefore a 'repeat while true' (forever) loop has been used. Some experimentation will be required to work out what value the light sensor should report to trigger the lights.



```
repeat while true
do
    if Light Level < 25
    do
        Front lights left: 10 right: 10
    else
        Front lights left: 0 right: 0
```

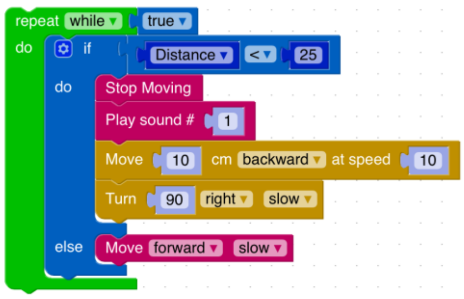
Note: See the user guide for more details on modifying 'if…do' blocks.

Key Focus 5 – Avoid obstacles

On the front of InO-Bot there is an ultrasonic range finder. This is a sensor that reports values from 0 to 255. The values are reported in centimetres and indicate how far an object is from the front of the sensor. InO-Bot can be programmed to move around and when the range finder reports an obstacle at a given distance in front it can back up and turn.

The ‘repeat while true’ (forever) loop contains a conditional check. If the distance sensor reports an object less than 20cm in front, the program makes InO-Bot reverse, turn and then start moving forward again.

There is a sound added (‘Play sound 1’) to indicate when an object is detected. The additional wait ensures commands run.

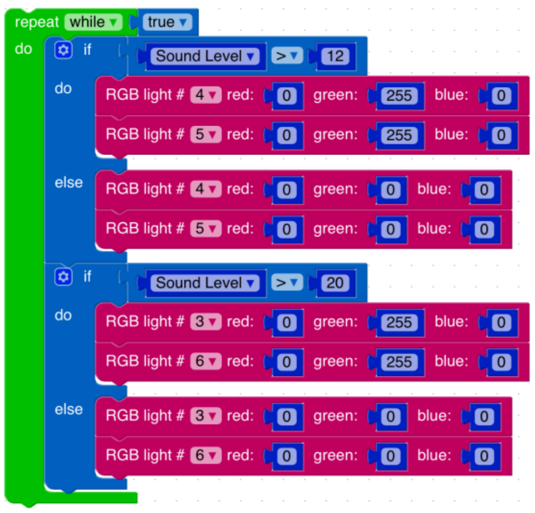


```
repeat while true
do  if  Distance < 25
    do    Stop Moving
          Play sound # 1
          Move 10 cm backward at speed 10
          Turn 90 right slow
    else  Move forward slow
```

Children should experiment with the commands above and spend time working out the optimum trigger distance. Should it be more or less than 20cm? There may be some discussion around the trigger point. Does InO-Bot react at exactly 20cm? If not, why not? Often it will be a little less and this is due to reaction time, i.e. the sensor sends data back to the PC, Scratch then needs to read that data and send commands to make InO-Bot do something else. Data needs to be sent back and forth. Although this data transmission is speedy there is still some lag. Are children able to modify their program to address this? For example, they may put 25cm when they want InO-Bot to stop at 20cm.

Key Focus 6 – Responding to sound

Just behind the pen tube is a sound sensor (a microphone reporting values 0 - 100). InO-Bot can be programmed to respond to sound. It might move, stop moving or light up different lights.



```
repeat while true
do  if  Sound Level > 12
    do    RGB light # 4 red: 0 green: 255 blue: 0
          RGB light # 5 red: 0 green: 255 blue: 0
    else  RGB light # 4 red: 0 green: 0 blue: 0
          RGB light # 5 red: 0 green: 0 blue: 0
    if  Sound Level > 20
    do    RGB light # 3 red: 0 green: 255 blue: 0
          RGB light # 6 red: 0 green: 255 blue: 0
    else  RGB light # 3 red: 0 green: 0 blue: 0
          RGB light # 6 red: 0 green: 0 blue: 0
```

An example is shown on the right. 'If' the sound level rises above 12 then 'do' light the front two LEDs 'else' turn them off. 'If' it rises above 20 then 'do' turn the next row of LEDs on 'else' turn them off. Because both conditional checks are in the same loop both can be true at the same time and therefore the first two rows of lights come on at the same time.

Note the commands used in the 'else' section. These are used to ensure the lights switch off again rather than switching on and staying on.

A further challenge could be to light another row of lights or make something else happen, e.g. at different sound levels change the colour of LEDs 4 and 5 only.

Getting Started with the InO-Bot app

The InO-Bot app should be downloaded from the Apple App Store or Google Play. InO-Bot is compatible with iPad 4 or newer (these devices have lightning connectors). InO-Bot is compatible with Android tablets with Bluetooth hardware running Android 5 or newer.

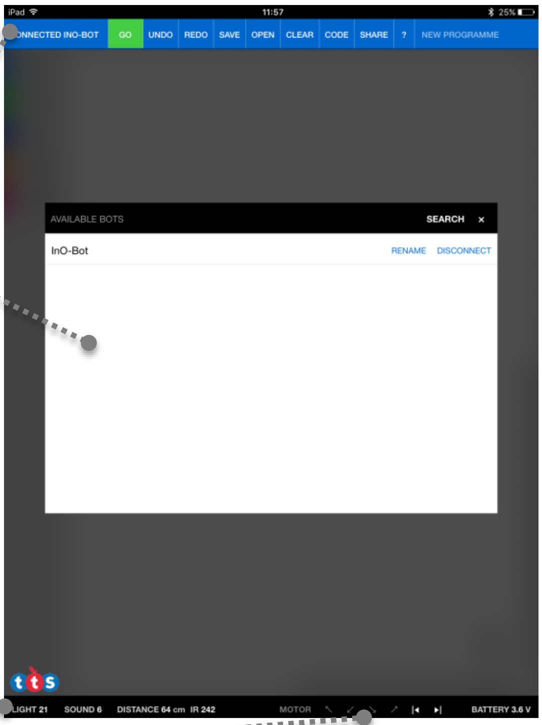
Bluetooth Connection

Note: InO-Bot does not needed pairing with an iPad or Android device. It should not be manually paired with one.

1) Switch on InO-Bot.
2) Launch the app.
3) The app should find and automatically connect to an available InO-Bot. Whilst it is connected, InO-Bot will not be available to other devices.

Advice when using a number of InO-Bots at the same time: Switch on all InO-Bots and all iPads. DO NOT open or launch the app. Open the app on one iPad only. That iPad will find an InO-Bot and connect to it. That pair of devices can then be used. Launch the app on another iPad and continue the process of launch and connect.

InO-Bot App Screen



This indicates an InO-Bo is connected. Tapping on this space will open the window shown.

Whilst InO-Bot is connected, readings from the sensors will be displayed on this bar.

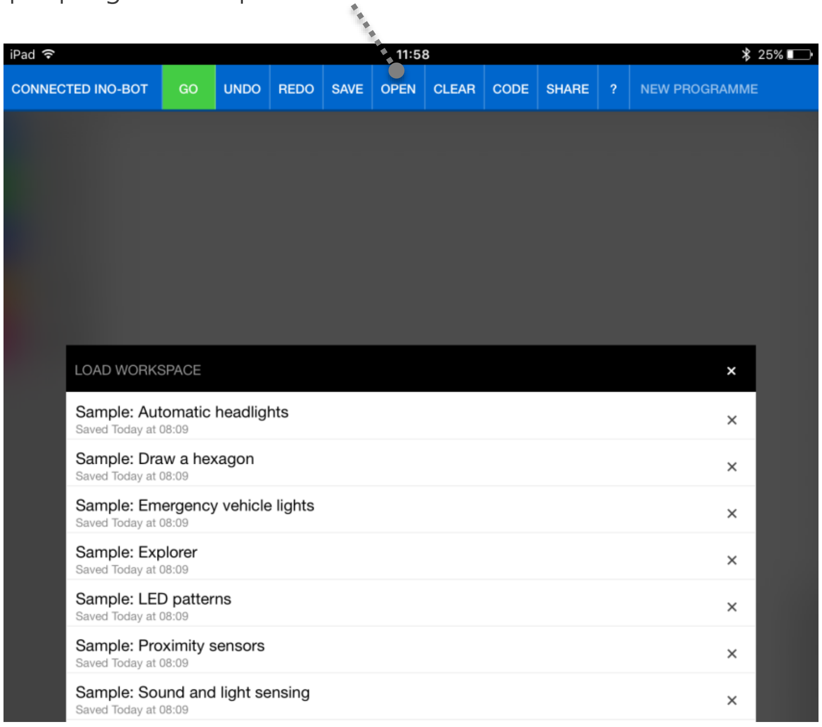
Corner proximity sensors and line follower sensors.

Sensor values

Sensor	Value
Light	0 to 255
Sound	0 to 255
Distance	0 to 255
IR – Infrared	255 to 0 (lower values = more IR light)

Example Programs

There are a number of example programs. Tap on OPEN to access them.



Sample: Automatic headlights
Saved Today at 08:09

Sample: Draw a hexagon
Saved Today at 08:09

Sample: Emergency vehicle lights
Saved Today at 08:09

Sample: Explorer
Saved Today at 08:09

Sample: LED patterns
Saved Today at 08:09

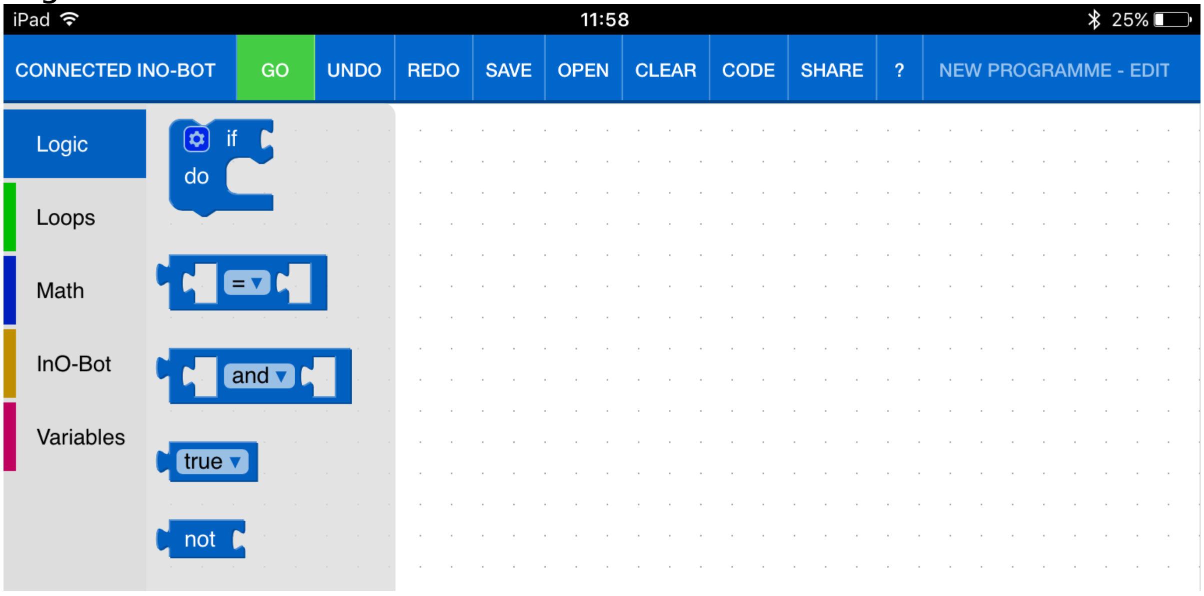
Sample: Proximity sensors
Saved Today at 08:09

Sample: Sound and light sensing
Saved Today at 08:09

Program	Description
Automatic headlights	Makes the headlights light in duller light conditions.
Draw a hexagon	Lowers the pen holder and moves in a hexagonal shape.
Emergency vehicle lights	Flashes LED 1, 4, 5 and 8 red and blue.
Explorer	Moves around avoiding obstacles detected by the range finder.
LED Patterns	Lights the top LEDs in different colours.
Proximity sensors	Lights LED 1, 4, 5 or 8 when nearest corner sensor is triggered.
Sound and light sensing	Lights LED 4 and 5 at given trigger sound level and lights headlights at given trigger light level.

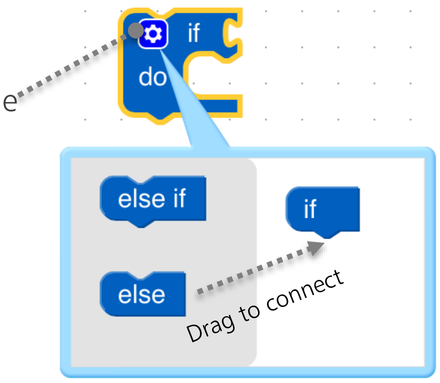
Blocks

Logic

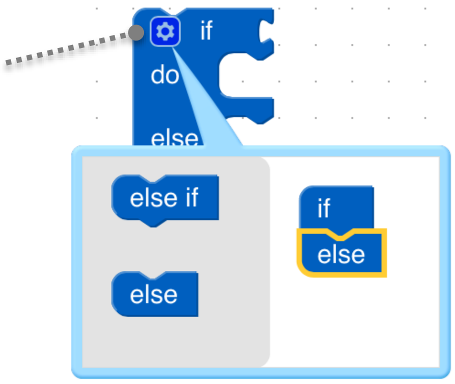


Modifying ‘if’ Blocks

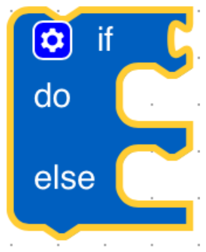
1. Drag an ‘if’ block onto the workspace. Tap on the cog to open the options panel.



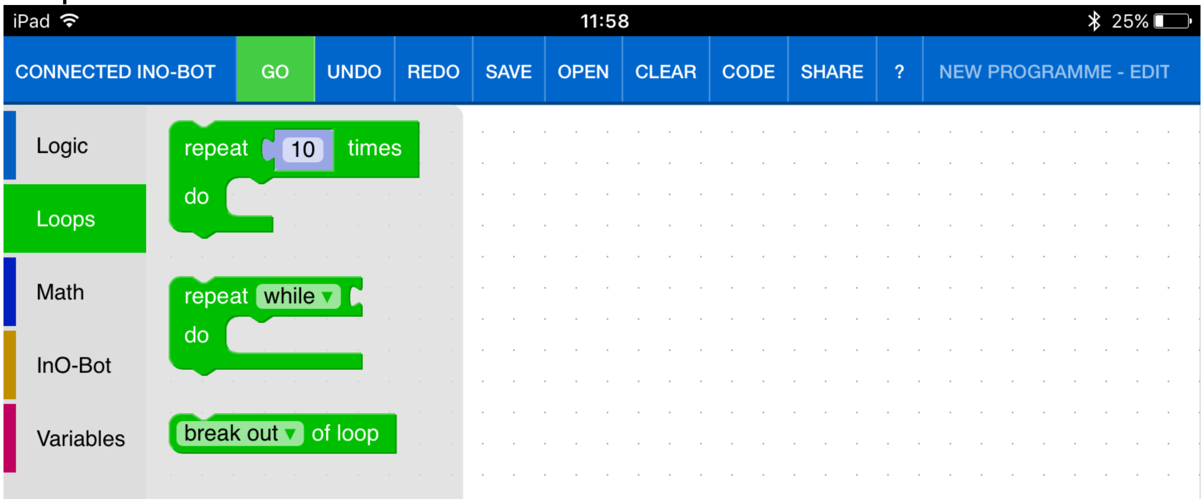
2. Tap on the cog again to close the option panel.



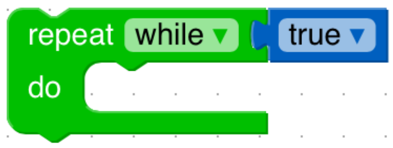
3. The modified ‘if’ block will remain on the workspace.



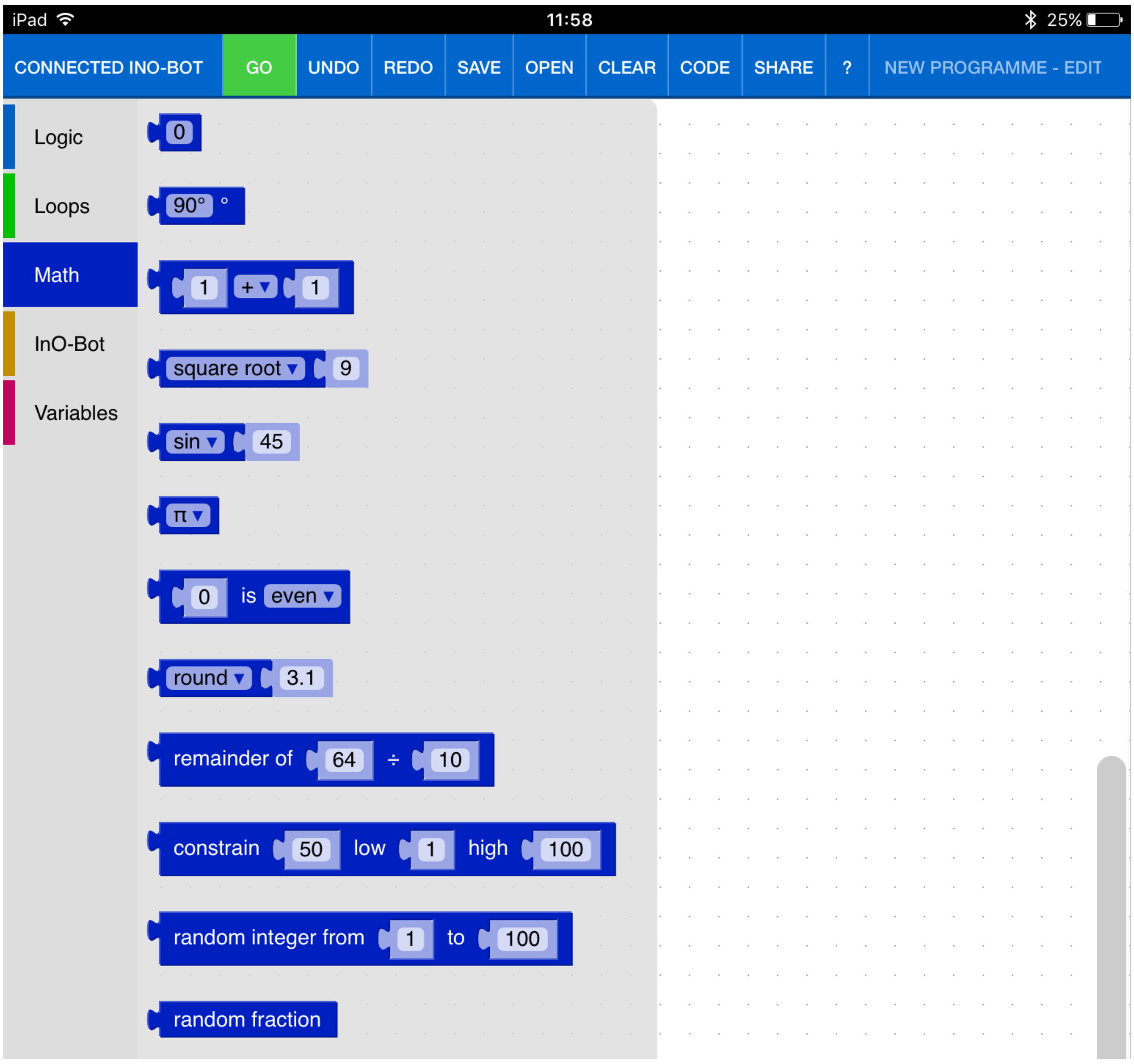
Loops



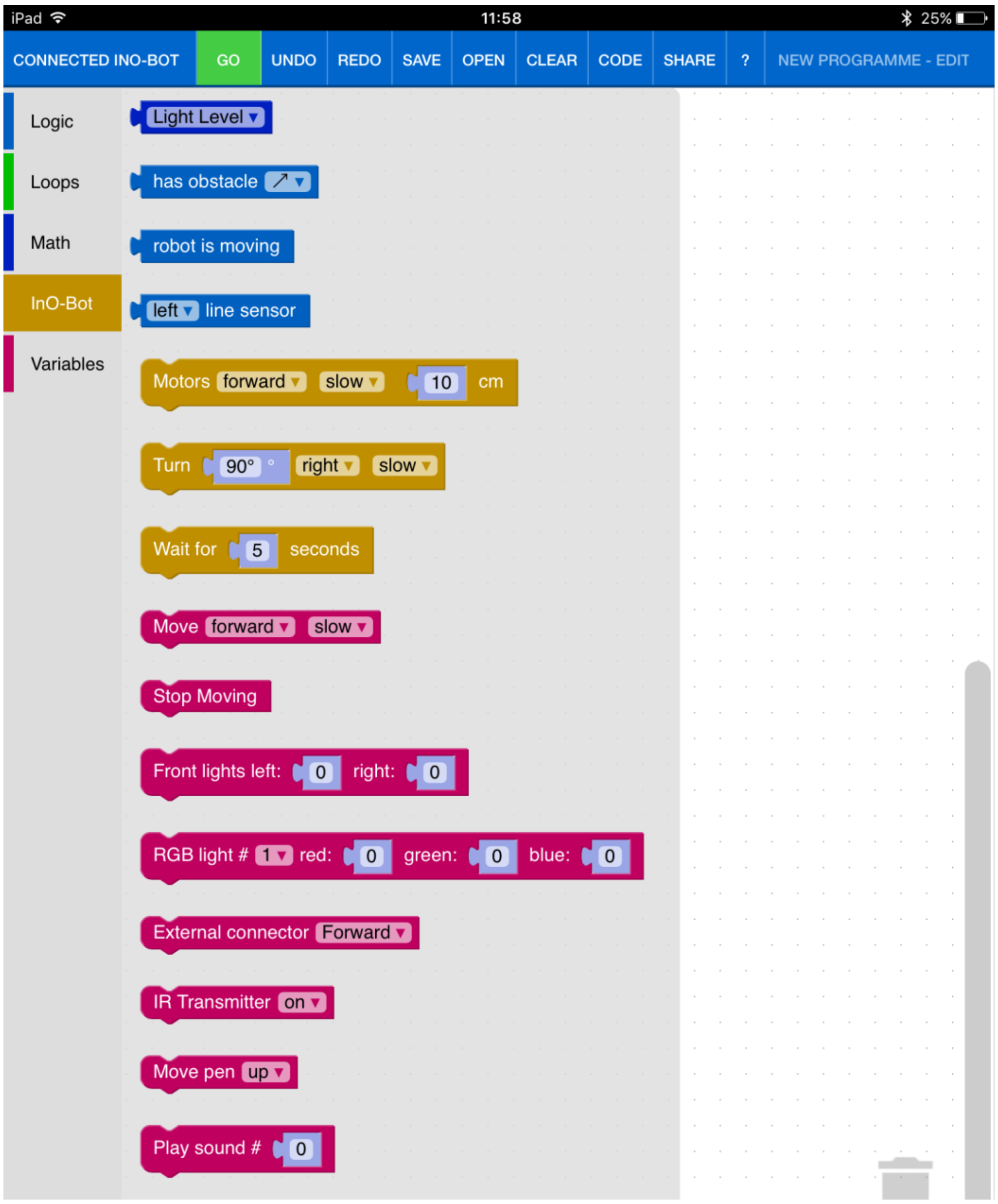
Repeat forever



Math

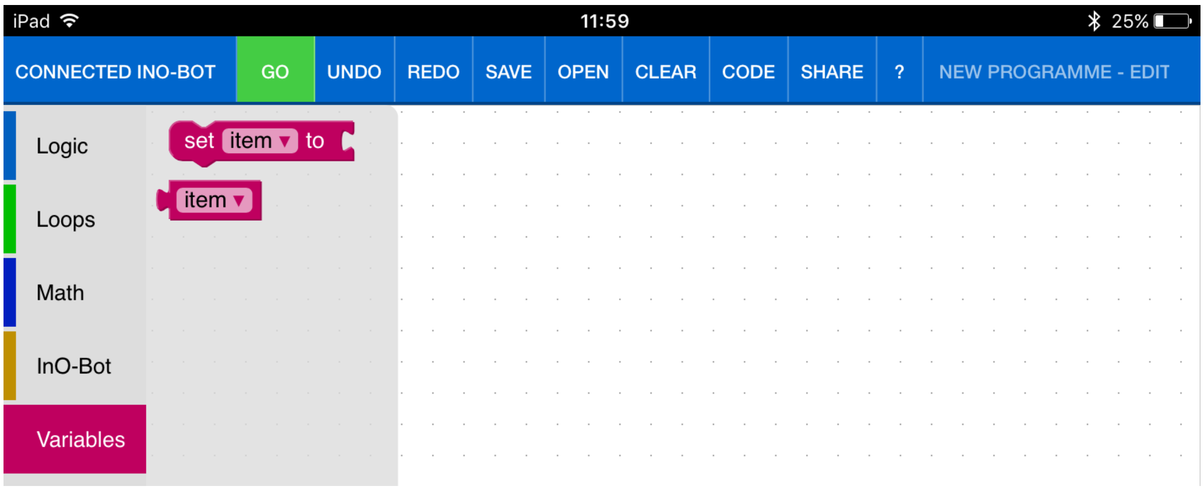

iPad
11:58
25%
CONNECTED INO-BOT
GO
UNDO
REDO
SAVE
OPEN
CLEAR
CODE
SHARE
?
NEW PROGRAMME - EDIT
Logic
Loops
Math
InO-Bot
Variables
0
90° °
1 + 1
square root 9
sin 45
π
0 is even
round 3.1
remainder of 64 ÷ 10
constrain 50 low 1 high 100
random integer from 1 to 100
random fraction

InO-Bot



Command	Accepted Values	Note
Motors	0 +	127cm is the maximum single movement. Values greater the 127 will result in a stepped movement.
Turn	0 +	180° in the maximum single movement. Values greater than 180 will result in a stepped movement.
Front lights	0 to 10	
RGB light #	0 – 255	
Play sound #	0 - 29	See index of sounds later in this document.

Variables


iPad
11:59
25%
CONNECTED INO-BOT
GO
UNDO
REDO
SAVE
OPEN
CLEAR
CODE
SHARE
?
NEW PROGRAMME - EDIT
Logic
Loops
Math
InO-Bot
Variables
set item to
item

Sounds

Effect		Piano		Xlyo	
Index	Sound	Index	Sound	Index	Sound
0	1	10	a	17	1c
1	2	11	b	18	1d
2	3	12	c	19	1e
3	4	13	d	20	1f
4	5	14	e	21	1g
5	6	15	f	22	2a
6	7	16	g	23	2b
7	8			24	2c
8	9			25	2d
9	10			26	2e
				27	2f
				28	2g
				29	3a

Technical Support

Please visit www.tts-group.co.uk for the latest product information.

Email feedback@tts-group.co.uk for technical support.

TTS Group Ltd.
Park Lane Business Park,
Kirkby-in-Ashfield,
Nottinghamshire,
NG17 9GU, UK.

Freephone: 0800 318686
Freefax: 0800 137525